UNSUPERVISED EVALUATION OF DEEP AUDIO EMBEDDINGS FOR MUSIC STRUCTURE ANALYSIS

Axel MARMORET (axel.marmoret@imt-atlantique.fr) (0000-0001-6928-7490)[1]

[1]*BRAIN team, MEE*, **IMT Atlantique**, Brest, France

ABSTRACT

Music Structure Analysis (MSA) aims to uncover the high-level organization of musical pieces. State-of-the-art methods are often based on supervised deep learning, but these methods are bottlenecked by the need for heavily annotated data and inherent structural ambiguities. In this paper, we propose an unsupervised evaluation of nine open-source, generic pre-trained deep audio models, on MSA. For each model, we extract barwise embeddings and segment them using three unsupervised segmentation algorithms (Foote's checkerboard kernels, spectral clustering, and Correlation Block-Matching (CBM)), focusing exclusively on boundary retrieval. Our results demonstrate that modern, generic deep embeddings generally outperform traditional spectrogram-based baselines, but not systematically. Furthermore, our unsupervised boundary estimation methodology generally yields stronger performance than recent linear probing baselines. Among the evaluated techniques, the CBM algorithm consistently emerges as the most effective downstream segmentation method. Finally, we highlight the artificial inflation of standard evaluation metrics and advocate for the systematic adoption of "trimming", or even "double trimming" annotations to establish more rigorous MSA evaluation standards.

1. INTRODUCTION

Beyond just sound, music is an intricate organization of scales and rhythms. At its core, musical composition relies on repetition, contrast, and variation to organize sound into coherent forms, establishing recognizable sections separated by distinct transitions. Music Structure Analysis (MSA) is the task of identifying this high-level organization. Specifically, it aims to partition a musical piece into meaningful, non-overlapping sections (*e.g.*, intro, verse, chorus, or bridge), and, in particular, to locate their precise temporal boundaries [1]. Beyond its established value in guiding applications like music summarization, cover song identification, and data-driven musicology, we believe that MSA will find renewed significance in the era of generative AI. By providing a structural framework, accurate MSA can enhance the editability, user control, and long-term coherence of AI-generated audio [2, 3].

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Yet, despite its practical importance, MSA remains a notoriously challenging task: musical form is inherently hierarchical, and multiple structural interpretations can be simultaneously valid depending on the listener's focus, the annotation protocol, and the desired level of granularity. Early work on MSA focused on self-similarity matrices (SSMs) [1, 4], *i.e.* square matrices indicating the pairwise similarity of all time instances in a song. From this viewpoint, sections are often understood as large, homogeneous and/or repeating regions, and boundaries as salient ruptures between subsequent sections. Comparing time instances to detect change points was the root of MSA for years, with significant research in the previous decade dedicated to refining the similarity computation between these instances [5–7]. This line of work was significantly advanced by the rise of deep learning, particularly representation learning [8–11]. Researchers have begun to use deep models in conjunction with standard downstream self-similarity segmentation algorithms to improve audio understanding. For instance, Salamon *et al.* [9] used deep embeddings to replace the traditional acoustic features in [6], improving the performance.

Parallel to these developments, supervised deep learning models [12–15] emerged as a dominant approach, explicitly learning to predict boundaries from labeled data and achieving state-of-the-art performance in MSA. However, this success comes with significant trade-offs. Supervised models rely heavily on large, meticulously annotated datasets, which are time-consuming and costly to produce. Furthermore, the inherent ambiguity of musical structure becomes a major bottleneck: supervised training biases estimates towards a single "ground truth" annotation protocol. This effectively discards other potentially valid, equivalent structural interpretations and limits the model's ability to generalize across different musical distributions.

Recent Self-Supervised Learning (SSL) paradigms offer a promising path to alleviate these issues. By leveraging pretext tasks (such as Masked Acoustic Modeling), these networks bypass the annotation bottleneck and learn directly from massive, unlabeled audio corpora. SSL dedicated to MSA (notably via MSA-specific inductive biases) has been shown to be effective in recent work [11]. Such specialized pretraining, however, requires a tailored architecture and training pipeline dedicated to structure. This raises an intriguing open question: do broad, general-purpose audio representations naturally encode structure-related information as a byproduct of their pre-training, even without being explicitly designed for structure? There is reason to expect so, as their pre-training tasks require

capturing rich timbral, temporal, and contextual characteristics. If they do, MSA could be performed without any task-specific training while directly benefiting from the rapid progress of general audio foundation models. Recent work by Toyama *et al.* [16] began to address that question by exploring whether generic deep audio models (SSL-based but not only) could benefit MSA. However, their study evaluated learned representations using linear probing, which involves training a supervised linear head on top of the frozen embeddings to estimate structure. While computationally efficient, linear probing remains fundamentally supervised and thus sensitive to the annotation drawbacks discussed above.

Our work shares the fundamental premise of Toyama *et al.* [16], but we employ strictly unsupervised downstream segmentation. In this paper, we study unsupervised boundary estimation using frozen embeddings extracted from nine open-source, generic, pre-trained deep learning audio models [17–25]. Following our previous work [7], these embeddings are computed at the bar scale. We evaluate these embeddings by computing barwise SSMs, and segment them using three standard and state-of-the-art unsupervised segmentation algorithms: Foote's checkerboard kernels [4], spectral clustering (LSD) [6], and the Correlation Block-Matching (CBM) algorithm [7]. We focus on the boundary retrieval task only, and do not attempt to label the sections. Our goal is two-fold: on one hand, to assess to what extent modern deep audio embeddings are rich enough to provide structured information about the data, and in turn improve unsupervised MSA; on the other hand, to compare the potential of traditional unsupervised segmentation algorithms to be improved when complemented with deep representation learning.

The remainder of this paper details our methodology for extracting and segmenting barwise embeddings (Section 2), outlines the experimental setup and evaluation metrics (Section 3), discusses our comparative results (Section 4), and concludes (Section 5).

2. METHODOLOGY

2.1 Deep Audio Models

To evaluate the structural information captured by modern representation learning, we extract embeddings from nine distinct open-source deep audio models [17–25]. These deep audio models differ significantly in their architectures and training objectives. Three of these nine models (MERT [18], MusicFM [19], and MuQ [20]) are only trained on music data, and their respective authors explicitly noted their potential utility for MSA.

The majority of the evaluated architectures rely on Masked Acoustic Modeling, an SSL paradigm where the network optimizes the reconstruction of masked portions of the audio signal. Within this category, AudioMAE [17] learns by reconstructing masked spectrogram patches. Meanwhile, MERT [18], MusicFM [19], and MuQ [20] learn to reconstruct indirect representations of the audio. Specifically, MERT uses pseudo-labels from acoustic teacher models, MusicFM predicts targets via a random projection quantizer, and MuQ predicts discrete tokens generated by a single-layer Residual Vector Quantization (RVQ). Additionally, M2D [21] and MATPAC++ [22] employ Masked Latent Prediction, using two models: one computing latent features of masked patches, the other one estimating these latent features from visible patches. MATPAC++ further incorporates Multiple Choice Learning to model prediction ambiguity, which could turn out to be very relevant for tackling the inherent ambiguity of MSA.

A second category of evaluated models consists of Neural Audio Codecs, which are generally designed as autoencoders to compress audio as aggressively as possible while maintaining high reconstruction fidelity. The extreme compression bottleneck forces these networks to learn highly abstract and compact latent representations of the acoustic space. DAC [23] is a prominent example of this approach, operating as a CNN-based autoencoder that utilizes RVQ to compress audio into discrete hierarchical tokens. To address the codebook collapse sometimes associated with traditional RVQ, CoDiCodec [24] utilizes Finite Scalar Quantization to project continuous representations onto a fixed integer grid. This architecture allows CoDiCodec to simultaneously extract both continuous summary embeddings and discrete acoustic tokens.

Finally, we evaluate models trained with cross-modal contrastive learning, which aligns audio and text latent spaces to bridge acoustic features with semantic concepts. CLAP [25] is the only evaluated model not pretrained via SSL reconstruction objectives: its audio encoder was first trained for supervised audio classification, then aligned with text via contrastive learning. M2D [21] and MuQ [20], while primarily focused on acoustic reconstruction, also incorporate a contrastive objective; both therefore yield two distinct representations: a unimodal audio space and a joint multimodal one.

2.2 Barwise Processing

In the context of Western modern music, we adopt the bar-scale as a musically motivated temporal resolution for MSA. Musical form is fundamentally tied to this metrical hierarchy, as macroscopic structural changes and repetitions frequently align with bar boundaries. Beyond this musical intuition, the validity of downbeat-synchronized boundaries is supported empirically: our previous work [7] demonstrated that barwise alignment maintains or improves the performance of Foote's checkerboard kernels [4] and CBM [7] algorithms, when compared with beat-scale processing. In that sense, embeddings will be individually computed on the different bars of the signal, thus resulting in barwise embeddings.

2.3 Downstream Segmentation Algorithm

After the computation of barwise deep embeddings, we apply three state-of-the-art unsupervised algorithms to perform the boundary retrieval task. All of these algorithms are based on SSMs [1] . Foote's algorithm [4] is a novelty-

[1] Authors of [6] prefer to mention "affinity matrices", with a clear definition of their computation, but the main principle remains the same.

based method that identifies abrupt structural transitions by correlating a square checkerboard kernel along the main diagonal of the SSM. Conversely, the LSD approach [6] takes a graph-theoretic perspective to capture long-term repetitions; it treats the SSM as an adjacency matrix and analyzes the eigenvectors of its normalized Laplacian to estimate boundaries via K-means change-points. Finally, the CBM algorithm [7] frames boundary retrieval as a global optimization problem solved via dynamic programming. Conceptually similar to Foote's method but optimized for detecting homogeneous regions rather than abrupt novelty, CBM identifies optimal boundaries by maximizing a score function that represents block structures around the main diagonal.

2.4 Barwise TF – Baseline Representation

To situate deep embeddings against a non-deep-learning reference, we compare them to a baseline built from classical acoustic features. MSA has traditionally relied on low-level features such as Mel spectrograms, chromagrams, or MFCCs [1]. We adopt the *Barwise TF* matrix [7], which represents the signal as a sequence of barwise vectors, each flattening the intra-bar time-frequency content of a bar into a single vector. Following [7], this content is computed on Log-Mel spectrograms, a representation shown in that work to perform strongly with the Foote and CBM algorithms, then reaching levels close to those of supervised deep-learning models [8–10].

3. EXPERIMENTAL SETTINGS

3.1 Implementation Details

We operate entirely at the bar scale, estimating downbeats using the Beat This! [26] model. To ensure compatibility, raw audio signals are segmented into bar-length chunks before computing embeddings from each model's final latent layer. Because the evaluated models operate at different native temporal resolutions, we homogenize the representations by averaging across the temporal dimension, yielding a single vector per bar. We adopt this framework for consistency, despite the contradiction between the gains reported in [16] and the temporal preservation favored by [7]. Investigating this specific impact is beyond our current scope.

Using these barwise embeddings, we compute barwise SSMs via both RBF and Cosine similarity measures, then segmented using Foote's and CBM algorithms. The exception is the LSD algorithm, which inherently computes its own specialized affinity matrix to emphasize diagonal stripes. The same SSM computation is applied to the Barwise TF baseline representation introduced in Section 2.4.

Regarding the deep learning models, we used *HuggingFace* for pretrained models whenever possible (*i.e.* for AudioMAE, CLAP, DAC, MERT, and MuQ) [2] . All other pretrained models were downloaded from their official repositories: CoDiCodec [3] , M2D [4] , MATPAC++ [5] , and MusicFM [6] (MSD checkpoint). We release our code for reproducing experiments [7] .

3.2 Hyperparameter Selection

To ensure optimal and fair downstream performance, we performed a grid search over the hyperparameters of each segmentation algorithm. For Foote's algorithm, we searched the kernel size and the novelty curve median filtering over $\{8, 12, 16\}$. For LSD, we varied the number of clusters $k \in \{4, 6, 8, 9, 10, 11, 12, 13, 14, 16\}$ [8] and set the median filtering parameter equal to k. For the CBM algorithm, we compared the Full and 7-band kernels. Crucially, we disabled the CBM penalty function that enforces specific segment sizes; omitting this penalty limits empirical structural priors and ensures the resulting segmentation better reflects the discriminative power of the evaluated embeddings. A study of the influence of these hyperparameters is presented in the Supplementary Material. LSD and Foote are computed using the *MSAF* [27] toolbox, and the CBM using the original implementation.

This search yields, per model, a set of candidate configurations, from which we report two conditions. In the *per-dataset* condition, the best configuration (averaging $\mathrm{F}_{0.5\mathrm{s}}$ and $\mathrm{F}_{3\mathrm{s}}$) is selected per model *and* dataset; this optimistic estimate reflects each embedding's intrinsic potential. In the *cross-dataset* condition, a single configuration per model is selected, maximizing performance *on average across all datasets*; this mimics a deployment setting where hyperparameters are fixed once and applied without re-tuning, though as the configuration still sees all datasets, it approximates rather than strictly tests generalization. Figures 1 and 2 use the *per-dataset* condition, Table 1 the *cross-dataset* one.

3.3 Metrics & Datasets

Following standard protocols, we assess boundary detection using the hit-rate F-measure at 0.5s and 3s tolerances ($\mathrm{F}_{0.5\mathrm{s}}$ and $\mathrm{F}_{3\mathrm{s}}$ respectively). Scores are computed using *mir_eval* [28]. We handle datasets via *mirdata* [29] and evaluate our approach on three standard MSA benchmarks: RWC-Pop [30] (now open-source [31]), SALAMI [32], and Harmonix [33]. For RWC-Pop, we use the MIREX10 annotation set across its 100 popular music tracks. From the SALAMI dataset, we restrict our evaluation to the 884 tracks that possess two independent annotations as in [9, 14]; we evaluate against the coarse-level annotations and report the best score among both annotations. Finally, we utilize the 912 Western popular music tracks comprising the Harmonix set.

[2] Respectively using the checkpoints `hance-ai/audiomae`, `laion/clap-htsat-unfused`, `descript/dac_44khz`, `MERT-v1-95M`, and `MuQ` | `MuQ-MuLan-large` (for audio and multimodal spaces)

[3] `https://github.com/SonyCSLParis/codicodec`

[4] `https://github.com/nttcslab/m2d`

[5] `https://github.com/aurianworld/matpac`

[6] `https://github.com/minzwon/musicfm`

[7] `https://github.com/ax-le/msa_deep_embeddings`

[8] We observe a high variability of the results given this parameter; detailed results are available in the Supplementary Material.

3.4 Trimming

Trimming annotations and predictions consists of removing the first and last segments before computing scores. This practice remains largely overlooked in the MSA literature: to the best of our knowledge, only Buisson *et al.* [11, 14] systematically trim annotations, and only the recent study of Korzeniowski and Vogl [15] explicitly investigates the impact of trimming on segmentation results. Yet evaluating this impact is critical for a rigorous analysis: the absolute first and last boundaries carry no information about segmentation quality and can artificially inflate performance metrics.

In our experiments, we will present results **without trimming** first, to remain consistent with existing literature. Nonetheless, we will also present comparative results of our best scores with and without trimming to study the apparent drop in performance that occurs when this artificial inflation is removed.

When analyzing the RWC-Pop (MIREX10) and SALAMI datasets, we observed that annotations systematically include silent segments at the file extremities (specifically, segments between 0 and the start of the audio signal, and between the end of the signal and the end of the file). We argue that these segments do not provide meaningful information regarding the quality of a structural prediction and should therefore be excluded. We term this preprocessing step "double trimming": first, removing the silent segments, and subsequently trimming both annotations and predictions to the active signal boundaries. We also evaluate the impact of this condition on the resulting performance scores. Since the annotations for the Harmonix dataset do not systematically contain these silences, we remove them by default; consequently, trimming on Harmonix is *de facto* considered double trimming.

Ideally, we expect that future standards in MSA will adopt trimming, or even double trimming, by default.

4. RESULTS AND DISCUSSION

4.1 Deep Models *vs.* Barwise TF features

Figure 1 compares peak deep audio models' performance against the Barwise TF baseline. Deep embeddings achieve the highest scores across all datasets and algorithms, demonstrating clear improvements over spectrogram-based features.

Figure 2 ranks the nine evaluated deep models. Notably, three of them (and one multimodal space) are outperformed by the baseline. The fact that nearly a third of the models fail to beat traditional acoustic features suggests that their specific pretraining objectives or latent spaces may not align well with structural musical properties. Among the successful models, MATPAC++ achieves the highest overall performance, with M2D and the continuous space of CoDiCodec closely following, indicating their latent spaces are particularly well-suited to disambiguate barwise audio representations. Although we observe differences when using multimodal versus unimodal audio spaces, the results are contradictory across conditions and yield no conclusive advantage for either approach. Both discrete embedding spaces (DAC and the discrete version of CoDiCodec) yielded relatively poor performance, with CoDiCodec's discrete representation notably underperforming its continuous counterpart. This suggests that discrete embeddings are ill-suited for our methodology. Finally, models trained exclusively on music data (MERT, MusicFM, and MuQ) consistently underperform, with results ranging from medium to poor.

Overall, these results demonstrate that while the rich semantic and temporal contexts of learned representations provide a distinct advantage for boundary detection, off-the-shelf deep audio embeddings do not universally guarantee improved segmentation over standard feature-based representations.

4.2 Downstream Segmentation Algorithm

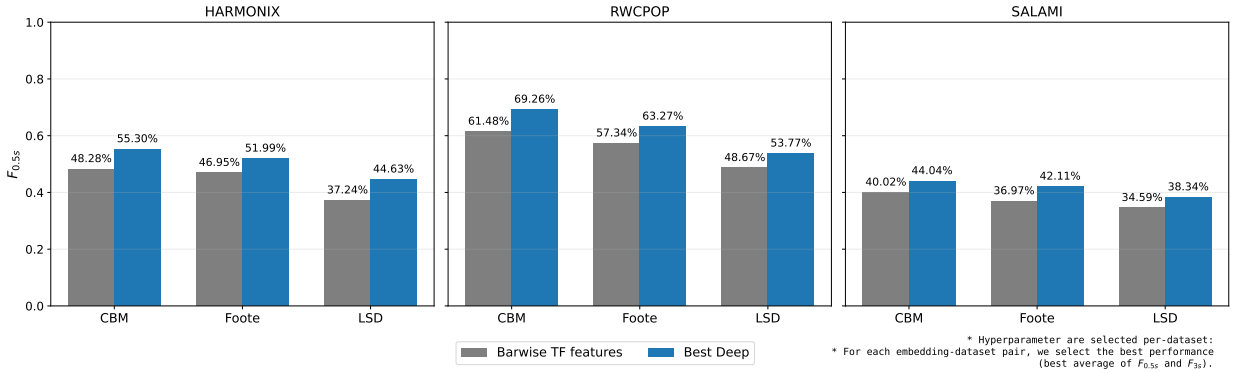
As indicated by the results in Figure 2, CBM is the most effective downstream segmentation algorithm for the vast majority of embedding-dataset pairs. Although the Foote algorithm occasionally achieves superior performance (notably with CoDiCodec), CBM performs best in 33 out of 36 conditions. These results demonstrate the interest of the CBM in MSA, even in conjunction with deep embeddings.

This study also marks the first application of the LSD algorithm at the bar scale. Unlike the Foote and CBM algorithms, designed to identify novelty and homogeneity (*i.e.* abrupt changes or local similarity), LSD is specialized for detecting "stripes" (*i.e.* repeating patterns). Given its proven effectiveness in the literature [11], we hypothesize that LSD's relative underperformance here stems from a misalignment with the barwise representations (a premise for barwise processing is a higher homogeneity between bars than beats) rather than a limitation of the algorithm itself to segment deep embeddings.

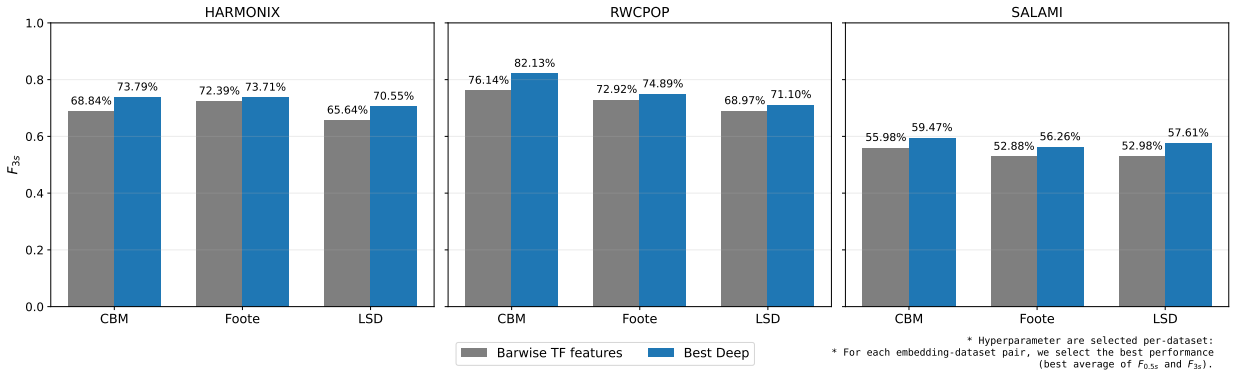
4.3 Results compared with literature

Next, we compare our approach against state-of-the-art models in Table 1. For this comparison, we report results using only the CBM downstream algorithm, under the *cross-dataset* condition for hyperparameters (Section 3.2). We also report results obtained by running locally the All-In-One [13] model, on RWC-Pop and SALAMI only, since Harmonix is part of its training data.

Five of the nine models (AudioMAE [17], MusicFM [19], MERT [18], DAC [23], and CLAP [25]) were compared in [16], where their embeddings were studied in a linear probing fashion. Comparing our CBM-based downstream approach to these linear probing baselines on the Harmonix dataset, we observe that our methodology yields significantly stronger performance for almost all shared models. Most notably, CLAP and MERT show substantial improvements, rising from 29.21% to 49.32% and from 42.23% to 52.43% for $\mathrm{F}_{0.5\mathrm{s}}$, respectively. This suggests that the CBM algorithm and barwise processing are highly effective at extracting structural information from these deep audio embeddings, notably when compared with linear probing.



(a) Best results obtained for $\mathrm{F}_{0.5\mathrm{s}}$



(b) Best results obtained for $\mathrm{F}_{3\mathrm{s}}$

Figure 1: Comparison of the best results obtained with deep models and the Barwise TF features (non-deep learning baseline), according to the segmentation algorithm and the dataset. The hyperparameters of the downstream segmentation algorithms are selected under the *per-dataset* condition (Section 3.2).

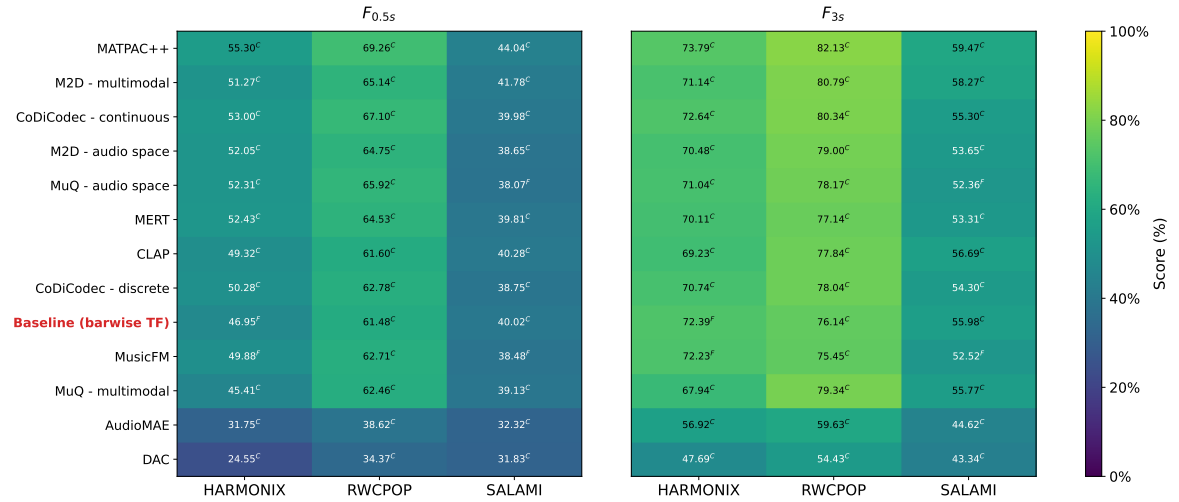


Figure 2: Best results obtained with all deep learning models, and their best downstream segmentation algorithm. Rows are ordered by decreasing average of $\mathrm{F}_{0.5\mathrm{s}}$ and $\mathrm{F}_{3\mathrm{s}}$. Superscript denotes the downstream segmentation algorithm used to obtain these results (C: CBM, F: Foote). The hyperparameters of the downstream segmentation algorithms are selected under the *per-dataset* condition (Section 3.2).

Overall, three methods stand out: the self-supervised approach of Buisson *et al.* [11], obtaining the best $\mathrm{F}_{3\mathrm{s}}$ results by a wide margin; the supervised All-In-One, obtaining the best $\mathrm{F}_{0.5\mathrm{s}}$ results on the datasets it did not train on; and, among training-free generic embeddings, MATPAC++. In particular, we observe that MATPAC++ proves highly competitive with specialized literature on both Harmonix and RWC-Pop: only the two models from Buisson *et al.* [11, 14] and All-In-One [13] surpass its performance, while it required no supervision. In addition, the performance on the RWC-Pop dataset is on par with the best-performing model (All-In-One).

Focusing on Buisson *et al.*, we stress that their approach and ours differ methodologically: they train a model

Table 1: Deep model segmentation performance across datasets (%), using the CBM algorithm as the downstream segmentation algorithm (shown to be the best one on average). Hyperparameters are selected under the *cross-dataset* condition (Section 3.2). Best results per column are highlighted in **bold** (differentiating baselines and our methods). *represents models where annotations were trimmed. †represents results where the dataset was used for training (in cross-validation settings). ‡represents oracle condition for hyperparameter fitting.

Deep Model		HARMONIX		RWCPOP		SALAMI	
		$\mathbf{F_{0.5}}$	$\mathbf{F_3}$	$\mathbf{F_{0.5}}$	$\mathbf{F_3}$	$\mathbf{F_{0.5}}$	$\mathbf{F_3}$
AudioMAE		31.75± 18.21	56.92± 18.46	38.62± 16.19	59.63± 18.01	25.86± 12.98	43.51± 15.76
CLAP		49.32± 21.38	69.23± 17.26	61.60± 17.98	77.84± 17.01	35.18± 16.70	51.72± 17.32
CoDiCodec	Discrete	50.28± 20.52	70.74± 16.02	62.78± 16.98	78.04± 15.27	35.01± 16.74	50.51± 17.72
	Continuous	53.00± 20.70	72.64± 16.34	67.10± 18.25	80.34± 15.11	36.54± 17.31	51.73± 17.55
DAC		24.10± 19.38	47.08± 21.57	34.37± 16.83	54.43± 19.45	25.59± 13.17	42.87± 15.71
M2D	Audio space	52.05± 18.79	70.48± 14.30	65.14± 13.77	75.79± 12.70	36.02± 16.88	49.29± 17.95
	Multimodal	50.90± 21.50	70.59± 17.22	65.14± 15.97	80.79± 15.42	36.10± 17.24	52.58± 17.71
MATPAC++		**55.30**± 20.83	**73.79**± 15.75	**69.26**± 16.42	**82.13**± 14.88	38.90± 17.66	53.74± 17.74
MERT		52.43± 19.25	70.11± 14.98	64.53± 15.41	77.14± 13.79	36.81± 16.80	49.52± 17.98
MuQ	Audio space	51.97± 18.66	69.67± 14.62	65.92± 13.99	78.17± 13.06	35.68± 16.89	48.73± 17.70
	Multimodal	45.41± 20.40	67.94± 17.27	62.46± 16.57	79.34± 14.93	33.95± 15.73	51.30± 16.97
MusicFM		51.55± 18.18	70.52± 13.86	59.66± 11.58	71.50± 11.35	38.37± 16.02	51.95± 16.88
Barwise TF baselines							
Foote		46.95± 19.18	72.39± 14.63	57.50± 13.70	71.41± 13.72	36.96± 15.67	52.87± 16.98
LSD		37.41± 16.07	65.21± 14.24	48.17± 12.73	67.48± 14.68	33.70± 12.61	52.51± 14.65
CBM		48.28± 17.56	68.84± 13.25	58.51± 12.60	71.27± 11.71	**40.02**± 14.76	**55.98**± 15.16
Literature results							
AudioMAE scores from [16]		36.95	58.11	-	-	-	-
CLAP scores from [16]		29.21	46.60	-	-	-	-
DAC scores from [16]		19.10	39.63	-	-	-	-
MERT scores from [16]		42.23	60.99	-	-	-	-
MusicFM scores from [16]		49.76	63.91	-	-	-	-
Salamon *et al.* [9]		45.74	68.84	-	-	33.78	55.65
Wang *et al.* [10]		49.7	73.8	-	-	-	-
Buisson *et al.* (supervised) [14]*		56.8†	71.7†	58.5†	75.0†	-	-
All-In-One [13]	from [13]	**66.0**†	-	-	-	-	-
	run locally	(Not reported - train/test leakage)		**71.10**± 14.28	**80.07**± 11.01	**47.75**± 16.62	60.06± 15.30
Buisson *et al.* (SSL) [11]*‡		48.5	**80.8**	-	-	39.8 ± 16	**70.1** ±17

purpose-built for MSA, with structure-specific inductive biases, whereas MATPAC++ is a *generic* audio model, simply paired with a standard, unsupervised segmentation algorithm. Their scores also benefit from a favorable evaluation protocol: their model produces a hierarchy of segmentations (by varying the number of clusters in LSD), and only the level maximizing each metric is reported per track [11]. This oracle selection is a legitimate and explicitly stated choice, but one that our single-level protocol does not exploit, suggesting the true gap is even smaller than the raw scores indicate.

All-In-One, in contrast, was used out-of-the-box, without modification or favorable tuning. Its performance comes at a cost, however: it is fully supervised and relies on rich, task-specific inputs, including source-separated instrument stems. MATPAC++ uses none of this, yet trails All-In-One by only a few points on RWC-Pop. That MATPAC++, a generic, training-free embedding using none of this, stays within reach of both a purpose-built self-supervised model and a fully supervised state-of-the-art system is, in our view, the central finding: general-purpose representations already encode rich structural information.

Table 2: MATPAC++, Barwise TF-CBM, and All-In-Oneresults from Table 1, in different trimming conditions.

Method		HARMONIX		RWCPOP		SALAMI	
		$\mathbf{F_{0.5}}$	$\mathbf{F_3}$	$\mathbf{F_{0.5}}$	$\mathbf{F_3}$	$\mathbf{F_{0.5}}$	$\mathbf{F_3}$
MATPAC++	No trimming	55.30± 20.83	73.79± 15.75	69.26± 16.42	82.13± 14.88	38.90± 17.66	53.74± 17.74
	Trimming	–	–	65.39± 18.48	79.87± 16.83	30.05± 19.51	47.12± 19.70
	Double trimming	54.27± 23.35	70.96± 17.83	65.04± 21.67	77.88± 19.10	28.20± 22.21	43.33± 21.93
Barwise TF CBM	No trimming	48.28± 17.56	68.84± 13.25	58.51± 12.60	71.27± 11.71	40.02± 14.76	55.98± 15.16
	Trimming	–	–	53.18± 14.33	67.59± 13.26	28.80± 16.78	47.65± 17.98
	Double trimming	45.55± 19.79	65.10± 15.14	51.00± 16.67	64.08± 15.09	25.68± 19.78	42.38± 21.67
All-In-One [13]	No trimming	(Not reported - train/test leakage)		71.10± 14.28	80.07± 11.01	47.75± 16.62	60.06± 15.30
	Trimming	–	–	67.13± 16.22	77.27± 12.59	37.50± 19.46	52.23± 18.00
	Double trimming	–	–	65.30± 18.00	74.61± 13.89	32.70± 24.03	46.73± 22.74

We conclude that while SSL representations dedicated to MSA still offer significant benefits, there is room for refinement. Specifically, we hypothesize that incorporating barwise processing could substantially improve $\mathrm{F}_{0.5\mathrm{s}}$ performance. Developing future bar-scale SSL models, segmented with a standard algorithm (notably CBM), may thus improve current results.

4.4 Trimming results

Finally, we present trimmed results in Table 2. These results demonstrate a consistent loss of performance when trimming annotations, which was expected and consistent with [15]. While trimming has only a marginal impact on the Harmonix dataset, the degradation can be severe elsewhere; for instance, $\mathrm{F}_{0.5\mathrm{s}}$ performance drops by more than 11% for the Barwise TF CBM on the SALAMI dataset, and degrades even further with double trimming. Because standard evaluation without trimming can artificially inflate scores by rewarding trivial boundary matches, we advocate that trimming should be adopted as the new standard for rigorous MSA evaluation.

5. CONCLUSION

In this work, we investigated the training-free capabilities of nine generic deep audio models for Music Structure Analysis, specifically focusing on boundary retrieval. By leveraging unsupervised downstream segmentation algorithms at the bar scale, we demonstrated that deep audio embeddings generally provide a distinct advantage over traditional barwise acoustic features. Among the evaluated representations, MATPAC++ proved particularly well-suited for estimating structural boundaries. Furthermore, our comparative analysis established the Correlation Block-Matching algorithm as the most effective downstream segmentation method, notably outperforming recent linear probing approaches.

Despite these advancements, the strongest results are still obtained by models specifically designed for the task: a self-supervised and a fully supervised approach. Our contribution is therefore not to surpass these specialized systems, but to show that generic, off-the-shelf representations reach competitive performance at no task-specific training cost. Future work could bring the effective ingredients identified here (bar-scale processing and standard segmentation algorithms such as CBM) to specialized, notably SSL, models, which could help close the remaining gap. Additionally, we demonstrated that traditional evaluation metrics are often artificially inflated by initial and final boundary matches. We strongly advocate for the community to adopt standard trimming and double trimming practices for more rigorous future evaluations.

6. ACKNOWLEDGMENTS

With support of ANR JCJC MusAIc ANR-25-CE23-5316.

7. REFERENCES

[1] O. Nieto *et al.*, "Audio-based music structure analysis: Current trends, open challenges, and applications," *Trans. Int. Soc. Music Information Retrieval*, vol. 3, no. 1, 2020.

[2] F. Morreale, M. A. Martinez-Ramirez, R. Masu, W. Liao, and Y. Mitsufuji, "Reductive, exclusionary, normalising: the limits of generative ai music," *Trans. Int. Soc. Music Information Retrieval*, vol. 8, no. 1, 2025.

[3] Y. Choi, J. Moon, J. Yoo, and J.-H. Hong, "Understanding the potentials and limitations of prompt-based music generative ai," in *Proc. 2025 CHI Conf. on Human Factors in Computing Systems*, 2025.

[4] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Int. Conf. Multimedia and Expo*. IEEE, 2000.

[5] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised music structure annotation by time series structure features and segment similarity," *IEEE Trans. Multimedia*, vol. 16, no. 5, 2014.

[6] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2014.

[7] A. Marmoret, J. E. Cohen, and F. Bimbot, "Barwise music structure analysis with the correlation block-matching segmentation algorithm," *Trans. Int. Soc. Music Information Retrieval*, vol. 6, no. 1, 2023.

[8] M. C. McCallum, "Unsupervised learning of deep features for music segmentation," in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[9] J. Salamon, O. Nieto, and N. J. Bryan, "Deep embeddings and section fusion improve music segmentation," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2021.

[10] J.-C. Wang, J. B. Smith, W.-T. Lu, and X. Song, "Supervised metric learning for music structure feature," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2021.

[11] M. Buisson, B. McFee, S. Essid, and H. C. Crayencour, "Self-supervised learning of multi-level audio representations for music segmentation," *IEEE/ACM Trans. Audio, Speech and Language Processing*, vol. 32, 2024.

[12] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2015.

[13] T. Kim and J. Nam, "All-in-one metrical and functional structure analysis with neighborhood attentions on demixed audio," in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2023.

[14] M. Buisson, B. Mcfee, and S. Essid, "Using pairwise link prediction and graph attention networks for music structure analysis," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2024.

[15] F. Korzeniowski and R. Vogl, "Simple and effective semantic song segmentation," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2025.

[16] K. Toyama *et al.*, "Do foundational audio encoders understand music structure?" in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2026.

[17] P.-Y. Huang *et al.*, "Masked autoencoders that listen," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

[18] Y. Li *et al.*, "MERT: Acoustic music understanding model with large-scale self-supervised training," *arXiv preprint arXiv:2306.00107*, 2023.

[19] M. Won, Y.-N. Hung, and D. Le, "A foundation model for music informatics," in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

[20] H. Zhu *et al.*, "MuQ: Self-supervised music representation learning with mel residual vector quantization," *IEEE/ACM Trans. Audio, Speech and Language Processing*, 2025.

[21] D. Niizumi *et al.*, "M2D-CLAP: Masked modeling duo meets clap for learning general-purpose audio-language representation," in *Interspeech*. ISCA, 2024.

[22] A. Quelennec, P. Chouteau, G. Peeters, and S. Essid, "MATPAC++: Enhanced masked latent prediction for self-supervised audio representation learning," *arXiv preprint arXiv:2508.12709*, 2025.

[23] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved RVQGAN," *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[24] M. Pasini, S. Lattner, and G. Fazekas, "CoDiCodec: Unifying continuous and discrete compressed representations of audio," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2025.

[25] Y. Wu *et al.*, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

[26] F. Foscarin, J. Schlüter, and G. Widmer, "Beat This! accurate beat tracking without dbn postprocessing," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2024.

[27] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2016.

[28] C. Raffel *et al.*, "mir_eval: A transparent implementation of common MIR metrics," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2014.

[29] R. M. Bittner *et al.*, "mirdata: Software for reproducible usage of datasets." in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2019.

[30] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2002.

[31] S. Balke *et al.*, "RWC revisited: Towards a community-driven MIR corpus," *Trans. Int. Soc. Music Information Retrieval*, vol. 9, no. 1, 2026.

[32] J. B. Smith *et al.*, "Design and creation of a large-scale database of structural annotations," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2011.

[33] O. Nieto *et al.*, "The Harmonix set: Beats, downbeats, and functional segment annotations of western popular music," in *Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2019.