A Contrastive Self-Supervised Learning scheme for beat tracking amenable to few-shot learning

Antonin Gagnere, Geoffroy Peeters, Slim Essid

A CONTRASTIVE SELF-SUPERVISED LEARNING SCHEME FOR BEAT TRACKING AMENABLE TO FEW-SHOT LEARNING

Antonin Gagneré **Slim Essid** **Geoffroy Peeters**
LTCI - Télécom Paris, Institut Polytechnique de Paris, France
antonin.gagnere@telecom-paris.fr

ABSTRACT

In this paper, we propose a novel Self-Supervised-Learning scheme to train rhythm analysis systems and instantiate it for few-shot beat tracking. Taking inspiration from the Contrastive Predictive Coding paradigm, we propose to train a Log-Mel-Spectrogram-Transformer-encoder to contrast observations at times separated by hypothesized beat intervals from those that are not. We do this without the knowledge of ground-truth tempo or beat positions, as we rely on the local maxima of a Predominant Local Pulse function, considered as a proxy for Tatum positions, to define candidate anchors, candidate positives (located at a distance of a power of two from the anchor) and negatives (remaining time positions). We show that a model pre-trained using this approach on the unlabeled FMA, MTT and MTG-Jamendo datasets can successfully be fine-tuned in the few-shot regime, *i.e.* with just a few annotated examples to get a competitive beat-tracking performance.

1 Introduction

Beat-tracking, *i.e.* locating the times in a musical audio signal where beats are perceived or notated in the corresponding score, is still one of the most challenging subjects in the Music Information Retrieval (MIR) research field. This is owing to the large use of the beat information in many applications and to the complexity of the task: beats belong to a hierarchy/tree of rhythmic accentuations (hence entailing ambiguities), arise both from perceptual and cognitive cues. It, therefore, requires knowledge of the cultural specificities of the studied music.

To alleviate these issues, data-driven systems purely rely on training data composed of music tracks that have been annotated (supposedly) by experts. However, this labeling process remains costly and as a consequence, the amount of data annotated into beats (at most a few thousands tracks) remains extremely low in MIR, as compared to other research fields (speech or computer vision). For this reason, developing approaches that allow training beat-tracking systems without annotated data, a.k.a. Self-Supervised Learning (SSL), is important. This is the goal of this paper.

By alleviating the need of large annotated datasets, SSL, has recently gained significant attention in the field of machine learning. The goal is to learn meaningful representations of the input data without the need for human annotations. To do so, the target outputs are directly inferred from the dataset itself, and often referred to as "pretext-task labels". Such supervision can be obtained by masking some part of the input and asking the model to predict it [1–4] or to generate two views of the same input and force a model to learn similar representations for the two views [5–8]. Another popular SSL approach is contrastive learning [9, 10] where one trains a network to predict whether two inputs are from the same class (or not) by forcing their trained embeddings to be more or less close from each other. Usually, upon pre-training completion, the model is fine-tuned in a supervised fashion for one or more downstream tasks, where the data is smaller in size. Our contributions are the following:

- We propose a novel contrastive SSL scheme producing representations which are useful for automatic rhythm analysis tasks, in particular the beat-tracking task. Its key component is the pretext-task design exploiting Predominant Local Pulse (PLP) local maxima to effectively sample anchor, positive, and negative time-steps for our contrastive loss function.
- We show that the pre-trained model can be fine-tuned in a few-shot learning setting to get competitive beat-tracking results. Moreover, we show that our approach yields, in most cases, at least better performance than Zero-Note Samba (ZeroNS) [11], which is, to the best of our knowledge, the only alternative SSL approach to this problem to date.
- Furthermore we show that our model outperforms ZeroNS in a cross-dataset generalization setting.
- Finally we compare our model to the state of the art in a 8-fold cross-validation setting and show that it is competitive.

Paper organization. The paper is organized as follows. In section 2 we present works related to our proposal. In section 3 we present our proposed contrastive SSL training strategy. Finally in section 4 we present the results of the different experiments we performed. To facilitate reproducibility, we make our code available. [1]

[1] https://github.com/antoningagnere/ssl_beat

2 Related Work

In the following, we provide a quick overview of related contrastive SSL techniques and review the attempts made along this line in the field of MIR, especially for beat and downbeat tracking. We also discuss the recent advances made towards solving these important MIR tasks.

2.1 Self-Supervised Representation Learning

Our approach takes inspiration from contrastive methods. In CPC (Contrastive Predicting Coding) [9], representations are learned from sequential data by predicting the future latent representations from the (aggregated) past ones. For this, an encoder is trained to produce latent representations with the task of making it easy to distinguish in the obtained latent space (positive) future latent representations from a set of negative samples. This encourages the model to capture meaningful information. Instead of predicting the future, in Wav2Vec2 [10] the task is to predict masked observations. In Wav2Vec2, features are extracted from an audio signal with a Convolutional Network and fed to a transformer encoder where some frames are masked. Additionally, the audio features are quantized and the model is trained to contrast the masked output with the quantized output and a set of distractors.

2.2 Self-Supervised Learning in MIR

Following the trend in speech processing research, SSL approaches have started to become popular in MIR. On the one hand, these approaches can be used to train general-purpose models, the so-called "foundation models" (such as MULE [12] or MERT [13]), which are supposed to be useful to solve a whole set of downstream tasks (see the MARBLE benchmark [14]). On the other hand, models can be developed to learn representations that are well aligned with a specific MIR task. Among those, learning representations that are equivariant to a semantic distortion of the audio signal has become a popular approach (*e.g.*, for pitch or tempo estimation using siamese networks [15–18]).

Few works have proposed to apply SSL for rhythm analysis tasks. Zero-Note Samba (ZeroNS) [11] leverages the synchronization of the various instrument stems in a music track. For this, they separate music tracks into their percussive and non-percussive parts and train an encoder to force the synchronization between the corresponding latent representations, which are then used for beat tracking. In [19] they used binary metric regularity to derive supervision for their CRF loss, enabling the network to model a hierarchical metrical structure.

2.3 Beat and Downbeat tracking

Before the rise of deep-learning approaches, beat and downbeat tracking systems were based on two-step systems: first audio features were extracted from the audio signal (including an onset detection function, Predominant Local Pulse (PLP), spectral features or a novelty function); then those were used as "observations" to a probabilistic model (such as Hidden Markov Models or Dynamic Bayesian Network) [20–22].

The shift toward data-driven approaches started with [23] where the authors proposed to process spectral features with bi-directional Long Short-Term Memory (LSTM) networks. [24] then proposed to replace the LSTM with a Temporal Convolutional Network (TCN) to process the spectral features. Later on, the model was improved by solving jointly multiple tasks (beat and downbeat positions, as well as tempo) [25, 26]. Currently, models based on the Transformer architecture, used in a multi-task setting (joint beat-downbeat tracking) are the most successful. In [27] the authors apply the Spectral-Temporal Transformer (SpecTNT) architecture [28] to tackle this task. This architecture combines a spectral transformer that processes harmonic features and a temporal transformer that aggregates the processed features over time. To further improve the performance, the authors combined SpecTNT with a Temporal Convolutional Network (TCN). Beat Transformer [29] incorporates dilated self-attention to capture long-range dependencies. Furthermore, in the middle layers, they alternate time-wise dilated self-attention with instrument-wise self-attention [2] .

3 Proposed Contrastive Learning SSL scheme

In this paper, we propose a novel SSL approach to learn representations useful for rhythm analysis tasks, and instantiate it for the beat tracking downstream task. We aim to learn a projection (an encoder) such that the resulting projections of observations at PLP peaks whose distance from each other is a power of 2 are close to each other, and different otherwise. The two key insights behind this is that: i) a significant fraction of the PLP peaks (supposedly aligned to the tatum grid) is expected to represent beat positions, with high probability, and ii) most of the musical recordings tend to have a binary metric structure (i.e. beats can be musically divided by two and grouped by two). We conjecture that despite being over-simplistic, these ingredients are "good-enough" to define a pretext-task that will be effective for training representations useful for various rhythm analysis tasks, especially beat tracking, provided that a downstream fine-tuning phase is anyway envisaged. In the following we will refer to the distance between two PLP peaks as *tatum-unit* and denote it by tu.

We solve this pretext-task using contrastive learning. We learn to distinguish observations at times separated by an interval of a power of 2 in tu units, from those that are not. Once computed, the PLP function is used to select an anchor, its associated positive, and a set of negative samples. We further explain the procedure in section 3.1. We then train our encoder to attract the anchor and the positive while repelling the set of negatives in the latent space using a contrastive loss. We describe the architecture of

[2] The instrument-wise attention is conducted along the stems of a demixed audio signal, contributing to a comprehensive analysis of the audio data.

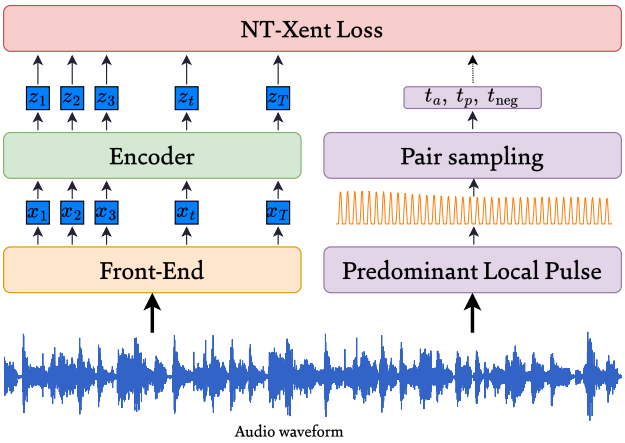

Figure 1: Our proposed contrastive SSL scheme for beat tracking. The left part displays our processed audio waveform to obtain the representations z_t. The right part displays our mining of positive and negatives.

our encoder in section 3.2. Our approach is summarized in Figure 1.

3.1 Mining positive and negatives

The key part of our work is to learn representations in a contrastive way. Therefore we need to define an anchor, a positive and multiple negative samples within each given audio excerpt. We rely on the Predominant Local Pulse (PLP) [30] function to extract local pulse information (see 3.1.1). Given such information, we sample positive and negative times for a selected anchor (see 3.1.2).

3.1.1 Predominant Local Pulse

The PLP method analyzes the Onset Strength Function (OSF) of an audio signal in the frequency domain to find a locally stable tempo for each frame. For this, a "tempogram" (a Short-Time Fourier-Transform, STFT) of the OSF is computed. At each time position, the maximum of the "tempogram" indicates the dominant pulse frequency. Using the corresponding amplitude and phase of this maximum, one can re-synthesize the corresponding temporal signal (a sinusoidal component). Using the usual overlap-and-add (OLA) inverse STFT method, a smooth temporal signal is formed by overlapping-and-adding the sinusoidal components with various dominant pulse frequencies over time. This temporal function is termed PLP and represents a localized enhancement of the original novelty function's periodicity.

Computation. Given an audio signal, we compute the PLP function with the same frame rate as our audio front-end (*i.e* 20ms). We used the *beat.plp* function from Librosa [31]. The function is fed with an OSF computed from a spectrogram [3] with 2048 points and the default minimum and maximum tempo parameters. We then estimate the local maxima peak y_k of the PLP using the *find_peaks* function from scipy.

[3] In a preliminary experiment, we found that using the spectrogram to get the OSF was working better than using the Mel-spectrogram

3.1.2 Sampling from PLP

In the following we will refer to the distance between two PLP peaks as *tatum-unit* and denote it by tu. For simplification, we do the following assumptions. We assume that the tatums correspond to the 8-th note and that most tracks are in a 4/4 meter. [4] Following this, we consider that the positives have a time distance Δ from the anchor which is a power of two of the tatum unit: $\Delta = i \times \alpha \times tu$ with $\alpha = 2^n$ and $i \in \mathbb{Z} \setminus \{0\}$. In this work we consider $n = 2$ (which corresponds to an inter-distance of two beats).

We define by $Y = \{y_1, ..., y_K\}$ the set of PLP peaks within a given audio segment. We first sample an anchor a uniformly in $[1, K]$. We denote by y_a the time associated to a (blue arrow in Fig.2). Given this anchor, we sample its associated positive time step p. This positive must be situated $i \times \alpha$ peaks away from the query. For a given anchor a, we therefore sample p uniformly from $Y_a = \{y_{a \pm i \times \alpha}, 0 \leq a \pm i \times \alpha \leq K\}$.

We denote by y_p the time associated to p (green arrow in Fig.2, green empty arrows are all the elements of Y_a).

We then sample N negative time steps at which we define hard negative and easy negative examples. An *easy negative* corresponds to a time step that is not a PLP peak. They are sampled uniformly in $[0, T] \setminus Y$. We also apply a "safety window" (whose duration was empirically determined to one frame) around peak time steps to avoid sampling negatives that are too close to a peak. A *hard negative* corresponds to a time step that is a peak but that is not in Y_a. They are sampled uniformly in $Y \setminus (Y_a \cup \{y_a\})$. We sample $N = 10$ negatives, half of them are hard negatives, and the other half are easy negatives.

To prevent any errors coming from the PLP function we discard audio segments where the inter-peak distance is not almost constant. We empirically set the allowed variation to 20 percent of this inter-peak distance within a segment (more details about this are given on the companion website).

3.2 Architectures

Front-end. We compute Mel spectrogram features from audio sampled at 16kHz using 128 bands, a window size of 2048 samples, and a hop size of 320 samples (20ms frame rate). We apply log compression and normalization [5]. Subsequently, a linear layer projects the frames to the embedding dimension. The resulting sequence x_t serves as the input to the encoder. We use audio segments of 20s long to ensure the model sees a sufficiently large context. However, we did not explore varying the length of audio segments fed into the encoder.

Encoder. For the encoder, we use a Transformer architecture similar to the one used in Wav2Vec2 or Hubert [10, 32]. It is composed of a stack of Transformer

[4] In a preliminary experiment, we delve deeper into determining the metrical level that the peaks of the PLP correspond to. Our findings suggest that these peaks align with either the beat, the 8-th note or the 16-th note level.

[5] For normalization, we use the mean and standard deviation computed over the training set.

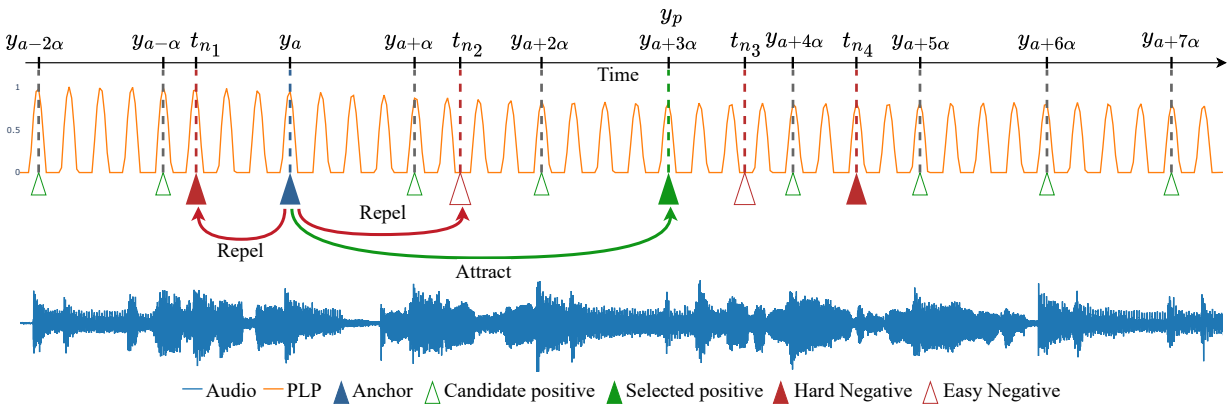

Figure 2: Proposed mining strategy of Positives and Negatives (easy and hard) given an Anchor time in the PLP function. Positive are sampled among peaks of the PLP whose time index is distant from the Anchor by a power of two tatum units tu (here $\alpha = 4 \times tu$); Negatives are the remaining times and are considered Easy if not peaks of the PLP and Hard if peaks of the PLP. Here we sample two hard and two easy negatives.

encoder layers. Each layer is composed of a multi-head self-attention mechanism followed by a feed-forward network. We use 8 layers each of which has 8 attention heads and apply a 0.1 dropout in the attention layer. The encoder outputs the embedding sequence z_t. The embedding dimension is set to 512 and the hidden dimension of the feed-forward network is set to 1024. In total, the model has 19.1M learnable parameters. We did not explore other architectures as the focus of our work was to study the proposed SSL scheme.

3.3 Contrastive Loss

Among the various formulations of the contrastive losses, we have chosen to use the NT-Xent loss [5] one. We define the similarity measure between two vectors u and v as $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$. Given an anchor y_a, a positive y_p and a set of N negatives time-steps $t_{\text{neg}} = \{t_{n_1}, ..., t_{n_N}\}$, we compute the contrastive loss as follows:

$$\mathcal{L}_{\text{NT-Xent}}(y_a, y_p, t_{\text{neg}}) = -\log \frac{\exp(sim(z_{y_a}, z_{y_p})/\tau)}{\sum_{i=1}^{N} \exp(sim(z_{y_a}, z_{t_{n_i}})/\tau)}. \quad (1)$$

We set the temperature to $\tau = 0.1$. For each audio in a batch, we use 80% of the available peaks as anchors. For each of them, we sample their corresponding positive and negatives. We compute the above contrastive loss over each pair and each audio. We then average the losses to obtain the global loss for the batch, that is if we have a total of M pairs in the batch:

$$\mathcal{L} = \frac{1}{M} \sum_{y_a, y_p, t_{\text{neg}}} \mathcal{L}_{\text{NT-Xent}}(y_a, y_p, t_{\text{neg}}). \quad (2)$$

4 Evaluation

To evaluate our model, we performed three experiments. In all three experiments, the model is pre-trained in a SSL way using unlabeled data.

In Experiment 1, we test the Few-Shot Learning (FSL) abilities of our model using only a few data for fine-tuning. Experiment 2 tests the generalization of our model on unseen conditions and serves as comparison to ZeroNS. Finally, Experiment 3 compares our performance to the ones obtained using fully-supervised beat-tracking models.

4.1 Datasets

For SSL pre-training, we use a combination of unlabeled datasets (in terms of beat positions): the Free Music Archive (FMA) [33], MTG-Jamendo [34], and MagnaTagaTune (MTT) [35]. FMA contains 106,574 full tracks spanning 161 genres. MTG-Jamendo contains around 55,000 full audio tracks. Finally, MTT contains approximately 26,000 excerpts of 30-s duration from 5223 unique tracks. Overall the combined datasets offer around 165k full audio tracks and a total of 8,000 hours.

For fine-tuning and testing, we used the following labeled (into beats) datasets, commonly used in previous works: SMC [36], Ballroom [37] and Hainsworth [38], GTZAN [39,40], RWC [41] and Harmonix [42]. The Harmonix dataset is mainly composed of pop music tracks, whereas the Ballroom, GTZAN, RWC, and Hainsworth datasets offer a wider variety of musical genres.

4.2 Evaluation Metrics

We report the commonly used metrics in the literature including the F-measure with a tolerance window of $\pm$ 70ms, continuity-based measures at the correct metrical level (CMLt & CMLc), and at alternate metrical levels such as double/half and offbeat (AMLt & AMLc) [43].

4.3 Implementation details

4.3.1 Pre-training

For SSL pre-training we kept 0.05% of the data for validation (9,000 tracks). Our model is pre-trained during 200 epochs (equivalent to around 270,000 steps). Training was conducted on 4 A100 GPUs utilizing float 16 precision and a global batch size of 96. We employed the Adam optimizer [44] with an initial learning rate set at 1e-4 and

applied a polynomial decay learning rate scheduler. The learning rate gradually increased to 5e-4 within the first 32,000 steps, then reverted to its initial value over the subsequent 250k steps. Additionally, gradient clipping was employed. We keep the model that gives the best validation loss.

4.3.2 Fine-tuning

After SSL pre-training, we need to adapt the model to the downstream task of beat tracking. This is done by adding a linear classification probe $g(.)$ and fine-tuning both the encoder and the linear probe. $g(.)$ projects the embedding into the scalar beat activation function. Instead of feeding $g(.)$ with the output of the encoder, we feed it with a weighted sum of the outputs of each layer of the Transformer [45]. That is $z = \sum_{l=1}^{8} \alpha_l z^{(l)}$, where $z^{(l)}$ is the output of layer l. The weights α_l are jointly learned with the linear probe $g(.)$.

The system is trained to minimize the binary cross-entropy loss between the beat activations and the target. Following the literature we widened the beat targets by a window [0.25, 0.5, 1, 0.5,0.25] [26]. We used the Adam optimizer [44] with an initial learning rate of 1e-5 and a polynomial decay learning rate scheduler.

During fine-tuning, we utilized audio chunks of sizes similar to those used during pre-training (20s). However, during inference, to avoid potential out-of-memory errors, we split audio excerpts exceeding 45 seconds into 20-second chunks with 5-second overlap. Subsequently, we overlap-add the activations to derive the beat activations for the whole track. These beat activations are then fed into a Dynamic Bayesian Network (DBN) [46] to predict the beat positions. The DBN is configured to model a tempo range of 40-270 beats per minute with transition lambda set to 45, observation lambda to 9, and a threshold of 0.15.

4.3.3 Data Augmentation

We found that both pre-training and fine-tuning could benefit from data augmentation, in particular time-stretching. We apply time-stretching in two manners: constant factor and time-varying factor. In both cases, we constrain the time-stretching factor to lie in the interval $[0.8, 1.2]$. For the constant factor case, we used sox effects in Torchaudio [47], and for time-varying factor we used Libtsm [48]. When using a time-varying factor we randomly sample time instants at which the stretching factor is modified (also randomly, see the repository for details). This was found to be particularly beneficial for the pre-training stage. Indeed because we have filtered out tracks where the inter-peak distance is not almost constant, the SSL training data does not contain examples of time-varying tempo. Using time-varying time-stretching allows us to simulate this in a controlled fashion.

We found that it was better to compute the Predominant Local Pulse (PLP) curve before time-stretching and shift the peaks accordingly, rather than on the time-stretched audio.

4.4 Experiment 1: Few-shot learning

Protocol. The goal here is to test the ability of our model to learn with only few examples, Few-Shot Learning (FSL). To be able to compare our results with previously published ones, we replicate the evaluation protocol proposed in ZeroNS [11]. We consider *individually* each dataset (both for fine-tuning and testing): $T \in \{\text{SMC, Ballroom, Hainsworth, GTZAN}\}$. For each dataset T, we split it into 8 folds, we use one for testing T_{test}, one for validation T_{valid} and perform FSL with the remaining ones T_{train}. The FSL ability is evaluated by selecting randomly $k \in \{1, 2, 3, 4, 6, 8, 12, 16, 24, 48, 64, 96\}$ items from T_{train}. For each k we sample 10 variations: $T^k_{train,i}$ For each choice of k, we fine-tune our pre-trained model on each variation $T^k_{train,i}$ and keep the one that performs the best on T_{valid}

Results. We give the results in Figure 3 for the T_{test} of each dataset (SMC Mirex, Ballroom, Hainsworth, and GTZAN) and each value of k (x-axis). We report the mean and standard deviation of the metrics over the training set variations $T^k_{train,i}$. Our model performs at least as well as ZeroNS on almost all metrics and datasets. The exceptions are with AMLt on SMC and AMLc and AMLt on GTZAN and SMC. We observe that our model performs significantly better on Hainsworth, with up to 10% absolute improvement in F1 score and almost 20% absolute improvement in CMLt and CMLc. Also, the performance gap is significant on Ballroom when using very few data (less than 10 tracks) where we can observe almost 10% absolute improvement in F1 score and up to 15% improvement in AMLc.

4.5 Experiment 2: Generalization

Protocol. The goal here is to test the generalization ability of our model, *i.e.* training our model on one dataset and testing on another. For this, we replicate the protocol proposed in ZeroNS [11]. For each choice of dataset $T \in \{\text{SMC, Hainsworth, Ballroom}\}$, we split it into 8 folds, we use one for validation T_{valid}, and the remaining seven for training T_{train}. We then use the best-performing model on T_{valid}. Instead of using the linear probe described above, we obtained better results using a MLP (two linear layers interleaved with a ReLU), also fed by the weighted sum of layer sequences (sec 4.3.2). Whatever the choice of T, the test is performed on the GTZAN dataset.

Trained on	Method	F1 (%)	AMLt (%)	CMLt (%)
SMC	Ours	$\mathbf{79.5 \pm 0.5}$	$\mathbf{88.0 \pm 0.6}$	$\mathbf{64.4 \pm 0.9}$
	ZeroNS	74.8 ± 2.1	86.3 ± 2.3	51.0 ± 2.1
Hainsworth	Ours	$\mathbf{85.1 \pm 0.8}$	$\mathbf{89.9 \pm 0.9}$	$\mathbf{73.2 \pm 1.8}$
	ZeroNS	80.6 ± 0.9	89.4 ± 0.7	62.8 ± 2.3
Ballroom	Ours	$\mathbf{83.9 \pm 0.3}$	88.4 ± 0.5	$\mathbf{72.3 \pm 0.9}$
	ZeroNS	82.6 ± 0.5	$\mathbf{89.0 \pm 0.8}$	67.6 ± 1.1

Table 1: Results of Experiment 2: Generalization

Results. We indicate the results in Table. 1. We report the mean and standard deviation of F1, AMLt, and CMLt

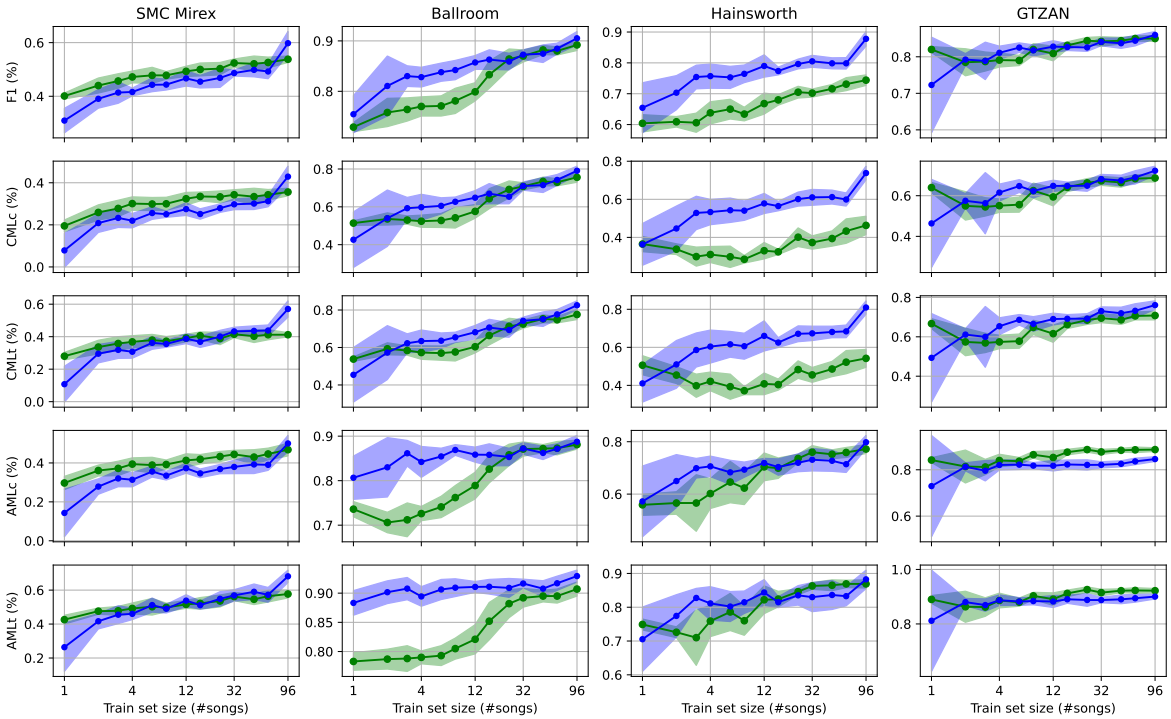

Figure 3: Results of Experiment 1: Few-Shot Learning. Shaded areas representation the standard deviation. (ZeroNS in green and our method in blue)

Method	F1	CMLt	AMLt
Böck [26]	0.885	**0.813**	**0.931**
Hung [27]	**0.887**	0.812	0.920
Zhao [29]	0.885	0.800	0.922
Ours	0.876	0.802	0.918

Table 2: Results of Experiment 3: Comparison with supervised baseline

scores across the different folds. Overall our model performs better than ZeroNS on all datasets except when for the AMLt metric when trained with Ballroom, but the difference is not statistically significant. This means that our model can generalize well to unseen data. Precisely we observe a 5% improvement in F1 score and more than 10% improvement in CMLt when training on SMC or Hainsworth. We nearly reach the F1 score of fully supervised models (presented next) when training solely on $7/8$ of Hainsworth (*i.e* 194 tracks).

4.6 Experiment 3: Comparison with supervised baseline

Protocol. The goal here is to compare the performance of our model to the ones provided by fully-supervised models. For this we replicate the commonly used 8-fold cross validation set-up after [26,27,29]. GTZAN is kept as a test set and is never seen in training. We average the metrics over the 8 training folds to obtain the final results.

Results. We give the results in Table 2. It is clear that the proposed beat tracking approach using our self-supervised pre-training can be competitive with state-of-the-art methods on GTZAN, a dataset covering a wide diversity of genres. While our method does not outperform the best-performing method, it achieves comparable results across all metrics, proving the quality of the learned representations.

5 Conclusion

In this paper, we proposed a novel Self-Supervised Learning approach to learn representations useful for the task of beat tracking using contrastive learning where the selection of anchor, positive and negative peaks derives from a Predominant Local Pulse function.

We assess our proposal positively based on a series of experiments. In a first experiment, we showed that our proposed approach was superior on some datasets to the previous SSL approach, ZeroNS, in a few-shot learning setting. In a second experiment, we show that our model has better generalization capabilities to unseen data. In the last experiment, we show that our model also yields comparable performances to the fully supervised baseline, indicating that our pre-training scheme effectively learns meaningful beat-related representations.

To further improve our method, future work will focus on developing a more sophisticated sampling mechanism that can handle other metrical structures than the binary one used-here (such as 6/8, 3/4). One potential approach is to incorporate additional audio features, such as self-similarity matrices, to gain a deeper understanding of the rhythmic structure within an audio segment and adaptively select positive positions for a given anchor.

6 Acknowledgements

This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013924R1 made by GENCI. The material contained in this document is based upon work funded by the ANR-IA and Hi! PARIS.

7 References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[2] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked autoencoders are scalable vision learners," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 06 2022, pp. 15 979–15 988.

[3] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *International Conference on Learning Representations*, 2022.

[4] P.-Y. Huang, H. Xu, J. B. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," in *Advances in Neural Information Processing Systems*, 2022.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 1597–1607.

[6] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 271–21 284.

[7] X. Chen and K. He, "Exploring simple siamese representation learning," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, 2021, pp. 15 750–15 758.

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019.

[9] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[10] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, 2020.

[11] D. Desblancs, V. Lostanlen, and R. Hennequin, "Zero-note samba: Self-supervised beat tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[12] M. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. Ehmann, "Supervised and unsupervised learning of audio representations for music understanding," 10 2022.

[13] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C. Lin, A. Ragni, E. Benetos, N. Gyenge, R. B. Dannenberg, R. Liu, W. Chen, G. Xia, Y. Shi, W. Huang, Y. Guo, and J. Fu, "MERT: acoustic music understanding model with large-scale self-supervised training," *CoRR*, vol. abs/2306.00107, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2306.00107

[14] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. B. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, "MARBLE: music audio representation benchmark for universal evaluation," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023.

[15] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirovic, "Spice: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, p. 1118–1128, 2020.

[16] E. Quinton, "Equivariant self-supervision for musical tempo estimation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, 2022*, 2022.

[17] A. Riou, S. Lattner, G. Hadjeres, and G. Peeters, "Pesto: Pitch estimation with self-supervised transposition-equivariant objective," 2023.

[18] A. Gagneré, S. Essid, and G. Peeters, "Adapting pitch-based self supervised learning models for tempo estimation," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[19] J. Jiang and G. Xia, "Self-supervised hierarchical metrical structure modeling," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[20] D. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, pp. 51–60, 03 2007.

[21] G. Peeters, "Beat-marker location using a probabilistic framework and linear discriminant analysis," 09 2009.

[22] M. Alonso, B. David, and G. Richard, "Tempo and beat estimation of musical signals," 10 2004.

[23] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proceedings of the 14th International Conference on Digital Audio Effects, DAFx 2011*, 09 2011.

[24] E. P. MatthewDavies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[25] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *International Society for Music Information Retrieval Conference*, 2019.

[26] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *International Society for Music Information Retrieval Conference*, 2020.

[27] Y.-N. Hung, J.-C. Wang, X. Song, W.-T. Lu, and M. Won, "Modeling beats and downbeats with a time-frequency transformer," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[28] W. Lu, J.-C. Wang, M. Won, K. Choi, and X. Song, "Spectnt: a time-frequency transformer for music audio," in *International Society for Music Information Retrieval Conference*, 2021.

[29] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022.

[30] P. Grosche and M. Muller, "Extracting predominant local pulse information from music recordings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1688–1701, 2011.

[31] 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.

[32] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, p. 3451–3460, oct 2021.

[33] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[34] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," in *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning (ICML 2019)*, 2019.

[35] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *International Society for Music Information Retrieval Conference*, 2009.

[36] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[37] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1832–1844, 2006.

[38] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 15, pp. 2385–2395, 2004.

[39] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[40] U. Marchand and G. Peeters, "Swing Ratio Estimation," in *Digital Audio Effects 2015 (Dafx15)*, Trondheim, Norway, Nov. 2015.

[41] M. Goto, H. Hashiguchi, T. Nishimura, and R. ichi Oka, "Rwc music database: Popular, classical and jazz music databases," in *International Society for Music Information Retrieval Conference*, 2002.

[42] O. Nieto, M. McCallum, M. Davies, A. Robertson, A. Stark, and E. Egozy, "The Harmonix Set: Beats, Downbeats, and Functional Segment Annotations of Western Popular Music," in *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 2019, pp. 565–572.

[43] M. Davies, N. Degara Quintela, and M. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," 2009.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[45] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T. hsien Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe,

A. rahman Mohamed, and H. yi Lee, "Superb: Speech processing universal performance benchmark," in *Interspeech*, 2021.

[46] F. Krebs, S. Böck, and G. Widmer, "An Efficient State-Space Model for Joint Tempo and Meter Tracking." in *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 2018.

[47] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, Y. Tao, R. Scheibler, S. Cornell, S. Kim, and S. Petridis, "Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch," 2023.

[48] J. Driedger and M. Müller, "TSM Toolbox: MATLAB implementations of time-scale modification algorithms," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Erlangen, Germany, 2014, pp. 249–256.