A SEMI-SUPERVISED LEARNING APPROACH FOR BEAT TRACKING

Jakob Abeßer[1] *Christon-Ragavan Nadar*[1] *Sascha Grollmisch*[1,2]

[1]Semantic Music Technologies Group, Fraunhofer IDMT, Ilmenau, Germany
[2]Institute of Media Technology, Technische Universität Ilmenau, Ilmenau, Germany

ABSTRACT

In this paper, we propose a semi-supervised learning approach for Deep Neural Network (DNN) based beat tracking. In addition to a commonly used supervised loss function, we introduce two additional loss terms to enforce sparsity and periodicity of the predicted beat salience function. Our results show that by adding a sparsity-based loss term, spurious peaks in the predicted salience function located between the true beat positions can be reduced significantly. Such peaks can potentially lead to erroneous tempo estimates in a subsequent rhythm inference step. We compare both Recurrent Neural Network (RNN) and Temporal Convolutional Network (TCN) models and show that while both perform comparably well, the TCN model can be trained much faster.

Index Terms— beat tracking, recurrent neural networks, temporal convolutional networks, semi-supervised learning

1. INTRODUCTION

The human perception of periodic rhythmic accents (pulses) as beats is an overarching characteristic of many musical cultures and styles. Algorithms to detect beat positions in music recordings commonly involve *two processing steps* as shown in Figure 1. First, an audio signal or a time-frequency representation thereof is mapped towards a salience function, which measures the local beat likelihood. Then, a probabilistic model is used to infer the beat time positions as well as related rhythmic properties such as the downbeat position and the meter. Beat tracking algorithms nowadays mostly rely on deep neural networks (DNN). In this paper, we solely focus on improving the beat salience modeling.

Training beat tracking algorithms to perform well regardless of the musical style remains challenging for various reasons. First, the human concept of *beat perception* differs across music styles and is often more complex due to *microtemporal deviation* in the music performance. Secondly, in contrast to popular music recordings, beats in other music genres do not necessarily coincide with *strong percussive accents*. Taking Jazz as an example, the exact beat positions are often *ambiguous* and can be allocated to either the ride cymbal hits or the onset of accompanying instruments such as the bass or the piano. As a third reason, existing audio datasets with beat annotations such as the Ballroom dataset, the Real Music Computing (RWC) database, or the Beatles dataset *mostly cover popular music* styles. Datasets including other styles exist but only to a limited extent, which hinders the progress in developing supervised DNN-based beat tracking approaches.

To compensate for the *lack of training data*, different strategies can be pursued. *Data augmentation* allows to multiply the available amount of data by applying basic audio manipulations such as time stretching and pitch shifting. *Transfer learning* allows to adapt existing neural networks to analyze audio data from a previously unseen music style. *Semi-supervised learning* strategies can be used to leverage both training data with explicit beat-level annotations as well as unlabeled data, which is usually available in abundance.

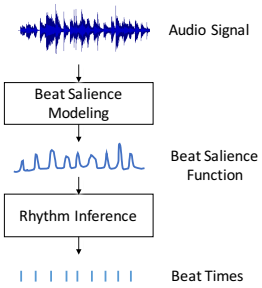


Fig. 1: Common two-stage approach for beat tracking systems including beat salience modeling and rhythm inference.

As the main contribution of this paper, we propose a semi-supervised learning approach for DNN-based beat salience modeling. We introduce two additional unsupervised loss terms in addition to a commonly used supervised loss term. These loss terms do not require any beat-level annotation and enforce the predicted beat likelihood function to be *sparse* and *periodic*, which are two commonly known characteristics of beat sequences.

2. RELATED WORK

Traditional methods for automatic beat tracking are based on combining audio signal processing and machine learning methods [1]. Nowadays, state-of-the-art algorithms rely on deep neural network architectures, which are trained in a purely data-driven manner. Audio signals are commonly transformed into time-frequency representations such as Short-time Fourier Transform (STFT) [2, 3], Mel Spectrogram [4], Constant-Q spectrogram [5], or Chromagram [6, 7] before being input to the neural network. Since beats often coincide with transient signals, the first order time derivatives of such spectrograms are used as additional input channels to the networks [2, 4]. Various neural networks architectures were applied for this task ranging from Convolutional Neural Networks (CNN) [5, 7] over Recurrent Neural Networks (RNN) [8, 6, 4, 3] to combined Convolutional Recurrent Neural Networks (CRNN) [9]. Similar to [10], we evaluated the proposed unsupervised loss functions on both an RNN as well as a Temporal Convolutional Networks (TCN), which will both be detailed in Section 3.2. As a post-processing step, probabilistic models such as Dynamic Bayesian Networks (DBN) [11, 10] are commonly used to infer the beat times from the beat salience function. Given a set of predicted beat time estimates, metrics for

This work has been supported by the German Research Foundation (AB 675/2-1, BR 1333/20-1).

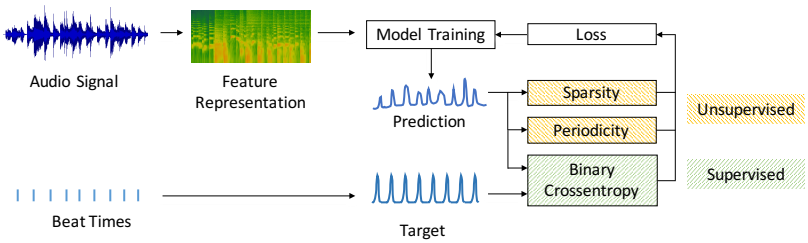


Fig. 2: Summary of proposed semi-supervised training method. In addition to the binary crossentropy, two additional (unsupervised) loss terms enforce sparsity and periodicity in the predicted beat salience function.

time instance evaluation such as precision, recall, and f-measure are commonly used.

Semi-supervised learning approaches have been successfully applied for different Music Information Retrieval (MIR) tasks such as musical instrument recognition [12], genre classification [13], and music transcription [14] to improve upon purely supervised approaches. In [15], unsupervised and supervised triplet loss functions have been combined to a semi-supervised version which increased the quality of the embeddings learned for the task of audio tagging. However, none of these approaches used semi-supervised loss functions for training DNN models as will be presented in this paper.

3. PROPOSED METHOD

3.1. Input Features & Targets

We process audio signals at a sample rate of 44.1 kHz and convert stereo to mono signals. As input feature $X \in \mathbb{R}^{N_T \times N_F}$ to the deep neural networks, we compute the mel spectrogram using the `librosa` python package[1] with a window size of 1024, a hopsize of 441, and an FFT size of 1024. N_T and N_F denote the number of time frames and frequency bins, respectively. X is normalized to zero-mean and unit-variance before processed to the neural network on a file level. Temporal down-sampling is avoided to assure that the predicted beat salience function $\hat{y} \in \mathbb{R}^{N_T}$ has the same temporal resolution as the input feature X.

Given the annotated beat times in seconds, we set the elements in the corresponding target vector $y \in \mathbb{R}^{N_T}$, which are closest to the true beat times, to 1. Furthermore, we introduce a temporal blurring to the target vector and apply a one-dimensional Gaussian filter with standard deviation of kernel: $\sigma = 2$ to this signal as inspired by [4]. By applying this temporal blurring step, y is continuously distributed between 0 an 1, which facilitates the training of the final sigmoid layer in the networks (cf. Section 3.2).

3.2. Neural Network Architectures

In this section, we briefly describe the Recurrent Neural Network (RNN) and Temporal Convolutional Network (TCN) architectures, which we compare for the beat salience modeling.

3.2.1. Recurrent Neural Network

As a baseline system, we adopt the RNN architecture from the state-of-the-art beat tracking methods proposed by [11, 6]. It consists of three layers of bidirectional gated recurrent units (BiGRU) with 25 units per layer and a *tanh* activation function. The last layer is a time-distributed dense layer with a sigmoid activation function which outputs the frame-level beat salience estimates $\hat{y}$.

3.2.2. Temporal Convolutional Network

TCNs have lately been lately used for sequence modeling tasks as an alternative to RNN variants such as Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU). Also, TCNs were successfully applied in generative models for waveform-based audio synthesis as WaveNet [16] or for machine translation [17]. We use three TCN layers with 128, 32, and 12 filters per layer and kernel sizes of 3, 5, and 3 respectively. For regularization, we use dropout with ratios of 0.1, 0.2, and 0.1 per layer. We also use dilation factors of 2^i with $i \in \{0, ..., 5\}$.

3.3. Loss Functions

In this section, we describe three loss functions which are used for training the neural networks. In addition to the commonly used supervised binary crossentropy loss L_B, we propose two unsupervised loss terms L_S and L_P based on the notions of sparsity and periodicity, respectively. Both loss terms do not require any target function y. We evaluate three configurations: a fully supervised loss function $L_1 = L_B$ as well as two semi-supervised loss functions $L_2 = L_B + \alpha_S \cdot L_S$ and $L_3 = L_B + \alpha_P \cdot L_P$. Section 5 provides detailed results for the optimization of the mixing coefficient α_S. Due to high computational costs, we could not systematically optimize α_P for L_3, but found $\alpha_P = 1$ to perform best based on a small hold-out validation set.

3.3.1. Binary Crossentropy

We interpret the predicted beat salience $\hat{y} \in \mathbb{R}^{N_T}$ as probability values and compute the binary crossentropy loss

$$L_B = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}). \quad (1)$$

This loss function enforces $\hat{y}$ to be as similar as possible to y in a supervised fashion.

3.3.2. Sparsity

Our first assumption is that beat salience functions have a sparse nature, i. e., they have only few non-zero values (cf. Figure 2). Therefore, we aim to enforce the sparsity of the predicted beat salience function $\hat{y}$ by adding an additional loss term. First, we segment $\hat{y}$ into multiple subsegments $\hat{y}_i$ using a window size of 5 s and a hopsize of 2.5 s. Given this window length, we ensure that multiple peaks of the beat salience function are captured by a single analysis

[1] `https://librosa.github.io/librosa`, version 0.7.0

window independent of the global tempo. Then, we compute a loss term L_S as a global non-sparsity measure over $\hat{y}$ by averaging over all N_S subsegments:

$$L_S = \frac{1}{N_S} \sum_{i=1}^{N_S} 1 - \frac{\sqrt{N_T} - (||\hat{y}_i||_1)/(||\hat{y}_i||_2)}{\sqrt{N_T} - 1} \quad (2)$$

This non-sparsity measure was proposed in [18] and measures the relationship of the l^1 norm and the l^2 norm.

3.3.3. Periodicity

Our second assumption is that the tempo of the analyzed music recordings remains constant over time or changes only slowly. Therefore, we measure the periodicity of the estimated beat salience function $\hat{y}$ as follows: First, we generate a reference beat salience function y_{ref} from regular beat pulses at a constant tempo of 120 bpm. Then, we align both sequences via Dynamic Time Warping (DTW) [1] and derive the cost matrix $C \in \mathbb{R}^{|\hat{y}| \times |y_{ref}|}$. We obtain an upper bound c_{max} for the alignment costs by traversing C along the longest possible path and accumulating the respective cost values. Finally, we derive the loss term L_P by normalizing c as

$$L_P = c/c_{max}. \quad (3)$$

Independent of the true tempo, we expect the optimal alignment path in C to be almost diagonal and the resulting loss value to be low if $\hat{y}$ also has a periodic structure. In contrast, non-periodic predictions $\hat{y}$ lead to a higher alignment cost and a higher loss value.

4. DATASET

As shown in Table 1, we train multiple models using three public beat tracking datasets. In addition to the Beatles dataset[2] [19], we used the Ballroom dataset [20], which includes eight ballroom dance music genres, as well as the Popular, Jazz, and Classic Music Databases (5 albums) taken from the Real World Computing (RWC) dataset [21]. We did not perform any type of data augmentation. The validation set was randomly selected as 20% of the training dataset.

The test set was selected in a way such that we can study the performance of the proposed loss function across various music genres. The Robbie Williams dataset [22] mostly contains pop and rock recordings with a clearly defined beat structure. However, we could only use a subset of 26 files due to the limited availability of the original artist recordings. The Sound and Music Computing (SMC) dataset [23] includes recordings from a diverse set of music genres and is therefore considered to be very challenging for beat tracking [4, 11]. Finally, the Weimar Jazz Database (WJD) [24] includes solo sections from 456 jazz recordings covering a wide range of jazz styles and instrumentations. This dataset is also very challenging due to microtemporal deviations (often referred to as swing ratio) as well as ambiguities of the perceived beat times as discussed in Section 1.

5. EVALUATION

5.1. Metrics

Initially, we derive the optimal alignment between $\hat{y}$ and y from the lag of the maximum of the cross-correlation between both. We allow an absolute tolerance of 70 ms inspired by the commonly used tolerance for evaluating beat time estimates [19, 6, 25]. Zero-padding is applied if required.

As first evaluation metric, we measure the similarity between the aligned $\hat{y}$ and y using the Pearson correlation coefficient r. For high correlation values, we expect the peak structures of both functions to resemble well and a subsequent rhythm inference to yield good predictions of the beat times. As second metric, we compute the "within-beat energy ratio"

$$\gamma = \frac{||\hat{y} \cdot y||}{||\hat{y}||}. \quad (4)$$

By filtering $\hat{y}$ with the true salience function y, we estimate the salience ratio close to the true best times. Lower values of γ indicate the tendency of a model to predict beat salience "between" the true beats. Such spurious peaks can misguide a beat tracking algorithm since the peak structure indicates wrong tempo values.

Table 1: Overview of all training & test datasets.

Dataset		# Files	Duration (h)
Training Dataset			
`Bs`	Beatles	127	5.8
`Bl`	Ballroom	698	6.1
`RWC`	RWC	328	23.6
Test Dataset			
`RW`	Robbie Williams	26	2.1
`SMC`	SMC	217	2.4
`WJD`	WJD	456	13.3

5.2. Experimental Procedure

We use the following general procedure for all training procedures: We split the training datasets into training and validation sets using a 80% - 20% split on a file-level. Each model is trained for a maximum of 100 epochs using early stopping with a patience of 10 epochs. We use the Adam optimizer [26] with a learning rate of 10^{-4} for the RNN model and 10^{-5} for the TCN model. For both loss functions L_1 and L_2, we process full files (batch size of 1) during training. To lower computational demands, we only update the model parameters every 32 batches for the loss function L_3.

5.3. Influence of Sparsity-Based Loss

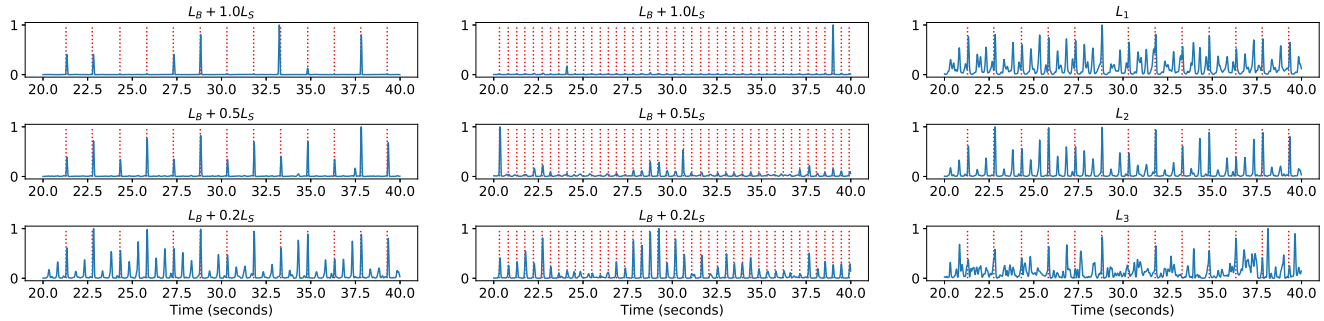
For the loss function L_2, we systematically optimized the mixing ratio α_S between the supervised binary crossentropy loss term L_B and the sparsity-based loss term L_S. Table 2 summarizes the evaluation measures for the three test datasets `SMC`, `RW`, and `WJD` for $\alpha_S \in \{0.2, 0.5, 1, 2\}$. One can observe that $\alpha_S = 0.2$ results in the best evaluation scores with the binary crossentropy loss L_B mostly dominating the total loss by factor 5.

To get further insights into the influence of α_S, we show in Figure 3 the beat salience predictions $\hat{y}$ for three different values $\{0.2, 0.5, 1\}$ for excerpts from a pop music recording with clearly defined drum beat taken from the `SMC` dataset[3] (left subplot) and an excerpt from a swing jazz solo[4] with more complex rhythmic accents from the `WJD` dataset (middle subplot). For the rhythmically

[2] `http://isophonics.net/datasets`

[3] SMC 275.wav includes a drum set (kick drum, snare drum, and hi-hat), female vocals, and electric piano.

[4] "Walkin'" by Miles Davis comprises of a drum set (snare drum and ride cymbal), trumpet, upright bass, and piano.



(a) Influence of α_S of L_2 for a pop song. (b) Influence of α_S of L_2 for a jazz song. (c) Influence of loss function for a jazz song.

Fig. 3: Beat salience predictions $\hat{y}$ of TCN models for 20 second segments taken from two music recordings as described in Section 5.3. The dashed red line represents the true beat times y and blue function represents the predicted $\hat{y}$.

Table 2: Evaluation results for TCN model using the loss function L_2 given different mixing ratios α_S (see Section 3.3). Best values are denoted in bold print.

	r				γ			
α_S	2	1	0.5	0.2	2	1	0.5	0.2
SMC	0.03	0.04	0.07	**0.15**	0.17	0.17	**0.16**	0.14
RW	0.05	0.06	0.29	**0.58**	0.29	0.34	**0.54**	0.50
WJD	0.03	0.04	0.10	**0.29**	0.32	0.32	0.34	**0.37**

more simple pop song, $\alpha_S = 0.5$ shows the best predictions with the main peaks covering all true beat times. In contrast, a smaller value of $\alpha_S = 0.2$ gives the best results for the jazz song. In general, we observe that for higher values of α_S, the sparsity-based loss overshadows the supervised loss and the beat salience predictions shows only a few non-zero peaks.

5.4. Loss Function Configuration

In this experiment, we wanted to compare the influence of the individual loss terms on the predictive performance of the trained RNN and TCN models. Table 3 summarizes the results for the evaluation measures r and γ. First, for all models, the correlation coefficient r is highest for the purely supervised training using the loss function L_1. At the same time, the loss function L_2 gives the highest γ values for most configurations. Therefore, it is possible to reduce spurious peaks by additionally enforcing the sparsity of the predicted beat salience function. However, at the same time, the peak heights at the correct beat times show a larger variance (cf. right subplot of Figure 3), which likely causes the slightly smaller r values compared to L_1. As confirmed by the results discussed in Section 5.3, the optimal mixing ratio α_S depends on the instrumentation and must therefore be determined for each testset individually. Adding the periodicity-based loss term L_P in loss function L_3 did not improve the quality of the beat salience curve.

When comparing the performance between both network architectures, we observe a similar performance between RNN and TCN models with a much faster training time for the TCN models as also found in [10]. We observed that the TCN model outputs fewer spurious peaks but with higher confidence levels. These peaks often coincide with transient hits from the snare drum or hi-hat, which do not fall on the beat times. In general, the SMC dataset achieves the lowest scores followed by the WJD dataset, which confirms that both

Table 3: Evaluation results for salience functions predicted using the RNN and TCN models based on the three loss functions L_1, L_2, and L_3 (see Section 3.3). Results are displayed separately for each test set. Best values are denoted in bold print.

Model	**Testset**	r			γ		
		L_1	L_2	L_3	L_1	L_2	L_3
RNN	SMC	**0.23**	**0.23**	0.20	0.11	**0.18**	0.09
	RW	**0.62**	0.61	0.52	0.26	**0.29**	0.19
	WJD	**0.45**	0.31	0.37	**0.30**	0.24	0.25
TCN	SMC	**0.16**	0.07	0.07	0.09	**0.16**	0.08
	RW	**0.64**	0.58	0.26	0.29	**0.54**	0.15
	WJD	**0.41**	0.29	0.17	0.28	**0.34**	0.20

are clearly more challenging for beat tracking than the RW dataset.

6. CONCLUSIONS

This paper introduces two loss function terms, which can enforce sparsity and periodicity of a beat salience function predicted by a DNN model. Both terms can be computed in an unsupervised fashion without requiring ground truth annotations. Our findings show that only the sparsity-based loss term allows to significantly reduce spurious peaks, which can lead to erroneous beat time and tempo estimates. However it seems crucial to find the optimal mixing ratio to combine the supervised and unsupervised loss terms. We found that TCN models are as effective as commonly used RNN models, but are much faster to train.

In future work, we aim to combine the proposed method for beat salience modeling with a Dynamic Bayesian Network for rhythm inference to evaluate the beat tracking performance for different loss function configurations. Also, we plan to investigate the scenario of adapting DNN based beat tracking models to novel datasets in a purely unsupervised fashion by exclusively using the proposed loss terms L_S and L_P.

Acknowledgements

This work has been supported by the German Research Foundation (AB 675/2-1, BR 1333/20-1).

7. REFERENCES

[1] Meinard Müller, *Fundamentals of Music Processing*, Springer, 2015.

[2] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, NY, USA, 2016, pp. 255–261.

[3] Davide Fiocchi, Michele Buccoli, Massimiliano Zanoni, Fabio Antonacci, and Augusto Sarti, "Beat Tracking using Recurrent Neural Network: a Transfer Learning Approach," in *Proc. of the 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, pp. 1915–1919.

[4] Tian Cheng, Satoru Fukayama, and Masataka Goto, "Convolving Gaussian Kernels for RNN-Based Beat Tracking," in *Proc. of the 28th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, 2018, pp. 1905–1909.

[5] Simon Durand, Juan P. Bello, Bertrand David, and Gael Richard, "Feature adapted convolutional neural networks for downbeat tracking," in *Proc. of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Shanghai, China, 2016, pp. 296–300.

[6] Florian Krebs, Sebastian Böck, Matthias Dorfer, and Gerhard Widmer, "Downbeat Tracking Using Beat-Synchronous Features and Recurrent Neural Networks," in *Proc. of the 17th International Society for Music Information Retrieval Conference (ISMIR)*, New York, NY, USA, 2016, pp. 129–135.

[7] Simon Durand, Juan Pablo Bello, Bertrand David, and Gael Richard, "Robust Downbeat Tracking Using an Ensemble of Convolutional Networks," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 25, no. 1, pp. 72–85, 2017.

[8] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "A Multi-Model Approach To Beat Tracking Considering Heterogeneous Music Styles," in *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 603–608.

[9] Magdalena Fuentes, Brian McFee, Hélène C. Crayencour, Slim Essid, and Juan Pablo Bello, "A Music Structure Informed Downbeat Tracking System Using Skip-chain Conditional Random Fields and Deep Learning," in *Proc. of the 44th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 481–485.

[10] Matthew E. P. Davies and Sebastian Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. of the 27th European Signal Processing Conference (EUSIPCO)*, A Coruña, Spain, 2019.

[11] Sebastian Böck, Florian Krebs, and Gerhard Widmer, "Joint Beat and Downbeat Tracking with Recurrent Neural Networks," in *Proc. of the 17th International So- ciety for Music Information Retrieval Conference (ISMIR)*, New York, NY, USA, 2016, pp. 255–261.

[12] Aleksandr Diment, Toni Heittola, and Tuomas Virtanen, "Semi-supervised learning for musical instrument recognition," in *Proc. of the 21st European Signal Processing Conference (EUSIPCO)*, Marrakech, Morocco, 2013.

[13] Soujanya Poria, Alexander Gelbukh, Amir Hussain, Sivaji Bandyopadhyay, and Newton Howard, "Music genre classification: A semi-supervised approach," 2013.

[14] Jakob Abeßer, Stefan Balke, and Meinard Müller, "Improving Bass Saliency Estimation Using Label Propagation and Transfer Learning," in *Proc. of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 306–312.

[15] Nicolas Turpault, Romain Serizel, and Emmanuel Vincent, "Semi-supervised Triplet Loss Based Learning of Ambient Audio Embeddings," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 760–764.

[16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv:1609.03499*, 2016.

[17] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu, "Neural Machine Translation in Linear Time," *arXiv:1610.10099*, 2016.

[18] Patrik O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[19] Matthew E. P. Davies, Norberto Degara Quintela, and Mark Plumbley, "Evaluation Methods for Musical Audio Beat Tracking Algorithms," 2009.

[20] Fabien Gouyon, Anssi Klapuri, Simon Dixon, Miguel Alonso, George Tzanetakis, Christian Uhle, and Pedro 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.

[21] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka, "Rwc music database: Popular, classical, and jazz music databases," in *Proc. of the 3rd International Conference on Music Information Retrieval (ISMIR)*, oct 2002, pp. 287–288.

[22] Bruno Di Giorgi, Massimiliano Zanoni, Augusto Sarti, and Stefano Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," *Proc. of the 8th International Workshop on Multidimensional Systems*, pp. 145–150, 2013.

[23] André Holzapfel, Matthew E. P. Davies, José R. Zapata, João Lobato Oliveira, and Fabien Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[24] Martin Pfleiderer, Klaus Frieler, Jakob Abeßer, Wolf-Georg Zaddach, and Benjamin Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*, Schott Campus, 2017.

[25] Martin Jenckel, Sourabh Sarvotham Parkala, Syed Saqib Bukhari, and Andreas Dengel, "Impact of training LSTM-RNN with fuzzy ground truth," *Proc. of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pp. 388–393, 2018.

[26] Diederick P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2015.