SINGING BEAT TRACKING WITH SELF-SUPERVISED FRONT-END AND LINEAR TRANSFORMERS

Mojtaba Heydari   **Zhiyao Duan**

Department of Electrical and Computer Engineering
University of Rochester, 500 Wilson Blvd, Rochester, NY 14627, USA
mheydari@ur.rochester.edu   zhiyao.duan@rochester.edu

ABSTRACT

Tracking beats of singing voices without the presence of musical accompaniment can find many applications in music production, automatic song arrangement, and social media interaction. Its main challenge is the lack of strong rhythmic and harmonic patterns that are important for music rhythmic analysis in general. Even for human listeners, this can be a challenging task. As a result, existing music beat tracking systems fail to deliver satisfactory performance on singing voices. In this paper, we propose singing beat tracking as a novel task, and propose the first approach to solving this task. Our approach leverages semantic information of singing voices by employing pre-trained self-supervised WavLM and DistilHuBERT speech representations as the front-end and uses a self-attention encoder layer to predict beats. To train and test the system, we obtain separated singing voices and their beat annotations using source separation and beat tracking on complete songs, followed by manual corrections. Experiments on the 741 separated vocal tracks of the GTZAN dataset show that the proposed system outperforms several state-of-the-art music beat tracking methods by a large margin in terms of beat tracking accuracy. Ablation studies also confirm the advantages of pre-trained self-supervised speech representations over generic spectral features.

1. INTRODUCTION

Music tempo and beat detection are two of the core and well-defined MIR topics, and scholars have proposed many approaches to addressing different aspects of them for various music genres. For instance, some works such as [1–4] proposed some unsupervised approaches to detect music beat and tempo by using some low-level features like onset strengths. More recently, several more recent approaches employ neural networks to address the mentioned tasks [5–8]. Extracting singing rhythmic parameters makes it possible to address several MIR problems such as automatic instrumental and singing tracks alignment, au-

tomatic music mix, and remix, interactive content creation on social media platforms, etc.

Many of the proposed music rhythmic analysis approaches [1, 5, 7] perform in an offline fashion while others [6, 9, 10] are capable of extracting music rhythmic features causally and even in real-time.

In contrast to all of the mentioned models, beat and tempo detection for singing voice is an untapped MIR task. There are significant inherent differences between the natures of complete music and singing voices. One of those differences is that complete music pieces usually contain rich percussive and harmonic profiles while singing tracks usually lack such beneficial parameters making their rhythmic analysis more demanding compared to the former group. Their other important difference is that music beat tracking models usually only use acoustical clues such as magnitude spectrogram as input features while singing tracks are more similar to speech signals where the models may require to deal with para-linguistics, semantic, and phonemic level inputs in addition to acoustical features.

Our contributions in this paper are as follows:

1- We introduce singing beat tracking as a novel MIR task. We propose two strategies to tackle the lack of annotated data for this task by leveraging pre-existing datasets and beat tracking and source separation techniques. We also propose a new evaluation scheme that can account for phase ambiguities of beat annotation for vocal music.

2- We propose two neural models for the singing beat tracking task. These models leverage pre-trained speech self-supervised models to extract feature embeddings, which are then fed into a linear transformer network to output beat predictions.

3- We evaluate the proposed models on hundreds of vocal tracks with diverse genres. Experiments show that the proposed models outperform three representative baselines designed or trained for general music beat tracking by a large margin. An ablation study is also performed to investigate the effect of speech self-supervised models over commonly used generic spectral features, and results again show an outperformance by a large margin.

The remainder of the paper is organized as follows: In Section 2, we describe the proposed vocal beat tracking task with dataset curation and a new evaluation scheme. In Section 3, we describe our proposed models with details. In Section 4, we present experimental comparisons with baselines and an ablation study on the feature extraction

front end. Finally, Section 5 concludes the paper.

2. THE PROPOSED TASK

We propose singing beat tracking as a new MIR task, which aims to design algorithms that can track beats of singing voices in online or offline fashion. In this section, we describe the two cornerstones for this task: datasets and evaluation metrics.

2.1 Singing Data and Label Generation

Data for training and evaluating singing beat tracking algorithms require both singing voice recordings and their beat annotations. As there is no existing datasets that meet this requirement, we need to design a mechanism to collect such data. Instead of recording singing voices and annotating their beats manually, we propose to leverage existing MIR datasets and techniques to collect data for singing beat tracking, as illustrated in Figure 1.
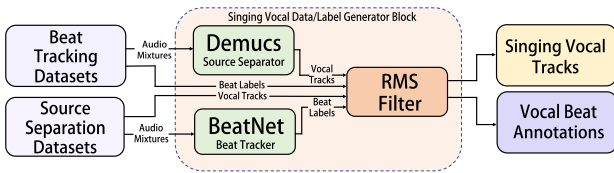

Figure 1. Singing data and label generation pipeline.

The first strategy we propose is to use pre-existing music beat tracking datasets in which the beat annotations are available. The idea is to obtain their singing tracks through music source separation. State-of-the-art (SOTA) singing voice separation methods have been shown to achieve outstanding results on popular genres of music, and the separated singing voice contains little interference from the background music. For this task, we employ Demucs Hybrid [11], one of the superior music source separation models. It uses a waveform-to-waveform convolutional auto-encoder with a U-Net structure and bidirectional LSTM layers, and is designed to separate the music mixture into four sources including bass, drums, vocals, and others. We use the pre-trained model [1] which was trained on all 150 songs of MusDB [12] dataset. It is noted that source separation is used in some related works e.g. [13–15] to improve music beat tracking while in this work, we utilize it as one of the suggested systematic ways to generate singing vocal beat tracking datasets.

The second strategy we propose leverages pre-existing music source separation datasets, in which the isolated vocal tracks are available as a part of those datasets. To obtain beat annotations, we apply BeatNet [6] offline version on the complete songs (i.e., music mixtures). BeatNet is a SOTA online beat tracking method that uses a convolutional recurrent neural network (CRNN) and a hidden Markov model (HMM) decoder to extract music beats and downbeats. We modify the HMM decoder from particle

---

[1] https://github.com/facebookresearch/demucs

filtering inference to Viterbi algorithm to improve its performance in the offline scenario. The reason that we run BeatNet on music mixtures instead of the separated vocal tracks is because it, the same as all other beat tracking methods, is trained on complete music pieces. While BeatNet has been shown to be fairly accurate on many songs with different genres, there are still annotation errors. To fix them, we also perform a manual revision by listening to the separated vocal track together with synthesized beeping sounds of the beat annotations and correcting errors.

While the beat annotations are obtained for the entire song, the ground-truth or separated vocal tracks show long chunks of silence due to the inactivity of singing. These long silent chunks are not useful and even are distracting for singing rhythmic analysis. Therefore, we normalize the energy of each vocal track using the root-mean-square (RMS), calculate a frame-wise RMS, and set a threshold to detect long silent chunks and split the vocal track into vocal segments. The RMS threshold is set to 0.01, and silent chunks shorter than 8 seconds are kept.

We apply these two strategies to a total of 8 existing beat tracking and music source separation datasets to obtain a total of 2248 vocal excerpts with beat annotations. The entire length of the vocal segments is 34h 35m. The datasets are summarized in Table 1.

2.2 Evaluation Metrics

For evaluation metrics, we adopt the commonly used F-measure in beat tracking. A beat is considered correctly detected if it is matched with a ground-truth beat with a time deviation smaller than 70 ms. In addition, we employ three additional metrics including P-score, Cemgil and Goto to provide a more detailed evaluation. Details about these three metrics can be found in [3].

For many music pieces, vocal tracks can align with the offbeat position (i.e., middle point between two adjacent ground-truth beats) rather that the beats. In fact, it also can be quite natural for humans to clap on the 180-degree phase shifted positions While this inherent ambiguity also exists in some instrumental music, it is much more common for singing voices. Apparently, the off-beat predictions would

Dataset	# Number of vocal excerpts	Total Length
Ballroom [16, 17] *	452	2 h 38 m
GTZAN [18, 19] *	741	5 h 44 m
Hainsworth [20]*	154	1 h 47 m
MUSDB18 [12]†	263	6 h 21 m
Rock Corpus [21]*	315	9 h 23 m
RWC pop [22, 23]*	188	5 h 06 m
RWC Royalty free [22, 23]*	29	19 m
URSing [24]†	106	3 h 17 m

Table 1. Datasets collected and adapted for singing beat tracking. ∗ denotes beat tracking datasets and † denotes music separation datasets.

be evaluated poorly against the ground-truth beat annotations for all of the abovementioned metrics.

To address this problem, we propose an additional Phase Inclusive (PI) evaluation scheme with existing metrics. In this scheme, a metric is computed twice, one comparing the predicted beat positions with the ground-truth beat annotations and the other comparing with the 180-degree shifted ground-truth. The maximum is reported as the final metric under the PI evaluation scheme.

3. THE PROPOSED METHOD

To address the singing beat tracking task, we propose two models that take advantage of pre-trained WavLM and DistilHuBERT SSL representations respectively to extract the input features. Then we build the same linear multi-head self-attention network on top of them to fine-tune the models for our task. Finally, a hidden Markov model decoder is used to infer the singing beat positions using the activations provided by the respective neural networks. The source code and system demos are available [2] .Figure 2 demonstrates the overall structure of all proposed models and Figure 3 demonstrates the neural network structures of all proposed models.
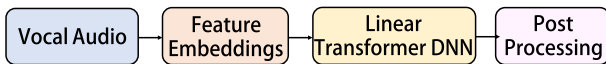

Figure 2. General pipeline of the proposed method.

3.1 Feature Embedding

Neural networks benefit from large quantities of labeled training data. However, in many cases including that of our task, labeled data is much harder to obtain than unlabeled data. Self-supervised learning has emerged as a paradigm to learn general data representations from unlabeled examples and fine-tuning the model on labeled data [25]. On the other hand, as demonstrated in [6], selecting appropriate input features plays an important role in music rhythmic analysis tasks. Due to acoustic and linguistic similarities between singing voices and speech, our main assumption is that self-supervised speech representations are helpful as feature embeddings in singing rhythmic analysis.

SSL achieves great success in speech-related tasks such as Automatic Speech Recognition, Phoneme Recognition and Emotion Recognition. Each task can be addressed by fine-tuning a universal pre-trained self-supervised model or training a neural network on top of the contextualized self-supervised representation of the input. In recent years, several pre-trained self-supervised models (e.g. [25–30]) attempt to provide such universal speech representations. The most successful models such as [25, 26] usually encode speech audio through a multi-layer convolutional neural network and then mask some chunks of the resulting latent speech representations. Then, a contrastive learning

step is performed by feeding the latent representations to a transformer network to distinguish true latents [25].

WavLM: WavLm [26] is one of the recent universal self-supervised pre-trained models on 94k hours of data to solve full-stack downstream speech tasks. It jointly predicts masked speech and performs denoising in the pre-training stage. The denoising module helps to enhance the potential for non-ASR tasks. Moreover, it uses gated relative position bias in its transformer structure to improve capturing sequence ordering of input speech. WavLM has three different versions, Base, Base+, and Large. WavLM Base and WavLM Base+ include 12 encoder layers, 768-dimensional hidden states, and 8 attention heads. The WavLM Large contains 24 encoder layers, 1024-dimensional hidden states, and 12 attention heads. According to the SUPERB [31], which is an evaluation benchmark designed to provide a standard and comprehensive testbed for pre-trained models on several speech tasks, WavLM outperforms the other counterparts for several speech downstream tasks.

Since WavLM model has demonstrated promising results for several downstream speech tasks, we employ it as the front-end model for our first proposed approach to prepare contextualized input embeddings. Among three pre-trained options, we chose the Base+ model, becasue of its better performance than the Base model and similar performance with the Large model on many downstream tasks.

According to some works on the layer-wise analysis of self-supervised models (e.g. [32, 33]), using their last layer's output is not necessarily the best option to leverage the SSL model's maximum capability. This is because the desired feature aspects of the input signal are distributed differently through internal encoder layers. For example, [33] demonstrated that in Wav2Vec2 self-supervised model, for some properties like word meaning, intermediate layers play a much more important role than the output layer, while for acoustical features, early layers carry the most important information. Therefore, in order to capture different aspects of the input signal, we use a weighted sum of all encoder layers of WavLM in which each weight is a learnable parameter during the training phase.

DistilHuBERT: Although WavLM demonstrated the most successful results for several downstream tasks, it is a large model which requires large memory and high pre-training and inference costs imposing a speed bottleneck for the system. DistilHuBERT [29], is a novel multi-task learning framework to distill hidden representations from a HuBERT [27] model directly. It is initialized with the HuBERT's parameters. Then, its prediction heads learn to generate the teacher's hidden representations by minimizing the loss function. DistilHuBERT reduces the model size by 75% and increases the inference speed by 73% from HuBERT while retaining most performance in ten different speech tasks. Moreover, it requires little training time and data, opening the possibilities of pre-training personal and on-device SSL models for speech.

Therefore, we propose a second model that switches the heavy WavLM front-end model with DistilHuBERT,
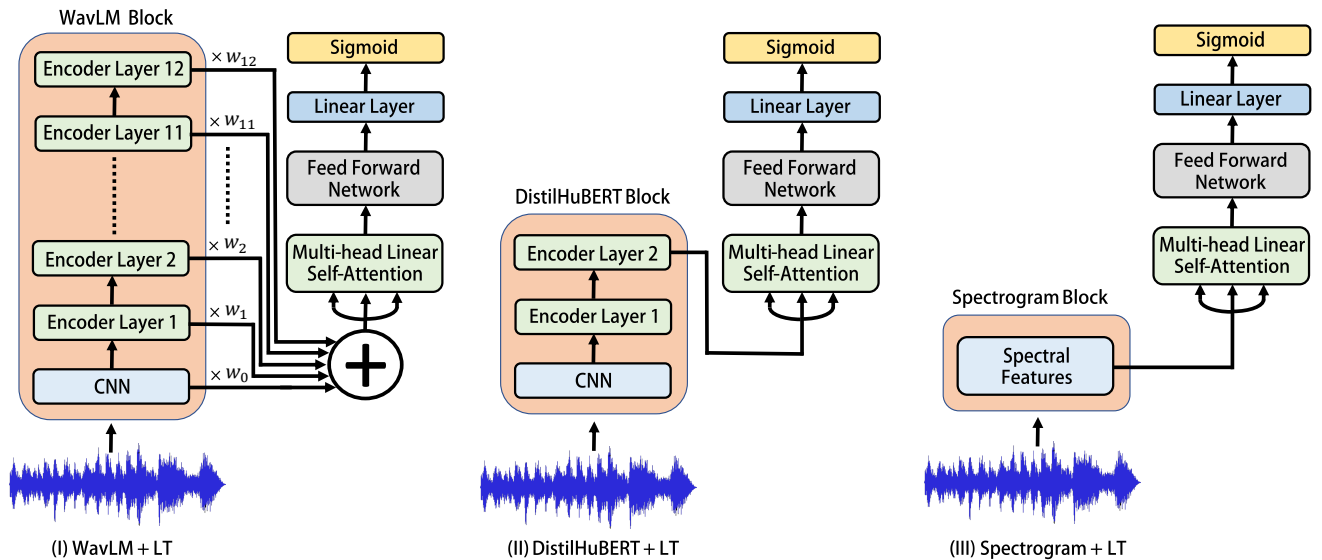

Figure 3. Neural network structures of the proposed models. (I), (II) and (III) use WavLM, DistilHuBERT and Spectrogram front-ends blocks, respectively, followed by the same linear transformer network.

a much lighter model, leading to a much faster inference process. DistilHuBERT includes a CNN network and two self-attention layers. According to its evaluation results [29], DistilHuBERT delivers comparable performance to the teacher HuBERT model with much less computational cost. It is noted that as the training objective of the DistilHuBERT student model is to learn directly from several internal encoder layers of the teacher model, the weighted sum strategy is not needed for this model and its last layer's output is used as the feature embeddings.

Spectrogram: We also perform an ablation study to evaluate the effect of using self-supervised models on the system performance by replacing the self-supervised pre-trained feature embeddings with spectral features that are commonly used in SOTA music beat tracking methods [6, 7]. These spectral features include log-magnitude mel-frequency spectrogram with three window sizes of 1024, 2048, and 4096 samples and their first-order time differences.

3.2 Linear Transformer Network

Transformers achieve remarkable performance in many tasks, but due to their quadratic complexity with respect to the input length, they are prohibitively slow for long sequences [34]. To address this limitation and to improve transformers' computational cost, several approaches are proposed. For instance, Linformer [35] is a model that reduces the transformer complexity in both time and space from $O(N^2)$ to $O(N)$ by approximating the self-attention mechanism by a low-rank matrix. It is noted that N is the sequence length. Another model [34]reduces the complexity to linear by expressing the self-attention as a linear dot-product of kernel feature maps and making use of the associativity property of matrix products. It achieves similar performance to vanilla transformers and they are up to 4000x faster on auto-regressive prediction of very long sequences. Another recent model [36] removes the softmax in self-attention by using the Gaussian kernel function to replace the dot-product similarity without further normalization. It enables a full self-attention matrix to be approximated via low-rank matrix decomposition.

In our proposed method, we build a linear multi-head self-attention layer using [34] on top of the feature embedding blocks for all three proposed models.

Our encoder comprises a self-attention layer with four attention heads with query and value dimensions of 192 and a feed-forward network with the size of 1024. The encoder network is followed by a linear layer and a sigmoid layer. The final output can be viewed as the beat salience for the input audio frame.

To train the proposed models, we use binary cross-entropy with logits between model output and the ground-truth beat annotations as the loss function. Only the linear transformer and its following layers are trained, while the self-supervised front ends are pre-trained and fixed.

4. POST-PROCESSING

In order to decode the beat positions, we employed a DBN approximated using HMM from Madmom library [37] with the default parameters. It is noted that the mentioned inference block is shared between all reported models in the evaluation table except BeatRoot [2].

5. EXPERIMENTS

5.1 Experimental Setup

We use all of the datasets collected in Section 2 except GTZAN for training the models with 80% of the songs randomly allocated for training and 20% for validation. Following several previous music beat tracking works, we keep the entire GTZAN dataset as the test set. It contains

1000 music excerpts covering 10 different music genres, out of which 741 of them contain vocal tracks. It ensures genre diversity in the test set and such a training/test dataset partition reduces the possibility of overfitting as well.Moreover, it is unseen in the training blocks of all compared methods. However, one drawback of it is that GTZAN's vocal tracks are separated from the whole songs. Hence, they may contain some information leakage from other musical instruments, which may boost the system performance compared to testing on isolated vocal tracks. To address this issue, we took the following steps. First, We used one of the best-performing supervised source separation models [11] and listened through the separated vocal tracks to ensure that they do not contain obvious music signal leakage. Second, when tuning the RMS filter, we discarded some sparse, abrupt and short signal segments that are more likely to be percussive sounds instead of vocal signals. Third, we compared with several high performing music beat tracking methods in the evaluation section; The leakage of musical signals into the separated tracks, if any, would improve the performance of these comparison models as well.

For the mentioned reason and given that singing beat tracking is a novel MIR task with no specifically design existing models, in this paper, we compare our proposed models against general music beat tracking models including two supervised models, BeatNet [6] and Böck [7] and an unsupervised signal-processing model, BeatRoot [2]. BeatNet is the SOTA online joint beat, downbeat, and meter tracking model that uses convolutional recurrent neural networks and particle filtering. It can also operate in an offline fashion by switching its particle filtering inference block with an offline hidden Markov model decoder. Böck uses a recurrent neural network to predict beat and downbeat saliences of each audio frame, and uses a Dynamic Bayesian Network (DBN) to infer beat and downbeat positions from the salience function. In our implementation, we use the same HMM decoder as that in BeatNet to replace the DBN. BeatRoot is a unsupervised method with a multiple agent architecture that simultaneously considers several different hypotheses of beat positions and tempi.

5.2 Training Details

To train the proposed models, we freeze the self-supervised front-end blocks and train the rest of the model including the multi-head self-attention layer and its following feed-forward and linear layers. For the WavLM+LT model, in addition to the abovementioned items, the 12 weights to combine different encoder layer's output are also trained.

The mentioned weights are initialized as ones and the rest of the weights and biases are initialized randomly. We use the Adam optimizer with a learning rate of 5×10^{-5} and a batch size of 10. The batches comprise 15-second long excerpts randomly sampled from all audio files in the training set. As the audio files have different length, to ensure a fair distribution, we sample with a probability that is proportional to the length of the files.

5.3 Results and Discussions

To assess the computational cost of the models, we report the inference speed for all of them. The reported numbers are the average computational time for all 741 excerpts with the average duration of 28 seconds each. Note that we measure the speed of all methods using CPU processing on the same Windows machine with an AMD Ryzen 9 3900X CPU and 3.80 GHz clock. Table 2 demonstrates the evaluation results.

Table 2 shows the evaluation results of the baselines and the proposed models using the four metrics presented in Section 2.2. It also includes the Phase Inclusive (PI) evaluation scheme for the proposed models. Several interesting observations can be made. First, all of the three proposed models outperform the three baselines by a large margin. This difference is especially pronounced for more strict metrics such as Goto [3]. This confirms our hypothesis that vocal tracks show very different patterns from complete songs, and the baseline models that are trained on complete songs do not perform as well as the proposed models that are trained on the vocal tracks. This hypothesis is further validated by the fact that BeatRoot outperforms BeatNet and Böck, while the latter two models are shown to outperform BeatRoot in previous studies on complete songs [6, 7]. It is noted that BeatNet and Böck are data-driven approaches and are trained on complete songs, while BeatRoot is a signal processing model without training. The mismatch between complete songs and vocal tracks does impose a strong negative bias to the data-driven baselines.

Second, the two proposed models that leverage speech SSL feature embeddings (i.e., WavLM+LT and DistilHu-BERT+LT) improves over Spectrogram+LT by a large margin, and this improvement is even bigger than that from the best baseline to Spectrogram+LT. This shows the significant advantage of using feature embeddings learned on speech data and suggests that this advantage is even more significant than training on the vocal tracks. As the spectrogram features used in Spectrogram+LT are the same as those in BeatNet and Böck and transformers have been shown to outperform RNNs in various tasks, it is reasonable to believe that even if BeatNet and Böck are trained on the vocal tracks, their performance would be only comparable to that of Spectrogram+LT.

Third, WavLM+LT delivers the best scores for all of the evaluation metrics except the computation time, showing that stronger feature embeddings lead to better vocal beat tracking accuracy. However, the attention layers' complexity in WavLM is quadratic to the length of input; while the average computational time (4.09 seconds) is acceptable for vocal segments (28 s long on average) in this experiments, as the length increases, the computational time will soon become prohibitive. Figure 4 shows the learned weights of different encoder layers of the WavLM SSL model. It can be seen that earlier layers generally contribute more to forming the feature embeddings compared to latter layers. Layers 3, 2 and 1 contribute the most. Note that Layer 0 represents the CNN network output.

	Method	F-Measure	P-Score	Cemgil	Goto	Comp. Time
	BeatNet	0.243	0.327	0.173	0.003	0.13 (s)
Baseline	BeatRoot	0.301	0.394	0.22	0.066	**0.03 (s)**
	Böck	0.171	0.195	0.122	0.009	1.56 (s)
	WavLM + LT	**0.733**	**0.704**	**0.618**	**0.560**	4.09 (s)
Proposed	DistilHuBERT + LT	0.703	0.668	0.593	0.516	1.83 (s)
	Spectrogram + LT	0.454	0.438	0.367	0.223	0.32 (s)
	WavLM + LT	0.745	0.715	0.627	0.574	4.09 (s)
Proposed (PI Results)	DistilHuBERT + LT	0.721	0.684	0.608	0.537	1.83 (s)
	Spectrogram + LT	0.489	0.477	0.391	0.265	0.32 (s)

Table 2. Average performance and speed across segments of several methods on the GTZAN separated vocal tracks.
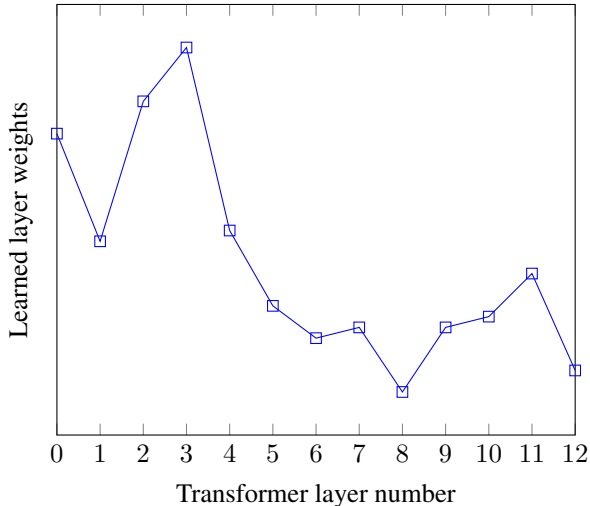

Figure 4. Illustration of the learned weights for different encoder layers of WavLM in the WavLM+LT model.

Fourth, with slight performance drop, the second proposed model which employs LT and DistilHubert delivers the next best performance among all models. DistilHubert uses a much lighter structure with fewer parameters than WavLM, making it ideal for time-sensitive inferences.

Finally, the PI evaluation scheme shows a slight improvement of all of the proposed models on all accuracy metrics. In particular, the improvement of Spectrogram+LT is greater than that of the other models. This suggests that the SSL pretrained speech embedding features are helpful to reduce the phase shift ambiguity.

Figure 5 shows the performance of our proposed WavLM + LT model for different music genres. According to the table, the best and worst performances belong to *disco* and *blues* respectively. One reason for that may be the stronger and punchier stresses in *disco* vocal tracks comparing to that of *blues* tracks. Another important reason may be the different syncopation ratio among vocal track of different music genres. Syncopation is a musical term that defines the placement of rhythmic stresses or accents where they would not normally occur, making part or all of a tune or piece of music off-beat [38]. Therefore, a higher syncopation ratio leads to a more difficult beat tracking process. Nate that in Figure 5 we only report the evaluation results of the genres that contain at least 80

separate singing tracks. It leaves out the *classical* and *jazz* genres which have only two and zero separated tracks.

6. CONCLUSION

In this paper we introduced singing beat tracking as a novel MIR task. We proposed two approaches to obtain labeled data for the mentioned task and introduced two methods to accomplish the mentioned task using pre-trained speech self-supervised models and multi-head linear self-attention networks. We also conducted an ablation study to investigate the effect of speech self-supervised models on the system performance. Experiments on a collection of vocal tracks with diverse genres show that the proposed models that combine self-supervised models and transformers outperform three representative baselines designed or trained for beat tracking for general music. The ablation study also shows that the self-supervised speech embeddings outperform generic spectral features that are commonly used in music beat tracking. In the future we will extend this research to cover real-time applications and more challenging singing tracks such as human humming and operas.
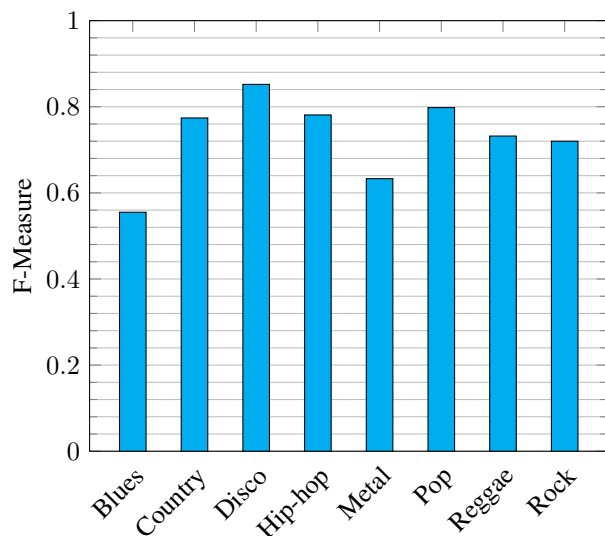

Figure 5. F-measure performance of WavLM + LT model on the GTZAN separated vocal tracks for different genres.

8. REFERENCES

[1] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[2] S. Dixon, "Evaluation of the audio beat tracking system beatroot," *Journal of New Music Research*, vol. 36, no. 1, pp. 39–50, 2007.

[3] M. E. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.

[4] A. Mottaghi, K. Behdin, A. Esmaeili, M. Heydari, and F. Marvasti, "Obtain: Real-time beat tracking in audio signals index terms—onset strength signal, tempo estimation, beat onset, cumulative beat strength signal, peak detection," *International Journal of Signal Processing Systems*, pp. 123–129, 2017.

[5] S. Böck and M. E. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR), Montreal, QC, Canada*, 2020, pp. 12–16.

[6] M. Heydari, F. Cwitkowitz, and Z. Duan, "Beatnet: Crnn and particle filtering for online joint beat downbeat and meter tracking," 2021.

[7] S.Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *In Proc. of the 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, 2016.

[8] M. Heydari and Z. Duan, "Don't look back: An online beat tracking method using RNN and enhanced particle filtering," in *In Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021.

[9] J. L. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis, "Ibt: A real-time tempo and beat tracking system." in *ISMIR*, 2010, pp. 291–296.

[10] M. Heydari, M. McCallum, A. Ehmann, and Z. Duan, "A novel 1d state space for efficient music rhythmic analysis," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 421–425.

[11] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[12] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[13] D. Desblancs, R. Hennequin, and V. Lostanlen, "Zeronote samba: Self-supervised beat tracking," 2022.

[14] C.-Y. Chiu, J. Ching, W.-Y. Hsiao, Y.-H. Chen, A. W.-Y. Su, and Y.-H. Yang, "Source separation-based data augmentation for improved joint beat and downbeat tracking," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 391–395.

[15] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stajylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 421–424.

[16] F. Gouyon, A. P. 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, 2006.

[17] F. Krebs, S. Böck, and G. Widmer., "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *Proc. of the 14th Int. Society for Music Information Retrieval Conf.*, 2013.

[18] U. Marchand and G. Peeters, "Swing ratio estimation," in *In Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx)*, 2015.

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

[20] S. W. Hainsworth and M. D. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 15, pp. 1–11, 2004.

[21] T. de Clercq and D. Temperley., "A corpus analysis of rock harmony," *Popular Music*, vol. 30, no. 1, pp. 47–70, 2011.

[22] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Popular, classical and jazz music databases." in *Ismir*, vol. 2, 2002, pp. 287–288.

[23] M. Goto *et al.*, "Development of the rwc music database," in *Proceedings of the 18th international congress on acoustics (ICA 2004)*, vol. 1, 2004, pp. 553–556.

[24] B. Li, Y. Wang, and Z. Duan, "Audiovisual singing voice separation," *arXiv preprint arXiv:2107.00231*, 2021.

[25] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[26] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.

[27] 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 Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[28] P. Peng and D. Harwath, "Fast-slow transformer for visually grounding speech," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7727–7731.

[29] H.-J. Chang, S.-w. Yang, and H.-y. Lee, "Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7087–7091.

[30] S. Ling and Y. Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[31] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[32] L. V. Prasad, A. Seth, S. Ghosh, and S. Umesh, "Analyzing the factors affecting usefulness of self-supervised pre-trained representations for speech recognition," *arXiv preprint arXiv:2203.16973*, 2022.

[33] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," *arXiv preprint arXiv:2107.04734*, 2021.

[34] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are rnns: Fast autoregressive transformers with linear attention," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5156–5165.

[35] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[36] J. Lu, J. Yao, J. Zhang, X. Zhu, H. Xu, W. Gao, C. Xu, T. Xiang, and L. Zhang, "Soft: Softmax-free transformer with linear complexity," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[37] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[38] M. Hoffman, "Syncopation," *National Symphony Orchestra. NPR. Retrieved*, vol. 13, 2009.