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.

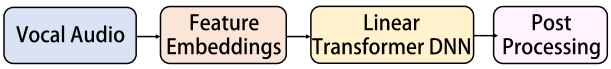

Figure 2. General pipeline of the proposed method.

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,

[2] https://github.com/mjhydri/singing-vocal-beat-tracking

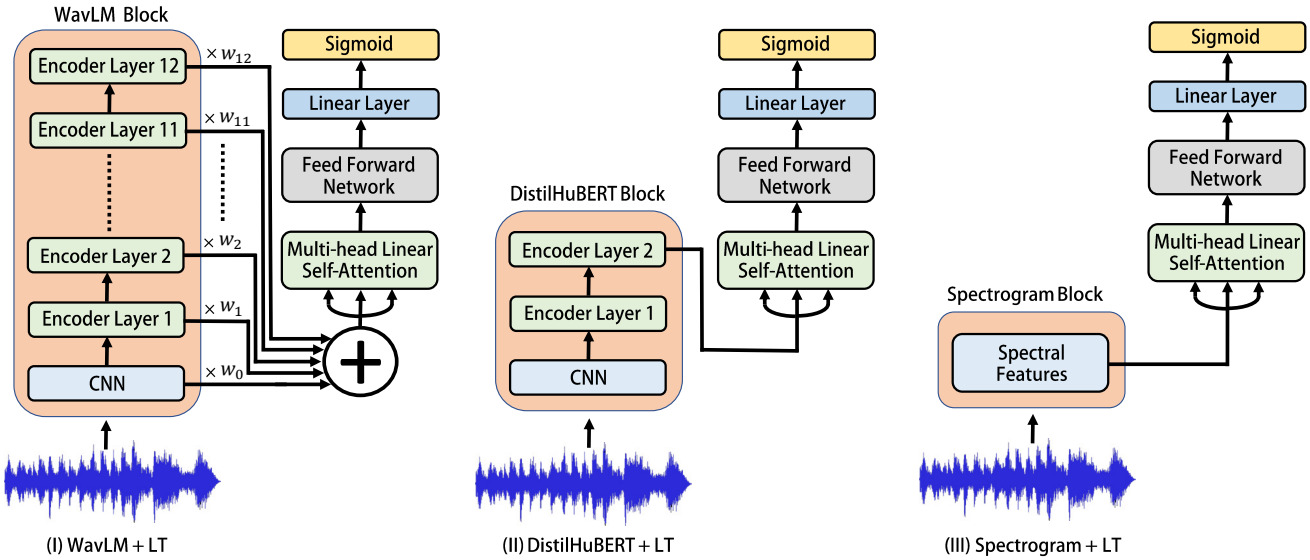

Figure 3. Neural network structures of the proposed models. (I), (II) and (III) use WavLM, DistilHuBERT and Spectrogram front-ends blocks, respectively, followed by the same linear transformer network.

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 DistilHuBERT+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
Baseline	BeatNet	0.243	0.327	0.173	0.003	0.13 (s)
	BeatRoot	0.301	0.394	0.22	0.066	**0.03 (s)**
	Böck	0.171	0.195	0.122	0.009	1.56 (s)
Proposed	WavLM + LT	**0.733**	**0.704**	**0.618**	**0.560**	4.09 (s)
	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)
Proposed (PI Results)	WavLM + LT	0.745	0.715	0.627	0.574	4.09 (s)
	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.

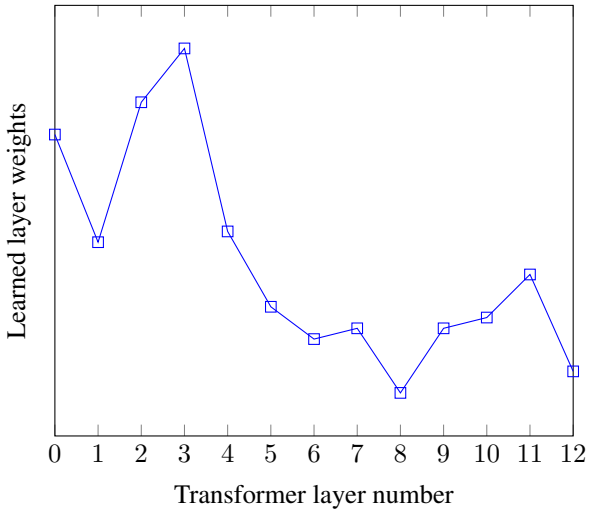

Figure 4. Illustration of the learned weights for different encoder layers of WavLM in the WavLM+LT model.

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.

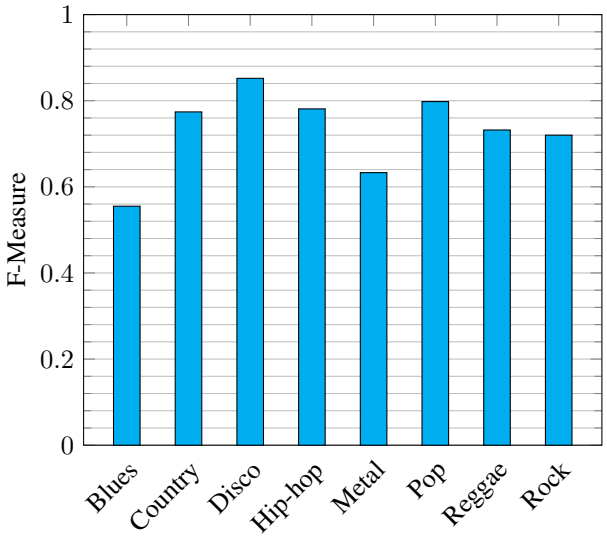

Figure 5. F-measure performance of WavLM + LT model on the GTZAN separated vocal tracks for different genres.

7. ACKNOWLEDGEMENT

This work was supported by National Science Foundation grant No. 1846184.

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, "Zero-note 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.

ENSEMBLESET: A NEW HIGH QUALITY SYNTHESISED DATASET FOR CHAMBER ENSEMBLE SEPARATION

Saurjya Sarkar **Emmanouil Benetos** **Mark Sandler**
Centre for Digital Music, Queen Mary University of London, UK
{saurjya.sarkar, emmanouil.benetos, mark.sandler}@qmul.ac.uk

ABSTRACT

Music source separation research has made great advances in recent years, especially towards the problem of separating vocals, drums, and bass stems from mastered songs. The advances in this field can be directly attributed to the availability of large-scale multitrack research datasets for these mentioned stems. Tasks such as separating similar-sounding sources from an ensemble recording have seen limited research due to the lack of sizeable, bleed-free multitrack datasets. In this paper, we introduce a novel multitrack dataset called EnsembleSet generated using the Spitfire BBC Symphony Orchestra library using ensemble scores from RWC Classical Music Database and Mutopia. Our data generation method introduces automated articulation mapping for different playing styles based on the input MIDI/MusicXML data. The sample library also enables us to render the dataset with 20 different mix/microphone configurations allowing us to study various recording scenarios for each performance. The dataset presents 80 tracks (6+ hours) with a range of string, wind, and brass instruments arranged as chamber ensembles. We also present our benchmark on our synthesised dataset using a permutation-invariant time-domain separation model for chamber ensembles which produces generalisable results when tested on real recordings from existing datasets.

1. INTRODUCTION

Audio source separation aims to extract individual sound sources from a digital audio mixture. Based on the constituents of the input mixture and the target output, the problem definition can be further refined to specific audio separation tasks like speech separation, speech enhancement, and music source separation [1]. While specific subtasks in the speech-domain like speech denoising, multi-speaker separation and dereverberation have been thoroughly explored, music separation research has largely been focused on the demixing challenge [2] aided by the popular MUSDB dataset [3]. The demixing challenge is targeted at solving the problem of separation of vocals, bass and drums from mixed and mastered pop songs. This has greatly benefited the field by demonstrating that source separation is indeed possible at a commercial scale with state-of-the-art deep learning based architectures. Unfortunately, this also has resulted in the research towards this specific task to dwarf other problems that would also fall under the umbrella of music source separation, to the extent that music source separation has become synonymous with the task of separating vocal, drums and bass stems from mastered songs.

In this paper, we explore a different area in music source separation with a focus on separation of chamber ensembles, where the target sources are harmonised and have very high spectral overlap. The music demixing challenge has shown successful separation of instruments with distinct spectro-temporal cues like vocals, drums and bass. Separating monotimbral ensembles is an inherently challenging task as they combine challenging aspects of both speech and music separation. In chamber ensembles we find the sources to occupy similar frequency ranges, may have label ambiguity [4, 5] due to multiple sources belonging to the same instrument family meanwhile being temporally and harmonically correlated, due to their musical structure which further increases their spectral overlap.

To address this challenge, we present a novel chamber ensemble dataset titled EnsembleSet [6]. EnsembleSet was synthesised using a realistic sample library Spitfire BBC Symphony Orchestra (BBCSO) [7] utilising the MIDI transcriptions from the RWC Classical Music Database [8] and lilypond scores from Mutopia [9] (refer to Section 3.4 for details). In Section 4 we utilise EnsembleSet to train a source separation model based on the Dual-path Transformer architecture (DPTNet) [10, 11] for separating mixtures of chamber ensembles. We achieve very good and generalisable separation performance which we exhibit through cross-dataset performance evaluation in Section 5. Other applications of EnsembleSet may include topics such as multi-instrument transcription [12], instrument recognition [13], score-informed source separation [14], microphone translation [15], automatic mixing [16] and other tasks that may benefit from score-aligned multitrack multi-instrument data.

2. BACKGROUND

Although the term "Music Source Separation" sounds like an umbrella-term for all applications of source separation

Attribution: S. Sarkar, E. Benetos, and M. Sandler, "EnsembleSet: a new high quality synthesised dataset for chamber ensemble separation", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

in music, in research the use of this term is largely used to describe the specific task of vocals/drums/bass instrument stem separation from mastered tracks [2, 17]. A music source separation task which is relatively unexplored is the challenge of separating similar sounding instruments from a mixture. This problem has two significant differences from the aforementioned task. Firstly, if the sources in the mixtures are similar sounding (e.g., mixture of a strings section), it results in high spectral energy overlap. This is further compounded by the fact that such sources often play in a very synchronised fashion while harmonising each other. Secondly, often in such mixtures we may have multiple sources of the same type present, which makes it an unsuitable problem to be solved using *class-based* separation methods [4]. We define the task of separating such mixtures with constituent sources suffering from label ambiguity and high timbral similarity as *monotimbral ensemble separation*.

2.1 Datasets

Training supervised source separation models typically requires datasets which provide the clean target sources as a reference for the deep learning models to learn from. While the majority of popular music can be recorded in separate takes for different performers, with a reference metronome or a backing track, ensembles are usually recorded together in the same take. This is due to the fact that ensemble performers rely on being able to hear each other during performance to be able to synchronise perfectly. This raises the problem that the majority of stems available from recording projects of monotimbral ensembles contain bleed [1] from non-target sources [18, 19]. This becomes problematic for training models for source separation as we do not have the clean ground truth as a target result for the model. This lack of clean and sizeable datasets for ensembles has affected the amount of research seen in this domain.

The URMP dataset [20] addresses this problem by making the performers record isolated takes and then subsequently re-align, dereverberate and downmix them to be present in a physical space. Another work [21] presented a dataset recorded by minimising the amount of bleed using soundproofing across booths while recording multiple performers in the same take. Bach10 [22] also presents multitrack recordings of chamber ensembles where each song consists of four parts (Soprano, Alto, Tenor and Bass) which were performed by violin, clarinet, saxophone and bassoon, respectively. The TRIOS dataset [23] consists of 5 bleed-free multitracks and synchronised MIDI files of 4 classical music pieces and 1 jazz piece.

2.2 Prior work

There are some tasks which fit our definition of monotimbral separation that have been explored recently. One is vocal harmony separation [11,24–26]. While the label ambiguity problem does exist for this task, some approaches have circumvented it by looking at the problem in a class-based separation fashion by categorising the constituent sources based on their registers i.e. alto, soprano, bass and tenor. One method of solving the label ambiguity problem is to tackle this problem in a score-conditioned fashion as in [25]. Another method of tackling this problem is called permutation invariant training (PIT) [27] which has been the preferred solution to tackle the label ambiguity problem for speech separation research [10, 28, 29]. PIT has been utilised for choral separation in [11]. Another approach has been to use multi-task learning by utilising score-information to simultaneously separate and transcribe mixtures of 2-source chamber ensembles which has shown some success for scenarios with small datasets [30].

3. DATASET

To overcome the challenge of bleed-free real recorded datasets for ensembles, we introduce a novel dataset “EnsembleSet", which utilises a highly realistic orchestral sample library by Spitfire Audio called "BBC Symphony Orchestra" (BBCSO) [7]. We use this sample library to render digital chamber ensemble scores from MIDI and MusicXML format to 18 unique multi-mic recordings and 2 professional mixes. For this work, we utilised the RWC Classical Music Database [8] and Mutopia [9] to source our chamber ensemble MIDI and MusicXML (converted from lilypond) scores. It must be noted that MIDI data are not ideal to capture string, wind and brass instrument scores as they do not encapsulate articulation information. On the other hand lilypond scores contain minimal dynamics (velocity) information, which is essential for realistic rendering using virtual instruments. In order to address these challenges, we utilise expression maps provided by Dorico [31], a scorewriter software which allows us to determine the articulation mode for each note in the piece.

3.1 Collecting Digital Music Scores

3.1.1 RWC Classical Music Database

The RWC Classical Music Database [8] consists of 50 public-domain classical pieces performed by musicians and then manually transcribed to MIDI with high-quality tempo and velocity mapping. Since the database only provides the final mix of these performances, its applications are limited especially in the context of source separation. We choose a subset of these pieces which contain chamber ensembles which can be rendered using our method. Our 9 selected pieces (1h 3m 34s) [2] consist of 4 string quartets, 2 clarinet quintets, 2 piano trios and 1 piano quintet. It must be noted that for the piano trios and quintet, we only render the string instrument parts. Because MIDI files lack articulation information, we modify the MIDI files using Dorico to automatically add it using keyswitches, which are then subsequently rendered as multitracks on Reaper [32].

[1] Sound picked up by a microphone from a source other than that which is intended.

[2] Rendered duration in dataset.