a trajectory-tracking stage through maximum filtering, instead of simply calculating the difference between spectral bins over time. Trajectory tracking helps to suppress spurious spectral peaks, especially arising from vibrato. For further information, we refer to [14] (see also Figure 2e).

2.2 DL-Based Activation Functions

2.2.1 CNN Onset Detector

Schlüter and Böck [16] approach the onset detection task as a computer vision problem, where magnitude spectrograms of the audio recordings are used as the input to a CNN. Onsets are often characterized by rapid transient changes in the spectrum, resulting in sharp edges that are clearly visible in a spectrogram. Using convolutional kernels, one can easily detect these sharp edges of onsets. Similar to SF-based methods, the proposed CNN model computes spectro–temporal differences and captures percussive and pitched onsets. The resulting activation function is referred to as `DL-O` (see Figure 2f for an example).

2.2.2 RNN Beat Detector

In the case of unreliable and noisy onset cues, using beat activation functions constitutes a more feasible solution to improve the temporal alignment. To compute such activation functions, we use the BLSTM model by Böck and Schedl [17] for framewise beat detection. BLSTMs can effectively model the temporal context of the data and is therefore suitable for beat tracking. In the proposed approach, magnitude spectrograms computed with three different window lengths, and their first order differences are used as the input to the network. The network outputs encode the likelihoods of beat positions, as illustrated by Figure 2g. In our experiments, the resulting beat activation functions are denoted as `DL-B`.

2.2.3 RNN Downbeat Detector

Böck et. al [18] present an RNN model to jointly detect beat and downbeats. Like the previously mentioned onset and beat detectors, this model also operates on magnitude spectrograms. The downbeat detector uses an RNN similar to the proposed network in [17] to model beats and downbeats. In our experiments, we only use the probability of downbeats as the activation cues, which we denote as `DL-D` (see Figure 2h for an illustration).

2.3 Combined Synchronization with Activation Functions

To find the optimal alignment between two feature sequences $X := (x_1, \ldots, x_N)$ and $Y := (y_1, \ldots, y_M)$, where $n \in \{1, \ldots, N\}$ and $m \in \{1, \ldots, M\}$, we rely on DTW. By comparing each pair of elements in the feature sequences, we obtain a cost matrix $\mathbf{C}(n, m) := c(x_n, y_m)$ of size $N \times M$, where c defines a local cost measure. Then, an *optimal warping path* is determined via dynamic programming. We refer to [9] for a detailed account on DTW for music synchronization. For efficient implementations, we refer to [21, 22].

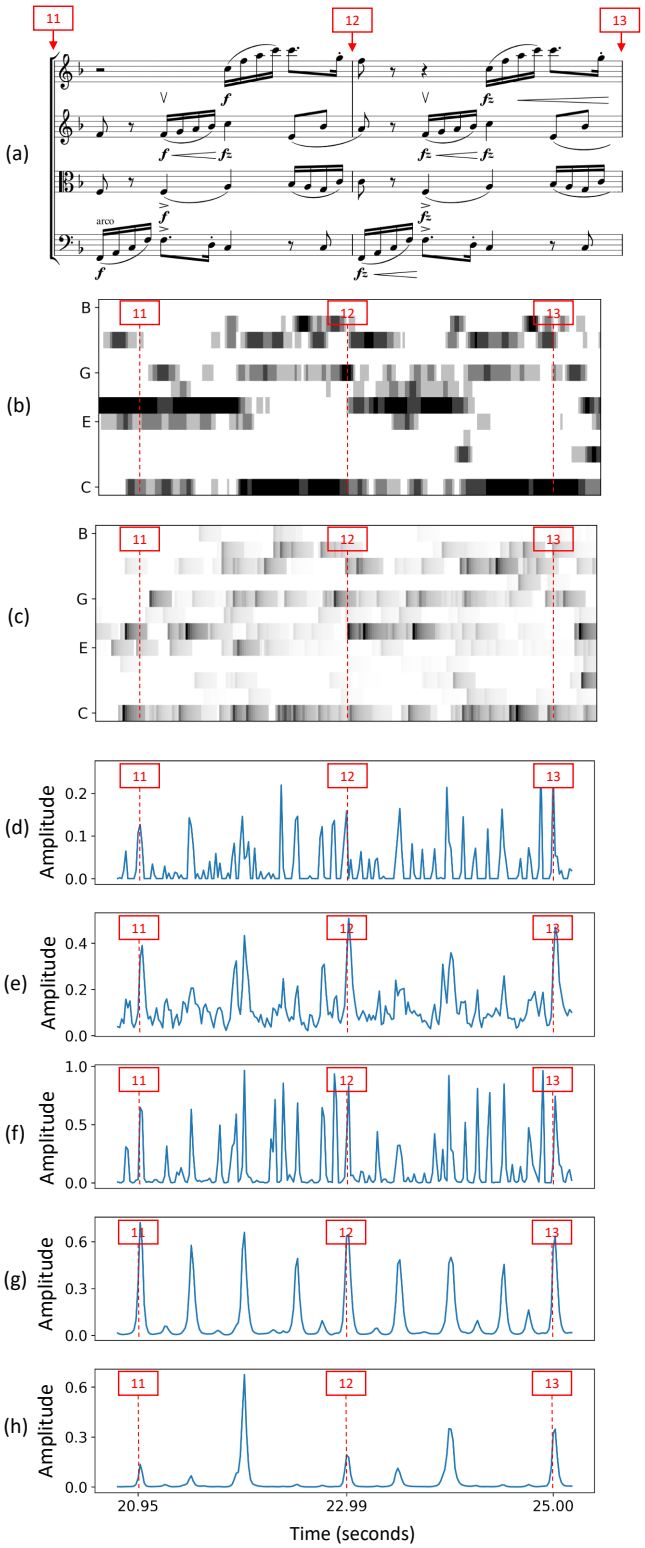

Figure 2: Chroma features and activation functions computed for an excerpt of the String Quartet No. 12 in F major, Op. 96 (first movement) composed by Antonín Dvořák, performed by the Borromeo Ensemble. Activation functions are shown in blue and ground-truth measure positions in red. **(a)** Sheet music representation of the measures 11–13. **(b)** Chroma **(c)** DLNCO **(d)** SF **(e)** SF* **(f)** Onsets (`DL-O`) **(g)** Beats (`DL-B`) **(h)** Downbeats (`DL-D`)

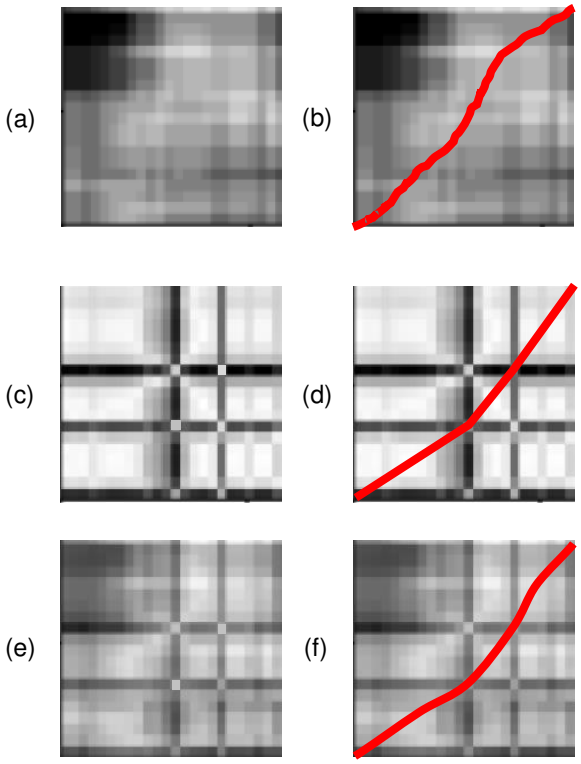

Figure 3: Excerpts from cost matrices and corresponding warping paths computed with DTW **(a) / (b)**: $\mathbf{C}_{\texttt{CHROMA}}$, **(c) / (d)**: $\mathbf{C}_{\texttt{ACT}}$, **(e) / (f)**: $\alpha\mathbf{C}_{\texttt{CHROMA}} + (1-\alpha)\mathbf{C}_{\texttt{ACT}}$, with $\alpha = 0.5$.

We now adapt the combined synchronization idea by Ewert et al. [12], integrating conventional onset-based and DL-based activation functions into the synchronization pipeline. Building upon this approach, we introduce three cost matrices. The first one $\mathbf{C}_{\texttt{CHROMA}}$ is a cost matrix based on the normalized chroma features and the cosine distance, see Figure 3a for an example. The second cost matrix $\mathbf{C}_{\texttt{ACT}}$ is computed using the Euclidean distance and conventional onset-based or DL-based activation functions as introduced in Section 2.1 and Section 2.2, respectively. Note that $\mathbf{C}_{\texttt{ACT}}$ exhibits grid-like structures. The visible horizontal and vertical grid lines of high cost (shown in black) correspond to high values in the first and second function, respectively. Only when a horizontal grid line intersects with a vertical grid line, the cost matrix has a small cost value at this intersection point (and also in a small neighborhood). This is where a high activation value of the first sequence meets another high activation value of the second sequence. In other words, these *intersection cells* encode a pair of time positions where two musical events (onsets, beats, downbeats) meet (see Figure 3c). Furthermore, the sections in the activation functions which have low values lead to homogeneous, zero-cost regions in the cost matrix $\mathbf{C}_{\texttt{ACT}}$. The third matrix is the sum of two cost matrices

$$\mathbf{C} = \alpha\mathbf{C}_{\texttt{CHROMA}} + (1-\alpha)\mathbf{C}_{\texttt{ACT}}, \qquad (1)$$

where $\alpha \in [0,1]$ is a weighting parameter. The sum $\mathbf{C}$ accounts for both harmonic or melodic information of the representations via $\mathbf{C}_{\texttt{CHROMA}}$ and additional activation cues via $\mathbf{C}_{\texttt{ACT}}$. Figure 3e illustrates an example using $\alpha = 0.5$.

Comparing the resulting optimal warping paths using $\mathbf{C}_{\texttt{CHROMA}}$ in Figure 3b, $\mathbf{C}_{\texttt{ACT}}$ in Figure 3d, and $\mathbf{C}$ in Figure 3f, we can observe an enhancement of the temporal alignment. The inclusion of DL-based onset, beat, and downbeat cues leads to an improvement of the warping path guided by grid structure's intersection points. Note that $\mathbf{C}_{\texttt{ACT}}$ remains zero in the regions without any novel events, and the overall alignment of $\mathbf{C}$ is mainly guided by $\mathbf{C}_{\texttt{CHROMA}}$.

3. EXPERIMENTS

3.1 Dataset

The genre of string quartet is composed for a small conductor-less ensemble, which consists of two violins, a viola and a violoncello. In our experiments, we use three versions (performances) of the String Quartet No. 12 in F major, Op. 96, by Antonín Dvořák, which comprises four movements. To give an insight of the musical properties of the string quartet, Table 1 provides the number of measures, time signature, and global tempo for each movement based on the recordings in our dataset. Note that the global tempo does not reflect any local tempo deviations. Its purpose is to indicate at what pace (average of three performances) a given movement is performed. In each recording, the repetitions are played as notated in the sheet music, thus ensuring structural consistency.

Table 1: Overview of four movements, including number of measures, time signature, and global tempo in BPM for each movement.

Movement	#Measures	Time signature	Global Tempo
M1	239	4/4	100
M2	97	6/8	84
M3	244	3/4	185
M4	382	2/4	140

For each recording (in the following referred to as version), we manually annotated the measure positions. In Table 2, the name of the version, the identifier, the recording year for each version, and the duration of each movement are listed. Note that each of the three performances last around 26 minutes in total, whereas durations of the movements may vary across different versions.

Table 2: Version, identifier, recording year, and duration of each movement.

			Duration (seconds)				
Version	ID	Year	M1	M2	M3	M4	Σ
Alban Berg	A	1991	599	410	238	323	1570
Borromeo	B	2012	540	465	228	338	1571
Prague	P	1973	584	424	250	322	1580
		Σ	1723	1299	716	983	4721

Table 3: Precision (P), Recall (R), and F-measure (F) based on methods DL-B for beat, and DL-D for downbeat tracking with DBN post-processing, evaluated on reference measure annotations and tolerance $\tau = 70$ ms. ϕ denotes the average accuracy over four movements M1, M2, M3, and M4.

	DL-B & DBN			DL-D & DBN		
	P	R	F	P	R	F
M1	0.194	0.806	0.312	0.648	0.586	0.612
M2	0.138	0.820	0.236	0.657	0.643	0.650
M3	0.327	0.539	0.407	0.524	0.224	0.314
M4	0.517	0.908	0.655	0.876	0.647	0.742
ϕ	0.294	0.768	0.403	0.676	0.525	0.580

3.2 Beat and Downbeat Tracking

As a baseline, we first introduce how DL-based beat [17] and downbeat trackers [18] perform in the detection of measure positions, where a dynamic Bayesian network (DBN) was used for post-processing (peak picking). Table 3 provides precision, recall, and F-measure values using a tolerance of $\tau = 70$ ms. Here, each entry indicates the mean value over different performances. Due to the higher density of beats, the beat tracker reveals a higher recall than the downbeat tracker for each movement, leading to a low precision and F-measure. Furthermore, each movement has a different time signature, which results in a different number of beats per measure and needs to be taken into consideration as prior information for the DBN.

3.3 Synchronization Results

In this section, we describe our experimental setting and evaluate audio alignments obtained using chroma features and different activation functions. In our experiments, we use the resulting warping path to transfer the measure positions annotated for the reference recording to the target recording. Given two versions of the same music piece with the time-continuous axes $[0, T_1]$ and $[0, T_2]$, the monotonous alignment can be modeled as a function

$$\mathcal{A} : [0, T_1] \to [0, T_2].$$

The pairwise alignment error ϵ_P for a given alignment of two recording is specified as the mean over the values

$$\epsilon_P(g_1) := |\mathcal{A}(g_1) - g_2|,$$

where $(g_1, g_2) \in [0, T_1] \times [0, T_2]$ denotes the ground-truth pairs of measure annotations. As an evaluation metric, we use *accuracy*, which is defined as the proportion of correctly transferred measure positions with a pairwise alignment error below a given tolerance τ [23].

In the following, we use eight different synchronization approaches based on conventional chroma features and the combination of chroma features with DLNCO features, SF, SF*, DL-based onsets (DL-O), DL-based beats (DL-B), DL-based downbeats (DL-D), and finally a 3-dimensional stacked activation function combining DL-based onsets, beats, and downbeats (DL-OBD) (see Section 2 for a detailed overview of activation functions). We use a feature rate of 50 Hz for the computation of chroma, and conventional onset features. To generate DL-based features, we use the *madmom* [24] library, for which we utilize the default setting 100 Hz as the feature rate, and downsample generated features to 50 Hz (after low-pass filtering).

Table 4: Accuracy values based on chroma and DL-OBD for different tolerances $\tau = 30,\ 70,\ 100$ ms. ϕ denotes the average accuracy over four movements M1, M2, M3, and M4.

	$\tau = 30$ ms		$\tau = 70$ ms		$\tau = 100$ ms	
	CHROMA	DL-OBD	CHROMA	DL-OBD	CHROMA	DL-OBD
M1	0.408	0.576	0.723	0.813	0.817	0.877
M2	0.396	0.600	0.694	0.842	0.789	0.917
M3	0.528	0.655	0.827	0.912	0.910	0.956
M4	0.485	0.662	0.773	0.884	0.861	0.930
ϕ	0.454	0.624	0.754	0.863	0.844	0.920

3.3.1 Overall Result

To get a first impression of the alignment behavior of different approaches, Figure 4 illustrates an overview of average accuracy values (averaged over all movements and all pairs of different performances) and for different tolerances τ. Obviously, one can observe that the synchronization accuracy improves with increasing threshold. For example, using $\tau = 30$ ms, the average synchronization accuracy is 0.454 when using only chroma features, and the accuracy increases to 0.624 when integrating the DL-OBD activation cues. This is a substantial improvement.

Next, we focus on the comparison of different approaches for $\tau = 70$ ms, which is a common tolerance value for the evaluation of music synchronization and beat tracking procedures. The inclusion of DLNCO slightly worsens the alignment since DLNCO features are not suited for soft onsets as occuring in string music. However, the integration of SF and SF* into the synchronization pipeline results in a better accuracy. Among conventional onset features, SF* shows a better performance than SF and DLNCO, owing to the fact that SF* features account for the detection of soft onsets and are therefore more suitable for string music. Furthermore, using DL-based methods leads to a better synchronization result compared to the conventional methods. Note that the integration of DL-OBD, which combines DL-O, DL-B, DL-D, reveals the best accuracy among all the synchronization approaches. It is also interesting to observe that DL-B, which is based on beats, is the second-best among DL-based approaches and leads to better accuracy values than downbeat-based DL-D, whereas DL-O reveals the lowest accuracy among the DL-based approaches.

In general, similar trends can be observed when using other thresholds. Using $\tau = 500$ ms, all synchronization approaches yield nearly perfect results.

3.3.2 Dependency on Movement

In our next experiment, we analyze the synchronization accuracy across different movements. Table 4 provides a comparison of alignment results based on the conventional chroma-based approach and our proposed combined

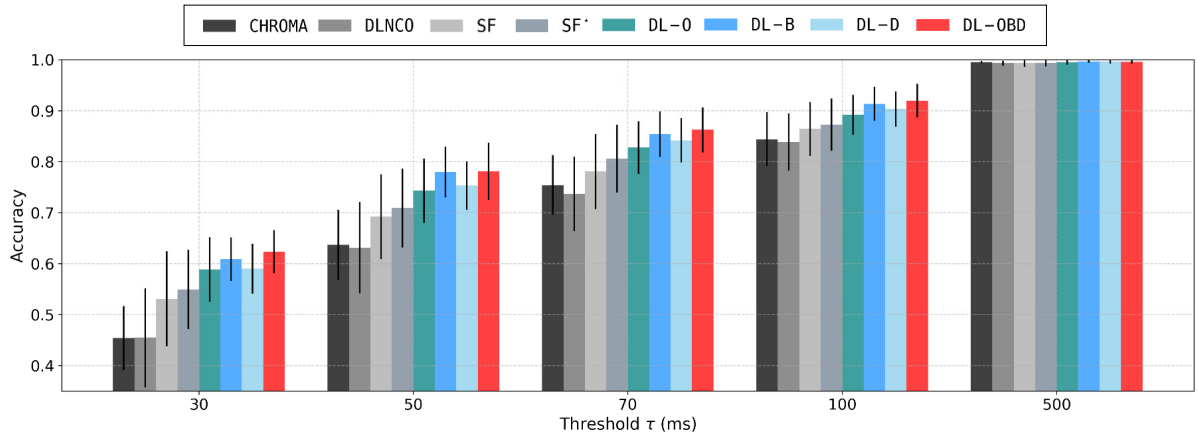

Figure 4: Comparison of the average accuracy values for different synchronization approaches and different threshold parameters τ. The accuracy denotes the proportion of correctly transferred measure positions having an error below a given tolerance τ.

Table 5: Accuracy based on chroma and DL-OBD across different synchronization pairs, for different tolerances τ. The first column indicates the pair of versions (see Section 3.1).

	$\tau = 30$ ms		$\tau = 70$ ms		$\tau = 100$ ms	
	CHROMA	DL-OBD	CHROMA	DL-OBD	CHROMA	DL-OBD
AP	0.494	0.636	0.774	0.864	0.839	0.928
PA	0.500	0.632	0.778	0.864	0.847	0.928
AB	0.434	0.576	0.722	0.849	0.830	0.907
BA	0.451	0.580	0.743	0.863	0.857	0.920
BP	0.429	0.662	0.762	0.870	0.850	0.919
PB	0.417	0.657	0.746	0.866	0.843	0.915
ϕ	0.454	0.624	0.754	0.863	0.844	0.920

method DL-OBD per movement for different tolerances τ. For example, considering the first movement and $\tau = 70$ ms, the accuracy of 0.723 for the chroma-based approach increases to 0.813 when using our combine approach DL-OBD. One can observe a similar trend across different movements for different tolerance parameters τ. The second movement tends to yield the lowest accuracy values when using only chroma features, while the integration of DL-OBD significantly improves the synchronization accuracy from 0.694 to 0.842 for the second movement. One reason may be that the beat and downbeat information leads to a significant improvement in synchronization accuracy of slower sections.

3.3.3 *Dependency on Performance*

As a final experiment, we provide a comparison of the accuracy values across different performances for different tolerance parameters τ in Table 5. In general, using a combination of chroma and activation functions significantly improves the accuracy for all the synchronization pairs and tolerances. For $\tau = 70$ ms, chroma reveals an average accuracy of 0.774 for AP and DL-OBD improves the accuracy to 0.864. Remarkably, across other synchronization pairs the synchronization accuracy values are similar. Nonetheless, deviations may occur due to soft onsets, slight inconsistencies in ground-truth annotations, and linear interpolation while measure transfer. Note that the synchronization accuracy rather depends on the musical complexity and structure, e.g., across different movements, but not on the performances.

4. CONCLUSION

In this article, we investigated the incorporation of conventional onset features, and activation cues obtained by recent DL-based onset, beat, and downbeat detectors to a conventional chroma-based synchronization pipeline. We showed that the integration of a combined version of onset, beat, and downbeat activation functions significantly improves the synchronization accuracy while maintaining the robustness of the original chroma-based synchronization approach. For the future, we will incorporate other features, which capture smoother note transitions for string quartets. We also aim at extending our string quartet dataset and evaluating the synchronization on beat-level annotations. Moreover, we will further investigate the role of the hyperparameter α (see Equation 1), which can be optimized as a time-dependent parameter.

Acknowledgements: This work was supported by the German Research Foundation (DFG MU 2686/10-2), and "Identification of the Czech origin of digital music recordings using machine learning" grant, which is realized within the project Quality Internal Grants of BUT (KInG BUT), Reg. No. CZ.02.2.69/0.0/0.0/19_073/0016948 and financed from the OP RDE. The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen IIS.

5. REFERENCES

[1] D. Schwarz, N. Orio, and N. Schnell, "Robust polyphonic midi score following with hidden Markov models," in *International Computer Music Conference (ICMC)*, Miami, Florida, USA, 2004.

[2] J. T. Foote, "Content-based retrieval of music and audio," in *Multimedia Storage and Archiving Systems II*, vol. 3229, International Society for Optics and Photonics. SPIE, 1997, pp. 138–147.

[3] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Computer Music Conference (ICMC)*, Paris, France, 1984, pp. 193–198.

[4] C. S. Sapp, "Comparative analysis of multiple musical performances," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 497–500.

[5] A. Lerch, C. Arthur, A. Pati, and S. Gururani, "Music performance analysis: A survey," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delft, The Netherlands, 2019, pp. 33–43.

[6] V. Konz, M. Müller, and R. Kleinertz, "A cross-version chord labelling approach for exploring harmonic structures—a case study on Beethoven's Appassionata," *Journal of New Music Research*, vol. 42, no. 1, pp. 61–77, 2013.

[7] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2919–2932, 2020.

[8] R. B. Dannenberg and N. Hu, "Polyphonic audio matching for score following and intelligent audio editors," in *Proceedings of the International Computer Music Conference (ICMC)*, San Francisco, USA, 2003, pp. 27–34.

[9] M. Müller, *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*, 2nd ed. Springer Verlag, 2021.

[10] B. Niedermayer and G. Widmer, "A multi-pass algorithm for accurate audio-to-score alignment," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 417–422.

[11] A. Arzt, G. Widmer, and S. Dixon, "Adaptive distance normalization for real-time music tracking," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, 2012, pp. 2689–2693.

[12] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 1869–1872.

[13] P. Grosche, M. Müller, and S. Ewert, "Combination of onset-features with applications to high-resolution music synchronization," in *Proceedings of the International Conference on Acoustics (NAG/DAGA)*, 2009, pp. 357–360.

[14] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detections," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Maynooth, Ireland, 2013.

[15] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal onset detection with bidirectional long short-term memory neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, August 2010, pp. 589–594.

[16] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, May 2014, pp. 6979–6983.

[17] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Paris, France, 2011, pp. 135–139.

[18] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, New York, USA, 2016, pp. 255–261.

[19] C. Weiß, V. Arifi-Müller, T. Prätzlich, R. Kleinertz, and M. Müller, "Analyzing measure annotations for Western classical music recordings," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York, USA, 2016, pp. 517–523.

[20] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[21] C. Tralie and E. Dempsey, "Exact, parallelizable dynamic time warping alignment with linear memory," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal, Canada, 2020, pp. 462–469.

[22] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software (JOSS)*, vol. 6, no. 64, pp. 3434:1–4, 2021.

[23] T. Prätzlich and M. Müller, "Triple-based analysis of music alignments without the need of ground-truth annotations," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Shanghai, China, March 2016, pp. 266–270.

[24] 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 ACM International Conference on Multimedia (ACM-MM)*, Amsterdam, The Netherlands, 2016, pp. 1174–1178.

A DEEP LEARNING METHOD FOR MELODY EXTRACTION FROM A POLYPHONIC SYMBOLIC MUSIC REPRESENTATION

Katerina Kosta **Wei Tsung Lu** **Gabriele Medeot** **Pierre Chanquion**
ByteDance
{name.surname}@bytedance.com

ABSTRACT

The task of identifying melodic lines in polyphonic music is a known active research topic in the symbolic and audio domain. Its importance has attracted the interest of researchers focusing on Music Information Retrieval and musicological applications and achieving high results is a common goal in industrial applications. Distinguishing the melody from the rest of the material in a written score can be a challenging task, however improvements have been reported in recent years using deep learning methods. In this paper, we present a lightweight deep bidirectional LSTM model for identifying the most salient melodic line of a music piece using handcrafted features without requiring the input score to be separated into multiple parts. We evaluate our model to measure the effectiveness of several data augmentation techniques and to compare performance to other state-of-the-art models. We also identify the features' importance and evaluate their incremental contribution on the model performance using evaluation metrics. Results on the POP909 dataset show that our model approximates or outperforms current state of the art models trained on the same dataset, based on different implemented metrics and observations.

1. INTRODUCTION

The concept of the note-to-note organization of music naturally evolves to perceiving larger structures such as phrases and melodic contours. Identifying what a listener perceives to be the melody in a piece of music and, more generally, examining the musicological concept of a melodic line has emerged in recent years as an important topic in the Music Information Retrieval community [1].

A melodic line incorporates musical properties which are rich in contextual information such as structure and rhythm and it embeds expressive characteristics from the perspective of the composer when constructing a piece as well as the perspective of the performer interpreting it. From a musicological point of view, being able to extract a melodic line with high accuracy can reveal new research directions, from analysing a composition style to classifying a performer's tendencies. In industrial or other research applications, high accuracy in melody extraction can improve the results of music search or recommendation algorithms as well as music generation systems.

In both audio and symbolic domains, the task of retrieving the notes of the melody from a polyphonic piece is not trivial. For example, during the act of listening, one may find it hard to distinguish note intervals from one another. It may happen that an interval judged as a major third when heard by itself, is judged as fourth in a separate music context instead [2]. During the act of reading a music score or rendering it using a single type of timbre, the note intervals within separate voices might be perceived as interchangeable. The aforementioned challenges have inspired the research on voice extraction, i.e. partitioning the polyphonic piece into a set of monophonic melodies (more details in [3] and [4]).

In this paper, we are focusing on the task of identifying the main melodic line in a symbolic score. Past work on this specific task can be divided into two broad categories. The first includes models predicting a melody part from a multiple-part score input, meaning that a series of notes have been separated to individual parts and the task is to predict which part, either globally or in segments, contains the main melodic line. Most common features used for this purpose are related to notes pitch, intensity, duration, as well as the total number of notes among a part. This type of melody identification is out of scope for this paper, however we refer to [5–7] for details.

Our approach belongs to the second category that is to identify the notes that constitute the main melody in the score without any prior knowledge of separated score parts. Hence, we process an unsegregated set of notes, as we believe that this approach give more flexibility on how it can be used in various applications.

Related work. There is a limited number of previous work on this matter. One of the first attempts is the skyline algorithm [8] which keeps the highest note starting at any time, from all simultaneous note events. The other existing algorithms can be separated in the way the music is represented as input. We can pinpoint two main strands. One handles the music as a piano-roll visualisation, which is a semantic segmentation where musical scores are treated as two-dimensional images. Melody extraction systems presented in [9] and [10] use this type of representation, with the aim to classify the notes represented as pixels. The

piano-roll representation is mainly used in deep learning methods using convolutional neural networks. The other strand, which our approach belongs to, treats the music sequentially, representing a series of tokens or low and mid-level features that best describe the input characteristics. Recurrent neural network based models employ the note-sequence representation. In the case of the system in [11], the input representation constitutes note-to-note affinity values coming directly from their contextual notes, constructing a weighted undirected graph, having as edge weights the corresponding affinity values. In order to obtain a single melody outcome, they employ spectral clustering to obtain one cluster over the learned graph.

The use of features is presented in [12], where each note is characterised by a set of note properties such as note dissonance, how close a note is to a note ranked with a high probability of being a melody note, or properties from the score metadata such as the musical instrument that a particular note has been assigned to. In our approach, we adapt a set of features which contain note information such as pitch and duration, as well as properties of a note in its polyphonic context.

A recent approach MIDIBERT [13] adopts the mask language model training strategy which is largely used in the field of natural language processing. The model accepts an input representation of sequence of tokens where each token represents a musical event. By training the model to reconstruct the masked input sequence, the transformer model can learn a latent representation of symbolic music. The experiment results show that with such pretraining, the model can be easily fine-tuned for other down stream tasks such as melody extraction.

Previous work reports high results in melody extraction accuracy, especially on recent systems that use deep learning methods. However, the resources used are either not publicly available or their use is copyright protected. Therefore, it is hard to compare and assess results due to unpublished datasets, different metrics reported and different test sets used. Datasets that clearly identify the primary melodic part from accompaniment are rare. For this publication, we have used the open-source POP909 dataset [14], where the salient melody notes have been distinguished. Pop music has been also used in [6, 9]. Folk music has been used in [10, 11] and classical in [9, 10]. The system in [13] adapts POP909 for the task of melody extraction.

Proposed method. The melody extraction model that we introduce in this paper is a supervised bidirectional Long-Short Term-Memory (biLSTM) model called LStoM, standing for Large Score to Melody. It receives a set of computed features as input derived from a MIDI score and classifies which notes constitute the main melodic line. We adapted the note events representation as a set of features in the form inspired by [4].

For our model input, we compute the following features for each note in our dataset, as described in Table 1. The first two features, `pitch` and `dur`, contain the basic pitch and duration information of a note, respectively, where pitch is expressed in a MIDI note number and the duration in a crotchet level. For `pitch_dist_below` and `pitch_dist_above`, we compute the distance to the pitch of the next (neighbouring) higher or lower note sounded simultaneously with the note in use. `pos_in_bar` records the score beat where the note onset is located within the score bar. `pitch_in_scale` records whether the note pitch is in the diatonic scale of the key signature. The features `pitch_dist_below` and `pitch_dist_above` are inspired by the work in [4]. The feature `pitch_in_scale` is inspired by the work in [12].

Feature name	Description
`pitch`	note pitch (MIDI number)
`dur`	note duration (crotchets)
`pitch_dist_below`	absolute pitch distance (semitones)
`pitch_dist_above`	absolute pitch distance (semitones)
`pos_in_bar`	note onset position in bar (crotchets)
`pitch_in_scale`	note pitch in key scale (boolean)

Table 1. The features that have been selected and computed, along with their description.

More on the data that we have processed and the model we have built in Sections 2 and 3 respectively. Section 4 includes a set of experiments and observations to assess the model's capabilities. Finally, in Section 5 we summarise our findings and suggest potential future directions.

2. DATA DETAILS

For our experiments, we have used the MIDI files from POP909 dataset [14]. This dataset includes piano arrangements of Chinese pop songs created by musicians playing on a MIDI keyboard, and the melody part has been identified by manually transcribing the lead vocal melody. The scores include the remaining parts "bridge" and "accompaniment" which we merge and consider as a single accompaniment part throughout our experiments.

The scores are rich in expressive characteristics, such as the note timing and duration. In our processing, we have used the dataset metadata information regarding the beats and the key. For our purpose, we have pre-processed the MIDI files, to rectify the following dysfunctions. Many scores include a time signature of '1/4', so we have updated the time signature by considering the meter information from the audio beat metadata. Time signatures of "2/4" and "2/2" have been mapped to "4/4". Also, many scores include a misalignment in the downbeat level, so we have fixed the start time of the piece to reduce this issue.

As mentioned before, the scores are performative and this means that they accommodate fine details or imprecision in timing. In order to setup our model, we have created a time-grid of the notes' onset and duration that is flexible enough to keep performative characteristics and strict enough to not explode the dimensions of possible values which would be hard to be trainable later on. The time-grid is setup to align onset times in triplets resolution. The note durations are adjusted to the minimum between the distance of current note onset and next note onset, and the current note duration in semiquaver resolution.