Project Report

Name: CHEUNG Chi Wang	**SID:** 1155158048	**Major:** AIST Year 3
Name: CHIU Long Him	**SID:** 1155143195	**Major:** AIST Year 4

Automatic Accompaniment Generation

1. Abstract

The goal of our project is to create a technology that will automatically create accompaniment for solo musical performances. Our technology makes use of time stretching and synchronization methods by MIDI-inputting the original tune and accompaniment. We will test four synchronization techniques in this study: Dynamic Time Warping, onset detection, tempo detection, and tempo and phase vocoder. In the absence of an accompanist, this tool seeks to give soloists a way to rehearse their performances with timed accompaniment. Through automated accompaniment production, our study enables musicians to improve their solo performances and advances the field of music synchronization.

2. Background

2.1 Project inspiration

In solo music performances, coordinating with an accompanist can be challenging. To overcome this, we developed a project to create a tool that allows soloists to practice without an accompanist. Using MIDI files of the original melody and accompaniment, our goal is to automatically generate accompaniment for live recordings, providing soloists with a valuable practice resource. This eliminates the need for extensive coordination and rehearsal time with an accompanist, enabling soloists to practice their performances more efficiently.

This project covers various topics from our lectures. We used the Dynamic Time Warping (DTW) method, which is taught in Lecture 4, to align the accompaniment with the melody. We explored tempo and tempogram alignment between the original melody and the live recording, which relates to Lecture 8. We also incorporated onset detection and the novelty function to further enhance alignment. We utilized Fourier Transform for frequency domain conversion and employed the phase vocoder for time stretching, relating to Lecture 3.

By combining these techniques, our project aims to provide soloists with an invaluable practice tool: automatic generation of accompaniment synchronized with their live recordings.

2.2 Similar projects

There are limited projects directly related to our topic. One noteworthy project in this domain implemented a machine learning approach to generate live interactive music accompaniment [1]. Their objective was to develop an application that aids musicians in practicing improvisation and musical interplay by generating dynamic accompaniment in response to a human player (see *Appendix 1*). Unlike our project, this system does not require user input, but rather focuses on providing accompaniment for users to practice their music improvisation skills.

Another relevant project is the Sync Toolbox [2], a Python package encompassing various components of a music synchronization pipeline. This toolbox utilizes the robust and efficient technology of DTW and offers functions for DTW and feature extraction. While the project does not explicitly detail the implementation of music synchronization using the DTW path, it provides valuable insights for our accompaniment alignment section.

These related projects serve as valuable references, inspiring our approach to autogenerating accompaniment and informing our understanding of music synchronization techniques.

3. Methodologies

3.1 Project description

The automatic accompaniment generation pipeline comprises six main steps: user input, preprocessing, audio synthesis, live music recording, audio synchronization, and audio output (see *Appendix 2*). These steps work together to generate synchronized accompaniment for solo performances.

User input is facilitated through the ability to input scores in the form of PDF files for different instruments. This allows users to provide the musical notation for their desired performance. In the preprocessing stage, the system scans and transforms the scores into MIDI files, separating them into different instrumental lines. This step can be facilitated using online tools such as ScanScore [3]. For the purpose of this project, the focus will not be on developing a score-scanning tool, and existing online tools will be utilized. Audio synthesis involves generating audio waveforms from MIDI files, aiming to make the synthesized audio more similar to live music recordings. This process ensures a more realistic and authentic accompaniment.

Audio synchronization is a crucial step in the project, where four different methods are implemented to map the accompaniment with the live recording of the user. These methods include DTW, Onset Detection, Tempo Detection, and Phase Vocoder. Each method contributes to achieving accurate synchronization between the accompaniment and the live recording. In the audio output stage, the system constructs the accompaniment while considering the tempo of the user's performance. This ensures that the generated accompaniment aligns properly with the timing and rhythm of the user's live recording.

To streamline the project and focus on its core functionality, the initial scope includes using "Twinkle Twinkle Little Star" as demonstration audio. This serves as a proof of concept, validating the effectiveness of the pipeline. Once the pipeline is functioning well, the project will expand to encompass different music pieces and combinations of instruments. Various types of audio recordings have been captured to simulate different user scenarios for testing purposes. These recordings include a normal version, one with an incorrect pitch, a faster tempo, and an irregular tempo. These diverse recordings enable the evaluation of the pipeline's performance under different conditions and user styles. Additionally, the recorded testing audio files are used to assess the results of the audio synchronization process. The four different methods applied to these recordings allow for a comparison of their advantages and disadvantages. This evaluation process provides valuable insights into the performance and effectiveness of each synchronization approach, informing decisions regarding their implementation in the pipeline. The inclusion of recordings with different characteristics allows for a thorough evaluation of the preprocessing algorithms' handling of these challenges in realistic scenarios.

3.2 Programming language

Our project utilizes Python due to its extensive range of libraries that support sound analysis. The main libraries employed include Numpy, Matplotlib, Librosa, FluidSynth, and midi2audio. FluidSynth and midi2audio are primarily used for data preprocessing and loading. Numpy is employed for data manipulation, Librosa serves as the main audio synchronization function, and Matplotlib is utilized for result visualization.

3.3 Algorithm

The algorithm that has been implemented in this project including Dynamic Time Warping, Fourier Transform, Inverse Fourier Transform, and Phase Vocoder.

Dynamic time warping (DTW) is an algorithm used to align two sequences of similar content but possibly different lengths. Given two sequences, $x[n], n \in \{ 0, \dots, N_x - 1\}$, and $y[n], n \in \{ 0, \dots, N_y - 1 \}$, DTW produces a set of index coordinate pairs {(i,j)...} such that x[i] and y[j] are similar. DTW requires the use of a distance metric between corresponding observations of x and y. One common choice is the Euclidean distance. [4]

A fast Fourier transform (FFT) is an algorithm that computes a sequence's discrete Fourier transform (DFT). Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. The DFT is obtained by decomposing a sequence of values into components of different frequencies. Inverse Fast Fourier transform (IFFT) is an algorithm to undoes the process of DFT. It is also known as the backward Fourier transform. It converts a signal of the frequency domain to a space or time signal. And we can reconstruct the filtered signal by using IFFT to transform it back to the time domain. [5]

Phase Vocoder is a type of vocoder-purposed algorithm which can interpolate information present in the frequency and time domains of audio signals by using phase information extracted from a frequency transform. The computer algorithm allows frequency-domain modifications to a digital sound file (typically time expansion/compression and pitch shifting). [6]

3.4 Data preprocessing

Our project involves the preprocessing of various live recordings of the performance of "Twinkle Twinkle Little Star." These recordings encompass different versions, including a normal rendition, recordings with incorrect pitch, faster tempo, and irregular tempo. To facilitate the preprocessing stage, we utilize libraries such as FluidSynth and midi2audio to convert the original MIDI files of "Twinkle Twinkle Little Star" into WAV files. This conversion is necessary as Librosa, one of the libraries used in the project, does not support MIDI file inputs directly.

Additionally, we leverage the ScanScore tool to transform the musical score of "Twinkle Twinkle Little Star" in PDF format into a MIDI file. This MIDI file serves as the basis for creating the accompaniment. To ensure the authenticity of the accompaniment in the live recordings, we employ a library called Music21 to convert the MIDI notes into a piano playing version, as the original accompaniment in the piece is played on a piano. Moreover, we explore the transformation of the MIDI notes to be played by other accompanying instruments, simulating the real accompaniment present in the live recordings. Next, we load these preprocessed audio files, including the converted WAV files and the live recordings, using the functions provided by Librosa. These loaded audio files are then utilized for further music retrieval and analysis processes (see *Appendix 3, Appendix 4*).

3.5 Methods

In our project, we implemented four different methods to try to map the accompaniment with the live recording of the user, namely, using Dynamic Time Warping, Onset Detection, Tempo Detection, and Phase Vocoder.

3.5.1 Dynamic Time Wrapping

The DTW method is employed in our project for audio synchronization between the live recording and the accompaniment. This involves several steps to align the two audio sources. The chroma representations of the original melody and the live recordings are computed, capturing the harmonic content (see *Appendix 5*). Visualizations are generated to showcase the harmonic patterns across different versions of the live performance. The DTW algorithm calculates the DTW and warping paths between the chroma representations. This identifies the optimal alignment between the original melody and each live recording. Plots of the warping paths illustrate the correspondence between the time frames and highlight temporal distortions (see *Appendix 6*).

To demonstrate alignment, the aligned audio waveforms are plotted alongside the original melody waveform for each live recording (see *Appendix 7*). The index mapping allows for a direct comparison of the synchronized audio, showcasing the achieved alignment through DTW. These DTW-based steps ensure accurate audio synchronization, aligning the accompaniment properly with the live performance. Visualizations and waveform comparisons aid in evaluating the effectiveness of the alignment process.

3.5.2 Onset Detection

The onset detection method employed in our project aims to identify the precise moments in the audio recordings where musical events, such as note onsets, occur. This information is crucial for aligning the accompaniment with the live recordings. In the project, the onset detection process involves several steps. Initially, the spectral flux is calculated by computing the absolute values of the Short-Time Fourier Transform (STFT) of the audio signal. The differences between consecutive frames are analyzed, retaining only positive changes in magnitude by setting negative differences to zero. The spectral flux is then normalized by subtracting the mean and dividing by the standard deviation to ensure an appropriate onset threshold relative to magnitude variations. The onset threshold is determined as half of the maximum spectral flux value. Any spectral flux values surpassing this threshold are identified as potential onsets. The onset frames are obtained by extracting the indices where the spectral flux exceeds the threshold. These frame indices can be converted to time values using the provided sampling rate and hop length.

To visualize the results, a spectrogram of the audio signal can be plotted, along with the spectral flux and the detected onsets (see *Appendix 8, 9*). The spectrogram displays the magnitude of the frequency components over time, while the spectral flux plot represents the magnitude changes in the spectrogram. Detected onsets can be indicated by red markers on the spectral flux plot. By applying this onset detection methodology to the live recordings, we can accurately identify the onset points and utilize this information for further alignment and synchronization processes.

3.5.3 Tempo Detection

The methodology of tempo detection is employed to align the live recording and the accompaniment by utilizing the beat and tempo information present in the live recording. The process begins by computing the tempogram for both the input recording and the accompaniment. This tempogram represents the variations in tempo over time (see *Appendix 10*). By calculating the cross-correlation matrix between the two tempograms, we can analyze the movement and relationship of these two time-series data. Using the maximum index of the cross-correlation, we determine the required time shift of the accompaniment audio and apply this shift to the accompaniment tempogram. This adjustment ensures the alignment between the accompaniment and the live recording. Next, we extract the beat times from the shifted tempogram, which serve as reference points for the tempo variations in the accompaniment.

To achieve synchronization, the accompaniment audio is time-stretched using the extracted beat times from the adjusted tempogram. This process allows the accompaniment to match the tempo fluctuations present in the live recording. By combining the shifted accompaniment with the recording, we effectively simulate the live performance with accompaniment, ensuring a coherent and synchronized musical experience. These revisions provide more clarity in describing the steps involved in tempo detection and alignment between the accompaniment and the live recording.

3.5.4 Phase Vocoder

The methodology of phase vocoder is employed as a similar approach to tempo detection, as mentioned in the previous section. The first step involves computing the time-stretch factor, which is essentially the ratio between the tempo of the accompaniment and the tempo of the recording. This factor determines the amount of time stretching or compression needed to align the accompaniment with the live recording. Next, the accompaniment audio is transformed into the frequency domain using the Fast Fourier Transform (FFT). This allows us to analyze the spectral content of the accompaniment audio and perform manipulations in the frequency domain. By employing the phase vocoder technique, we can effectively stretch or compress the audio in the frequency domain based on the computed time-stretch factor. This process ensures that the accompaniment aligns with the tempo variations present in the live recording. Finally, the shifted FFT data of the accompaniment is converted back to the time domain using the Inverse Fast Fourier Transform (IFFT). This transformation restores the audio to its original time-domain representation, now with the desired time-stretching applied.

By utilizing the phase vocoder methodology, we can accurately adjust the tempo of the accompaniment to match the tempo variations in the live recording. This enables a synchronized and cohesive musical performance, where the accompaniment complements the tempo fluctuations of the live performance.**3.6 Result**

3.6.1 Dynamic Time Wrapping

Dynamic Time Warping (DTW) is one of the methods employed for aligning the accompaniment with the recording melody. However, one drawback of DTW is the potential introduction of jittering sounds due to forceful alignment of the accompaniment to the recording melody (see *Appendix 11*). To address this issue, we explored two solutions.

The first solution involved denoising the aligned audio using a low-pass filter (see *Appendix 12*). This approach aimed to reduce the jittering sounds caused by the forceful alignment. By applying a low-pass filter, high-frequency noise and unwanted artifacts introduced during the alignment process were attenuated, resulting in a smoother and more natural-sounding accompaniment.

However, a better solution was devised to overcome the limitations of the initial approach. Instead of aligning duplicated indices, we opted to generate a new accompaniment directly using the aligned indices obtained through the DTW process. This approach allowed for the creation of a fresh accompaniment that followed the tempo and timing of the recording melody more accurately. By leveraging the aligned indices, the new accompaniment avoided the jittering sounds associated with the forceful alignment, resulting in a more seamless and cohesive musical experience.

By implementing this improved approach, we were able to mitigate the issues related to jittering sounds in the aligned accompaniment. The generated accompaniment, derived from the aligned indices, offered a more precise and accurate representation of the intended musical performance, enhancing the overall quality and synchronization of the final result.

3.6.2 Onset Detection

Onset detection is an essential component of the project, which involves identifying the precise moments when musical events or note onsets occur in the recordings. However, we encountered a challenge in that the number of onsets detected varied for each recording, making it difficult to match them accurately (see *Appendix 8, 9*).

To address this issue, we initially considered an auto-determination approach for setting the threshold used in onset detection. This method aimed to automatically adjust the threshold based on the characteristics of each recording. However, upon further evaluation, we determined that this approach was not suitable for the current stage of the project.

As the project progressed, we realized that a more comprehensive solution was needed to handle the varying numbers of onsets in the recordings effectively. This involved exploring alternative techniques or algorithms that can address the challenge of matching onsets between the accompaniment and the live recordings accurately.

3.6.3 Tempo Detection

Tempo detection plays a crucial role in aligning the accompaniment with the live recording (see *Appendix 13*). It assumes that the recording is correctly in tempo with the original melody, albeit potentially faster or slower. However, a challenge arises when the performer does not follow the tempo consistently throughout the recording, leading to a poor performance.

To overcome this issue, we devised a solution that involves segmenting the audio into smaller parts and applying our tempo detection approach to each segment individually. By breaking down the audio into manageable segments, we can more accurately analyze the tempo variations within each part. Once the tempo detection is applied to each segment, the next step is to merge the segmented audio back together. This merging process ensures that the accompaniment aligns smoothly with the live recording, taking into account the tempo fluctuations observed in each individual segment. By combining the segmented parts, we achieve a more comprehensive and accurate alignment between the accompaniment and the live recording, improving the overall performance quality.

This segmentation and merging approach allows us to effectively address the challenge of inconsistent tempo in the live recording. By analyzing and aligning smaller segments individually, we can accommodate variations in tempo and achieve a more synchronized and cohesive musical experience between the accompaniment and the live performance.

3.6.4 Phase Vocoder

For using tempo detection and phase vocoder, we generated the graph of the shifted waveform before and after (see *Appendix 14*). Upon evaluating the result of using beat detection and phase vocoder for aligning live recordings with incorrect tempos, we discovered that the performance was significantly lacking. The underlying reason for this is that these methods rely on the overall tempo of the live recording to time-stretch the accompaniment audio for alignment. Consequently, while these methods may successfully match the start and ending of the two audios, they struggle to align the notes in the middle of the recording with the accompaniment effectively.

To address this challenge, a potential solution is to segment both the live recording and the accompaniment audio into smaller pieces. By breaking them down into manageable segments, we can apply tempo detection or phase vocoder individually to each sub-audio. This approach allows us to align the tempo of each segment between the live recording and the accompaniment more accurately. Following the segmentation and individual processing of the sub-audios, the next step is to merge these segments of the accompaniment back together. By doing so, we can obtain a better version of the accompaniment that aligns more precisely with the tempo variations in the live performance.

This segmentation and merging approach presents a potential solution to improve the alignment of accompaniment with live recordings that have incorrect tempos. By analyzing and aligning smaller segments individually, we can account for the tempo variations within each sub-audio and achieve a more synchronized and cohesive musical experience between the accompaniment and the live performance.

4. Future Development

In addition to the current functionality, there are several potential areas for future development and enhancement of the project. By incorporating these potential future developments, the project can evolve into a more versatile and comprehensive tool for musicians. These enhancements would provide users with a personalized and interactive practice environment, facilitating skill development, and promoting musical expression.

4.1 Adjustable tempo

Introduce the capability for users to modify the tempo of the accompaniment according to their preferences. This feature would enable more comfortable practice sessions and allow users to focus on specific sections of the music at their desired tempo.

4.2 Metronome integration

Incorporate a built-in metronome within the project's interface. The metronome would provide audible beats to help users maintain a steady tempo and stay in sync with the accompaniment. This feature would enhance the overall rhythmic accuracy and timing during practice sessions.

4.3 Looping sections

Implement the ability for users to easily loop specific parts of the musical score. This functionality would enable users to practice challenging sections repeatedly, aiding in the mastery of difficult passages and improving overall performance precision.

5. Conclusion

In this project, we implemented four strategies to generate accompaniment for live recordings, aiming to simulate performances without an accompanist. However, we encountered limitations and identified areas for improvement in aligning the accompaniment with the live recording. For the DTW approach, we addressed the jittering sound issue by applying a low-pass filter or generating a noiseless accompaniment using the optimal DTW path. Onset detection proved impractical due to the need for determining an optimal threshold for each recording. The tempo detection and phase vocoder methods worked well for recordings with consistent tempos but struggled with inconsistent tempos. We proposed segmenting the recording and aligning the accompaniment to each segment for better results. Overall, this project provided valuable insights into music synchronization, aiming to assist soloists in practicing without accompanists and enhancing their performances on stage.

6. Labour Distribution

Project Parts	Cheung Chi Wang	Chiu Long Him

Proposal sharing	✓	✓
Researching	✓	
Project presentation PowerPoint		✓
Coding: Data Preprocessing		✓
Coding: Dynamic Time Wrapping	✓	✓
Coding: Onset Detection		✓
Coding: Tempo Detection	✓	
Coding: Phase Vocoder	✓	
Coding: Further Studies - Metronome		✓
Project presentation	✓	✓
Report: 1. Abstract		✓
Report: 2. Background	✓	
Report: 3. Methodologies	✓	✓
Report: 4. Future Development		✓
Report: 5. Conclusion	✓	
Report: 6. References		✓
Report: Formatting		✓

6. References

[1] Generating live interactive music accompaniment using machine learning, https://www.duo.uio.no/bitstream/handle/10852/95694/UiO_Master_Thesis_benjamas.pdf?sequence=1 (accessed May 1, 2023).

[2] “Sync toolbox: A python package for efficient, robust, and accurate music synchronization¶,” Sync Toolbox: A Python Package for Efficient, Robust, and Accurate Music Synchronization - SyncToolbox 1.0.0 documentation, https://meinardmueller.github.io/synctoolbox/build/html/index.html (accessed May 1, 2023).

[3] “Noten Scannen und Bearbeiten: Ganz Einfach mit ScanScore,” ScanScore, https://scan-score.com/ (accessed April 13, 2023).

[4] “Dynamic time warpingâ¶,” dtw, https://musicinformationretrieval.com/dtw.html (accessed May 4, 2023).

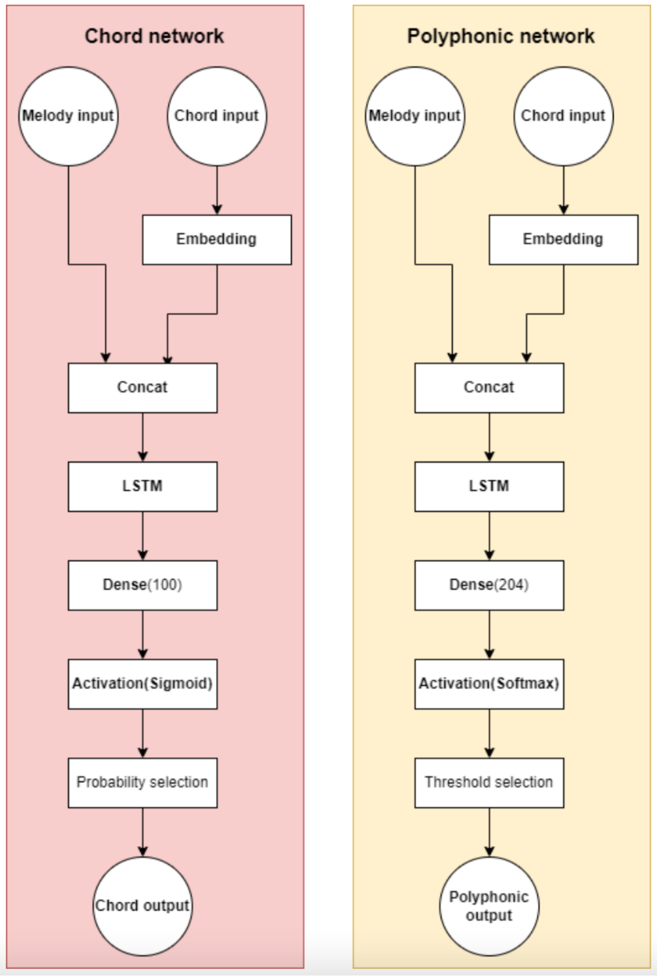
[5] “Fast fourier transform,” Wikipedia, https://en.wikipedia.org/wiki/Fast_Fourier_transform#Definition (accessed May 4, 2023).

[6] “Phase vocoder,” Wikipedia, https://en.wikipedia.org/wiki/Phase_vocoder (accessed May 6, 2023).

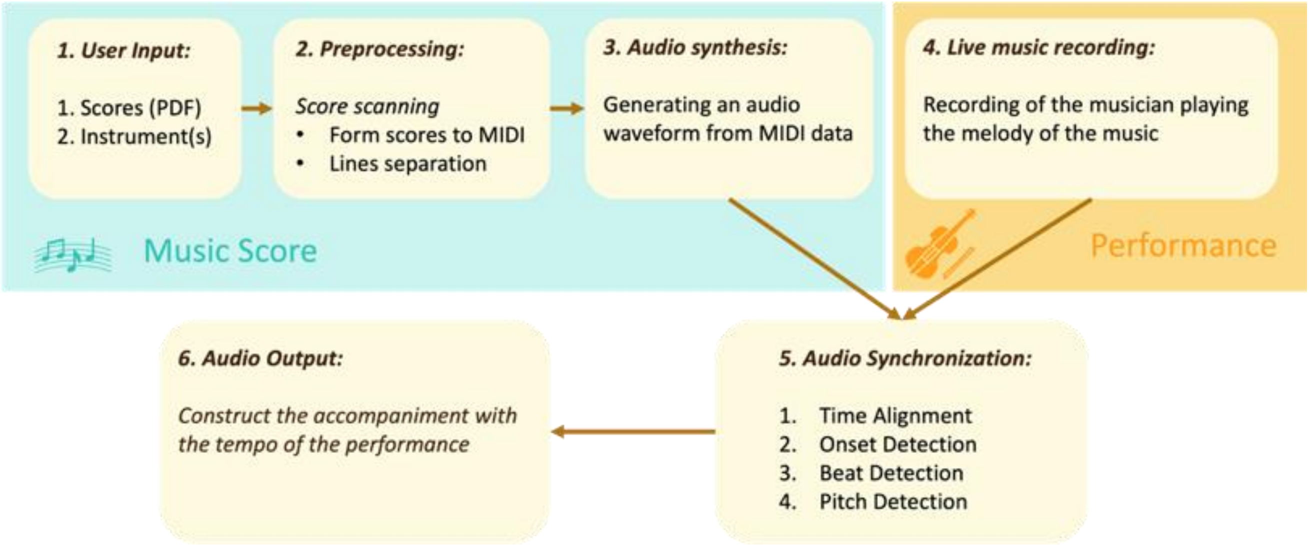
7. Appendix

This is the link to our project code in Google Colab:

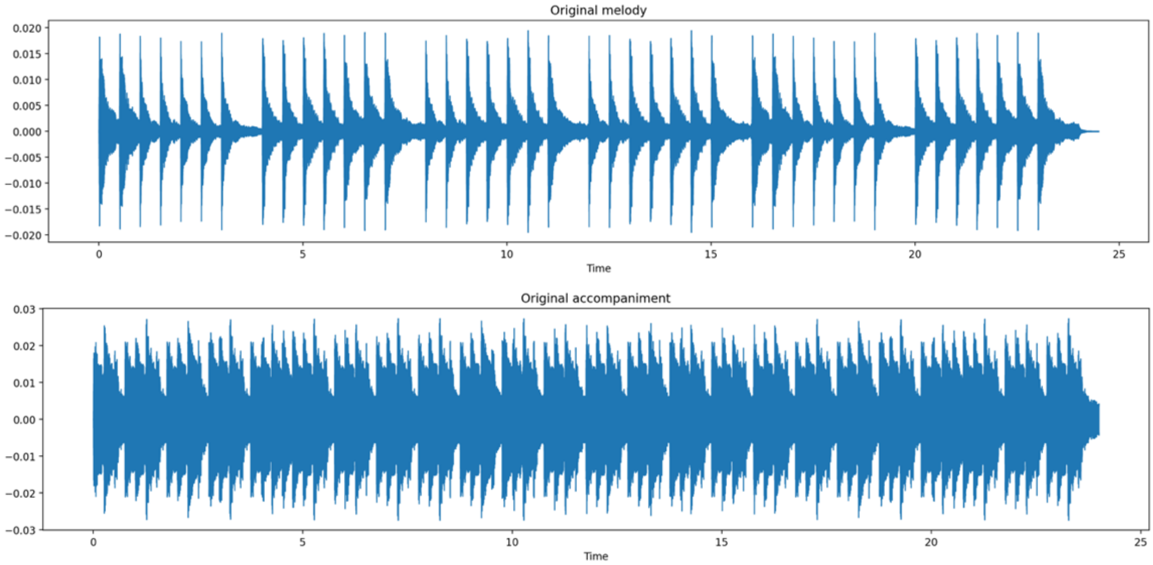
https://colab.research.google.com/drive/1F8eL37bPRGJy90J13iZ9yGzV6AAit3O6?usp=sharing



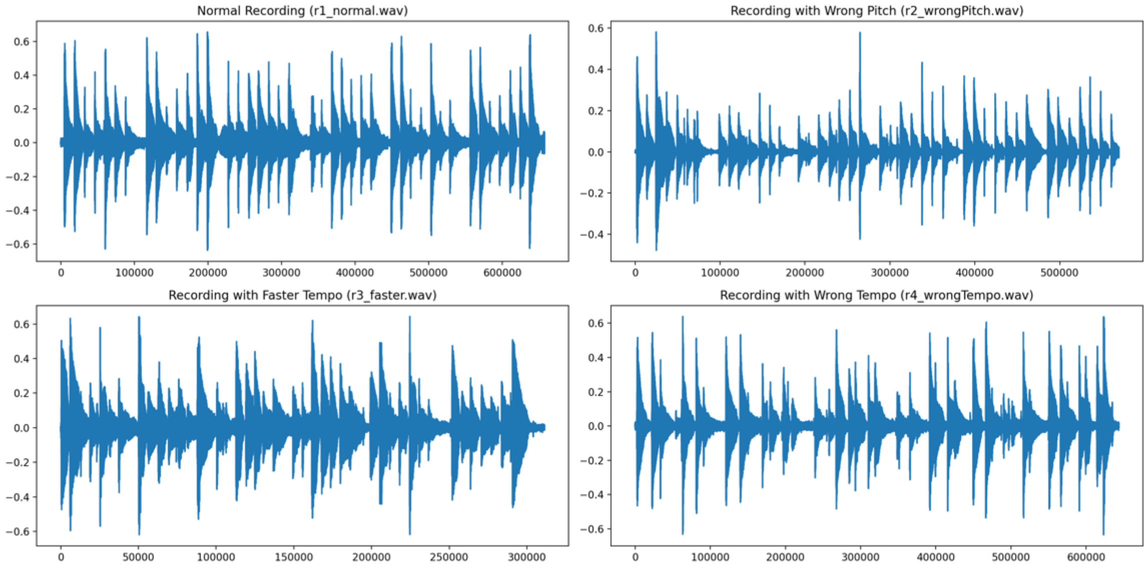
Appendix 1 Architecture of Chord and Polyphonic Generative Network



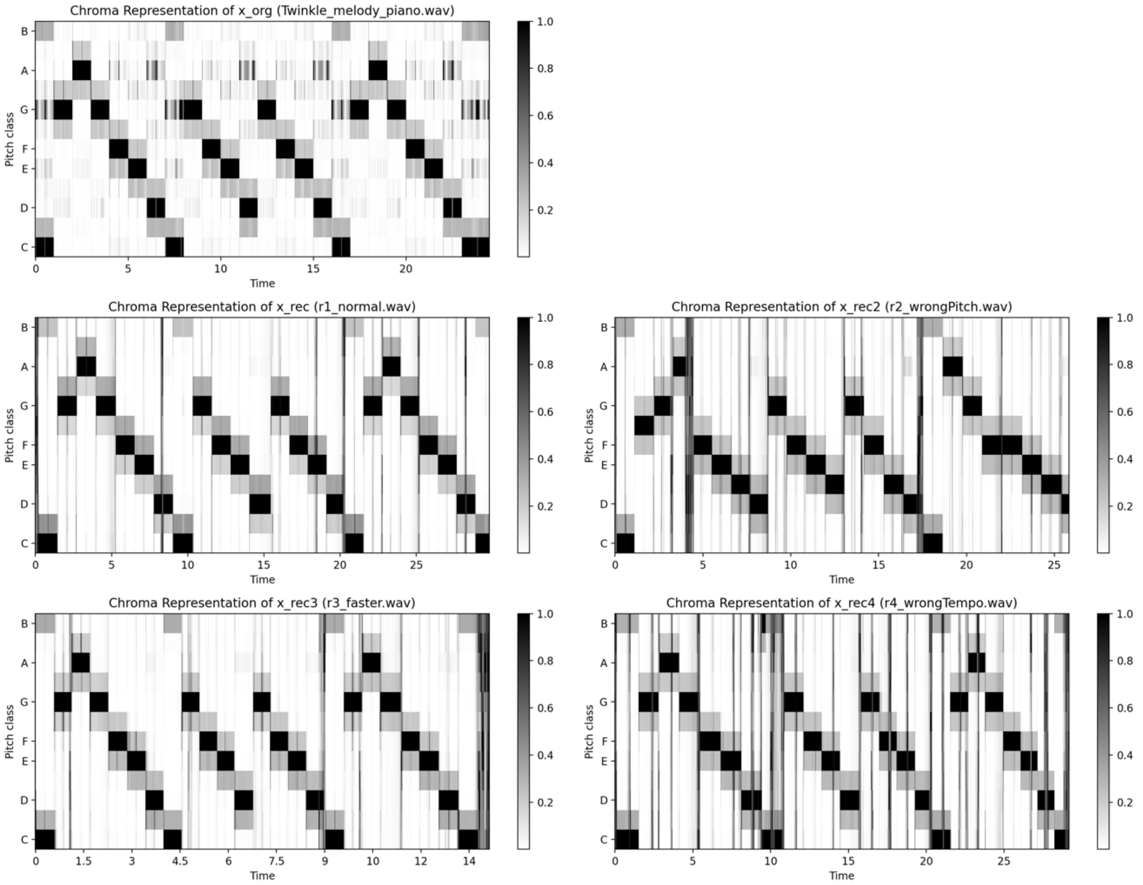
Appendix 2 Automatic Accompaniment Generation Pipeline



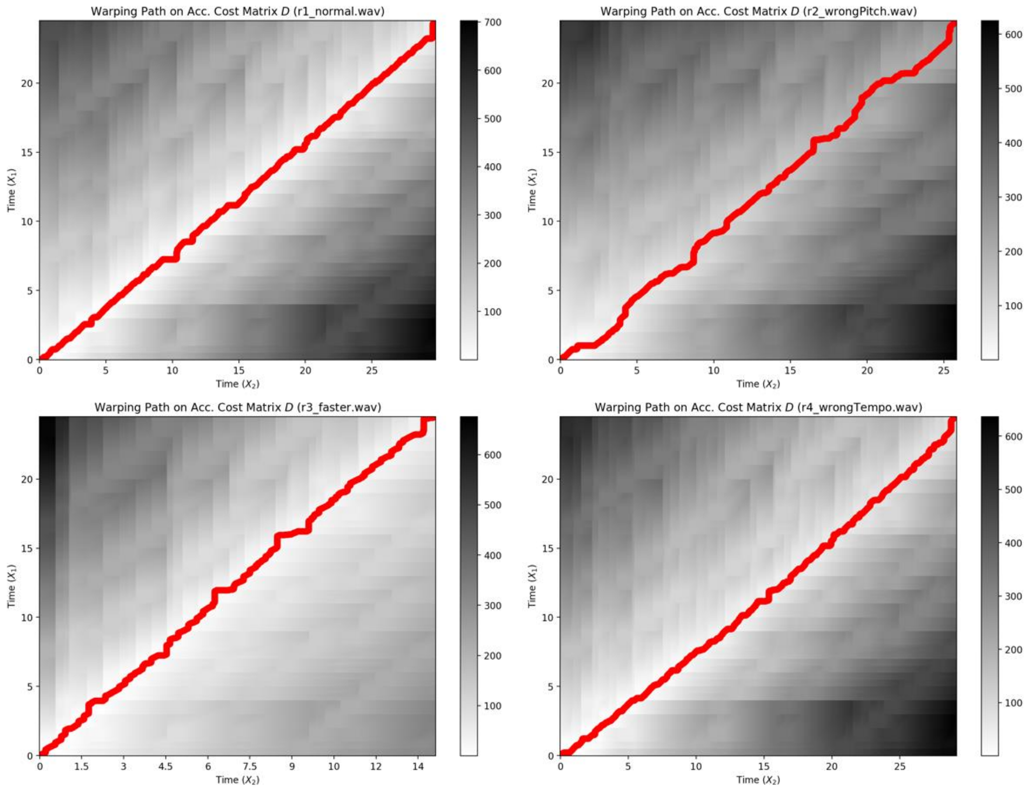
Appendix 3 The Waveform of Original Melody and Accompaniment



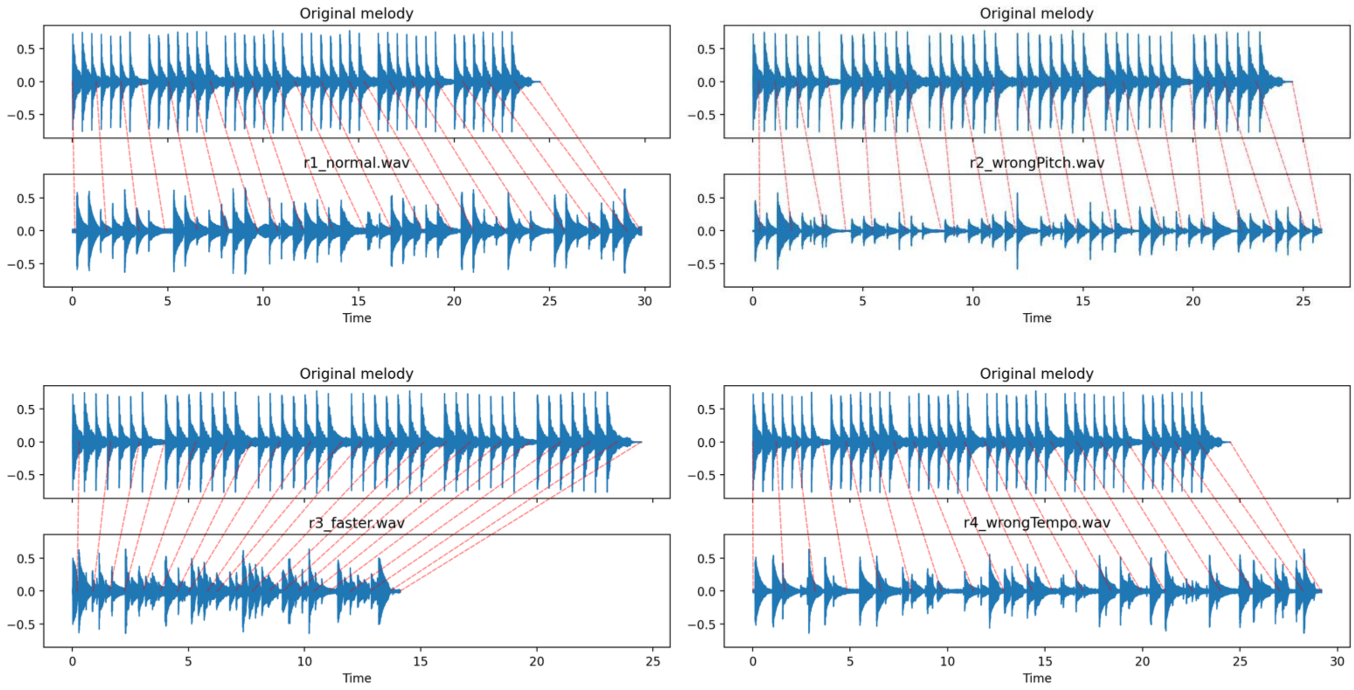
Appendix 4 The Waveforms of the Live Recordings



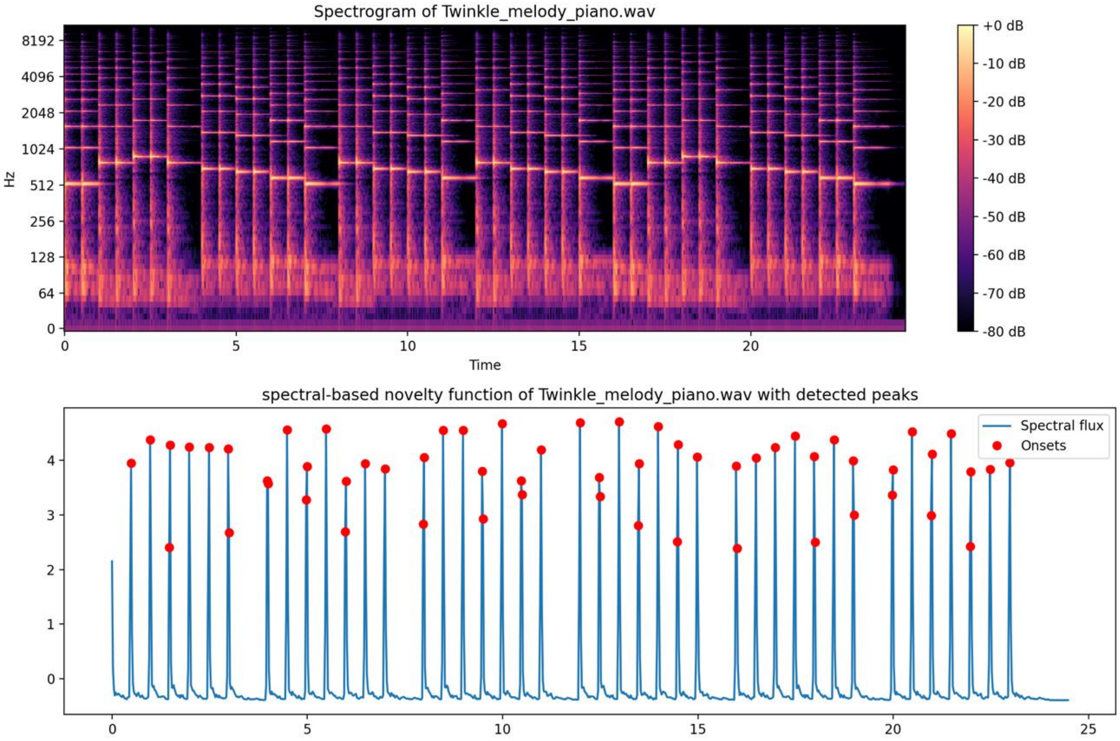
Appendix 5 Chromogram of Different Audio Data



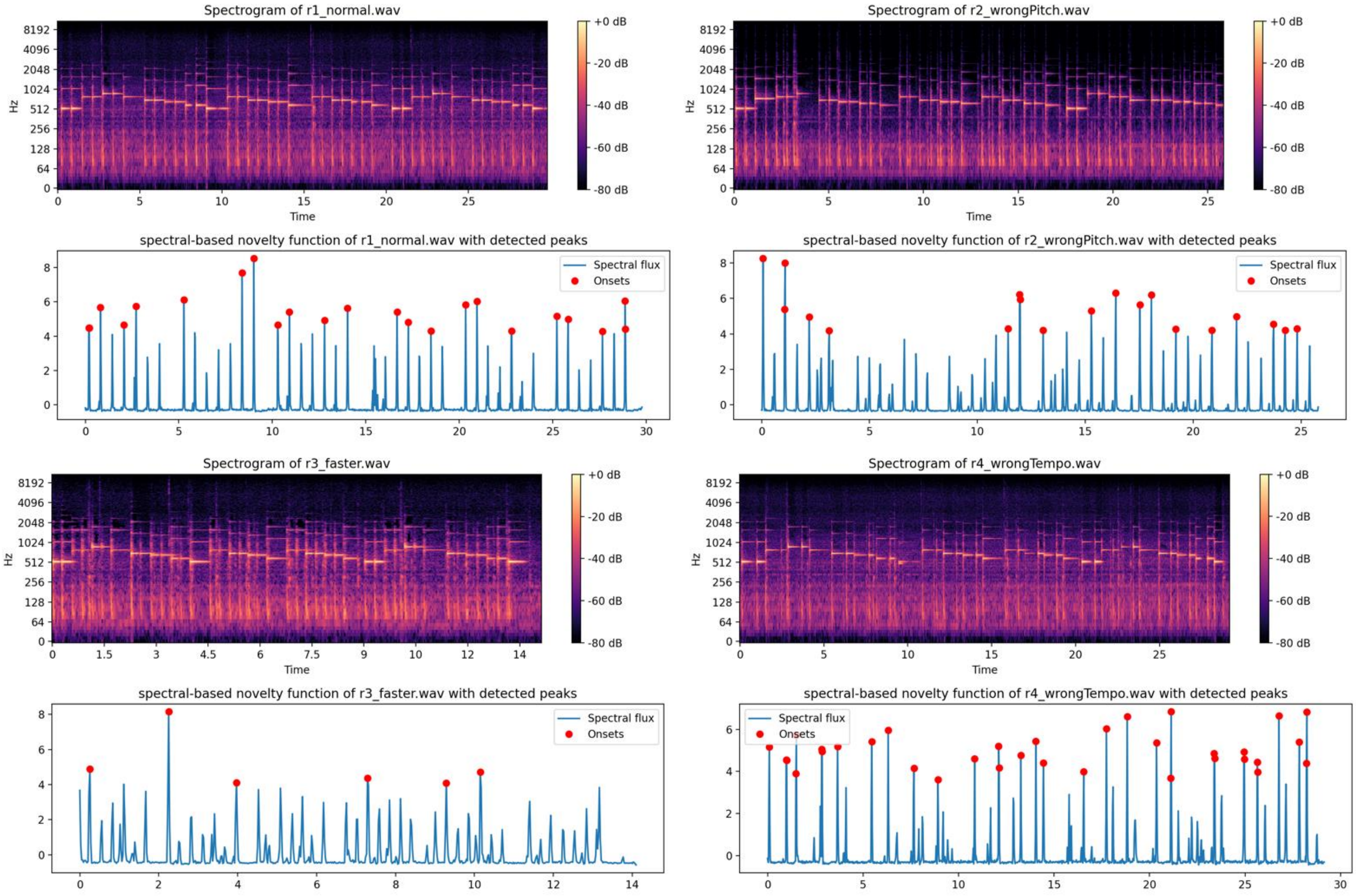
Appendix 6 Warping Path On Cost Matrix by the Melody and Different Audio Data



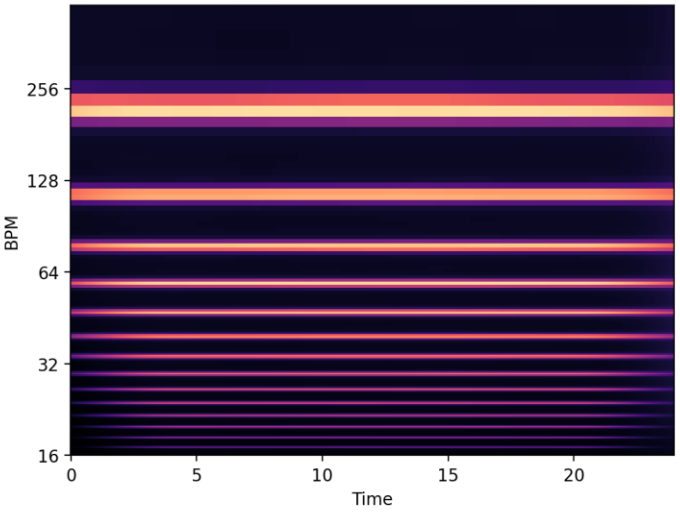
Appendix 7 Matching Waveform of the Melody and Different Audio Data using DTW



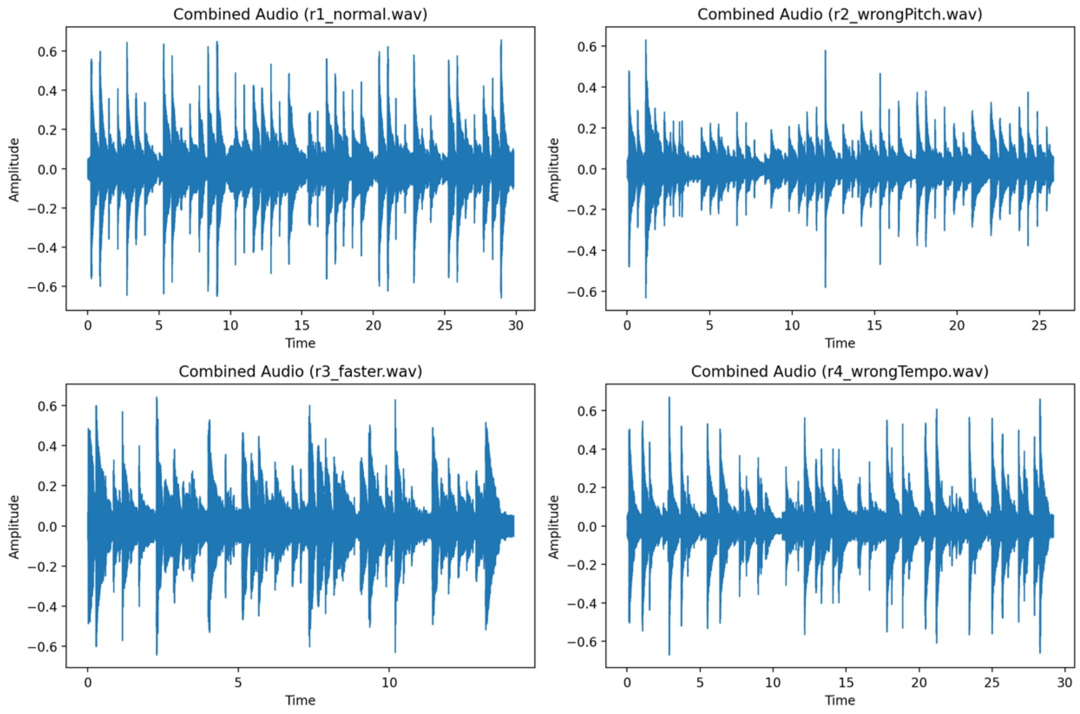
Appendix 8 Spectrogram and Spectral-based Novelty Function of the Melody



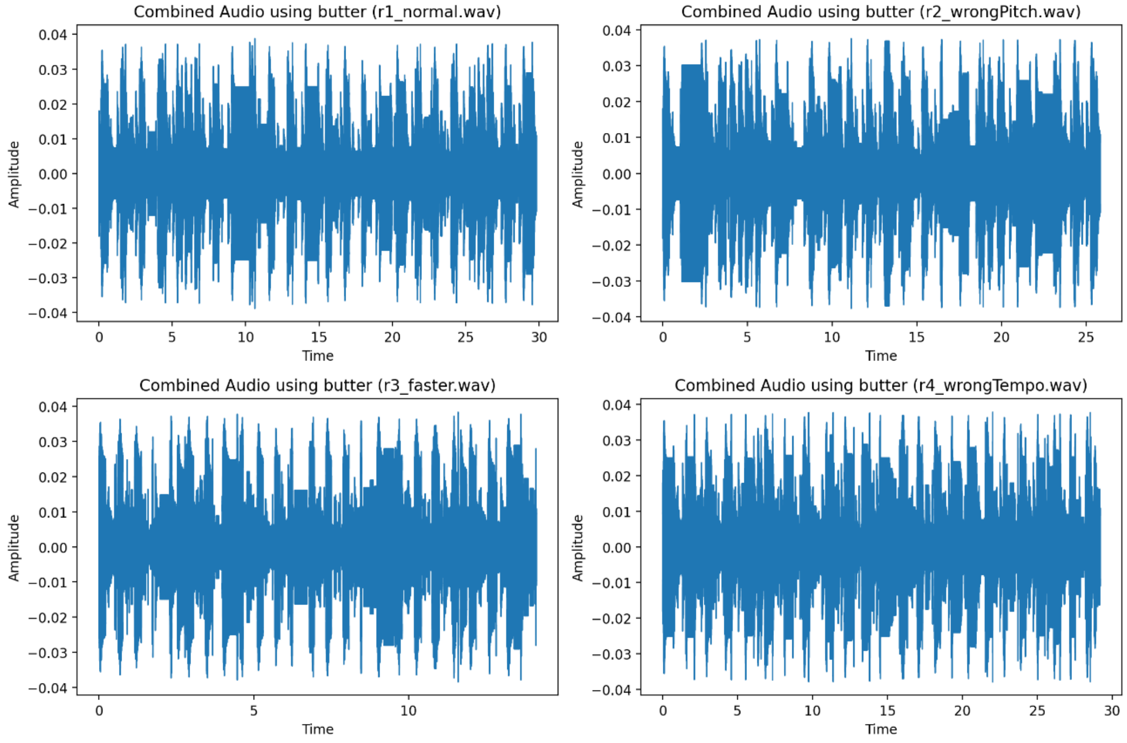
Appendix 9 Spectrogram and Spectral-based Novelty Function of Different Audio Recording



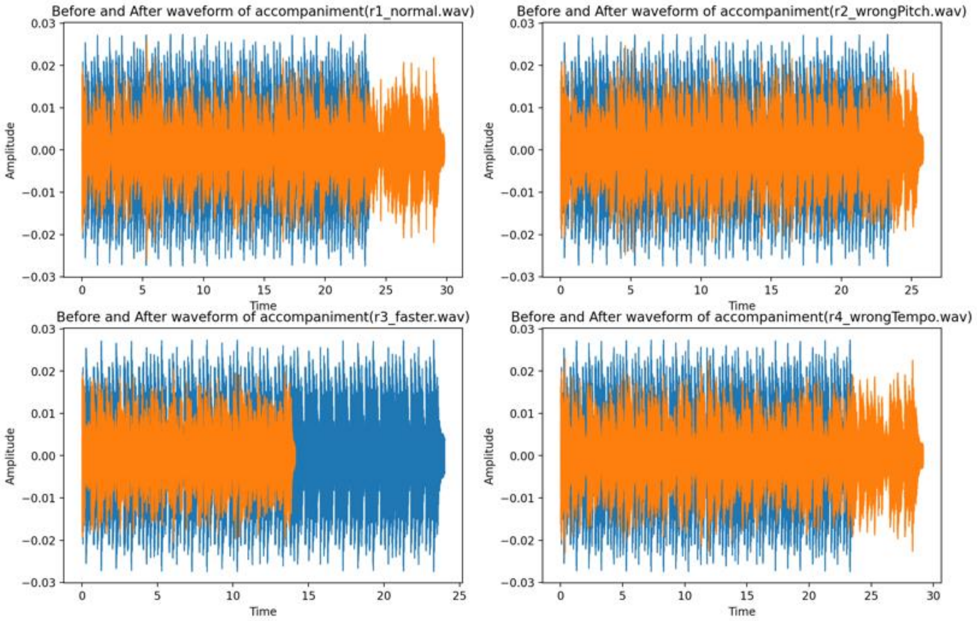
Appendix 10 Tempogram of the Accompaniment Audio



Appendix 11 Combined Waveform of the Melody and Different Audio Data using DTW

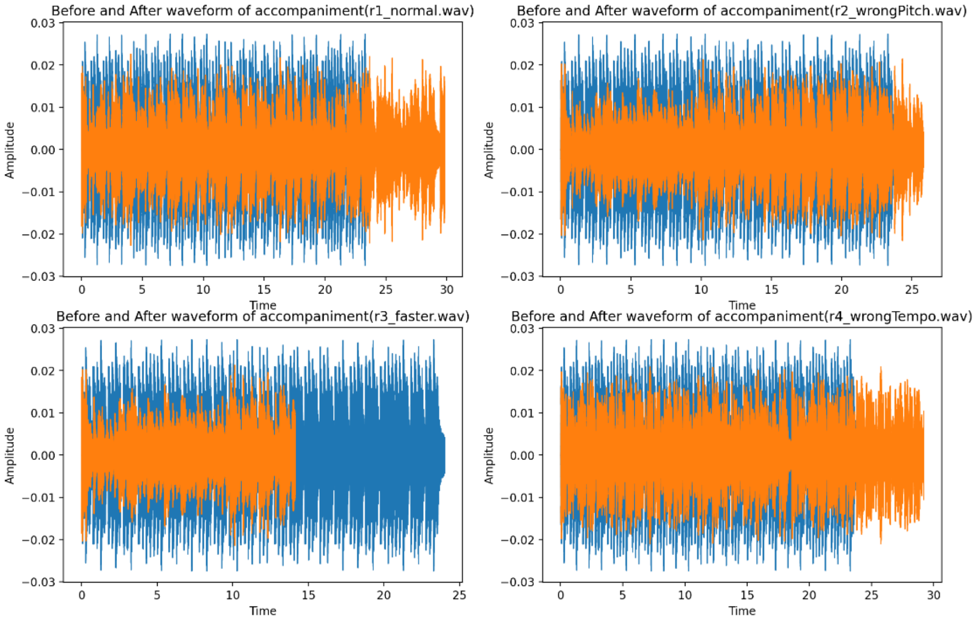


Appendix 12 Combined Waveform of the Melody and Different Audio Data using DTW Followed by Butter



Appendix 13 The Before-After Waveform of Recordings using Tempo Detection

(Blue: before, Orange: after)



Appendix 14 The Before-After Waveform of Recordings using Tempo and Phase Vocoder

(Blue: before, Orange: after)