# MOTION2MUSIC

**KONSTANTINOS PAPAANDREA**

University of Cyprus
Department of Computer Science

**Department of Computer Science**

**May 2026**

UNIVERSITY OF CYPRUS

Faculty of Pure and Applied Sciences

Department of Computer Science

# MOTION2MUSIC

**KONSTANTINOS PAPAANDREA**

Advisor:

Dr ANDREAS ARISTIDOU

The Thesis was submitted in partial fulfillment of the requirements for obtaining the Computer Science degree of the Department of Computer Science of the University of Cyprus

**May 2026**

## Acknowledgements

I want to thank Dr. Andreas Aristidou as my thesis advisor for his support, feedback and help during the creation of this project. I also want to thank the Department of Computer Science at the University of Cyprus for having an academic environment in which to complete my thesis. Finally, I want to thank my family for their patience, support and encouragement during my thesis and throughout my time in school.

## ABSTRACT

Motion2Music investigates how dance motion can be turned into prompts for automatic music compositions based on how that dance has been performed. This is done by developing a system that can generate music which matches the style of a dance motion's rhythm, energy and expressiveness through multi-levels of meaning rather than by simply mapping motion to audio. Each level of the Motion2Music will include some degree of interpretability when producing the final audio output. The last level of the Motion2Music process will take dance motion in BVH format, extract Laban Movement Analysis features from this motion, generate a beat and rhythm from the dance motion, create some kinematic motion descriptors, generate semantic audio descriptors, convert those into natural language prompt terms and then use MusicGen to generate the final audio output.

The system was trained and evaluated using dance motion and music data from AIST++, FineDance, and Simon Alexanderson/Motorica. FineDance was used as additional training data through its SMPL H motion representation, while the final web application uses BVH as the main input format. The final tempo model was evaluated as a clip-level tempo selector and obtained a strict Mean Absolute Error of 7.8184 BPM and an alias aware MAE of 3.6526 BPM on as set of 505 held out clips. The semantic prediction model used 256 motion derived features to predict six audio descriptors namely rms_mean, rms_std, centroid_mean, rolloff85_mean, bandwidth_mean, and flatness_mean. This resulted in an average $R^2$ of 0.4581, a median $R^2$ of 0.4086, and positive $R^2$ values for all six targets, with the best result on flatness_mean.

Through prompt wording experiments, it was discovered that MusicGen is sensitive to the language of the prompt. Compact, concrete musical wording worked better than raw technical labels, or overly dense prompt tags. The final application also introduces an auxiliary prototype selector, manual genre/style control and sectioned generation for longer dance motions. Our results suggest that the problem of motion-to-music generation is better viewed as an interpretable motion-to-prompt control problem. Motion cannot fully specify a single exact musical output, but can provide useful musical cues such as rhythm, energy, dynamics, brightness, spectrum width and texture. In summary, Motion2Music provides a practical framework to translate dance motion into a musical language that a text-to-music model can understand.

# TABLE OF CONTENTS

Acknowledgements ..... 4
ABSTRACT ..... 5
TABLE OF CONTENTS ..... 6
LIST OF TABLES ..... 9
LIST OF FIGURES ..... 10
LIST OF ABBREVIATIONS ..... 11
1 Introduction ..... 12
1.1 Background and Motivation ..... 12
1.2 Problem Statement ..... 12
1.3 Aim and Objectives ..... 12
1.4 Research Questions ..... 12
1.5 Thesis Contribution ..... 12
2 Research ..... 12
2.1 Motion to Music Generation ..... 12
2.2 Dance Motion Representation ..... 12
2.3 Motion Feature Extraction ..... 12
2.4 Laban Movement Analysis and Expressive Motion ..... 12
2.5 Rhythm and Tempo Estimation from Motion ..... 12
2.6 Audio Descriptors and Music Information Retrieval ..... 12
2.7 Semantic Representation of Music ..... 12
2.8 Text to Music Generation ..... 12
2.9 Related Work ..... 12
2.10 Research Gap and Thesis Direction ..... 12
3 Methodology ..... 12
3.1 Overview of the Proposed System ..... 12
3.2 Input Motion Data ..... 12
3.3 BVH Loading and Motion Preprocessing ..... 12
3.4 LMA Based Motion Feature Extraction ..... 12
3.5 Rhythm, Beat, and Tempo Feature Extraction ..... 12
3.6 Motion Feature Representation ..... 12
3.7 Tempo Reference and BPM Handling ..... 12

3.8 Learned Semantic Layer ..... 12
3.9 Auxiliary Prototype and Groove Phrase Selection ..... 12
3.10 Conversion from Audio Descriptors to Semantic Tags ..... 12
3.11 Prompt Generation ..... 12
3.12 Music Generation with MusicGen ..... 12
3.13 Sectioned Generation for Longer Motions ..... 12
3.14 Web Application Design ..... 12
3.15 Summary of the Methodology ..... 12
4 Results/ Findings ..... 12
4.1 Overview ..... 12
4.2 Tempo Model Results ..... 12
4.3 Tempo Generalization Across Datasets ..... 14
4.4 Semantic Audio Descriptor Prediction Results ..... 15
4.5 Per Target Semantic Results ..... 17
4.6 Prompt Wording Evaluation ..... 18
4.7 Prompt Wording Findings ..... 18
4.8 Sectioned Generation Findings ..... 19
4.9 MusicGen Continuation Findings ..... 21
4.10 Findings from Earlier Development Attempts ..... 21
4.11 Overall Findings ..... 22
5 Discussion/ Interpretation ..... 23
5.1 Overview ..... 23
5.2 Motion to Music as an Ambiguous Mapping ..... 23
5.3 Interpretation of Tempo Results ..... 23
5.4 Interpretation of Semantic Descriptor Results ..... 24
5.5 Prompt Engineering and MusicGen Behaviour ..... 26
5.6 Importance of the Multistage Pipeline ..... 27
5.7 Interpretation of Sectioned Generation ..... 27
5.8 Limitations ..... 28
5.9 Future Work ..... 28
5.10 Summary ..... 29
6 Summary of Findings/ Recommendations ..... 31
6.1 Overview ..... 31

6.2 Summary of the Thesis ..... 31
6.3 Summary of Main Findings ..... 31
6.4 Practical Recommendations ..... 32
6.5 Recommendations for Future Work ..... 33
6.6 Final Conclusion ..... 33
BIBLIOGRAPHY ..... 33

# LIST OF TABLES

Table 1: Base learned semantic word bank used for normal prompt generation. ..... 12
Table 2: Final tempo model overall results ..... 12
Table 3: Tempo model per dataset results ..... 14
Table 4: Semantic descriptor meanings. ..... 15
Table 5: Final semantic model summary ..... 17
Table 6: Semantic model per target results ..... 17
Table 7: Best aggregate prompt variants ..... 18
Table 8: Sectioned-safe tuned word bank derived from prompt wording evaluation. ..... 19
Table 9: Sectioned generation development findings. ..... 20

# LIST OF FIGURES

Figure 1: Motion2Music Pipeline ..... 12
Figure 2: BVH dance motion preview displayed as an animated skeleton in the web interface. ..... 12
Figure 3: Example of a generated auxiliary groove phrase. ..... 12
Figure 4: Mapping from predicted audio descriptors to semantic categories. ..... 12
Figure 5: Base Prompt vs Sectioned-safe Prompt ..... 12
Figure 6: Example of a generated prompt combining semantic descriptors, BPM, and fixed control phrases. ..... 12
Figure 7: Sectioned Generation for long motion clips pipeline. ..... 12
Figure 8: Motion2Music web application interface showing BVH upload, motion preview, prompt generation, and audio output controls. ..... 12
Figure 9: Tempo Estimation Summary ..... 13
Figure 10: Dataset Test Clips with Alias-Aware Error ..... 14
Figure 11: Tempo Accuracy Thresholds by Dataset ..... 15
Figure 12: Overview of the Motion2Music pipeline from BVH dance motion input to MusicGen audio synthesis and web application output. ..... 31

# LIST OF ABBREVIATIONS

| Abbreviation | Meaning |
|---|---|
| AI | Artificial Intelligence |
| AIST++ | AIST++ Dance Motion Dataset |
| BPM | Beats Per Minute |
| BVH | Biovision Hierarchy |
| CSV | Comma Separated Values |
| FFT | Fast Fourier Transform |
| ICCV | International Conference on Computer Vision |
| LMA | Laban Movement Analysis |
| MAE | Mean Absolute Error |
| MIR | Music Information Retrieval |
| ML | Machine Learning |
| NLP | Natural Language Processing |
| $R^2$ | Coefficient of Determination |
| RMS | Root Mean Square |
| SMPL | Skinned Multi Person Linear Model |
| SMPL H | Skinned Multi Person Linear Model with Hands |
| UI | User Interface |
| VR | Virtual Reality |
| WAV | Waveform Audio File Format |

# 1 Introduction

# 2 Research

# 3 Methodology

# 4 Results/ Findings

## 4.1 Overview

This chapter presents the main results and findings of the Motion2Music system. The evaluation focuses on four main parts of the project: tempo estimation, semantic audio descriptor prediction, prompt wording evaluation, and long clip sectioned generation. In addition to the quantitative results, this chapter also presents important findings discovered during development, because several important design choices were made after observing what worked and what failed in practice.

The system is evaluated as a motion to prompt control pipeline. Therefore, the results focus on tempo guidance, semantic descriptor prediction, prompt quality, and generation stability.

## 4.2 Tempo Model Results

Tempo is one of the most important musical properties for dance, because the generated music must follow the movement timing of the dancer. The tempo experiment evaluated whether motion derived rhythm signals could estimate the folded audio tempo of the corresponding music.

The final tempo model was evaluated as a clip level tempo selector. Each motion/audio item was treated as one clip for final evaluation, but each clip was expanded into several candidate tempo rows before training. The model generated candidate tempo from motion signals such as root speed, pelvis vertical motion, body energy, and left/right foot speed. These candidates were created using Fast Fourier Transform (FFT) peaks, autocorrelation peaks, local window tempo estimates, and octave related alternatives such as half time, original tempo, and double time. The train/test split was done at the clip level, so candidate rows from a test clip were not used during training. The final evaluation used 505 held out clips across AIST++, FineDance, and Simon Alexanderson/Motorica data.

***Table 2: Final tempo model overall results***

| **Metric** | **Value** |
|---|---|
| Train clips | 1180 |
| Test clips | 505 |
| Train candidate rows | 50514 |
| Test candidate rows | 21573 |
| Strict MAE | 7.81 BPM |
| Strict drift30 | 3.9 beats |

| | |
|---|---|
| Within 5 BPM | 80.59% |
| Within 2 BPM | 76.63% |
| Alias aware MAE | 3.65 BPM |
| Alias aware drift30 | 1.82 beats |
| Alias aware within 5 BPM | 83.56% |
| Alias aware within 2 BPM | 78.61% |

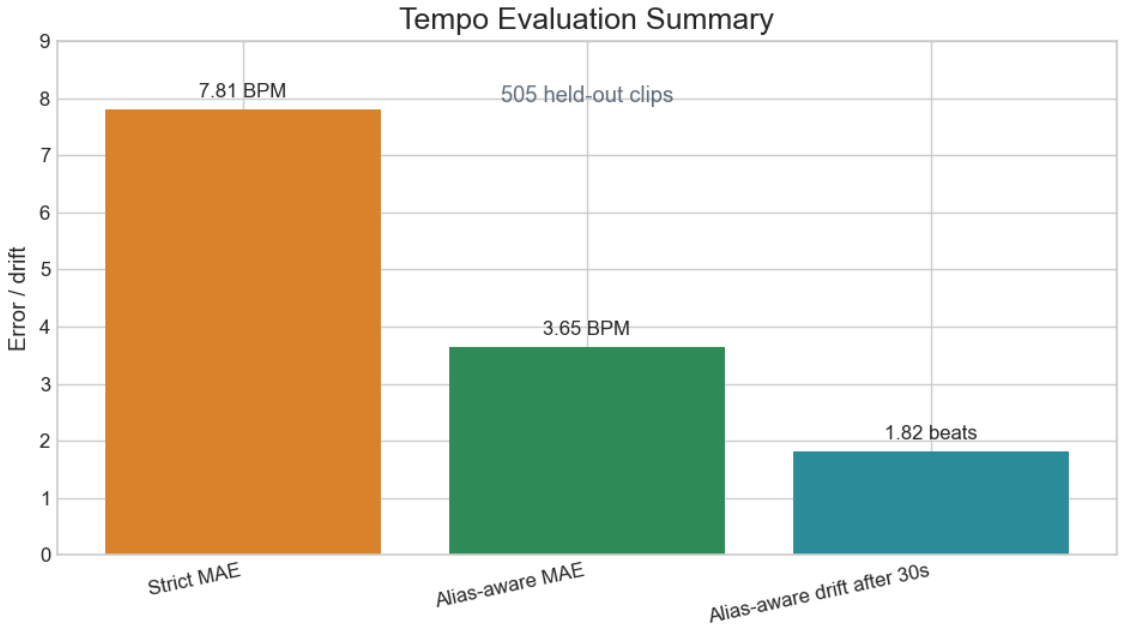


***Figure 9: Tempo Estimation Summary***

The strict MAE was 7.8184 BPM, while the alias aware MAE was 3.6526 BPM. This difference is important because dance tempo can be ambiguous. A dancer may move in a way that corresponds to half time or double time relative to the music. For example, a movement pattern may reasonably match both 80 BPM and 160 BPM depending on how the beat is interpreted. Therefore, the alias aware metric gives a more musically meaningful evaluation than strict BPM error alone.

Moreover, the drift after 30 seconds shows the practical effect of tempo error over time. Although the alias aware MAE of 3.65 BPM is relatively low, even small BPM differences can accumulate during playback. In this case, the alias aware drift after 30 seconds is approximately 1.83 beats, meaning that the generated music may gradually move out of perfect synchronisation with the original dance timing. This shows why tempo estimation is useful, but also why exact beat alignment remains challenging.

The result shows that the tempo model can often identify a musically close tempo candidate, even when the exact folded tempo is not perfectly predicted. This supports the use of tempo estimation as a guide for prompt generation, but it also shows that tempo from motion is not always exact.

### 4.3 Tempo Generalization Across Datasets

The tempo model did not perform equally across all datasets. The best performance was obtained on AIST++, while FineDance and Simon Alexanderson/Motorica were more difficult.

***Table 3: Tempo model per dataset results***

| Dataset | Test clips | Strict MAE | Within 5 BPM | Within 2 BPM | Alias aware MAE |
|---|---|---|---|---|---|
| AIST++ | 422 | 5.5978 | 88.15% | 85.78% | 2.2894 |
| FineDance | 61 | 16.8691 | 44.26% | 29.51% | 9.9420 |
| Simon Alexanderson / Motorica | 22 | 25.3173 | 36.36% | 31.82% | 12.3628 |

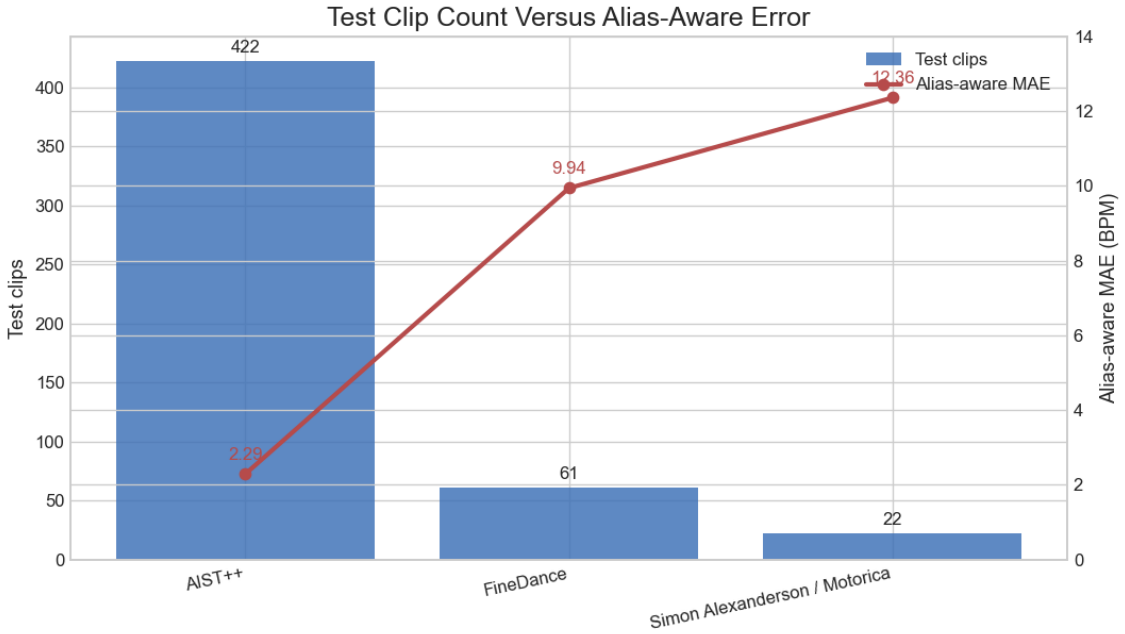


***Figure 10: Dataset Test Clips with Alias-Aware Error***

The AIST++ subset achieved the strongest result, with a strict MAE of 5.5978 BPM and an alias aware MAE of 2.2894 BPM. FineDance and Simon Alexanderson/Motorica were more challenging, with strict MAE values of 16.8691 BPM and 25.3173 BPM, respectively.

This suggests that tempo prediction from motion is highly dataset dependent. This happens because AIST++ contains more regular dance/music relationships, making tempo easier to infer from movement. FineDance and Simon Alexanderson/Motorica appear to contain more diverse or less rhythmically regular motion, which makes tempo prediction harder. This is an important limitation because it shows that motion to tempo estimation does not generalize equally well to every style of dance.

The finding also supports the decision to use BPM folding and fallback logic in the final application.

Also its important to mention that the error happens mostly not because the BPM is always a little far off but because of the few times that we are not catching the real BPM entirely. As mentioned in Table 3 on AIST++ 88% of clips had less than 5 BPM MAE where 85% had less than 2 BPM. This suggest that every time we have clear signal we are almost perfectly aligned with the original BPM.

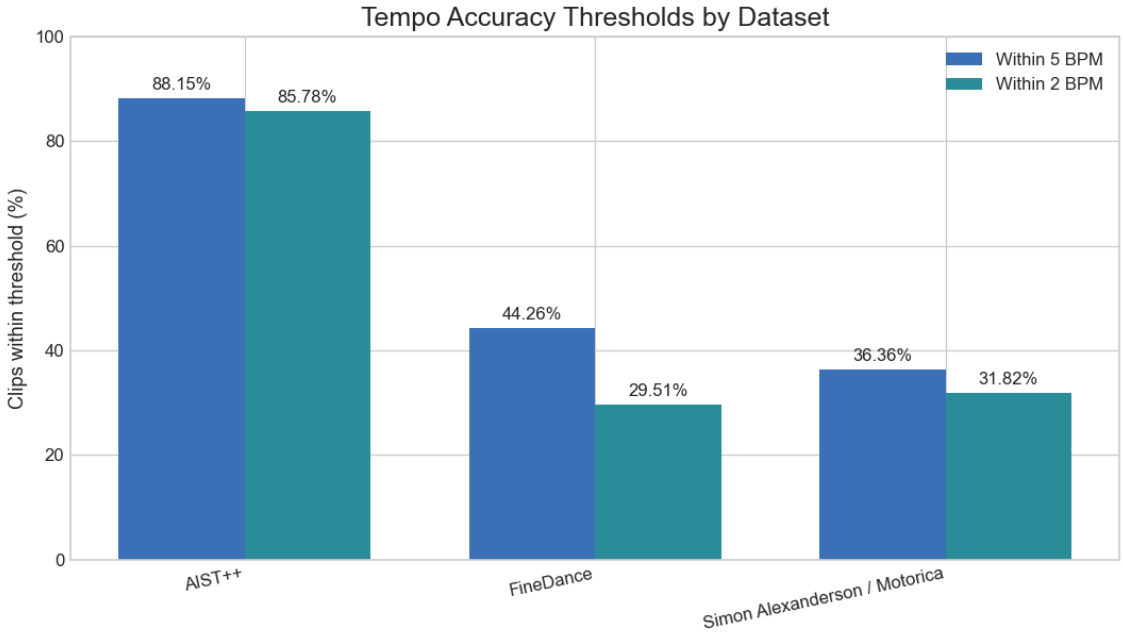


***Figure 11: Tempo Accuracy Thresholds by Dataset***

### 4.4 Semantic Audio Descriptor Prediction Results

The second major experiment evaluated whether motion derived features could predict useful audio descriptors. This stage is important because the final prompt uses semantic categories such as energy, dynamics, brightness, spectrum width, and texture. These categories are based on predicted audio descriptors.

The target audio descriptors were extracted from the paired music tracks using MusicExtractor from Essentia [10]. MusicExtractor produced a larger set of audio features, but the final model did not use all of them as prediction targets. Some descriptors were less stable, harder to predict from motion, or less useful for prompt generation. Therefore, the final semantic model focused on the six most useful and stable descriptors: rms_mean, rms_std, centroid_mean, rolloff85_mean, bandwidth_mean, and flatness_mean. These descriptors were selected because they can be translated into meaningful prompt categories such as energy, dynamics, brightness, spectrum width, and texture.

***Table 4: Semantic descriptor meanings.***

| Descriptor | Signal meaning | Plain language explanation | Meaning in our system |
|---|---|---|---|

| | | | |
|---|---|---|---|
| rms_mean | Average signal energy/loudness over the audio window. | Measures how strong the audio signal is on average. Quiet/sparse passages have lower RMS loud, powerful, or dense passages have higher RMS. | Controls how energetic or intense the music should feel. |
| Rms_std | Variation of loudness/energy over time. | Measures how much loudness changes during the window. A steady loop has low variation; music with swells, accents, drops, or changing intensity has higher variation. | Controls how dynamically the music changes over time. |
| Centroid_mean | Spectral centroid, often perceived as brightness. | The "center of gravity" of the sound spectrum. More high frequency energy makes the sound brighter, sharper, or more present, lower energy feels warmer or darker. | Helps decide whether the sound should feel bright or warm. |
| Rolloff85_mean | Frequency below which 85% of the spectral energy lies. | Shows how far the sound extends into high frequencies. Low rolloff suggests darker/muffled sound, high rolloff suggests clearer top end, brighter percussion, or airy synths. | Helps decide the amount of high frequency top end. |
| Bandwidth_mean | Spread of the spectrum around the centroid. | Measures whether the sound is concentrated in a narrow band or spread across bass, mid, and highs. Low bandwidth feels focused/compact, high bandwidth feels fuller, wider, or more open. | Controls how narrow, wide, full, or open the arrangement/mix should feel. |
| Flatness_mean | Spectral flatness/noisiness. | Distinguishes smooth pitched sound from noisy/percussive sound. Low flatness is tonal, like clean synth or bass, high flatness is noise like, like percussion, grain, hiss, or textured rhythmic detail. | Controls the perceived texture of the music. |

The final semantic model used 4 second windows with a 4 second hop. Each window contained motion features from the corresponding dance segment and audio descriptors from the matching audio segment. The combined final model used AIST++, FineDance, and Simon Alexanderson/Motorica data, producing 11,484 training windows and 4,747 test windows. The input vector contained 256 motion derived features, including LMA features, beat/motion alignment features, and kinematic features such as root speed, pelvis motion, body energy, foot speed, foot contact balance, periodicity, and motion

derived tempo summaries. The output vector contained the six selected audio descriptors listed above.

***Table 5: Final semantic model summary***

| Metric | Value |
|---|---|
| Train rows | 11,484 |
| Test rows | 4,747 |
| Train unique stems | 1,199 |
| Test unique stems | 486 |
| Input features | 256 |
| Output descriptors | 6 |
| Mean $R^2$ | 0.4581 |
| Median $R^2$ | 0.4086 |
| Targets with positive $R^2$ | 6 / 6 |

The final semantic model achieved a mean $R^2$ of 0.4581 and a median $R^2$ of 0.4086 across the six targets. All six targets achieved positive $R^2$ scores. This is an important result because it shows that the model learned meaningful relationships between dance motion and audio descriptors. However, the scores also show that the mapping is not perfect, which is expected because movement does not uniquely determine the exact audio content.

### 4.5 Per Target Semantic Results

The model predicted some audio descriptors better than others.

***Table 6: Semantic model per target results***

| Target | MAE | $R^2$ |
|---|---|---|
| flatness_mean | 0.0497 | 0.7793 |
| rms_mean | 0.0564 | 0.5032 |
| rms_std | 0.0261 | 0.4370 |
| rolloff85_mean | 1093.0413 | 0.3802 |
| centroid_mean | 490.8618 | 0.3348 |
| bandwidth_mean | 360.7832 | 0.3143 |

The strongest target was flatness_mean, with an $R^2$ of 0.7793. This suggests that motion features were especially useful for predicting texture related audio qualities. The RMS related targets also performed reasonably well, with rms_mean achieving an $R^2$ of 0.5032 and rms_std achieving 0.4370. Spectral descriptors such as rolloff, centroid, and bandwidth were more difficult, but they still achieved positive $R^2$ values.

This result supports the final semantic prompt design. Motion features appear to capture broad musical qualities such as texture, energy, and dynamics better than exact spectral detail. Therefore, the model is useful for guiding descriptive prompt terms, but it should not be described as reconstructing the original music. The correct interpretation is that the model learns approximate musical tendencies from movement.

A key development finding was that window level modelling worked better than treating the whole clip as one single sample. Earlier clip level semantic experiments

were weaker, especially for descriptor prediction. Moving to 4 second windows increased the number of training examples and allowed the model to capture local changes in the dance. This helped the final semantic layer become more useful for prompt generation.

### 4.6 Prompt Wording Evaluation

Prompt wording was one of the most important practical parts of the project. MusicGen does not respond equally to all descriptions, even when the descriptions mean similar things to a human. Therefore, different prompt wordings were tested to find which phrases produced generated audio closer to the intended descriptor profile.

The main prompt evaluation used 30 clips from AIST++, FineDance, and Simon Alexanderson/Motorica. For each case, multiple MusicGen outputs were generated using different prompt variants. The same audio feature extractor was then applied to both the generated audio and the original/reference audio. The generated outputs were ranked using feature distance, where lower distance means the generated audio was closer to the reference audio according to the selected descriptors. This was an automatic proxy evaluation, not a full human listening evaluation. Human listening was used qualitatively to check whether the automatic winners were also musically usable.

All prompt variants in the final sweep used the five semantic variables but changed the wording and sentence structure. This tested whether the issue was the variables themselves or the way they were expressed in natural language.

***Table 7: Best aggregate prompt variants***

| Rank | Variant | Clips | Mean overall distance | Wins |
|---|---|---|---|---|
| 1 | all5_energy_light_sentence | 30 | 0.3397 | 1 |
| 2 | all5_instrument_compact | 30 | 0.3440 | 2 |
| 3 | all5_motion_tone_sentence | 30 | 0.3441 | 3 |
| 4 | all5_instrument_sentence | 30 | 0.3474 | 1 |
| 5 | all5_plain_sentence | 30 | 0.3511 | 4 |

The best aggregate variant was all5_energy_light_sentence, with a mean overall distance of 0.3397. However, the wins were distributed across several variants, which suggests that no single wording was best for every clip. This is important because it shows that prompt wording is not only a technical detail. It directly affects the behaviour of MusicGen.

### 4.7 Prompt Wording Findings

The prompt wording experiments showed that concrete musical words usually worked better than abstract motion words, especially for long sectioned generation. The following word bank does not replace the base learned semantic prompt. It was used to

create a safer prompt style for sectioned generation, after the prompt wording sweep showed that compact musical wording reduced instability.

***Table 8: Sectioned-safe tuned word bank derived from prompt wording evaluation.***

| Variable | Low | Medium | High |
|---|---|---|---|
| Energy | restrained motion feel | steady beat | strong kick pulse |
| Brightness | warm, rounded synth tone | warm clear synth color | bright synth color |
| Dynamics | steady groove contour | flowing | evolving drum and bass movement |
| Spectrum | focused midrange mix | even frequency balance | open mix shape |
| Texture | polished percussion | light textured surface | busy percussion detail |

A major finding was that more tags did not always improve the output. Early versions of the system used raw or excessive tags, such as direct labels for energy, dynamics, brightness, spectrum, and texture. These prompts often overcondition MusicGen and could make the generated music unstable or messy, especially during sectioned generation. Removing or simplifying some tags sometimes improved stability, which showed that prompt compactness mattered.

However, later experiments showed that the problem was not the five variables themselves. The problem was how they were translated into the prompt language. The final all five-sweep showed that all five variables could be useful if each one was mapped to stable musical wording. This supports the final design, where energy, dynamics, brightness, spectrum width, and texture are all kept, but the system uses different wording depending on the generation context.

These words were not the original base semantic prompt words. The base learned prompt used broader descriptors such as “driving,” “wide dynamic swells,” and “bright crisp upper frequencies.” The wording shown in this table was introduced later as a tuned sectioned-safe version, because long sectioned generation was more sensitive to prompt instability.

This was one of the most important development findings. It changed the project from simply “put all extracted variables into the prompt” to “translate variables into musical language that MusicGen responds to reliably.” It also led to the distinction between the base learned semantic prompt used for normal generation and the sectioned-safe prompt used for long generation.

### 4.8 Sectioned Generation Findings

Longer generations were another major practical challenge. MusicGen often becomes less stable after approximately 30 seconds. For longer dance motions, a single

generation call may drift away from the prompt, lose musical consistency, or fail to complete generation reliably. Because of this, the application includes sectioned generation.

During development, the first sectioned generation approach added extra section specific prompt words and used crossfade between generated chunks. This did not work well. Adding section specific words created audible instability and sometimes made the music sound as if multiple musical layers were stacked together. Importantly, this problem could appear from the start of the generated output, not only at section boundaries.

As a result, a sectioned-safe prompt wording was introduced. This prompt used the same semantic variables as the base learned prompt, but expressed them using more compact and stable musical phrases selected from the prompt wording sweep. The final decision was to remove extra section specific prompt tags, remove crossfade, keep the same sectioned-safe prompt across sections, and expose MusicGen continuation as an optional UI switch.

***Table 9: Sectioned generation development findings.***

| Tested approach | Finding | Final decision |
|---|---|---|
| Extra section specific prompt words | Caused prompt drift and musical layering | Removed |
| Crossfade between chunks | Did not solve the main instability | Removed |
| Same sectioned-safe prompt across sections | Improved consistency | Kept |
| MusicGen continuation | Sometimes smoother, sometimes unstable | Made optional |

The main interpretation is that the issue was not only the sectioning algorithm itself. The problem was also prompt drift. When each section receives additional or changing prompt information, MusicGen may reinterpret the music for every chunk. This can create a layered or inconsistent result. The final approach keeps the same sectioned-safe prompt across sections so that the generated track remains closer to one musical identity.

This finding is also important because it clarifies the role of sectioned generation in the thesis. Sectioned generation is a practical runtime solution, not the main research contribution. The main contribution remains the motion to the semantic to prompt pipeline. Although the final implementation uses the same sectioned-safe prompt across sections for stability, the section level motion analysis could be useful in future work with a more controllable music generation model. Such a model could accept changing section level conditions, instead of relying on one global prompt for the entire dance.

### 4.9 MusicGen Continuation Findings

MusicGen continuation was tested to make sectioned chunks connect more smoothly. Continuation can help preserve musical context because the next section receives a short part of the previous audio as input. However, continuation was not always better. In some cases, especially when the prompts were too descriptive or changed between sections, continuation reinforced unwanted artifacts.

For this reason, the final app does not force continuation. Continuation is exposed as a user-controlled option. This gives the user the ability to test smoother transitions when useful, while also allowing independent section generation when continuation causes instability.

The finding here is practical but important: generative audio continuation is sensitive to both audio context and prompt wording. A sectioned-safe prompt with optional continuation was more reliable than forcing continuation with complex section-specific instructions.

### 4.10 Findings from Earlier Development Attempts

Several earlier attempts helped shape the final system, even if they were not kept as the central final method.

At the beginning, it seemed logical to use the five semantic variables directly as prompt tags, such as low energy, steady dynamics, bright, wide spectrum, and noisy texture. In practice, this often made MusicGen over conditioned and unstable. Removing extra tags reduced the “music on top of music” effect, especially in sectioned generation. This showed that the prompt had to be designed around how MusicGen reacts, not only around what the variables technically mean.

The project also tested individual variables. Early tests suggested that texture/noise information was especially useful, while the full five variable version was not reliable when the wording was too raw. Later, after prompt wording was improved, the five variable approach became more useful. This shows that the final success depended more on semantic translation and wording than on simply adding or removing variables.

Another important development direction was the move away from direct music reconstruction. Earlier direct regression experiments showed limited generalization when trying to predict exact audio properties or embeddings from motion at the clip level. This helped motivate the final design: instead of forcing the system to recover the original song, the system predicts broader audio descriptors and converts them into prompt language. This is more realistic because many different songs can fit the same dance.

The auxiliary prototype/groove phrase selector also came from earlier experimentation. In the final app, it is not the main semantic system, but it remains useful as an automatic style or groove hint. It helps the automatic prompt choose a broader direction when the user does not manually select a genre.

### 4.11 Overall Findings

The results show that Motion2Music is strongest when it is treated as an interpretable control pipeline rather than a direct audio prediction system.

The tempo model showed that motion contains useful rhythmic information, especially in datasets with clear dance/music alignment such as AIST++. However, the weaker results on FineDance and Simon Alexanderson/Motorica show that tempo prediction from motion is not equally reliable across all dance styles and datasets.

The semantic model showed that motion derived features can predict broad audio properties with meaningful accuracy. The strongest result was obtained for flatness_mean, followed by RMS related energy and dynamics descriptors. This supports the use of semantic descriptors for prompt control, while the spectral results show that detailed timbre and production choices remain difficult to infer from motion.

The prompt wording experiments showed that MusicGen is highly sensitive to wording. The same semantic idea can produce different results depending on how it is phrased. Concrete musical terms worked better than vague or overly technical language. This supports the final prompt generation design, where raw descriptor values are translated into musical phrases, with broader wording used for the base learned prompt and more compact wording used for sectioned-safe generation.

Finally, the long clip experiments showed that generation stability is a practical limitation of MusicGen. Sectioned generation is useful, but it must be handled carefully. Using the same sectioned-safe prompt across sections was more stable than adding extra section specific wording.

Overall, the findings support the final Motion2Music design: motion is first analysed through interpretable features, then converted into semantic musical descriptors, enriched with a broad groove direction when useful, and finally transformed into a MusicGen prompt. This makes the system understandable, adjustable, and more stable than a direct motion to audio generation approach.

# 5 Discussion/ Interpretation

## 5.1 Overview

This chapter interprets the results from Chapter 4 and explains what they reveal about motion to music generation, prompt-based control, and the limitations of the final system.

## 5.2 Motion to Music as an Ambiguous Mapping

One of the most important findings of this project is that the relationship between dance and music is ambiguous. The same dance motion can match many different musical outputs. For example, a fast and powerful dance sequence could match hip hop, electronic dance music, cinematic percussion, or even orchestral music. Similarly, a slow and smooth motion could be interpreted as ambient, cinematic, contemporary, or emotional music.

This means that the goal of the system should not be to recover the exact original song. Recovering the exact song from motion alone would be unrealistic because the motion does not contain enough information about instrumentation, harmony, melody, production style, or arrangement. Instead, the more realistic goal is to generate music that is compatible with the movement.

This is why the final system focuses on intermediate musical properties. It predicts/estimates properties such as tempo, energy, dynamics, brightness, spectrum width, and texture. These properties do not describe one exact song, but they describe the kind of music that may fit the dance. This makes the problem more manageable and more explainable.

This interpretation also explains why the final application uses prompt-based music generation. A prompt allows the system to describe musical intention in a flexible way. Rather than forcing the model to produce one exact target, the system gives MusicGen a structured description of the desired musical character.

## 5.3 Interpretation of Tempo Results

The tempo results show that dance motion contains useful rhythmic information, but also that tempo estimation from motion is difficult. The final tempo selector achieved a strict MAE of 7.8184 BPM and an alias aware MAE of 3.6526 BPM over 505 held out test clips. The difference between these two metrics is important because tempo in dance can often be interpreted at half time or double time. For example, a movement may reasonably fit both 80 BPM and 160 BPM depending on how the beat is counted. This means that a strict BPM error can sometimes make a musically acceptable prediction look worse than it is.

The alias aware result is therefore more meaningful for this project. An alias aware MAE of 3.6526 BPM suggests that the system often finds a tempo that is musically close to the target, even if it is not always the same BPM value. However, even a small BPM error can accumulate over time. The alias aware drift after 30 seconds was approximately 1.8263 beats, which means that the generated music may gradually move away from perfect synchronization if exact beat alignment is required.

The per dataset results also reveal an important limitation. The model performed much better on AIST++ than on FineDance or Simon Alexanderson/Motorica. AIST++ achieved a strict MAE of 5.5978 BPM, while FineDance and Simon Alexanderson/Motorica had higher errors of 16.8691 BPM and 25.3173 BPM respectively. This happens because AIST++ contains more data but also because it has more regular or clearer dance/music alignment motion, like hip hop, while FineDance and Simon Alexanderson/Motorica contain more diverse, expressive, or less rhythmically regular motion.

This finding shows that tempo estimation from motion is dataset dependent. It works better when movement has a clear relationship with the musical beat, but it becomes harder when the dance is more expressive, less repetitive, or less directly beat driven.

### 5.4 Interpretation of Semantic Descriptor Results

The semantic audio descriptor model produced meaningful results across all six targets, and the findings reveal both the potential and the boundaries of what motion based prediction can realistically achieve.

The final model used 256 motion derived features and predicted six audio descriptors: rms_mean, rms_std, centroid_mean, rolloff85_mean, bandwidth_mean, and flatness_mean. It achieved a mean $R^2$ of 0.4581, a median $R^2$ of 0.4086, and positive $R^2$ values for all six targets. These results should not be interpreted as the model failing to fully reconstruct the original music. That was never the goal. The correct interpretation is that motion features can capture meaningful tendencies in the audio, tendencies that are consistent enough across the data to be learnable, but not so deterministic that a single motion sequence maps to one exact musical output.

The strongest result was flatness_mean, with an $R^2$ of 0.7793. This is the most interpretable finding in the semantic model. Spectral flatness measures how noise like or tonal a sound is, and this quality appears to have a strong and consistent relationship with how a body moves. Sharp, fragmented, or highly active movement tends to correspond to noisier, more percussive audio, while smoother or more sustained movement tends to correspond to more tonal sound. Because this relationship is relatively direct and consistent across dance styles, the model can learn it reliably. The flatness result suggests that motion is genuinely informative about musical texture, and that this is one of the stronger signals available in the motion to music mapping.

RMS related descriptors also performed reasonably well, with rms_mean achieving an $R^2$ of 0.5032 and rms_std achieving 0.4370. These results make intuitive sense. Overall energy and dynamic variation in music are broadly connected to the physical intensity and variability of the dancer's movement. A high energy dance sequence tends to correspond to louder or more intense music, and a dance with large changes in movement energy tends to correspond to music with more dynamic variation. These relationships are not perfect, but they are consistent enough for the model to capture them.

The spectral descriptors, centroid_mean, rolloff85_mean, and bandwidth_mean, were more difficult to predict, with $R^2$ values of 0.3348, 0.3802, and 0.3143 respectively. This is expected, but it is worth examining why. Spectral properties depend strongly on instrumentation, mixing, and production style, all of which vary independently of movement. A sharp and fast dance sequence could be matched with bright electronic synths, metallic percussion, high strings, or distorted guitar. Each of these would be a musically valid choice for that motion, but they would produce very different spectral centroid and rolloff values. The model therefore faces a many to many problem for these targets, where the same motion can correspond to a wide range of spectral outcomes depending on genre and production choices.

One factor worth noting is that genre labels are available in AIST++ but were intentionally excluded from the semantic model inputs. Including genre would likely improve spectral predictions, since much of the spectral variation in the data is genre dependent rather than motion dependent. However, doing so would mean the system is relying on externally provided genre information rather than deriving musical properties purely from movement. Since the goal of this thesis is to generate music from motion alone, genre labels were excluded to keep the prediction grounded in what the body actually does. The weaker spectral results therefore partly reflect this deliberate constraint rather than a fundamental failure of the motion features.

The window level design also has implications for these results. Using four second windows increased the number of training examples and allowed the model to capture local motion changes, which was an improvement over clip level modelling. However, four second windows may still be too coarse for some motion patterns and too fine for others. Some expressive qualities in dance develop over longer time spans than four seconds, while others change faster. A more adaptive windowing strategy, or a model that combines local and global motion context, could potentially improve results across all targets.

Overall, the semantic model is best understood as a motion informed prior over musical properties rather than a precise predictor of audio content. It captures the broad tendencies that connect movement to sound, particularly texture, energy, and dynamics, while leaving the finer details of timbre, instrumentation, and production to the music generation model. This division of responsibility is appropriate given the ambiguous nature of the motion to music mapping, and it is reflected in the final prompt design,

where predicted descriptors guide the broad musical direction without attempting to specify every detail of the output..

### 5.5 Prompt Engineering and MusicGen Behaviour

The prompt wording experiments showed that MusicGen is highly sensitive to language. The same semantic idea can produce different results depending on how it is phrased. For example, a high energy movement can be described using several possible phrases, but MusicGen may respond differently to each one.

An important distinction is that the project used two prompt word banks. The first was the base learned semantic word bank, which generated broad descriptive phrases for normal prompts, such as “driving,” “wide dynamic swells,” and “bright crisp upper frequencies.” This base prompt was designed to describe the predicted audio character of the motion.

The second was a tuned sectioned-safe word bank, developed after the prompt sweep and long generation tests. This second word bank used more compact phrases such as “strong kick pulse,” “flowing movement contour,” and “bright synth color.” It was introduced because long sectioned generation was more sensitive to prompt instability, and compact wording helped reduce musical layering and prompt drift.

The 30-case prompt wording sweep showed that all tested variants used the five semantic variables, but different wording and sentence structures produced different feature distance results. The best aggregate variant was all5_energy_light_sentence, with a mean overall distance of 0.3397, but no single prompt variant won for every clip.

This shows that prompt engineering is not a small detail. It is part of the system design. Since MusicGen is controlled through text, the translation from motion descriptors into prompt language directly affects the final audio. Raw technical labels such as rms_mean, spectral centroid, or flatness are not ideal prompt terms. MusicGen responds better to musical language that is compact, concrete, and stable.

A major development finding was that more prompt tags did not always improve the result. Early prompts that used too many raw variables or repeated tags could make the output unstable, especially during sectioned generation. The generated music sometimes sounded layered, messy, or overconditioned. This led to an important design decision: keep all five semantic variables but express them differently depending on the generation mode.

This finding supports the final prompt generation approach. It converts descriptor values into stable musical phrases that MusicGen can interpret more reliably. For normal generation, the base learned semantic prompt is used. For long sectioned generation, the sectioned-safe tuned wording is used to improve stability.

### 5.6 Importance of the Multistage Pipeline

The final multistage design was necessary because each stage solves a different part of the motion to music problem.

A direct motion to the audio system would require the model to learn everything at once: rhythm, energy, timbre, genre, structure, and audio synthesis. This is difficult because the relationship between motion and music is not deterministic. In contrast, the multistage pipeline breaks the problem into smaller steps.

The motion feature extraction stage captures physical movement properties. The rhythm and tempo stage captures timing information. The semantic layer predicts audio related descriptors. The prompt generation stage translates these descriptors into musical language. Finally, MusicGen handles the actual audio synthesis.

This structure has two advantages. First, it is more interpretable. If the generated music does not match the motion, it is possible to inspect the intermediate stages and see whether the problem came from rhythm estimation, semantic prediction, prompt wording, or MusicGen generation. Second, it is more practical. The project uses a pretrained generative model and focuses on the motion to prompt translation rather than training a full music generation model from scratch.

### 5.7 Interpretation of Sectioned Generation

Longer music generation was a practical challenge. MusicGen often becomes less stable after approximately 30 seconds, so generating one long track in a single call can lead to drift, instability, or reduced prompt following. For this reason, the application includes sectioned generation for longer dance motions.

During development, several sectioned generation strategies were tested. The first approach added extra section specific prompt words and used crossfade between chunks. This did not work well. Extra section wording caused audible instability and sometimes made the generated music sound as if multiple layers were being stacked. Importantly, this problem could appear from the start of the generated audio, not only at the boundaries between sections.

The final design removed extra section prompt tags and removed crossfade. The same sectioned-safe prompt is used across sections, with MusicGen continuation exposed as an optional switch. This keeps the musical identity more consistent across the full output.

This finding is important because it shows that sectioned generation is not only an audio stitching problem. It is also a prompt consistency problem. If each section receives changing prompt instructions, MusicGen may reinterpret the music each time. This can cause style drift, layering, or instability. Using the same sectioned-safe prompt across sections reduces this risk.

However, the current sectioned generation approach is still limited. It does not dynamically change the music for each section based on local motion changes. This was a deliberate decision for stability. In future work, section level motion analysis could be more useful with a more controllable music generation model. Such a model could accept changing section level conditions instead of relying on one global prompt for the whole dance.

### 5.8 Limitations

The first limitation is that Motion2Music does not generate perfectly beat synchronized music. The tempo model provides useful BPM guidance, but even small tempo errors can accumulate over time. The system can suggest a musically reasonable tempo, but it does not guarantee frame level or beat level synchronization.

The second limitation is that the system does not reconstruct the original song. This is not the goal of the thesis, but it is important to state clearly. The system predicts broad musical properties and generates new music based on them. It should therefore be evaluated as a motion guided music generation system, not as an audio recovery system.

The third limitation is dataset generalization. The tempo results were much stronger on AIST++ than on FineDance or Simon Alexanderson/Motorica. This shows that motion to music relationships vary across datasets and dance styles. A model trained on one type of dance/music pairing may not generalize equally well to all other styles.

The fourth limitation is the dependence on MusicGen. MusicGen is powerful, but it is controlled mainly through text prompts. This limits precise control over timing, structure, instrumentation, and section level changes. Some prompt information may be ignored or interpreted unpredictably by the model.

The fifth limitation is that automatic prompt evaluation is only a proxy. The prompt sweep compared the generated and reference audio using extracted audio descriptors, which makes the evaluation internally consistent. However, feature distance does not fully capture musical quality, dance suitability, or listener preference.

The sixth limitation is that sectioned generation is a practical workaround rather than a complete solution for long form music generation. It helps the system handle longer dances, but it does not fully solve long term musical structure or exact motion synchronization.

### 5.9 Future Work

One future improvement would be to use a more controllable music generation model. A model that accepts lower-level control signals, section level embeddings, beat grids, or audio conditioning could make better use of the motion features than a text only prompt system. This would allow the music to change more dynamically with different sections of the dance.

Another future direction is stronger beat synchronization. The current system estimates BPM and uses it in the prompt, but it does not force generated beats to align exactly with the dancer's movement. Future work could explore beat conditioned generation or post processing methods that align generated audio events with motion beat events.

A third improvement would be to expand human evaluation. The automatic metrics are useful, but dancers, choreographers, and listeners should also evaluate whether the generated music feels appropriate for the dance. This would make the evaluation more realistic because motion to music quality is partly subjective.

Another direction is genre specific semantic mapping. The same motion descriptor may imply different musical choices in hip hop, contemporary dance, electronic music, or orchestral music. Future versions could learn different prompt mappings for different genres instead of using one general mapping.

The semantic model could also be improved with more data and more stable audio targets. Some descriptors were easier to predict than others, and the system may benefit from selecting or designing targets that are more directly related to dance perception. Instead of only using low level audio descriptors, future work could include higher level musical attributes such as groove strength, percussiveness, tension, or emotional intensity.

Finally, the sectioned generation system could be extended. In the current final version, the same sectioned-safe prompt is used across sections for stability. A future system could use section level prompts or control signals if the music generator supports them reliably. This would allow the music to follow changes in the dance more closely without causing prompt drift or musical layering.

### 5.10 Summary

The discussion shows that Motion2Music is most successful when understood as an interpretable motion to music control pipeline. The system does not solve the full problem of generating perfectly synchronized and fully structured music from dance motion. However, it demonstrates that dance motion can be analysed and translated into useful musical control information.

The tempo results show that motion contains rhythmic cues, but also that tempo estimation is affected by dataset style and half time/double time ambiguity. The semantic descriptor results show that motion can predict broad audio qualities such as texture, energy, and dynamics better than exact spectral or production details. The prompt experiments show that MusicGen is highly sensitive to wording, making prompt engineering a central part of the system. The sectioned generation findings show that long form generation requires careful prompt consistency and that extra prompt changes can cause instability.

Overall, the final system supports the thesis idea that dance motion can be converted into music generation prompts through interpretable intermediate stages. The strongest

contribution is not only the generated audio, but the structured process that connects physical movement to musical meaning.

# 6 Summary of Findings/ Recommendations

## 6.1 Overview

This chapter summarises the thesis findings and presents practical and future work recommendations.

## 6.2 Summary of the Thesis

Motion2Music is a multistage pipeline that generates music from dance motion by translating movement into musical control information. Rather than attempting a direct conversion from motion to audio, the system breaks the task into interpretable stages: BVH motion loading and preprocessing, LMA based feature extraction, rhythm and tempo analysis, semantic audio descriptor prediction, prompt generation, and finally music synthesis using MusicGen. The final application demonstrates this pipeline through a local web interface where a user can upload a BVH file, preview the motion, generate a prompt, and produce a music output.

The final Motion2Music pipeline follows this general structure:

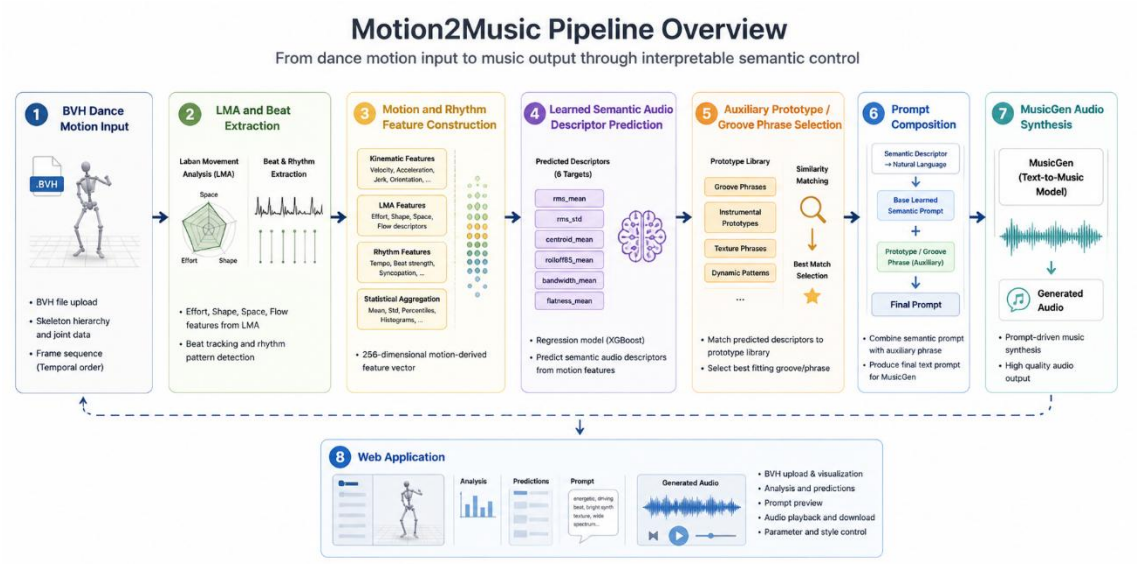


***Figure 12: Overview of the Motion2Music pipeline from BVH dance motion input to MusicGen audio synthesis and web application output.***

## 6.3 Summary of Main Findings

The findings of this thesis converge on one central conclusion: motion-to-music generation works better when treated as an interpretable control problem rather than a direct audio reconstruction task. Motion does not uniquely determine music, and earlier attempts to force a direct mapping produced limited and unstable results. The multistage design, which converts movement into semantic descriptors and then into prompt language, proved more practical and more explainable than a single end-to-end approach.

Tempo estimation showed that dance motion carries useful rhythmic information, though prediction accuracy varied considerably across datasets. The system performed well on structured dance data but struggled with more expressive or rhythmically irregular motion, which reflects the inherent ambiguity between physical movement and musical beat.

Semantic descriptor prediction produced meaningful results across all six targets, with motion features proving especially effective at capturing texture and energy related qualities. Spectral properties were harder to infer, which is expected given that timbre and production choices depend on factors that movement alone cannot determine.

Prompt engineering emerged as a more important design consideration than initially anticipated. MusicGen responded differently to semantically similar phrases, and compact musical wording consistently outperformed raw technical labels. This finding shaped the final distinction between the base learned prompt and the sectioned-safe prompt used for longer generations.

### 6.4 Practical Recommendations

Based on the findings of this thesis, the first practical recommendation is to keep the motion to music pipeline modular. Separating the task into motion feature extraction, rhythm analysis, semantic prediction, prompt generation, and audio synthesis make the system easier to understand, test, and improve. If the output is poor, the developer can inspect the intermediate stages instead of treating the system as one black box.

The second recommendation is to avoid overly technical prompts. MusicGen does not need raw terms such as rms_mean, spectral centroid, or flatness_mean in the final prompt. It responds better to natural musical language. Therefore, numerical descriptors should be translated into compact musical phrases such as “moderately energetic,” “steady groove,” “bright synth color,” or “light textured surface.”

The third recommendation is to keep prompts compact, especially during long sectioned generation. Adding more tags does not always improve the result. In this project, dense prompts sometimes made the generation less stable. A shorter prompt with carefully chosen terms was often more effective than a long prompt containing every available descriptor.

The fourth recommendation is to keep manual genre and style control available. The automatic prompt can provide a reasonable default direction, but the user should be able to manually change the genre or style. This is important because dance to music interpretation is subjective, and different users may want different musical outcomes for the same movement.

The fifth recommendation is to use alias aware tempo evaluation when working with dance motion. Strict BPM error alone can be misleading because half time and double time interpretations may still be musically valid. Alias aware evaluation provides a better understanding of whether the predicted tempo is close in a musical sense.

The sixth recommendation is to combine automatic metrics with human listening. Automatic feature distance evaluation is useful for comparing prompt variants, but it cannot fully measure whether the generated music feels appropriate for the dance. A complete evaluation should include listening tests and feedback from dancers, choreographers, or users.

The seventh recommendation is to treat MusicGen as a powerful but limited generator. It can create useful music from prompts, but it does not provide exact control over beat alignment, structure, or long-term development. For this reason, the system should guide MusicGen with clear prompts rather than expect perfect control from text alone.

### 6.5 Recommendations for Future Work

Future work should prioritise more controllable music generation models that accept beat grids, motion embeddings, or section level conditions, since text prompts alone limit precise timing and structural control. Stronger beat synchronisation is another important direction, as the current system estimates BPM and includes it in the prompt but does not force generated beats to align with the dancer's movement.

Genre specific semantic mapping would also improve results. The same motion descriptor may imply different musical choices depending on dance style, and future models could learn separate prompt mappings for different genres rather than relying on one general mapping. Finally, human evaluation by dancers and listeners should be included in future assessments, since motion to music quality is partly subjective and automatic feature distance metrics cannot fully capture whether generated music feels appropriate for a given dance.

### 6.6 Final Conclusion

Motion2Music demonstrates that dance motion can be analysed and converted into useful musical control information through interpretable intermediate stages. The system does not fully solve automatic music generation for dance, and it does not guarantee perfect beat synchronisation or complete musical structure control. However, it provides a practical framework for connecting physical movement to musical generation, and its strongest contribution is the structured process that translates dance motion into musical language a text to music model can understand. The findings support a motion to semantic to prompt approach as a realistic direction for this problem, while also highlighting the need for more controllable generation models in future work.

## BIBLIOGRAPHY

[1] K. Collins, Game Sound: An Introduction to the History, Theory, and Practice of Video Game Music and Sound Design, MIT Press, 2008.

[2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi and A. Défossez, "Simple and Controllable Music Generation," in *Advances in Neural Information Processing Systems 36*, 2023.

[3] R. Li, S. Yang, D. A. Ross and A. Kanazawa, "AI Choreographer: Music Conditioned 3D Dance Generation with AIST++," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[4] V. Tan, J. Nam, J. Nam and J. Noh, "Motion to Dance Music Generation Using Latent Diffusion Model," in *SIGGRAPH Asia 2023 Technical Communications*, 2023.

[5] C. Zhang and Y. Hua, "Dance2Music Diffusion: Leveraging Latent Diffusion Models for Music Generation from Dance Videos," *EURASIP Journal on Audio, Speech, and Music Processing,* 2024.

[6] J. Romero, D. Tzionas and M. J. Black, "Embodied Hands: Modeling and Capturing Hands and Bodies Together," *ACM Transactions on Graphics,* 2017.

[7] R. Li, J. Zhao, Y. Zhang, M. Su, Z. Ren, H. Zhang, Y. Tang and X. Li, "FineDance: A Fine grained Choreography Dataset for 3D Full Body Dance Generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[8] S. Alexanderson and E. Ericsson, "Motorica Dance Dataset," GitHub, [Online]. Available: https://github.com/simonalexanderson/MotoricaDanceDataset.

[9] A. Aristidou, E. Stavrakis, P. Charalambous, Y. Chrysanthou and S. L. Himona, "Folk Dance Evaluation Using Laban Movement Analysis," *ACM Journal on Computing and Cultural Heritage,* 2015.

[10] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. R. Zapata and X. Serra, "Essentia: An Audio Analysis Library for Music Information Retrieval," in *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013.

[11] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in *Proceedings of the 14th Python in Science Conference*, 2015.

[12] S. Li, W. Dong, Y. Zhang, F. Tang, C. Ma, O. Deussen, T. Y. Lee and C. Xu, "Dance to Music Generation with Encoder Based Textual Inversion," in *SIGGRAPH Asia 2024 Conference Papers*, 2024.

[13] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems 31*, 2018.

[14] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. v. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu and T. L. Scao, "Transformers: State of the Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.