International Journal of Intelligent Information Systems
2025, Vol. 14, No. 6, pp. 121–135
https://doi.org/10.11648/j.ijiis.20251406.12

Research Article

AI-Powered Music Generation from Sequential Motion Signals: A Study in LSTM-Based Modelling

Wisam Bukaita*, Nestor Gomez Artiles, Ishaan Pathak

Math and Computer Science Department, Lawrence Technological University, Southfield, United States

Abstract

This study presents an interactive AI-driven framework for real-time piano music generation from human body motion, establishing a coherent link between physical gesture and computational creativity. The proposed system integrates computer vision–based motion capture with sequence-oriented deep learning to translate continuous movement dynamics into structured musical output. Human pose is extracted using MediaPipe, while OpenCV is employed for temporal motion tracking to derive three-dimensional skeletal landmarks and velocity-based features that modulate musical expression. These motion-derived signals condition a Long Short-Term Memory (LSTM) network trained on a large corpus of classical piano MIDI compositions, enabling the model to preserve stylistic coherence and long-range musical dependencies while dynamically adapting tempo and rhythmic intensity in response to real-time performer movement. The data processing pipeline includes MIDI event encoding, sequence segmentation, feature normalization, and multi-layer LSTM training optimized using cross-entropy loss and the RMSprop optimizer. Model performance is evaluated quantitatively through loss convergence and note diversity metrics, and qualitatively through assessments of musical coherence and system responsiveness. Experimental results demonstrate that the proposed LSTM-based generator maintains structural stability while producing diverse and expressive musical sequences that closely reflect variations in motion velocity. By establishing a closed-loop, real-time mapping between gesture and sound, the framework enables intuitive, embodied musical interaction without requiring traditional instrumental expertise, advancing embodied AI and multimodal human–computer interaction while opening new opportunities for digital performance, creative education, and accessible music generation through movement.

Keywords

Artificial Intelligence, Human-Computer Interaction, Deep Learning, LSTM Networks, Computer Vision, Real-Time Systems, AI, Musical Expression

1. Introduction

The intricate relationship between physical movement and music expression is the cornerstone of artistic reception and performance throughout the centuries. Music is more than just an auditory experience; rather, it is profoundly felt, articulated, and modulated through physical movement. The presence of musical rhythm and dynamics is inherently linked to body movement [1], whether the subtle gesture of the conductor, the expressive dance of the performer, or the precise movement of the instrumentalist. The unfolding potential of artificial intelligence (AI), and particularly the understanding and

*Corresponding author: wbukaita@ltu.edu (Wisam Bukaita)

Received: 27 November 2025; **Accepted:** 9 December 2025; **Published:** 29 December 2025

creation of sequential data, is rapidly enabling new possibilities for the mapping of dynamic body movement to the creation of music in richly expressive and captivating ways. Herein, we present the design and evaluation of an emerging system dedicated to the real-time translation of human physical movement into adaptive and expressive piano music through the use of deep learning and computer vision.

The idea for the project came from a recognition of how traditional music-making software and hardware often demand formal education at high levels, specialized equipment, or extensive working familiarity with music theory. Our system seeks to offer a simple, intuitive means for anyone with a camera to be an expressive musician through movement alone. By taking advantage of the strong motion detection in MediaPipe and the generative powers of Long Short-Term Memory (LSTM) networks [2], we present in this paper a new paradigm for composition through movement. Ultimately, our goal is to take advantage of the temporal coherence of the body's velocity and map it without loss into musical tempo and thereby produce coherent, evocative, and emotionally resonant melodies in real-time. MIDI (Musical Instrument Digital Interface) is a standardized digital protocol used to represent musical performance data such as pitch, note-on/off, velocity, and duration. This format enables consistent representation and manipulation of musical sequences for AI-based training and generation.

Existing research in AI music generation has been focused mainly in two distinct areas: synthetic music generation from symbolic MIDI input, most typically employing rule-based methods or early-stage neural network solutions, and pose-based input method design mainly for gestural control of pre-recorded audio samples or software instruments. Various deep-learning models (GANs, Transformers) exist in music generation but we focus on LSTM-based methods [3, 4]. Meanwhile, gesture-based musical interfaces have largely focused on mapping physical movements to sound parameters rather than generating new music [5, 6].

2. Literature Review

Research at the intersection of human motion and music generation has advanced along two complementary axes: (1) improved multimodal representations that tightly couple visual/kinetic signals with audio, and (2) increasingly sophisticated sequential models that generate temporally coherent musical output conditioned on non-audio inputs. The selected corpus of recent contributions Gan et al. (2020) [17], Rhodes et al. (2020) [18], Li et al. (2021) [19], Li et al. (2022) [20], Jiang (2022) [21], and Christodoulou et al. (2024) [22] exemplifies these trends and highlights both opportunities and outstanding challenges for real-time, embodied music generation.

Gan et al. (2020) [17] advance the representational side with *Music Gesture*, a structured keypoint representation designed for visual sound separation. Rather than treating video as an unstructured signal, Gan et al. explicitly model body and finger keypoints with a context-aware graph network and an audio–visual fusion module to associate gestural events with audio onsets. Their results demonstrate that tightly coupling skeleton-based motion features with audio improves separation and alignment tasks; importantly for motion-to-music research, this work shows that explicit, anatomically grounded representations capture salient expressive cues (e.g., finger motion, torso dynamics) that correlate with musical events. For systems that translate movement into musical control or composition, Gan et al. provide strong evidence that structured motion encodings (rather than raw optical flow or image features) are a superior starting point for downstream musically relevant inference.

Rhodes, Allmendinger, and Climent (2020) [18] survey machine-learning approaches to gesture classification in musical contexts and explore new interfaces that convert biometrics and wearable/visual gesture data into musical control signals. Their work underscores practical considerations often omitted in purely generative research: the role of feature engineering for low-latency control, the tradeoffs between discrete gesture classification versus continuous mapping, and human factors such as learnability and perceptual immediacy. By highlighting both sensor modalities and ML pipelines used in interactive music systems, Rhodes et al. call attention to usability constraints that real-time motion-driven composition systems must satisfy—constraints that directly inform the design choices in our real-time LSTM framework (e.g., the choice of velocity as an intuitive continuous controller and the emphasis on low latency). Karpathy, Johnson, and Fei-Fei (2015) [12] demonstrate that individual LSTM neurons can learn meaningful temporal patterns and long-range dependencies. Although applied to language modeling, their findings are highly relevant to sequence-based tasks, offering theoretical support for using LSTM architectures to model and interpret temporal motion signals in AI-powered music generation systems.

AIST++ and the FACT model introduced by Li et al. (2021) [19] mark an important advance in multimodal data availability and cross-modal modeling. The AIST++ dataset provides large-scale paired music and 3D dance motion with multi-view video and carefully curated annotations. Li et al.'s FACT cross-modal transformer demonstrates that attention-based models can learn long-range correlations between audio and articulated motion, producing realistic music-conditioned motion sequences. For motion-to-music tasks, AIST++ has a twofold impact: it supplies a rich training corpus that supports learning complex temporal correspondences, and it demonstrates that transformer-style architectures with full cross-attention can outperform naive sequence models when learning cross-modal alignment across long time horizons.

Building on cross-modal transformer ideas, Li et al. (2022) [20] propose DanceFormer, which decomposes dance generation into key-pose prediction followed by parametric mo-

tion curve interpolation. This two-stage formulation improves rhythm alignment and fluent motion synthesis by separating coarse rhythmic alignment (key poses) from fine temporal interpolation. DanceFormer's success indicates that multi-stage or hierarchical modeling where discrete, rhythm-aligned events are first predicted and then rendered smoothly can yield better alignment to music than monolithic sequence predictors. The method is directly relevant to motion→music inversion: just as DanceFormer decomposes music motion into rhythm and motion primitives, motion→music systems may benefit from separating discrete musical-event triggers (beat, onset, motif) from continuous expressive control (tempo, dynamics, articulation).

Jiang (2022) [21] focuses explicitly on matching dance movement with music rhythm using human posture estimation techniques. Jiang's approach stresses feature extraction from keypoint trajectories and the use of rhythm features (e.g., beat, tempo measures) to learn alignment functions between movement and music. While Jiang's study is oriented toward analysis and matching rather than generative composition, it validates the core hypothesis used in our work: kinematic features especially temporal derivatives such as velocity and acceleration are strong predictors of musical tempo and rhythmic emphasis. Jiang's findings justify velocity-based conditioning as a principled starting point for real-time tempo modulation.

Finally, Christodoulou, Lartillot, and Jensenius (2024) [22] examine the broader dataset and benchmarking landscape for multimodal music research. Their survey highlights pervasive limitations in current datasets insufficient diversity across performance styles, uneven annotation quality, and a lack of standardized evaluation metrics for multimodal alignment tasks. The authors call for task-aware dataset design and more comprehensive benchmarks that include real-time and human-in-the-loop evaluations. This critique is important: it explains why many generative systems (including LSTM-based approaches) perform well in closed-corpus evaluations yet struggle with cross-dataset generalization and with subjective measures such as perceived expressivity.

Synthesis and gap analysis. Collectively, these works suggest several converging insights for embodied music generation. First, anatomically grounded keypoint representations and derived kinematic features (velocity, acceleration, smoothness) are effective inputs for cross-modal mapping [17, 21]. Second, large, carefully annotated multimodal corpora (e.g., AIST++) and cross-attention architectures enable robust long-range alignment between music and motion but also point toward transformer-style models as strong alternatives or complements to recurrent networks [19, 20]. Third, human-centered design constraints emphasize latency, intuitiveness, and perceptual alignment factors that often favor simpler, continuous control mappings (e.g., velocity tempo) in real-time settings [18]. Finally, dataset and evaluation shortcomings remain a bottleneck for assessing expressivity and generalization [22].

Positioning the present work. The LSTM-based, velocity-conditioned system presented in this paper sits at the nexus of these findings. We adopt the structured, keypoint-based motion encodings advocated by Gan et al. and Jiang [17, 21], prioritize continuous, interpretable controllers to satisfy human-in-the-loop constraints described by Rhodes et al. [18], and acknowledge the representational and dataset tradeoffs highlighted by Li et al. and Christodoulou et al. [19, 20, 22]. While transformer-style cross-modal models show promise for richer, long-range alignment, the LSTM approach remains attractive for real-time deployment due to its computational efficiency and proven capacity to model sequential musical structure; nonetheless, our review motivates future work that hybridizes LSTM conditioning with attention modules and richer gesture descriptors to close remaining gaps in expressivity and semantics.

3. Objectives

The overall objective of this research is to conceptualize and implement a deep learning model with the capability to produce expressive piano music in real-time through direct body manipulation. Achieving the ambitious goal of such a project requires the model to take advantage of both the highly temporal nature of music and the dynamic characteristics of perceived body gesture in order to achieve truly interactive audio synthesis. The success of the prediction system is based on how the various significant factors are combined and composed together, and in the following, each is elaborated in detail:

Velocity (Movement Speed) as Primary Modulator: Velocity (movement speed) is the most influential parameter to mediate physical movement and musical characteristics. Velocity of movement is the most important parameter to quantify the degree to which the user moves in discrete amounts of time. In our system, stronger, more aggressive movement directly translates to higher speed in music and results in more aggressive and accelerated musical phrasing. More subdued, more reflective movement results in more relaxed and spacious phrasing in the music, naturally simulating the subtle rise and fall of human gesture and the dynamics of music. The direct, single-to-one relationship provides the user with an intuitive model of control.

Accurate pose landmark detection: In order to detect the nuances of body movement accurately, the system leverages advanced computer vision tools, in this case, MediaPipe's Pose Landmarker [7-9]. The technology is used to extract highly accurate anatomical keypoints (i.e., shoulders, wrists, hips, elbows, and knees) within three-dimensional space from real-time video streams. The resulting landmarks provide dense spatial encoding of the user's body and stance. In addition, by tracking the temporal evolution of the landmarks frame by frame, the system can detect reliably complex movement patterns and subtle stance changes, and be used as the basis data for speed calculation and further, richer analy-

sis.

Supporting Temporal Coherence and Real-Time Reactivity: The major design challenge for any real-time interactive system is sustaining temporal coherence. To preserve musical coherence and give the user a smooth experience, both input motion data and output MIDI sequences must be rigidly time-synchronized. This strict synchronism enables effective real-time interaction where any detectable change in the user's gesture immediately and smoothly impacts the resulting musical output. Low-latency feedback is crucial to the system's natural feeling and actually reacting to the user's physical gestures.

The generative core of the system is based on Long Short-Term Memory (LSTM) networks, themselves a specific subfamily of Recurrent Neural Networks (RNNs), which were chosen for the project specifically for their superior long-distance sequence learning and prediction properties. LSTMs possess a distinctive strength of maintaining long-range dependencies in sequential data, allowing the model to see and maintain the underlying structural and stylistic phrasing characteristics of music. Movement-derived information (velocity, primarily) is used to condition the LSTM in training, then subsequently supplied to the LSTM as a dynamic condition, in which the network is able to modulate the sequences learned in real-time to create tempo variation and maybe other parameters of music. Enabling Embodiment and Co-Creativity: In addition to use as control input, the ultimate potential of the project is to see how movement can be made an expressive medium in itself. It is different from pure command-and-control, in which the user can embody the music they create on the most fundamental level. Through the creation of an even stronger and more intuitive relationship between physical gesture and auditory outcome, the system acts as an active creative partner rather than a static music generator. This approach enables a new paradigm of co-creation: the user's expressive gestures are directly embedded into the music generated by the AI. In other words, the human movement becomes an integral part of the composition, leading to novel forms of human AI artistic collaboration.

4. Methodology

The real-time method of generating expressive piano music from human body gesture, and more specifically motion velocity mapped to musical temporal features, is rooted in a cutting-edge deep learning architecture. The architecture is naturally divided into two main modules: the body motion analysis pipeline, and as shown in Figure 1 the LSTM network music generation model. The two modules are linked in a dynamic, real-time feedback loop, wherein the physical movement of the user directly and continuously affects the generated musical output.

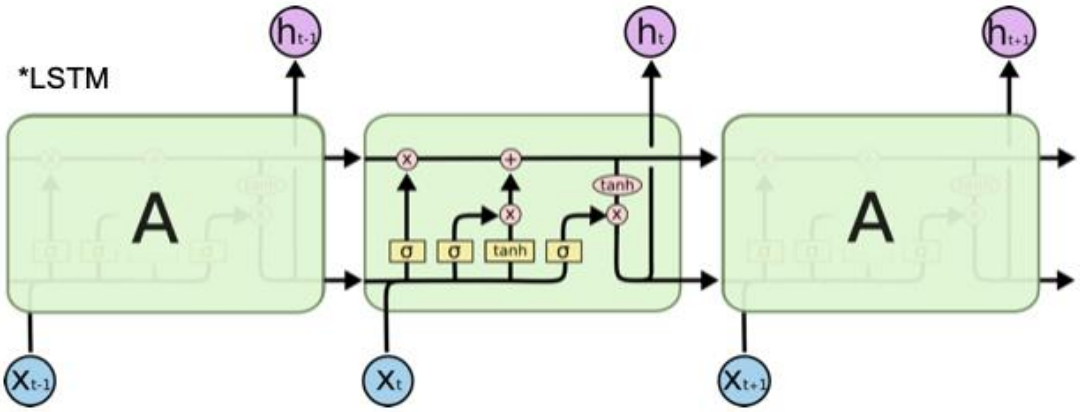

Figure 1. *Internal Structure of a Long Short-Term Memory (LSTM) unit. [11].*

This figure shows the internal architecture of an LSTM unit. Each unit contains input, forget, and output gates, which control the flow of information and gradients across time steps. The horizontal connections across cells represent the memory state that carries temporal context, while the gating functions allow selective updates. This design enables the model to retain or discard information dynamically, making it particularly effective for sequence modeling tasks such as music generation [9].

4.1. Body Movement Analysis Pipeline

The initial component of the system is the effective detection and proper processing of body movement through real-time video stream. This pipeline is tasked with performing the transformation of raw visual data into informative kinetic features to be utilized in driving the process of music generation.

Pose Detection and Landmark Tracking

Technology Employed: The system employs MediaPipe's

Pose Landmarker, which is an extremely capable and accurate machine learning-based real-time system for the detection of the pose of the human body. MediaPipe offers various real-time tracking solutions (pose and hand tracking [9]). So it provides 3D skeletal landmark detection, 33 key anatomical points, as shown in Figure 2, (nose, eyes, shoulders, elbows, wrists, hips, knees, ankles) tracked on the entire body across video frames. 3D estimation is crucial here as it portrays the movement in a better and stable way compared to 2D projections, especially when the camera moves or the user moves [10].

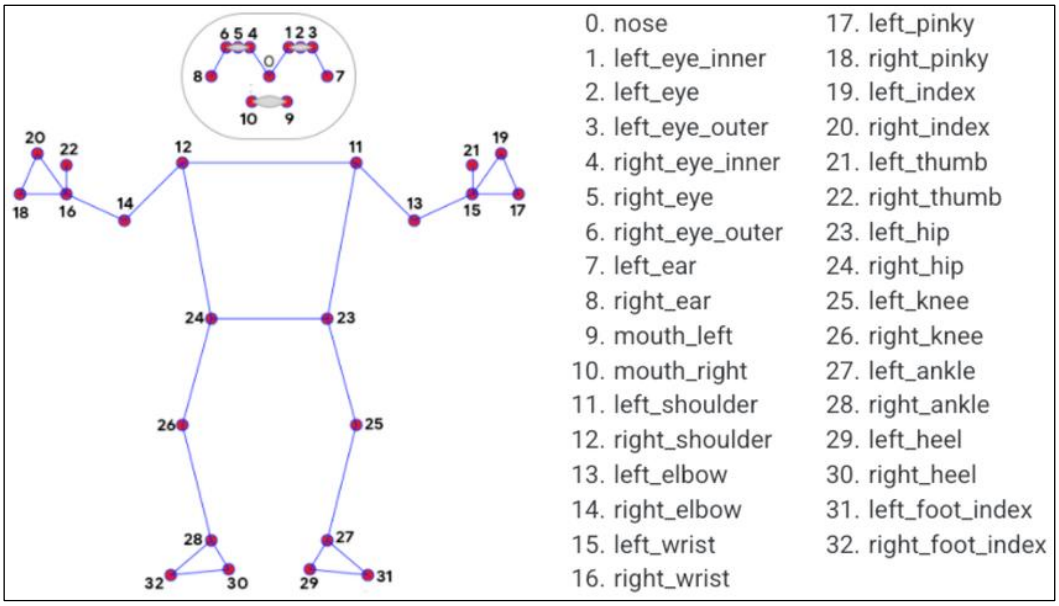

Figure 2. *33 Skeletal Keypoints Used in MediaPipe Pose Detection, Visualized on a Human Body Outline. [2, 3].*

This figure shows the full set of 33 anatomical keypoints extracted by the MediaPipe Pose model, including nose, eyes, shoulders, elbows, wrists, hips, knees, ankles, and feet. These keypoints are tracked in 3D across video frames, enabling precise estimation of human posture and motion. The spatial consistency offered by 3D tracking is particularly valuable in dynamic scenes where either the subject or the camera is moving. MediaPipe's landmark model outputs not only (x, y) coordinates, but also depth (z), which enhances robustness and continuity in the downstream motion-to-music generation process. This image is licensed under CC BY 4.0 [2, 3].

Data Extraction: The 3D coordinates (x,y,z) of all 33 landmarks for an incoming video frame are returned by the Pose Landmarker. The coordinates are normalized to provide improved consistency across different camera settings and subject distances.

Temporal Tracking with OpenCV: Temporal tracking of these landmarks is then performed using OpenCV. The system calculates precise displacement vectors based on the positions of key landmarks (e.g., shoulder, wrist, hip centers) between consecutive frames. Special care is taken in tracking the composite motion, rather than the motion of each separate joint, in order to derive a combined estimate of the user's total kinetic energy.

Velocity Calculation

Definition: The most significant movement feature derived is velocity, which is utilized as the primary continuous control input to the music generation model. Velocity is the measure of the rate over time of change of displacement.

Normalization: The next step is to normalize the resulting velocity signal into a fixed range (e.g., (0, 1) or some target MIDI velocity range) for consistent input into the neural network and for easier tempo mapping to music. Normalization also enables varied scales of movement for varying users. Real-time Synchronization: The derived velocity signal is in constant time-synchronization with arriving MIDI sequences generated by the LSTM. This close synchronization is a causal, real-time relationship where the tempo of the ongoing music is immediately affected by the change in perceived movement. This low-latency coupling lies at the heart of the system's responsiveness and the feeling of direct control the user has over the music.

Calculation Method: For each frame, the mean Euclidean distance covered by a group of significant landmarks, as shown in Figure 3, (e.g., shoulders, hips, and wrists) from their locations in the previous frame is computed. The quotient of mean displacement and the temporal interval between the frames (proportional to the inverse of frame rate) is an instantaneous velocity measure.

$$\text{Velocity} = \frac{\sum_{i=1}^{N} \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2 + (\Delta z_i)^2}}{N \times \Delta t}$$

Figure 3. *Frame-by-Frame Velocity Estimation Using Tracked Landmarks (e.g., Shoulders, Hips, Wrists).*

The diagram illustrates how displacement of these key points is calculated between consecutive frames. Longer arrows indicate greater movement between frames, which corresponds to higher velocity. This visualization demonstrates how the system quantifies user motion in real time.

4.2. LSTM Network Music Generation Model

The generative heart of the system is a Long Short-Term Memory (LSTM) network, which we thoroughly trained to produce expressive and coherent piano melodies. LSTMs are specifically well-suited to sequential data like music since they are founded on the management of long-term dependencies, a key feature in guaranteeing musical coherence and theme elaboration in the long term. [14, 15].

Data collection and preprocessing

Musical Dataset: The model is trained on a large corpus of classical piano performances in the form of MIDI data. Public datasets such as MAESTRO (MIDI and Audio Edited for Synchronous TRacks and Organization) and manually transcribed Chopin piano datasets are valuable resources. They comprise thousands of laboriously transcribed pieces, in the form of musical event streams: note-on, note-off, pitch, velocity, and duration. MIDI data's time-domain structure makes it naturally conducive to sequence modeling. The complete dataset totaled approximately 25 hours of classical piano performances, combining public sources such as MAESTRO and manually transcribed Chopin pieces. The data were partitioned into 70% for training, 20% for validation, and 10% for testing. The test subset was reserved exclusively for final performance evaluation and was not used during model optimization.

MIDI Parsing and Numeric Encoding: We parse MIDI and numerically encode the MIDI files for feeding into the neural network. Each MIDI file was parsed into a stream of symbolic musical events. Each event captures the following information:

1) Note-on and note-off events (indicating when a note starts and ends)
2) Pitch (as a MIDI number, 60 = Middle C)
3) Velocity (note intensity, 0–127)
4) Duration (in quarter lengths, compatible with music21 format)

Sequence Generation: The number sequences are segmented into fixed-length, overlapping sequences (e.g., 64 or 128 notes per sequence). Each sequence is the input to the LSTM, and the following musical event in the original stream serves as the target label, as shown in Figure 4, "Sequence-to-next-event" prediction is typical of music generation. Overlapping windows benefit the model in learning relations between various temporal contexts.

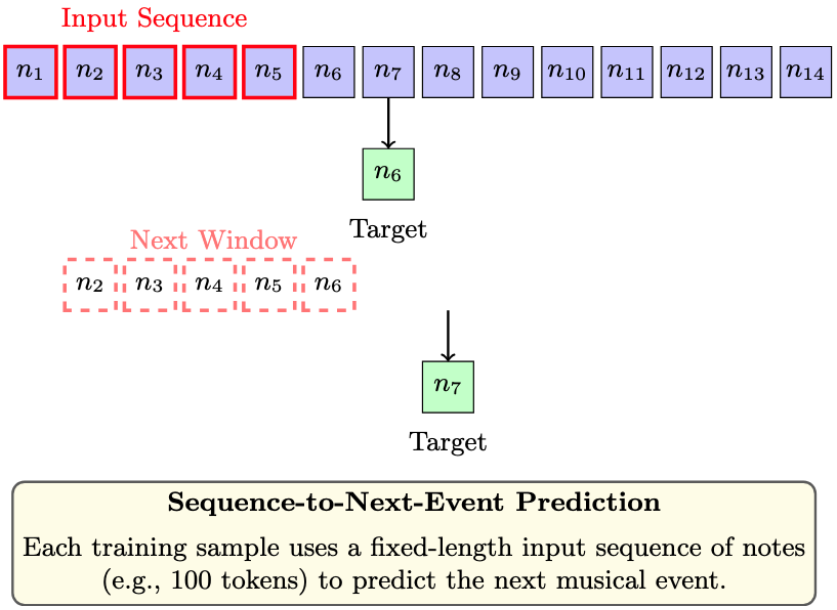

Figure 4. *Sequence-to-Next-Event Prediction with Overlapping Windows for LSTM Training.*

This figure illustrates the supervised learning setup used during training. Each training sample consists of a fixed-length input sequence of musical tokens (e.g., notes), and the goal is to predict the next token in the sequence. After each prediction, the window slides forward by one time step, creating an overlapping input-target pair. This overlapping method ensures that the model captures contextual dependencies over time, enabling it to learn musical structure, repetition, and development across different time scales.

Scaling Data: Numerical values such as velocity and pitch are scaled to the normalized range, typically [0, 1], as shown in Figure 5, Scaling is used to accelerate the learning efficiency and rate of the neural network while training to prevent features that favor higher numerical ranges from overpowering the training.

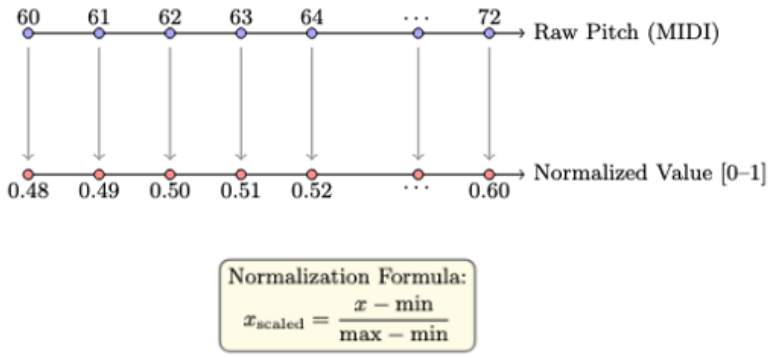

Figure 5. *Normalization of Raw MIDI Pitch Values to a [0–1] Range.*

This figure shows how raw MIDI pitch values, ranging from 60 to 72 in this example are linearly scaled to fit within a normalized range of [0, 1]. This form of feature normalization is applied to both pitch and velocity during preprocessing. By keeping all features within a similar numeric range, the model avoids the risk of high-valued inputs (like pitch or velocity) dominating learning. Normalization also accelerates convergence during training and stabilizes gradients, making the learning process more efficient and balanced.

Model Architecture:

The LSTM architecture that we choose, as shown in Figure 6, must be capable of capturing complex temporal relationships in music as well as be open to receiving real-time external modulation (speed of movement). A suitable architecture would be stacked LSTMs with dense output layers.

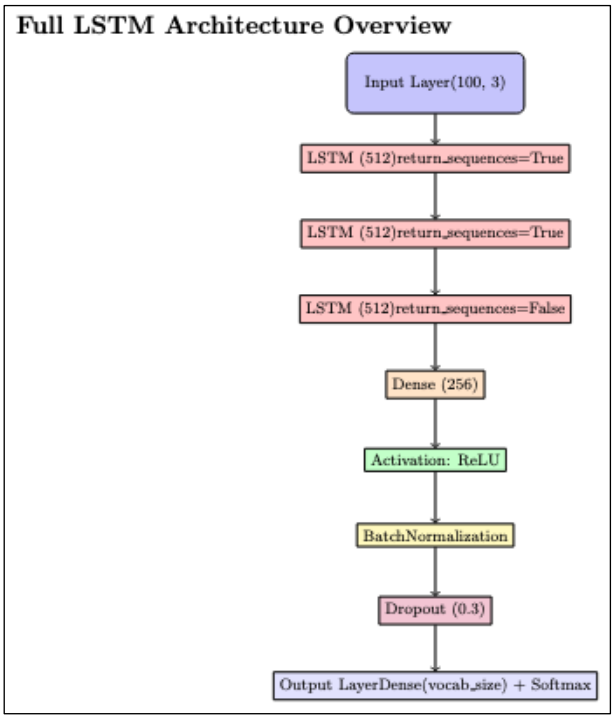

Figure 6. *Overview of the LSTM-Based Architecture Used for Sequence Modeling in Music Generation.*

This figure shows the full deep learning architecture used to generate music from sequential inputs. It begins with an input layer accepting sequences of encoded musical events (e.g., pitch, velocity, duration). The core modeling is handled by three stacked LSTM layers with 512 units each, where the first two return full sequences to retain temporal context, and the final one outputs a single vector summary. This is followed by a dense layer with ReLU activation to capture non-linear relationships, batch normalization to stabilize learning, and dropout for regularization. The output layer is a softmax-activated dense layer that predicts the next musical token from the vocabulary. This design balances complexity and generalization while supporting real-time conditioning via external features like movement speed.

Input Layer: Accepts the numerically encoded sequences of events in music. Typical input shape is (batch_size, sequence_length, number_of_features_per_note). The number_of_features_per_note could include pitch, duration, and even an extra for modulated velocity.

Stacked LSTMs: Two or more LSTMs (e.g., 2 or 3 LSTMs) are stacked to enable the network to acquire hierarchical representations of patterns in music.

Typically, an LSTM layer contains numerous memory cells (e.g., 256, 512, or 1024 units). The more units there are, the more complex patterns the model can be trained to identify, and the longer the dependencies.

return_sequences=True on inner LSTM layers ensures the output of the whole sequence of a layer serves as input to the next layer without losing temporal context. The final LSTM layer may set return_sequences=False to provide a single vector of learned sequence context before dense layers.

Dense Layers: There are fully connected (Dense) layers following the LSTM layers to process the acquired representations further.

Dense (256): It uses a dense hidden layer to create deep connections and higher-level feature learning.

Activation ('relu'): Adding the ReLU activation function provides non-linearity so that the model will be able to learn complex, non-linear relationships.

BatchNormalization (): It normalizes the activations of the prior layer in each batch, stabilizing the training process.

Dropout (0.3): A 0.3 (i.e., 30% of neurons are randomly dropped in every training step) dropout layer plays an essential role in regularization. It keeps the network away from overfitting through the learning of generalized features without relying on specific sets of neurons.

Output Layer: The final dense layer, with softmax activation, outputs a probability distribution across all the following musical events (all the possible pitches, for example, or pitch and duration combination if used as the only categorical output). It has the same number of units as the size of the music vocabulary.

The softmax ensures the probabilities output add up to 1, so the model is able to choose the next note as the most probable.

Model Training and Optimization

Loss Function: Multinomial Cross-Entropy is the default loss function for multi-classification problems properly tailored to predict the subsequent music event within the dictionary of notes. It computes the loss between the true one-hot encoded label and the predicted probability distribution.

Optimizer: The RMSprop optimizer is a widely used optimizer for RNNs. It employs a different learning rate for each parameter, stabilizing the vanishing/exploding gradient issue of RNNs and speeding up convergence. The learning rate is an important hyperparameter, usually a small value (e.g., 0.001 or 0.0001) for stable training.

Epochs and Batch Size: The training is done over many epochs (i.e., 100–200 or more) with a given batch size (we used 64). The sheer number of epochs allows the model to learn its weights by repeating passes between the data, while batch size sacrifices computation efficiency in favor of learning stability.

ModelCheckpoint: We use a ModelCheckpoint callback during training to track a certain metric (e.g., validation loss) and store the model's weights only if the model gets better. In this way, the best model in terms of generalization is kept.

Training Progression: Standard training curves illustrate an initial sharp drop in loss in early epochs, which signifies fast learning of fundamental musical patterns. Then comes a plateau in the curve as the model shifts to fine-tuning. Small oscillations demonstrate the optimization process balancing the learning of new patterns versus the structure already acquired. A steady range of loss (e.g., 2.80-2.81) indicates consistent performance in predicting sequences of notes with low error, which is required for consistent musical output.

4.3. Motion-to-Music Mapping and Real-Time Integration

The main emphasis is on the cohesive combination of the two modules to allow real-time adjustment of the generated music based on human movement.

Tempo Modulation: The velocity signal normalized out of the body movement analysis is used as the conditioning input to the LSTMs. The dynamic effect of the input is primarily to control the tempo of the output.

There is a continuous function of mapping whereby higher velocity measurements are mapped to higher tempo, accomplished by modifying the duration of the produced notes or the velocity of the following notes. Concurrently, lower velocity measurements are mapped to slower tempo.

Adaptive Generation: The LSTM, conditioned by this real-time velocity input, learns to produce note sequences that inherently reflect the desired tempo. This is not simply stretching or compressing pre-generated music; the model actively composes new musical phrases that align with the incoming velocity.

System architecture and data movement, as shown in Figure 7:

Input Layer: A camera is used to visually record the person.

Pose Estimation Module: MediaPipe processes video frames, extracts 3D landmarks, and passes them to the OpenCV-based velocity calculation.

Velocity Module: Calculates the instantaneous velocity.

Music Generation Framework

The model is fed the current musical context, including the sequence of notes generated so far, and the real-time velocity signal.

This predicts the probability distribution of the upcoming musical event.

A sampling technique, for instance, temperature-controlled sampling, relies on probabilistic measures to determine the next note, thus creating a balance between predictability and originality.

Then the generated MIDI note, including its pitch, duration, and velocity, is sent to a virtual piano or synthesizer.

The virtual instrument converts the MIDI events to audible piano sounds. Feedback Loop: The mechanism in the background runs in an ongoing loop without delay, thus ensuring the user feels an immediate and natural reaction from the system.

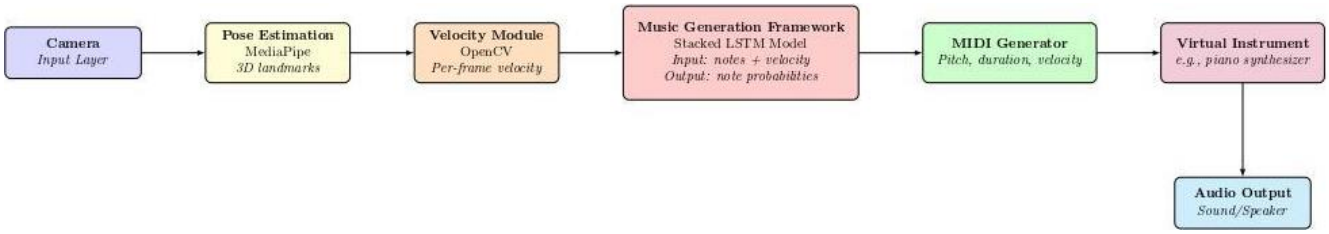

Figure 7. *System Architecture Showing the Data Flow from Motion Capture to Sound Output.*

This figure shows the overall system pipeline. It starts with video input, followed by pose estimation using MediaPipe and velocity computation via OpenCV. These motion features are passed to an LSTM-based music generation model, which predicts the next musical event. The output is converted to MIDI and played through a virtual instrument. The full process runs in real-time, enabling responsive, movement-driven music generation.

5. Experimental Setup and Evaluation

The experimental phase consisted of training the LSTM model and then combining it with the real-time motion capture system for interactive performance and evaluation. The success of the system was measured with quantitative metrics and qualitative user feedback.

5.1. Dataset Preparation and Augmentation

Deep learning system performances, especially on generative problems like music composition, significantly rely on the diversity and quality of the training data.

Classical Piano MIDI Corpus: As mentioned, the MAESTRO dataset serves as the primary basis. It is made up of recordings of piano performances under competitive settings around the world and is a precious source of expressively and musically meaningful performances. The dataset also includes precise timing information, note-on/off events, and MIDI velocities, which are essential in obtaining musical dynamics.

Data Cleaning and Standardization:

Quantization: Although there is exact timing in MIDI, there are small timing variations (humanization) in live performances. The notes may be quantized to a grid (for example, 16th notes) to simplify the learning task for the LSTM and ensure rhythmic consistency. This allows the model to learn simple rhythmic patterns.

Pitch Range Normalization: Although LSTMs are able to process a broad range, normalization to a standard pitch representation (i.e., converting all notes into an internal common scale or relative pitch) may, on occasion, facilitate learning.

Velocity Normalization: MIDI velocities (0-127) are normalized to a 0-1 range to be used by neural network inputs, as shown in Figure 8.

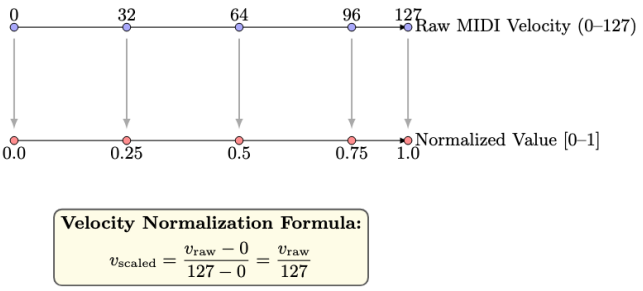

Figure 8. *Normalization of Raw MIDI Velocity Values to a [0–1] Scale.*

This figure shows how MIDI velocity values ranging from 0 to 127 are linearly scaled to a normalized range of 0 to 1. This normalization ensures that the neural network processes velocity values consistently, preventing large numerical differences from skewing training. A simple scaling formula is applied, making the input range suitable for efficient learning.

Sequence creation: Musical data is sequential in nature.

Sliding Window Method: The corpus is split into overlapping sequences of a specified length (e.g., 64 or 128 notes/events). For each sequence (input), the following note/event is marked as the target (output), as shown in Figure 9. This is the supervised learning setup where the model is learning to predict the subsequent element in a sequence based on preceding context.

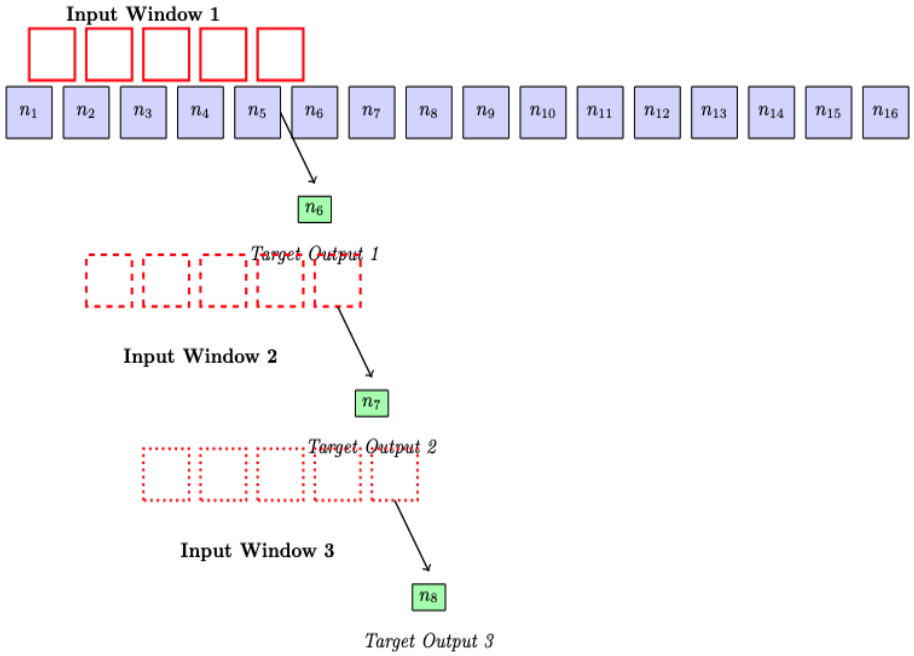

Figure 9. *Overlapping Input Windows Used for Supervised Sequence-to-Next-Event Training.*

This figure shows how sliding window segmentation is applied to create input-target pairs during training. Each input window consists of a fixed number of sequential tokens (e.g., 64 notes), and the model learns to predict the next note following each window. As the window shifts by one step at a time, the network is exposed to overlapping temporal contexts. This setup allows the model to generalize musical structure by learning dependencies between closely related sequences.

Batching: These sequences are all combined in batches to train in an efficient manner so that the model can process multiple sequences at once.

5.2. Training Configuration and Convergence

Training plays a vital role in tuning the LSTM model so that it generates musically coherent and contextually relevant sequences.

Training Environment: The model was trained using Google Colab's GPU runtime (NVIDIA T4, 16 GB VRAM). Training spanned approximately 10 hours across 150 epochs, using a batch size of 64 and learning rate of 0.001 with the Adam optimizer. The Colab environment provided sufficient performance for stable training and convergence of the LSTM architecture. After training, final model evaluation was performed on the held-out test subset to confirm generalization. In our experiments, the final LSTM model achieved a next-note prediction accuracy of 85% on the validation set and 83% on the test set, indicating strong generalization performance.

Hyperparameters:

Epochs: The model had been trained for a substantial number of epochs (i.e., more than 100 epochs, with some refinement epochs). One epoch is one pass through all of the training data.

Batch Size: Our batch size was 64 for training, chosen to balance gradient stability with memory constraints.

Loss Function: Categorical cross-entropy, as discussed earlier, was used to compute the difference in the model's predicted note probabilities and the correct next note, shown in Figure 10.

Optimizer: The adaptive learning rate feature of RMSprop made it the choice, as it performed well for stable learning on sequential data. A specific learning rate was established and perhaps refined through training.

Early Stopping: To prevent overfitting and shorten training time, the early stopping functionality was implemented. It monitored the validation loss and would halt training when no significant improvement was observed for 10 consecutive epochs. This allowed the model to terminate at its best generalization.

Model Checkpointing: The best model weights, depending on the validation metric being tracked, were stored while the model was being trained. This permitted the recovery of the model that performed best for deployment.

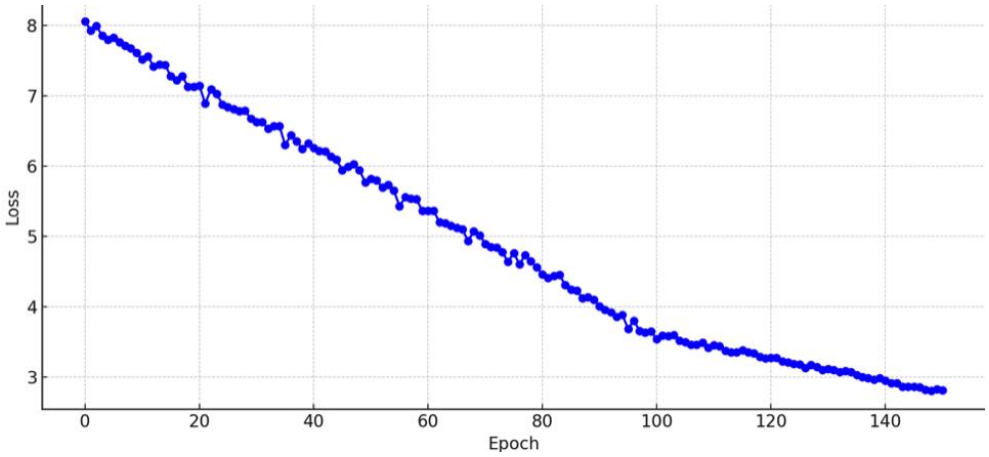

Figure 10. *Training Loss Curve Showing the Model'S Improvement Over 150 Epochs Using Categorical Cross-Entropy.*

This figure shows the reduction in training loss over time as the model learns to predict the next note in a musical sequence. The y-axis represents the categorical cross-entropy loss, while the x-axis tracks the training epochs. The steady decline indicates successful learning, with fewer prediction errors as training progresses. The slight tapering toward the end suggests the model is converging.

Training Progression and Loss Reduction:

As Figure 10 illustrates, the training curve experienced an immediate, steep loss drop. The rapid decrease illustrates how the model learned the overall patterns and rules of classical music composition from the MIDI data quickly.

Following the initial steep decline, the curve flattened more and more, illustrating that the model was starting to leave behind learning general patterns and was fine-tuning its predictions. Minor oscillations during the second half are to be expected in the optimization process whereby the model is trying to balance learning new, complex musical details against maintaining already acquired structural coherence.

The consistent loss values (i.e., maintaining between 2.80 and 2.81 at epoch 140) reflect consistent learning and predicting next notes. Stability is one of the essential factors in producing musically coherent and emotionally stable music pieces. The loss dropped from around 3.60 initially to around 2.79 by epoch 140.

5.3. Evaluation Metrics

Although reduction of loss is one of the most important quantitative measures during training, musical output quality and the responsiveness of the system to human input are the defining factors of a successful music generation system.

Quantitative Assessment (generation and during training):

Loss (Categorical Cross-Entropy): As described, it computes the prediction error between actual and predicted note probabilities.

Note Diversity: Measures the diversity of the notes, rhythms, and dynamics generated. An algorithm that generates the same set of few notes over and over again is not desirable. It can be calculated by considering the entropy of the distribution of the generated notes.

Confusion Matrix:

Because this model performs generative next-note prediction rather than categorical classification, a traditional confusion matrix is not directly applicable. Instead, we evaluated model accuracy through categorical cross-entropy loss and note diversity, which better capture generative performance.

Qualitative Analysis (User Studies):

Musical Coherence: Does the resulting music sound like "music"? Is it based on basic melodic, harmonic, and rhythmic principles? Is it free from abrupt, jarring transitions?

Expressiveness and Emotion Alignment: Is the music conveying the emotional intention of the movement? Is faster movement really more energetic music and slower movement more peaceful music? This is so subjective but so crucial to the purpose of the project.

Responsiveness/Latency: How immediate is the musical response to motion change? High latency breaks the sense of immediate control.

Intuitiveness of Control: How natural and easy is it for the users to control the music with their body movements?

User Engagement: How engaging and enjoyable is the overall experience? Measured through user feedback surveys and actual usage patterns.

Example of Generated Music Sequences (Shown in Figure 11):

To complement the quantitative and qualitative evaluations, we present a sheet of music generated by the system. This example was produced using the LSTM model conditioned on real-time motion-derived velocity input. The excerpt illustrates how the system responds to variations in body movement intensity by modulating musical dynamics and articulation in real-time.

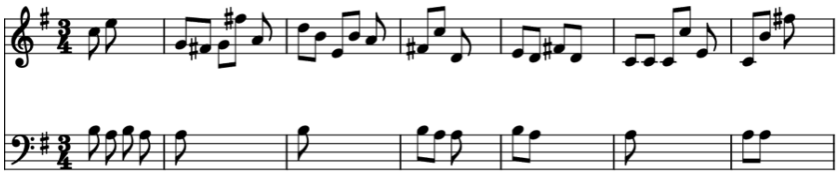

Figure 11. *Music Sheet Generated by the LSTM Model Based on Real-time Movement Input.*

This figure shows a short piano score produced by the trained LSTM model, conditioned on motion-derived velocity input. Louder or faster gestures tend to produce more energetic passages, while slower movements result in more relaxed musical phrases.

6. Results and Discussion

The experimental outcomes indicate a promising basis for the AI-Powered Piano system, proving its capability for real-time, gesture-based music composition. Quantitative as well as qualitative analysis were both successful in revealing the model's performance and how it can be enhanced in the future.

6.1. Model Performance and Musical Coherence

The training process, as indicated by the consistently reducing loss metric (see section 4.2), indicates that the LSTM model was able to learn intricate musical patterns from the classical piano MIDI dataset. The leveling off of the loss at a low point (around 2.80-2.81) after significant epochs indicates that the model acquired a strong understanding of musical context and structure, enabling it to produce high-confidence predictions of upcoming note sequences. This quantitative achievement directly correlates to the qualitative finding of musical coherence in the generated music. The system was able to produce melodies that, on the whole, maintained tonal coherence, rhythmic structure, and potential harmonic progressions, without lapsing into dissonant or random sequences seen in less constrained generation models. We attribute this inherent musicality to the LSTM's ability to model long-term dependencies fundamental to extended melodic lines and harmonic phrasing.

6.2. Responsiveness and Embodied Control

One key contribution of this work was an effective realization of motion-to-music responsiveness. Instant mapping of calculated velocity to corresponding tempo of ensuing piano pieces created an instant and direct relationship. That faster motion inherently created more rapid-paced, energetic musical passages and slower, more reflective motion created softer, more reflective musical settings. Embodied agency moves beyond simple button presses or pre-scripted sequences; it allows the user's kinetic energy and motion-expressed mood to directly shape the composition. This tight human-AI loop makes the interaction feel more participatory and gives the user a greater sense of personal engagement in the music creation. Low latency in the system, a key property of perceived real-time interaction, guaranteed that musical response was virtually simultaneous with input motion, affirming user agency.

6.3. Limitations and Observed Discrepancies

While the results were promising, some limitations were noted, especially regarding the emotional correspondence between musical output and user movement intent. While the model generates reliably and shows stable learning, we noticed that the musical output does not always perfectly match the emotional subtlety of intent of the movements." This mismatch implies that, while velocity as a starting point is a promising one for tempo modulation, it is not enough alone to convey the range of human emotional expression through movement. A rapid, wild movement may result in rapid music, but if the user intends the speed to convey a sense of urgency or anxiety, the music may still be generally "fast" but not necessarily "anxious." This implies the necessity of a more sophisticated interpretation of the movement's semantics beyond velocity alone. For example, the manner of the movement (e.g., sharp vs. smooth, heavy vs. light) or its path through space may convey meaningful emotional information that pure velocity cannot.

Moreover, melodic and rhythmic complexity in addition to standard patterns remains an issue. While succeeding in generating understandable sequences, it's hard to get it to generate complex harmonies, higher-level counterpoint, or interpret larger structures of music with coherence. The model essentially predicts next notes; it's a struggle to get it to write out an entire piece with a narrative trajectory based on sustained motion.

6.4. Comparison with Related Work

This work differs from existing research in several significant respects. Traditional rule-based music generation systems are rigid and their potential for emergent creativity is narrow because they are bounded in terms of static musical grammars [13]. They can generate structurally consistent music but their expressiveness is narrow and adding in dynamic, real-time human input is usually cumbersome. Early neural-network approaches (e.g., simple RNNs or Markov models) struggled with long-term dependencies and often produced incoherent music beyond very short phrases [14]. Our LSTM application addresses this in particular and enables one to generate long and consistent musical lines. However,

architectures specifically designed for long-term musical structure have also been explored [15].

Moreover, systems that only cater to gestural control of pre-stored sound libraries, if interactive at all, do not involve generative AI in the composition process. Such systems typically map gestures onto parameters like volume or filter cutoff [5, 6]. Our system, on the other hand, actually composes novel musical material based on movement and offers a richer level of co-creation. The integration of real-time pose estimation (MediaPipe) and a generative deep learning model (LSTM) for musical composition, rather than pure parameter control, is a new contribution to the state of the art in embodied music AI.

6.5. Implications and Applications

The potential applications for this study are vast and cut across many fields:

Entertainment and Interactive Experiences: The technology enables a new philosophy of interactive entertainment. Users can interact with games in which their motion dynamically affects the soundtrack, or with virtual reality worlds in which their movement has direct impact upon ambient music to create personal and dynamic soundscapes.

Performing Arts and Digital Art: The AI-Powered Piano enables new modes of artistic expression and live performance. Choreography of a dance can now also produce musical accompaniment in real-time to create genuinely interdisciplinary performances. It can also be used in digital art installations where movement of the audience dynamically creates shifting soundscapes.

Human-Computer Interaction (HCI) Research: The work makes a contribution to the overall field of HCI by showing a highly intuitive and embodied interaction model. It investigates how human-centered design, through the use of natural body expression, can result in more compelling and potent computing experiences.

This paradigm aligns with the evolving role of AI in musical ecosystems [16], demonstrating how human motion can become an integral part of creative music technologies.

7. Future Work and Enhancements

To expand on the AI-Powered Piano system and balance its existing limitations, several viable research and development avenues have been recognized. These could expand further the model's understanding of human intent, its expressiveness in music, and its technical proficiency.

7.1. Incorporating Additional Movement-Based Metrics

Current application of velocity, when successful for tempo modulation, is only a subset of what human movement conveys. To create more nuanced correlation between movement and musical affect, future studies should include a wider variety of measures derived from movement:

Gesture Types/Qualities: Defining specific gestures (e.g., sweeping, striking, flowing, sharp, sustained) would enable invoking specific musical phrases or style features. Different machine learning classifiers trained on diverse gestural databases would identify motion and map it to corresponding musical motifs or affective valences. This goes beyond continuous modulation to discrete musical events triggered by specific gestural vocabulary.

Acceleration/Deceleration of Movement: Besides velocity, acceleration or deceleration of velocity can additionally impart even more dynamic and expressive music direction. Sudden accelerations could suggest percussive timbres or crescendos and could even produce fading musical lines and ritardandos with only deceleration.

Movement Flow and Fluidity: Quantitative measures of smoothness or jerkiness of movement can be used to generate musical legato or staccato, or to inject chaos or order within the work.

Facial Expressions and Emotional Inference: Merging facial expression recognition with emotional inference (i.e., through visually trained emotion models) would be a straightforward route of emotional input. Mapping smiles to major key tonality and frowns to minor tonality.

Body Center of Gravity: Tracing a user's movement in his or her center of mass might affect low-frequency music parameters, bass lines, or rhythm accent, synchronizing the music with the user's stability or instability.

7.2. Advanced Model Architectures and Training Strategies

To improve the model's ability to learn complicated musical structures and emotional patterns, certain more advanced deep learning techniques might be explored:

Reinforcement learning: One could also explore reinforcement learning, where an agent is rewarded for producing musically pleasing sequences. This approach might help optimize for subjective qualities in the music.

More Variegated and Broader Training Data: Exposure to even larger and diverse musical databases (i.e., apart from classical piano music, for example, jazz pieces, electronically produced music, or experimental music) would enable the model to acquire a richer musical vocabulary and produce a diverse array of styles.

Longer Time for Training and Deeper Networks: Greater computing power would allow longer training and deeper networks, potentially leading to a more nuanced understanding of music and its relationship to movement.

7.3. Enhanced Real-time Interaction and Feedback

Personalization and Adaptation: Develop models that can

learn to adapt to the individual user's own movement tendencies and personal style over time through user-specific training or online learning.

Multi-Instrumental Generation: Extend the system to generate music for more than a single instrument or entire ensemble from different elements of complex movement.

Haptic Feedback: Include haptic feedback methods that allow participants to "feel" what they are making through vibrations or other sensory feedback to heighten their embodied experience.

User Interface Enhancements: Develop richer user interfaces that can visually display movement tracking, music parameters, and model predictions to enable users to better understand how their input affects the operation of the system.

7.4. Robustness and Generalization

Environmental Robustness: Improve the ability of the system to operate in different lighting conditions and situations, and with different camera qualities.

Cross-User Generalization: Ensure that the model can handle diverse body types, locomotion patterns, and cultures.

Computational Efficiency: Optimize pipelines and models for deployment on lower-capacity hardware (e.g., mobile devices or embedded systems) in order to improve accessibility.

Through these avenues of inquiry, the AI-Powered Piano can be a more versatile, dynamic, and expressive instrument of real-time creative expression, adaptive music therapy, interactive virtual and augmented reality entertainment, and a significant advancement towards the new field of embodied artificial intelligence. It's an exciting advance in making human expression through movement become part of how people compose and engage with music in intelligent systems.

8. Conclusion

In summary, this project demonstrated a system that translates human body movement into expressively rich piano music in real time. We integrate advanced computer vision tools (MediaPipe for pose estimation and OpenCV for movement analysis) with deep learning (LSTM networks) to enable dynamic, expressive audio feedback based on movement velocity.

Its central purpose has been to infer and interpret movement parameters of a human and use velocity as a primary controller for musical tempo. Training the LSTM model on a diverse corpus of classical piano MIDI pieces has equipped it with the ability to generate musically valid and contextually sensitive sequences. Notably, the closed-loop real-time feedback ensures that modifications in motion are directly translated to sound in real-time, enabling an intuitive and highly interactive creative process. Even though the current implementation is a significant improvement in providing an impressive proof-of-concept for motion-based interactive music generation, the research also delineates avenues for continued refinement. The described deviations from idealized emotional harmonies between motion and music production identify the imperative of surpassing velocity as a sole metric for movement. Subsequent research could include integration with a broader set of features, such as gestural quality and even perhaps physiological signals in an effort to gain an even finer sense of human expressive intent. The application of higher-order deep learning structures and larger processing capabilities could also be a key factor in enabling more complex musical structures and even nuanced shadings of emotion.

The potential of this work has broad-ranging applications from new venues for enjoyment and customized therapeutic devices to new methods of learning and new domains for digital performance art. In its role in embodied artificial intelligence, this work demonstrates how human expression can be directly integrated into generative AI systems. By bridging the gap between body movement and sound generation, the AI-Powered Piano system brings us to an age when human expression in movement is not just sensed, but dynamically adds to and informs the very foundation of our creative and sensory endeavors. The training and evaluation setup demonstrates that even within moderate computational constraints, generative LSTM models can effectively learn musical structure and adapt to human motion signals in real time. This work establishes a solid foundation for further research at the exciting confluence of music, AI, and human movement.

Abbreviations

AI	Artificial Intelligence
CV	Computer Vision
LSTM	Long Short-Term Memory
3D	Three-Dimensional
MIDI	Musical Instrument Digital Interface
RMSprop	Root Mean Square Propagation (Optimization Algorithm)
HCI	Human–Computer Interaction
ML	Machine Learning

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Jensenius, Alexander Refsum. 2013. “Some Video Abstraction Techniques for Displaying Body Movement in Analysis and Performance.” Leonardo 46 (1): 53–60. https://doi.org/10.1162/LEON_a_00478

[2] Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” Neural Computation 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

[3] Huang, Cheng-Zhi Anna, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Curtis Hawthorne, and Ian Simon. 2018. "Music Transformer: Generating Music with Long-Term Structure." arXiv, June 12, 2018. https://doi.org/10.48550/arXiv.1809.04281

[4] Yang, Li-Chia, Szu-Yu Chou, and Yi-Hsuan Yang. 2017. "MidiNet: A Convolutional Generative Adversarial Network for Symbolic-Domain Music Generation." In Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR), 324–331. https://doi.org/10.48550/arXiv.1703.10847

[5] Fiebrink, Rebecca, and Perry R. Cook. 2010. "Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance." Proceedings of NIME. https://doi.org/10.5281/zenodo.849810

[6] Bevilacqua, Frédéric, Norbert Schnell, and Romain Flety. 2005. "Gesture Control of Sound Synthesis: Approaches and Design." Gesture Workshop. https://doi.org/10.1007/11678816_5

[7] Kim, Jong-Wook, Jin-Young Choi, Eun-Ju Ha, and Jae-Ho Choi. 2023. "Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model." Applied Sciences 13 (4): 2700. https://doi.org/10.3390/app13042700

[8] Olah, Christopher. 2015. "Understanding LSTM Networks." Colah's Blog. August 27, 2015. https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[9] Lugaresi, Camillo, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. "MediaPipe: A Framework for Building Perception Pipelines." arXiv. June 14, 2019. https://doi.org/10.48550/arXiv.1906.08172

[10] Sengar, Sandeep Singh, Abhishek Kumar, and Owen Singh. 2024. "Efficient Human Pose Estimation: Leveraging Advanced Techniques with MediaPipe." arXiv. June 21, 2024. https://doi.org/10.48550/arXiv.2406.15649

[11] Zhang, Fan, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. "MediaPipe Hands: On-Device Real-Time Hand Tracking." arXiv. June 18, 2020. https://doi.org/10.48550/arXiv.2006.10214

[12] Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. 2015. "Visualizing and Understanding Recurrent Networks." arXiv. June 5, 2015. https://doi.org/10.48550/arXiv.1506.02078

[13] Briot, Jean-Pierre, Gaëtan Hadjeres, and François-David Pachet. 2017. "Deep Learning Techniques for Music Generation – A Survey." arXiv. https://doi.org/10.48550/arXiv.1709.01620

[14] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. https://www.deeplearningbook.org/

[15] Roberts, Adam, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck. 2018. "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music." ICML Workshop. https://doi.org/10.48550/arXiv.1803.05428

[16] Miranda, Eduardo Reck. 2021. Artificial Intelligence and Music Ecosystems. Springer. https://doi.org/10.1007/978-3-030-72116-9

[17] Gan, C., Huang, D., Zhao, H., Tenenbaum, J. B., & Torralba, A. (2020). Music Gesture for Visual Sound Separation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10478–10487. https://doi.org/10.1109/CVPR42600.2020.01049

[18] Rhodes, C., Allmendinger, R., & Climent, R. (2020). New interfaces and approaches to machine learning when classifying gestures within music. Entropy, 22(12), 1384. https://doi.org/10.3390/e22121384

[19] Li, R., Yang, S., Ross, D. A., & Kanazawa, A. (2021). AI Choreographer: Music Conditioned 3D Dance Generation with AIST++. IEEE/CVF International Conference on Computer Vision (ICCV) 2021. https://doi.org/10.1109/ICCV48922.2021.01315

[20] Li, B., Zhao, Y., Shi, Z., & Sheng, L. (2022). DanceFormer: Music Conditioned 3D Dance Generation with Parametric Motion Transformer. AAAI Conference on Artificial Intelligence (AAAI) 2022, 1272–1279. https://doi.org/10.1609/aaai.v36i2.20014

[21] Jiang, D. (2022). Matching model of dance movements and music rhythm features using human posture estimation. Computational Intelligence and Neuroscience, 2022: 7331210. https://doi.org/10.1155/2022/7331210

[22] Christodoulou, A. M., et al. (2024). Multimodal music datasets? Challenges and future goals in music-multimodal research. https://doi.org/10.1007/s13735-024-00344-6