Diploma Project

LMA-Based Motion Analysis Using Deep Learning

Loukia Shikki

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

LMA-Based Motion Analysis Using Deep Learning

Loukia Shikki

Supervisor

Dr Andreas Aristidou

This BSc Thesis was submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Science in Computer Science from the Department of Computer Science of the University of Cyprus.

May 2026

I declare that this dissertation and any accompanying software are my own original work, conducted in full compliance with the University of Cyprus Code of Conduct for Research (https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/), and with full accountability on my part for the accuracy, integrity, and validity of all content.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr Andreas Aristidou, for his guidance, support and invaluable feedback throughout the development of this thesis. His expertise in motion analysis was fundamental to the completion of this work.

I would also like to thank Chrysostomos Chadjiminas for encouraging me to explore and develop my own approach to the 3D CNN architecture. While his version served as an initial reference, the architecture presented in this thesis was independently designed and implemented.

I am grateful to Alexis Mylordos for providing the DanceDB motion capture data used to expand the training corpus.

Finally, I would like to thank my family and friends for their continuous support and encouragement during my studies.

Abstract

Laban Movement Analysis (LMA) is a widely adopted framework for describing and interpreting human movement across four components: Body, Shape, Space and Effort. Traditionally, the annotation of motion data according to LMA principles requires skilled human analysts and is a time-consuming, labor-intensive process. This thesis presents an automated pipeline for predicting continuous LMA annotation scores directly from BVH motion capture data using supervised deep learning.

The core contribution is MotionCNN3DV2, a body-part-aware 3D Convolutional Neural Network that divides the 25 selected skeleton joints into five anatomical groups – Torso, Left Arm, Right Arm, Left Leg and Right Leg – and processes each group through an independent tower. Each tower applies a stem convolution, a multi-scale temporal reasoning block with parallel convolutions of kernel sizes 3, 5 and 9 to capture motion patterns at different temporal scales, and two residual refinement blocks. The five tower outputs are concatenated and passed through a shared MLP fusion head to produce six regression targets: Body, Shape, Space, Effort: Time, Effort: Weight and Effort: Flow.

A secondary contribution is a skeleton retargeting pipeline that converts 22-joint BVH files from the DanceDB dataset into the 38-joint format required by the annotation pipeline, enabling dataset expansion beyond the original recordings. After integrating the DanceDB nohands data, the model was trained on a significantly larger corpus, and training stability improved substantially with the best validation MSE reaching 0.0187 at epoch 3. Evaluation on the test set shows that the model achieves low absolute errors across all six annotation scores, though R^2 scores remain close to zero for most targets, indicating that the model captures the general scale of the annotations but has limited ability to track their fine-grained temporal variation. These results confirm that dataset diversity is a key bottleneck and motivate future work on data augmentation, architecture refinement and annotation target engineering.

Table of Contents

Table Of Figures

1 Introduction

1.1 Motivation

Understanding and describing human movement is a fundamental challenge in computer science, animation, robotics, and digital health. The human body is capable of an enormous variety of movements that differ not only in their geometric trajectory but also in their expressive quality — the force behind a gesture, the fluidity of a transition, the spatial intent of a posture. These qualitative dimensions of movement are difficult to capture with simple kinematic measurements such as joint angles or velocities, yet they are precisely what distinguishes a dance performance from a sequence of mechanical poses.

Laban Movement Analysis (LMA) is a well-established framework developed by Rudolf Laban [1] that provides a structured, multidimensional vocabulary for describing human movement. It organises movement description into four components – Body, Shape, Space and Effort – each of which capture a different aspect of how a person moves. LMA has been applied across a wide range of domains including dance education, choreography, animation, human-computer interaction, and clinical movement therapy [2] [3] [4].

Despite its expressive power, applying LMA to motion capture data is largely a manual process. Trained movement analysts must observe recordings and assign annotation values through a careful and time-consuming watch-and-pause process [5]. This bottleneck severely limits the scalability of LMA-annotated datasets and makes automated motion analysis or real-time feedback systems difficult to realise. There is therefore a strong motivation to develop automated methods capable of predicting LMA annotation scores directly from motion capture data.

1.2 Problem Statement

This thesis addresses the problem of automating LMA annotation prediction from BVH motion capture data. Given a sequence of 3D joint positions recorded over time, the goal is to learn a mapping from these low-level positional measurements to the high-level

LMA descriptors that describe the expressive qualities of the motion. Specifically, the system predicts continuous scores for six annotation targets: Body, Shape, Space, Effort-Weight, Effort-Time and Effort-Flow.

The problem is framed as a supervised regression task. A labelled dataset of motion sequences is paired with LMA annotation scores generated by an algorithmic feature extraction pipeline based on the methodology of Aristidou et al. [2] [3]. A deep learning model is trained to reproduce these scores from raw joint positions, removing the dependency on manually engineered features at inference time.

The challenge is compounded by the limited size of available annotated datasets, the structural diversity of BVH skeleton formats across different motion capture systems, and the difficulty of capturing slow, expressive LMA features – particularly Effort:Time and Effort:Flow – from the short temporal windows used in training.

1.3 Contributions

This thesis makes the following contributions:

1. MotionCNN3DV2 – A body-part-aware 3D convolutional neural network that separates the 25 selected skeleton joints into five anatomical groups and processes each group through an independent tower with multi-scale temporal reasoning and residual refinement. The five tower outputs are fused by a shared MLP head to produce six LMA regression targets.
2. Skeleton Retargeting Pipeline – A script that converts 22-joint BVH files (from the DanceDB nohands dataset) into the 38-joint numpy format expected by the annotation pipeline. This enables dataset expansion without requiring re-export of motion data from the original capture systems.
3. End-to-End Training Pipeline – A complete supervised deep learning pipeline using Pytorch Lightning [6] [7], Hydra configuration [8],and Weights & Biases experiment logging [9], enabling reproducible training and systematic evaluation of model variants.
4. Empirical Evaluation – Training and test results on the expanded dataset, demonstrating positive R^2 scores for Shape, Space and Effort:Weight, and identifying Effort:Time and Effort:Flow as directions for future improvement.

1.4 Thesis Structure

Background and Related Workreviews the relevant background on Laban Movement Analysis and related work in automated motion analysis using deep learning. Dataset and Annotation Pipelinedescribes the datasets used, the LMA annotation pipeline, and the skeleton retargeting approach developed to expand the training corpus. Methodologypresents the MotionCNN3DV2 architecture and training setup in detail. Experimental Results reports the experimental results including training curves and per-annotation performance metrics. Conclusions and Future Work summarises the conclusions and outlines directions for future work.

2 Background and Related Work

2.1 Laban Movement Analysis

Laban Movement Analysis is a method for observing, describing, notating, and interpreting human movement developed by Rudolf Laban (1879-1958) [1]. It provides a multidimensional framework that describes movement quality rather than merely its geometric trajectory, making it suitable for capturing the expressive and intentional aspects of motion that underlie dance, theatre, and everyday gesture.

LMA organises movement description into four major components. The Body component addresses which body part is active, how they connect or support each other, and what sequences and patterns of movement the body follows. The Shape component describes the geometric and morphological changes of the body during movement – whether it is growing or shrinking, widening or narrowing. The Space component characterises the relationship of movement to the environment : the pathways taken, the directions explored, and the extent of the personal kinesphere. The Effort component is perhaps the most semantically rich: it describes the qualitative dynamic intention behind movement, organised into four sub-factors, each with two polarities [2] [3]:

- Space (Direct / Indirect): whether the movement is focused and specific in its spatial attention or multi-focused and flexible.
- Weight (Strong / Light): whether the movement has bold, forceful impact or is delicate and sensitive.
- Time (Sudden / Sustained): whether the movement has a sense of urgency and staccato, or stretches time in a legato, leisurely quality.
- Flow (Bound / Free): whether the movement is controlled, careful and restrained, or released, outpouring, and fluid.

The reliability of LMA as an annotation system has been studied by Bernardet et al. [4], who found that trained practitioners can reach acceptable inter-rater agreement on LMA categories, supporting its use as a ground truth for automated systems. The computational implementation of LMA typically involves extracting a set of kinematic features from motion capture data – distances, velocities, accelerations, volumes, and jerk – that

correlate with each LMA component [2] [5]. These features serve as intermediate representations that bridge the gap between raw joint positions and high-level movement descriptors. The following subsections describe the low-level features associated with each LMA component, following the feature set of Aristidou et al. [2].

Body features (f1-f19). These features characterise body connectivity and posture by measuring geometric relationships between key joints. f1 (feet-hips distance) and f2 (hands-shoulders distance) capture the extension of the limbs from the torso. f3 (hand-hand distance) and f4 (hands-head distance) describe the spatial relationship between the arms and the upper body. f5 (pelvis height) indicates whether the performer is standing, crouching, or airborne. f6 combines multiple displacements to characterize the overall postural extension of the body. f7 (centroid-ground distance) and f8 (centroid-pelvis distance) provide information about the performer's balance and centre of mass. f9 (gait size: right-left foot distance) captures the stride width and is indicative of locomotion style and expressive quality.

Effort features (f10-f18). These features capture the dynamic qualities of motion that map to the four Effort sub-factors. f10 (head orientation) addresses the Space factor. When the direction of the head aligns with the direction of travel, the movement is classified as Direct while misalignment indicates Indirect attention. f11 (deceleration of the root joint) maps to the Weight factor: pronounced deceleration peaks indicate Strong weight, while smooth deceleration indicates Light quality. f12 (root joint distance), f13 (average hand velocity), and f14 (average foot velocity) are all indicative of the Time factor, distinguishing Sudden from Sustained movement. f13 and f14 are particularly useful when the performer remains stationary while the choreography is expressed through arm and leg gestures. f15 (hip acceleration), f16 (hand acceleration), and f17 (foot acceleration) provide derivatives of the corresponding velocities and further refine the Time classification. f18 (jerk: rate of change of acceleration) is the primary feature for the Flow factor, with high jerk values indicating Bound motion and low jerk values indicating Free, fluid movement.

Shape features (f19-f25). These features describe the volumetric and morphological form of the body. f19 measures the overall bounding volume of all joints, capturing how much three-dimensional space the performer occupies. f20-f23 decompose this into four subvolumes: upper body, lower body, left side, and right side respectively, enabling

asymmetric shape changes to be detected independently. f24 (torso height: head-root distance) captures whether the performer is crouching or standing upright, reflecting changes in the vertical extent of the torso. f25 (hands level) encodes the vertical position of the hands relative to the body, distinguishing whether the arms are raised above the head, held at a middle level, or lowered below the chest.

Space features (f26-f27). These features characterize how much of the surrounding environment the performer engages. f26 (distance) is the total path length traced by the projection of the root joint onto the ground plane over a time window, measuring how far the performer travels. f27 (area) is the area of the polygon formed by the same ground-plane projection, capturing the breadth of the spatial region explored. These two features together distinguish confined, localised movement from expansive, spatially exploratory choreography.

2.2 Motion Capture and BVH Format

Motion capture is the process of recording the movement of a person or object and translating that movement into a digital representation. Optical systems use infrared cameras and reflective or active LED markers places on the performer's body to track joint positions at high frame rates. The motion capture system used in this thesis employs active LED markers and allows high-frequency optical tracking of up to 960 Hz [2]. Optical systems are the most widely used in professional animation and research due to their high sampling rates, freedom of movement for the performer, and ability to support numerous markers simultaneously.

The Biovision Hierarchical (BVH) file format is the standard interchange format for motion capture data [10]. A BVH file consists of two sections: a hierarchy section that defines the skeleton's joint structure and resting offsets, and a motion section that records the joint rotations and root translation for each captured frame. The hierarchy section defines a tree for joints rooted at the pelvis or hips, with each joint named and assigned a channel order specifying the sequence in which Euler angle rotations are applied. The order of the channels is significant since matrix multiplication is not commutative, and different bones within the same file may use different rotation orders [10]. The motion section contains one line per frame, with values for each channel in the order they appear

in the hierarchy. To recover the world-space position of any joint, forward kinematics must be applied recursively from the root: each joint's local transformation matrix is premultiplied by its parent's global transformation matrix up the chain to the root [10].

Different motion capture systems produce BVH files with different skeleton hierarchies, joint counts, and joint naming conventions [10]. The system used in this thesis produces a 38-node hierarchy with joint names such as Hips, LowerBack, Spine, LeftArm, and LeftHand. The DanceDB dataset produces a 22-joint hierarchy with different joint names and structural organisation. This incompatibility motivates the retargeting pipeline described in Dataset and Annotation Pipeline

2.3 Deep Learning for Motion Analysis

Aristidou et al. [2] presented a framework for folk dance evaluation based on LMA that extracts 27 kinematics features capturing Body, Effort, Space and Shape components from motion capture data, then uses these features to compare and evaluate dance performances. Their feature set – including joint distances, velocities, accelerations, volumetric measures, and jerk – forms the basis of the annotation pipeline used in this thesis. In a related work, Aristidou et al. [3] applied LMA features to classify the emotional content of dance performances, demonstrating that LMA components can discriminate between different expressive states.

Santos and Dias [5] investigated the use of Bayesian Networks to computationally implement LMA components for human-machine interaction, demonstrating that LMA provides useful semantic descriptors for automated movement characterisation. Their work showed that the Effort component in particular carries strong information about the expressive intent of a movement.

Wang et al. [11] proposed a CNN-LSTM hybrid deep learning model for dance emotion recognition using LMA features extracted from BVH data. Their approach first extracts kinematic features like relative distances, velocities and accelerations of joint points and then trains a CNN for spatial feature extraction followed by an LSTM for temporal modelling. Their results show that the combined CNN-LSTM model outperforms either component alone, highlighting the importance of both spatial and temporal reasoning for LMA-related tasks.

Guo et al. [12] compared four machine learning algorithms – Random Forest, K-nearest neighbours, Neural Network Decision Tree – for automatic LMA annotation from video, using skeleton keypoints extracted by Facebook Detectron2. Their system produced binary classifications across the four LMA dimensions and achieved high accuracy using Random Forest with cross-frame temporal features, noting that the Space dimension benefited most from including temporal context.

Du et al. [13] proposed a multi-branch spatiotemporal fusion network with attention mechanisms for recognising Labanotation actions from motion capture data. Their approach converts motion capture data into 3D coordinates, extracts skeleton vector, temporal difference, and skeleton angle features, and fuses them across branches. Their work is architecturally related to the body-part-aware multi-branch design of MotionCNN3DV2, though applied to distinct task of Labanotation symbol recognition rather than continuous LMA score regression.

2.4 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have become the dominant architecture for learning spatial features from structured data. Standard 2D CNNs operate on spatial inputs (images or feature maps), while 3D CNNs extend convolution to include a temporal dimension, making them naturally suited to video and motion sequence data [14]. In the context of motion capture, the three dimensions of a 3D convolution can capture patterns across time, joints, and spatial coordinates simultaneously.

Liu et al. [14] proposed a 3D CNN with multiscale spatial and temporal attention modules for EEG classification, using parallel convolution branches with different kernel sizes to capture features at multiple temporal resolutions. The key insight – that different temporal scales capture different types of patterns – directly motivates the multi-scale temporal reasoning block in MotionCNN3DV2, where kernel sizes 3, 5, and 9 capture fast gestures, medium-tempo patterns, and slow postural changes respectively.

He et al. [15] introduced residual connections (ResNet) as a solution to the vanishing gradient problem in deep networks. A residual block computes $F(x) + x$, where $F(x)$ is the output of a series of convolutional layers and x is the original input passed through a skip connection. This design allows gradients to flow directly through the network during

backpropagation, enabling the training of much deeper architectures than was previously practical. Residual connections are incorporated into each body-part tower in MotionCNN3DV2 following He et al. [15].

2.5 Evaluation Metrics

Three complementary metrics are used to evaluate the regression performance of the model: Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R^2 coefficient of determination. Each metric captures a different aspect of prediction quality, and together they provide a more complete picture of model behaviour than any single measure alone.

Mean Squared Error (MSE). MSE is the primary training loss used in this thesis. Given N samples with ground truth values $y_1, y_2, \ldots, y_n$ and corresponding predictions $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n$, it is defined as:

$$MSE = (1/N) \Sigma (y_i - \hat{y}_i)^2$$

MSE penalises large errors more heavily than small ones due to the squaring of residuals. This makes it sensitive to outlier predictions, which is desirable during training because it steers the optimizer away from large deviations. The output annotations are normalized to [0,1], so MSE values are interpretable on that scale.

Mean Absolute Error (MAE). MAE measures the average magnitude of prediction errors without squaring, giving equal weight to all deviations:

$$MAE = (1/N) \Sigma |y_i - \hat{y}_i|$$

Because MAE is expressed in the same units as the target variable, it is easier to interpret directly: an MAE of 0.013 on a [0,1] scale means the average absolute prediction is 1.3% of the full annotation range. MAE is used as a secondary evaluation metric in this thesis, reported per annotation on the test set alongside R^2.

R^2 Coefficient of Determination. R^2 measures the proportion of variance in the target annotation that is explained by the model's predictions. It is defined as:

$$R^2 = 1 - (SS_{res} / SS_{tot})$$

Where $SS_{res} = \Sigma(y_i - \hat{y}_i)^2$ is the residual sum of squares (total prediction error) and $SS_{tot} = \Sigma(y_i - \bar{y}_i)^2$ is the total sum of squares (variance of the ground truth around its mean $\bar{y}$). $R^2 = 1$ indicates a perfect fit; $R^2 = 0$ indicates the model explains no more variance than a constant predictor at the mean; $R^2 < 0$ indicates the model performs worse than always predicting the mean. In this thesis, R^2 is the most informative diagnostic metric because it reveals whether the model is genuinely tracking the structure of each annotation or merely predicting a near-constant value close to the target mean – a failure mode that can produce deceptively low MAE while offering no real predictive power.

3 Dataset and Annotation Pipeline

3.1 Data Sources

3.1.1 Baseline Motion Capture Data

Figure 3.1: 38-Joint Skeleton Hierarchy

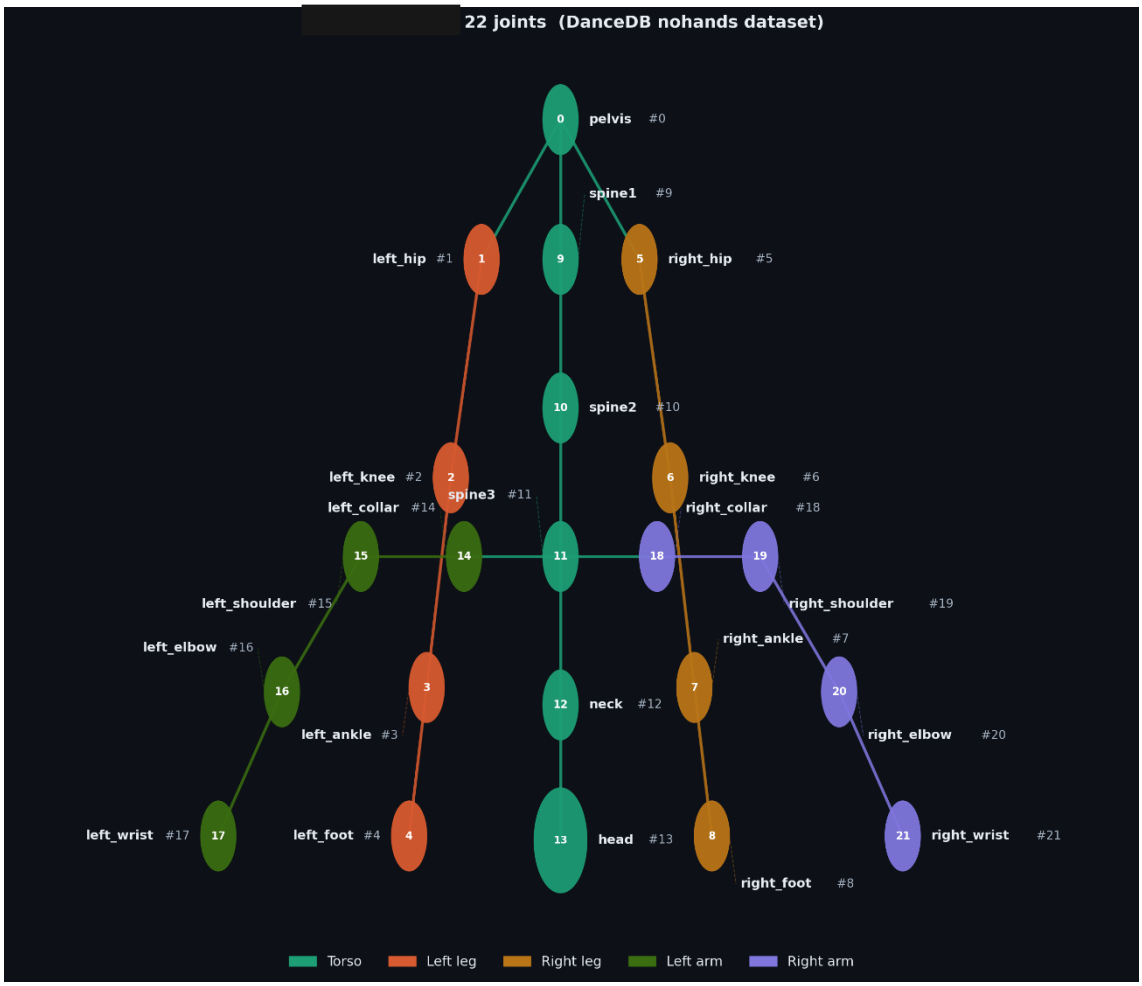

Figure 3.2: 22-Joint Skeleton Hierarchy

The primary dataset consists of BVH motion capture recordings produced by a commercial optical motion capture system [2]. The recordings capture contemporary and folk dance performances by experienced dancers at 120 Hz. The skeleton uses a 38-node hierarchy rooted at Hips, including joints for the full spine chain, both arms with finger segments, both legs with toe joints, and end sites as structural placeholders. Joint positions are available as world-space 3D coordinates (d-xyz) for each frame.

From the 38 available nodes, 25 are selected for the model input using the JOINT_INDICES defined in config.py. These 25 joints exclude structural end sites, intermediate hip joints (LHipJoint, RHipJoint), shoulder intermediaries (LeftShoulder, RightShoulder), and individual finger index joints, retaining the joints most relevant for LMA feature computation.

Figure 3.3: Selected 25-Joint Subset of the 38-Node Skeleton Hierarchy

3.1.2 DanceDB Nohands Dataset

To expand the training corpus, additional motion data was sourced from the Dance Motion Capture Database (DanceDB), which is publicly available at dancedb.cs.ucy.ac.cy [16]. The nohands variant of this dataset uses a 22-joint skeleton rooted at pelvis, without finger joints. The files are recorded at 120 Hz, matching the sampling rate of the baseline

data, but use a different skeleton hierarchy with different joint names and structural organisation.

The DanceDB files could not be processed directly by the existing annotation pipeline, which expects the 38-joint format. This incompatibility motivated the development of the retargeting pipeline described in section 3.3 Skeleton Retargeting

3.2 LMA Feature Extraction

The annotation pipeline converts raw BVH joint position data into per-window LMA score vectors. The pipeline was originally developed by Aristidou et al. [2] and is implemented in create_annotator.py and create_style_words.py. It operates in four stages.

Stage 1 – Texture creation: The joint positions are downsampled from 120 Hz to 5 Hz by selecting one frame every 24 frames. The 25 selected joints are extracted and arranged into a texture array of shape (75, n_downsampled_frames), where each joint contributes three rows (X, Y, Z coordinates).

Stage 2 – Style word extraction: A sliding window of 16 frames is applied to the texture with a step of 4. For each window, 85 kinematic features are computed following the feature set of Aristidou et al. [2]. These include joint distances (f1-f9), Effort features such as deceleration peaks, hip/hand/foot velocities and accelerations, and jerk (f10-f18), Shape features including volumetric bounding measures for the full body and subvolumes (f19-f25), and Space features for total distance and area covered (f26-f27). The resulting vector of 85 kinematic features for each window is called a style word.

Stage 3 – Scaling and averaging: A MinMedianMax scaler is fitted on the stacked style words across all training motions to normalise each feature. The scaler maps values below the median to [-1,0] and values above the median to [0,1]. Specific subsets of the 85 style word features are then averaged to produce single scores for each LMA annotation target, as specified in config.py. The feature indices used are:

- LMA_BODY_INDICES = [11, 23, 19, 31]
- LMA_SHAPE_INDICES = [53, 57, 72, 76, 80, 84, 61]
- LMA_SPACE_INDICES = [68]
- LMA_EFFORT_WEIGHT_STRONG_INDICES = [32]

- LMA_EFFORT_TIME_SUDDEN_INDICES = [35, 38, 41, 42, 44, 46]
- LMA_EFFORT_FLOW_BOUND_INDICES = [48]

Stage 4 – Normalisation and frame expansion: The six annotation scores are normalised from [-1, 1] to [0,1] and then expanded back to the full downsampled frame count by repeating each window's annotation values across the frames it covers. The result is a CSV file with one row per downsampled frame, with columns BODY, EFFORT_WEIGHT_STRONG, EFFORT_TIME_SUDDEN, EFFORT_FLOW_BOUND, SHAPE and SPACE.

85-Feature Style Word Breakdown

Index	Raw Variable(s)	Statistic(s)	LMA Category	Description
0–3	VFeetHip	max/min/std/mean	BODY	*Feet–hip distance (openness)*
4–7	VHandsShoulder	max/min/std/mean	BODY	*Hands–shoulder distance*
8–11	VHands	max/min/std/mean	BODY	*Inter-hand distance*
12–15	VHandsHead	max/min/std/mean	BODY	*Hands–head distance*
16–19	VHandsHip	max/min/std/mean	BODY	*Hands–hip distance*
20–23	Vhip	max/min/std/mean	BODY	*Pelvis height above floor*
24–27	VHipFeetHip	max/min/std/mean	BODY	*Hip elevation + feet spread*
28–31	VFeet	max/min/std/mean	BODY	*Feet mutual distance (gait size)*
32	DecelerationPeaks	count	EFFORT Weight	*Nr. of acceleration sign reversals*
33–35	Velocity (hip)	max/std/mean	EFFORT Time	*Hip velocity magnitude*
36–38	VelocityHand	max/std/mean	EFFORT Time	*Mean hand velocity*
39–41	VelocityFeet	max/std/mean	EFFORT Time	*Mean feet velocity*
42–43	Acceleration (hip)	max/std	EFFORT Time	*Hip acceleration magnitude*
44–45	AccelerationHand	max/std	EFFORT TIme	*Hand acceleration*
46–47	AccelerationFeet	max/std	EFFORT TIme	*Feet acceleration*
48–49	Jerk (hip)	max/std	EFFORT Flow	*Peak jerk (bound vs free flow)*
50–53	volume_5	max/min/std/mean	SHAPE	*5 end-effector convex hull volume*
54–57	volume_all	max/min/std/mean	SHAPE	*All-joints convex hull volume*
58–61	Vtorso	max/min/std/mean	SHAPE	*Torso height (head–pelvis distance)*
62–64	HandsLevel	upper/mid/low ct	BODY	*Hand height zone counts per window*
65	Total Distance	sum	BODY	*Total root path length in window*
66–68	AreaCovered	max/std/mean	SPACE	*Floor area swept (convex hull XZ)*
69–72	volume_upper	max/min/std/mean	SHAPE	*Upper-body directional volume*
73–76	volume_lower	max/min/std/mean	SHAPE	*Lower-body directional volume*
77–80	volume_right	max/min/std/mean	SHAPE	*Right-side directional volume*
81–84	volume_left	max/min/std/mean	SHAPE	*Left-side directional volume*

Total: 85 features → MinMedianMaxScaler → FeatureAverager → 6 LMA groups → NormaliseMinus1To1 → annotations.csv

Figure 3.4: Composition of the 85-Dimensional Style Word Feature Vector

3.3 Skeleton Retargeting

3.3.1 Problem

The DanceDB nohands BVH files use a 22-joint skeleton that is incompatible with the 38-joint format expected by the annotation pipeline. The differences include a different root joint name (pelvis vs Hips), a different spine chain (spine1, spine2, spine3, neck vs LowerBack, Spine, Spine1, Neck, Neck1), the absence of hip intermediary joints (LHipJoint, RHipJoint), the absence of the Neck1 joint, and the absence of finger joints (LeftFingerBase, LThumb, RightFingerBase, RThumb).

3.3.2 Solution

Figure 3.5: 22 joint-to-38 joint Skeleton Retargeting Pipeline

The retarget_bvh.py script addresses this by loading each DanceDB BVH file using the existing loadbvh.py parser, extracting the world-space joint positions (d_xyz) for each named joint and constructing a (38, 3, n_frames) array in the 38-joint order. The 38 output slots are filled as follows.

Direct Maps (22 slots): Joints that have a direct equivalent are copied by name.

Midpoint synthesis (3 slots): Three missing intermediary joints are approximated as midpoints of adjacent joints.

- LHipJoint (slot 1) is synthesized as a midpoint between pelvis and left_hip
- RHipJoint (slot 7) is the midpoint between pelvis and right_hip
- Neck1 (slot 17) is the midpoint between neck and head

Finger extension synthesis (4 slots): The four finger-related joints – LeftFingerBase (slot 24), LeftHandIndex1 (slot 25), LThumb (slot 27), and their right-side equivalents (slots 33, 34, 36) – are synthesised by extending past the wrist in the direction of the forearm. Specifically, given elbow position e and wrist position w, a synthesised finger joint is placed at w + (w-e) * a, where a is 0.30 for FingerBase, 0.40 for Thumb and 0.50 for HandIndex1.

End sites (9 slots): Structural end-site placeholders (slots 6, 12, 19, 26, 28, 35, 37) are filled with NaN, consistent with the original format.

Of the 25 selected joints for model input, 21 are direct maps and 4 are synthesised finger joints. The synthesised joints contribute to hand distance calculations in the Body and Shape annotations, introducing a small approximation error. All intermediary joints synthesised by midpoint (LHipJoint, RHipJoint, Neck1) do not belong to the selected joints and therefore they do not affect model input or annotation computation.

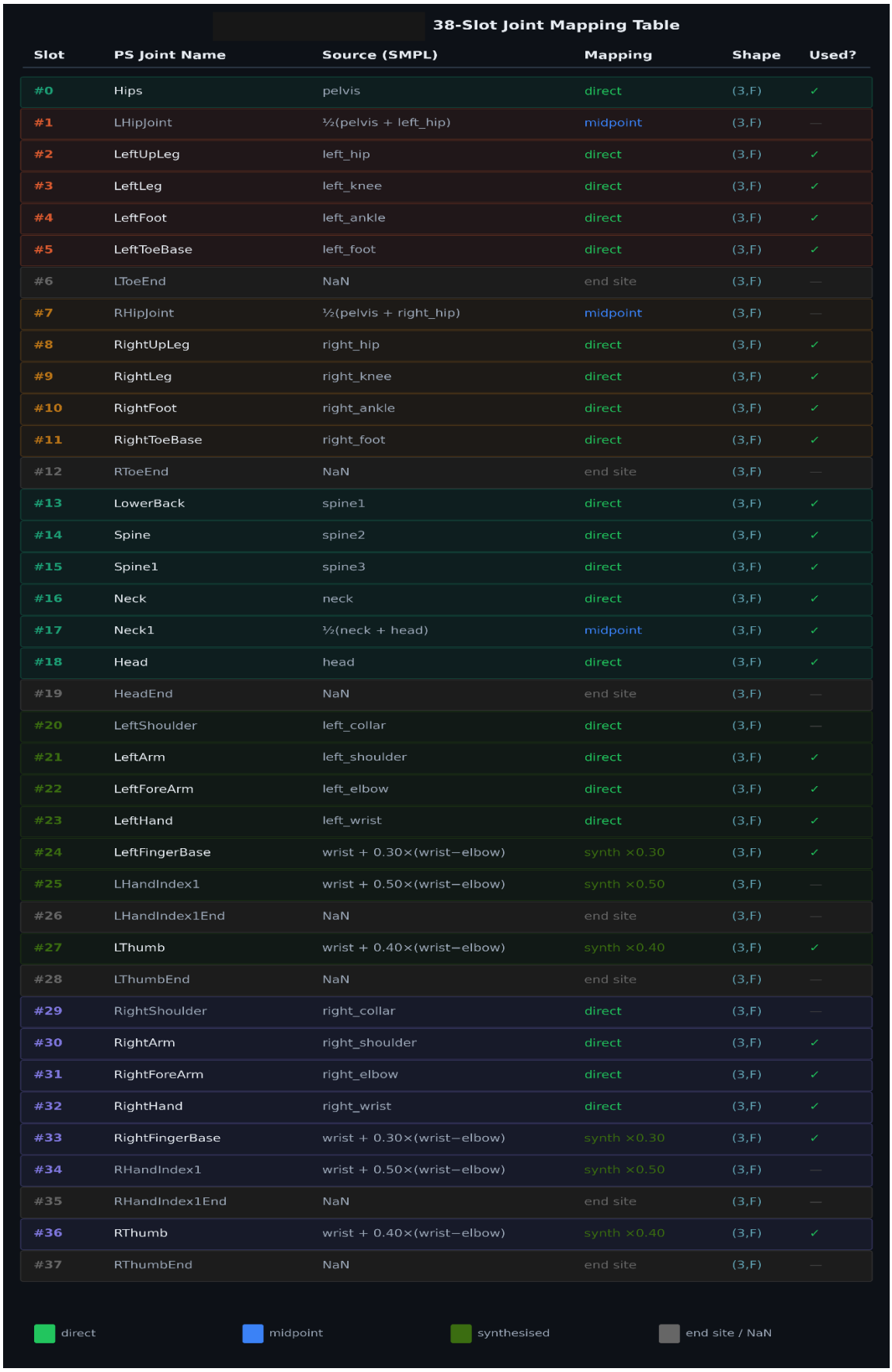

Figure 3.6: Joint Mapping from 22-joint to 38-joint Format

3.3.3 Integration

The script automatically detects the skeleton type of each BVH file by inspecting joint names. Files containing a Hips root joint are identified as 38-joint format and skipped (handled by create_annotator.py). Files containing a pelvis root and left hip joint are identified with 22-joint format and retargeted. Files that are too short (fewer than 768 frames, corresponding to less than two style word windows after downsampling) are also skipped to prevent degenerate annotation windows.

The retargeted numpy arrays are saved to the joints/ directory of the dataset, and the annotation pipeline is run on them using the fitted scaler from the existing lma-annotator.joblib, ensuring that the scaling parameters are consistent between the original and retargeted data.

3.4 Dataset Statistics and Preprocessing

After retargeting and annotation, the combined dataset consists of recordings from both the 38-joint collection and the DanceDB nohands dataset. Each recording is downsampled from 120 Hz to 5 Hz (one frame every 24 frames), giving a practical effective frame rate of 5 Hz. The 25 selected joints are extracted using AxisFeatureSelector, yielding motion tensors of shape (n_downsampled_frames, 25, 3).

A sliding window of length 16 frames is applied with a hop length of 1, generating training windows. Each window is paired with the annotation vector at the following frame (future_annotation_length = 1), framing the task as a one-step prediction problem. The dataset is split 60/20/20 into training, validation and test sets using stratified random splits at the motion level (not the window level) to prevent data leakage between splits.

4 Methodology

4.1 System Overview

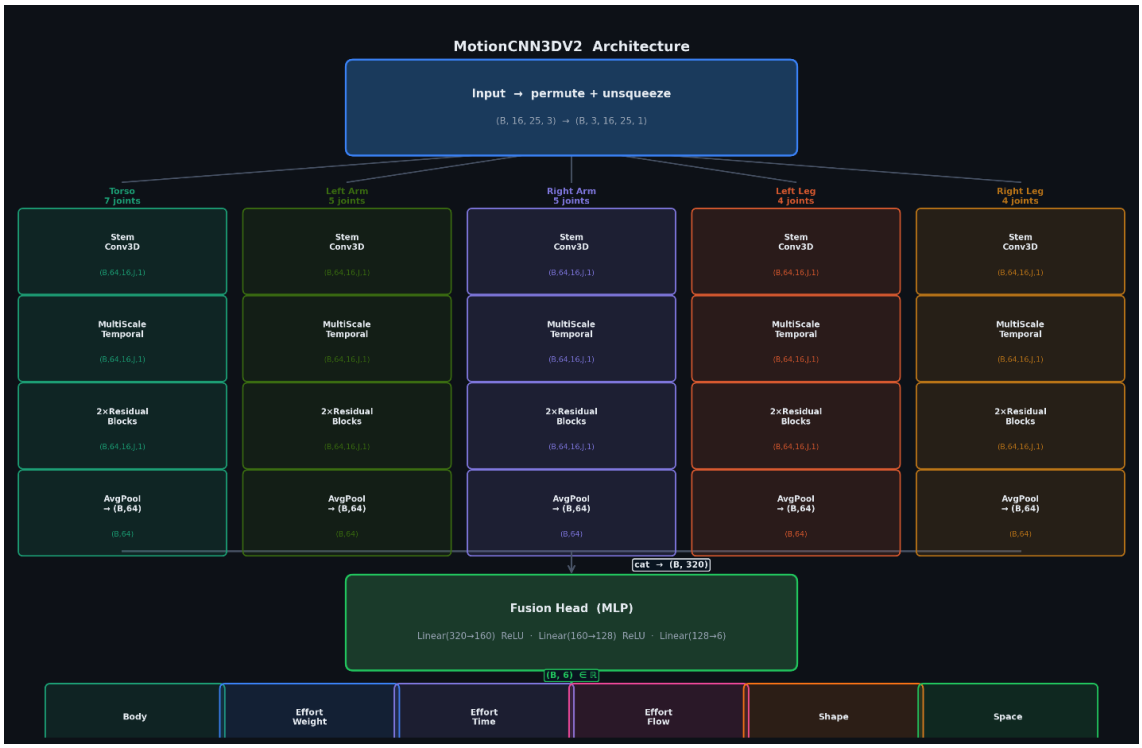

Figure 4.1: MotionCNN3DV2 End-to-End Architecture

The proposed system is an end-to-end supervised deep learning pipeline for LMA annotation prediction from motion capture data. Given a BVH motion file, the pipeline proceeds as follows:

1. The BVH is parsed to extract world-space joint positions
2. Skeleton retargeting is applied if the file uses a non-38-joint format
3. The annotation pipeline generates ground truth LMA scores
4. The model is trained on sliding windows of joint positions with the corresponding annotation scores
5. At inference time, the model produces continuous LMA scores for new motion sequences

The model input is a sliding window of T = 16 downsampled frames, each containing J = 25 joints positions in 3D, giving a tensor of shape (B, T, J, 3). The model output is a vector of 6 continuous values in [0,1] corresponding to the six LMA annotation targets.

4.2 MotionCNN3DV2 Architecture

4.2.1 Design Rationale

The key design insight of MotionCNN3DV2 is that different body parts contribute differently to different LMA descriptors. The legs are most relevant to Body and Space (locomotion, stance, gait), the arms to Body and Shape (openness, hand-head relationships), and the torso to Shape and Effort (volume, postural changes). Rather than treating all joints uniformly, the architecture separates them into five anatomical groups and learns specialised representations for each group through independent convolutional towers. This body-part-aware decomposition is inspired by work on body-segment analysis in LMA [2] and multi-branch spatiotemporal architectures [13].

4.2.2 Anatomical Groups

The 25 selected joints are assigned to five groups based on their anatomical location. The Torso group (7 joints) includes Hips, LowerBack, Spine, Spine1, Neck, Neck1 and Head. The Left Leg group (4 joints) includes LeftUpLeg, LeftLeg, LeftFoot and LeftToeBase and the Right Leg group (4 joints) is the symmetric counterpart. The Left Arm group (5 joints) includes LeftArm, LeftForeArm, LeftHand, LeftFingerBase, and LThumb and the Right Arm group (5 joints) is the symmetric counterpart. These compact indices within the 25-joint selection are [0, 9, 10, 11, 12, 13, 14] for Torso, [1, 2, 3, 4] for Left Leg, [5, 6, 7, 8] for Right Leg, [15, 16, 17, 18, 19] for Left Arm and [20, 21, 22, 23, 24] for Right Arm.

4.2.3 Input Formatting

The input tensor of shape (B, T, J, 3) is permuted to (B, 3, T, J) and an extra dimension is appended to give (B, 3, T, J, 1), making it compatible with Conv3D operators that expect

a 5D input. Each group's joints are then extracted by slicing along the joint dimension, yielding per-group tensors of shape (B, 3, T, J_group, 1).

4.2.4 Body-Part Tower

Each anatomical group is processed by an independent BodyPartTower consisting of four stages. The stem applies a single Conv3D layer with a 3x3x1 kernel and C = 64 output channels, followed by BatchNorm3D and ReLU, extracting initial spatial features from the group's joints. The multi-scale temporal block then processes the stem output through three parallel Conv3D branches with temporal kernels of sizes 3,5 and 9 respectively, each using same padding to preserve the temporal dimension. The three branch outputs are concatenated along the channel dimension (giving 3C channels) and reduced back to C channels by a 1x1x1 Conv3D followed by BatchNorm3D, ReLU and Dropout3D. This multi-scale design is motivated by Liu et al. [14], who demonstrated that parallel branches with different kernel sizes capture features at different temporal resolutions; the kernel-3 branch captures fast, fine-grained joint changes such as quick gestures; the kernel-5 branch captures medium-tempo patterns such as arm swings and weight shifts; and the kernel-9 branch captures slow, global postural changes most relevant to LMA Effort and Shape descriptors.

Following the multi-scale block, two sequential residual blocks provide deeper feature refinement. Each residual block applies the following structure: Conv3D (3x3x1) -> BatchNorm3D -> ReLU -> Dropout3D -> Conv3D (3x3x1) -> BatchNorm3D and then adds input x to the output F(x) before final ReLU, implementing the residual connection of He et al. [15]. This prevents vanishing gradients when stacking multiple convolutional layers and allows the network to learn residual corrections on top of the identity mapping. An AdaptiveAvgPool3D layer then collapses the spatial dimensions (T, J_group, 1) to (1, 1, 1) and the result is flattened to a C-dimensional vector.

4.2.5 Fusion Head

The five tower output vectors, each of dimension C = 64, are concatenated to form a single vector of dimension 5C = 320. This fused representation is passed through a three-

layer MLP: Linear(320->160) -> ReLU -> Dropout, Linear(160->128) -> ReLU -> Dropout, Linear(128->6). No final activation is applied, making the output compatible with both MSE loss for regression and BCEWithLogitsLoss for binary classification. The total number of trainable parameters is approximately 1.9 million.

4.3 Training Setup

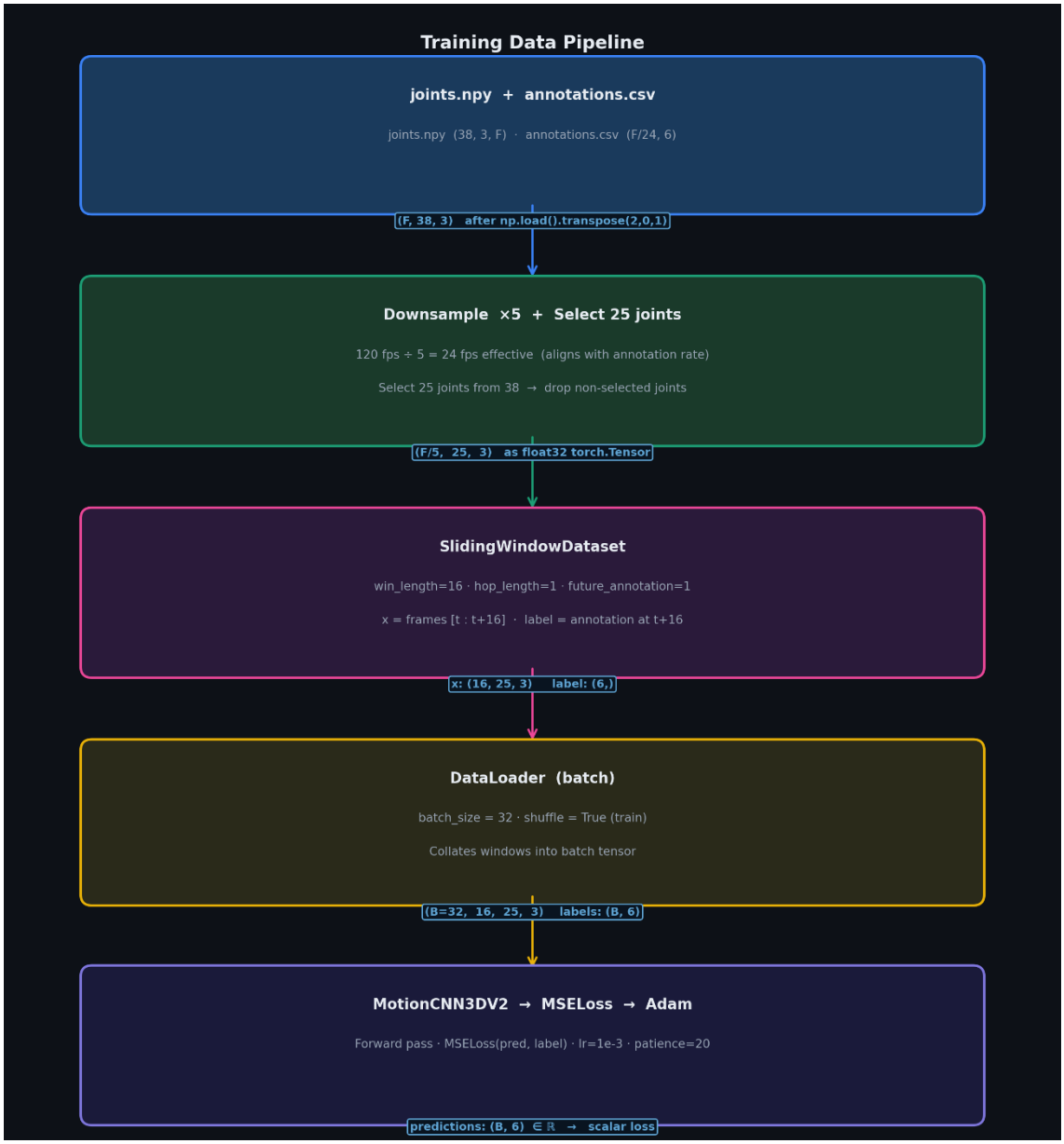

Figure 4.2: Supervised Training Pipeline Overview

The model is trained using Mean Squared Error (MSE) loss with the Adam optimiser. The training framework is PyTorch Lightning [6] [7], which handles the training loop, gradient accumulation, early stopping and model checkpoint. Hyperparameters are managed by Hydra [8] and experiment runs are tracked using Weights & Biases [9].

Early stopping is applied with patience of 20 epochs, monitoring validation loss. The best checkpoint by validation loss is restored at the end of training and used for test evaluation. Training uses a batch size of 32 with 4 DataLoader workers. The sliding window hop length is set to 1 during training for maximum data utilisation.

Evaluation metrics include MSE loss (the training objective), Mean Absolute Error (MAE) and R^2 coefficient of determination. R^2 measures the proportion of variance in the target explained by the model. A value of $R^2 = 1$ indicates perfect prediction, $R^2 = 0$ indicates the model is no better than predicting the mean and negative R^2 indicates the model is worse than a constant mean predictor. Per-annotation breakdown tables and visualisations are produced after each test run.

5 Experimental Results

5.1 Baseline: Original Data Only

The baseline experiment was conducted using only the original motion capture recordings, without the DanceDB data. The model was trained with a learning rate of 1e-3, batch size of 32, MSE loss, and early stopping with patience of 20 epochs. The training curves revealed severe overfitting: training loss decreased rapidly and stabilised near zero, while validation loss spiked repeatedly and failed to converge, reaching values above 0.5 before briefly touching a best of 0.0222 at epoch 32 before diverging again. The annotation name ordering in this experiment also contained a mismatch in the output head, so the per-annotation metrics shown in Figure 5.1 correspond to the correct column order BODY, EFFORT_WEIGHT_STRONG, EFFORT_TIME_SUDDEN, EFFORT_FLOW_BOUND, SHAPE, and SPACE despite the x-axis labels displaying them in a different sequence.

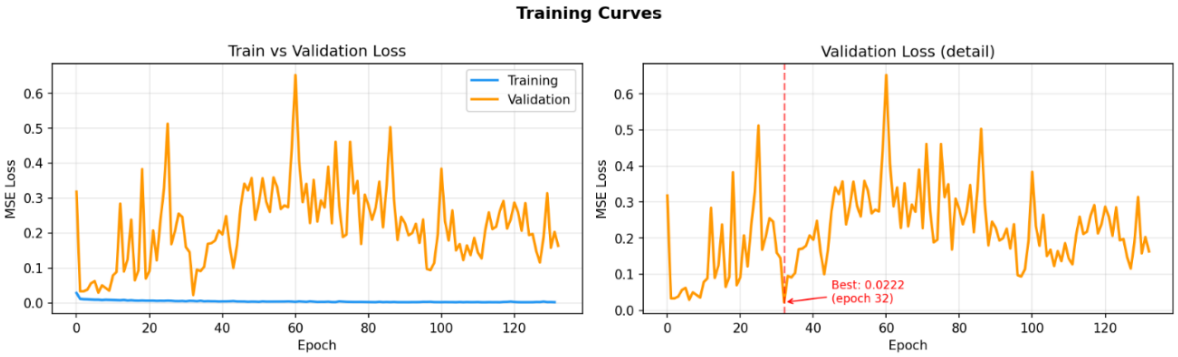

Figure 5.1: Training and Validation Loss Curves – Only Original 38-joint Format Recordings

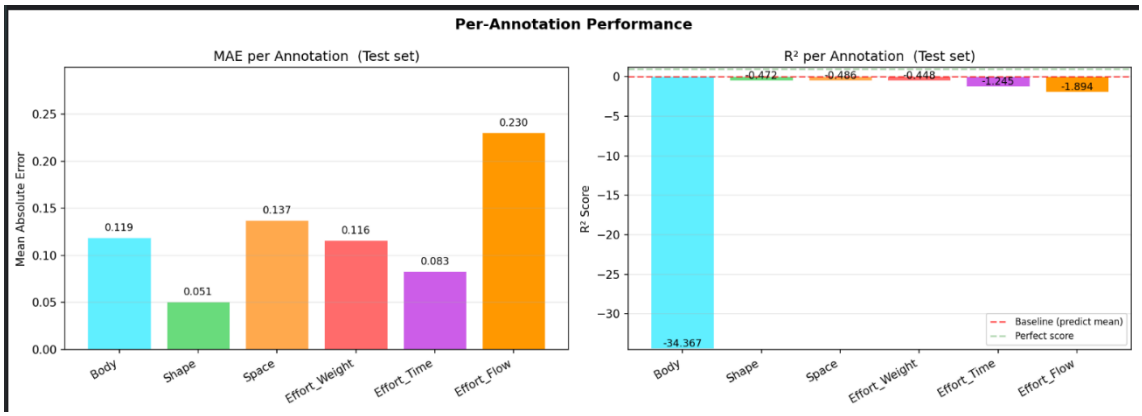

Figure 5.2: Per-annotation MAE and R^2 on the Test Set – Only Original 38-joint Format Recordings

The test-set per-annotation breakdown confirmed that the model was not learning meaningful representations. All six R^2 scores were negative, with Body reaching −34.367, indicating that the model performed dramatically worse than a constant mean predictor for that annotation. MAE values ranged from 0.051 (Shape) to 0.230 (Effort_Flow). The gap between training and validation loss, and the uniformly negative R^2 values, are consistent with a model that has memorised the small training set but failed to generalise. The primary cause is the limited size of the 38-joint-only corpus, which provided insufficient diversity for a model with approximately 1.9 million trainable parameters. These results motivated the dataset expansion described in Chapter 3.

5.2 Results with Expanded Dataset

After expanding the dataset with retargeted DanceDB recordings, a second training run was conducted with identical hyperparameters. The training curves show a qualitatively different behaviour: both training and validation loss decreased rapidly within the first few epochs and stabilised at similar low values, with the best validation loss of 0.0187 achieved at epoch 3. Importantly, the validation loss remained stable throughout all 100+ epochs, with no divergence or instability. This indicates that the addition of the DanceDB data successfully mitigated the overfitting problem observed in the baseline.

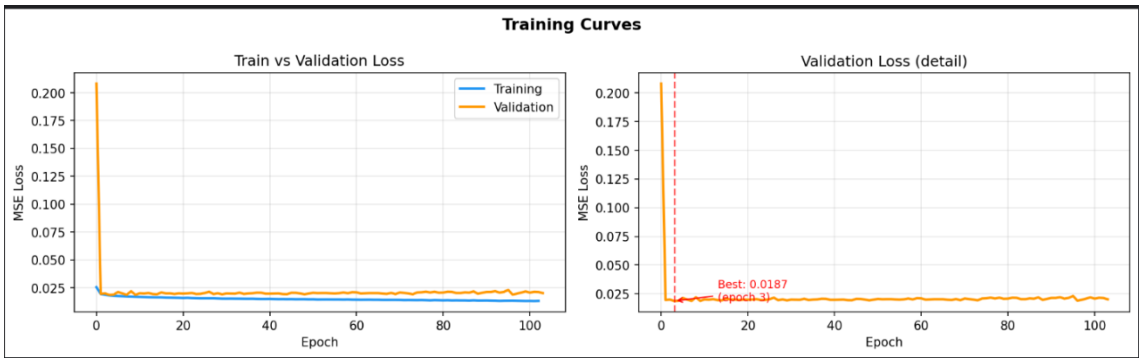

Figure 5.3: Training and Validation Loss Curves – Expanded Dataset

The per-annotation test-set results are reported in Table 5.1 and Figure 5.4. The model achieved positive R^2 scores for Effort_Time (0.030), Effort_Flow (0.025), Effort_Weight (0.005) and Space (0.015), indicating that the model explains some variance in these annotations beyond a constant mean predictor. Shape achieved the best R^2 of all annotations, and its MAE of 0.068 was also the lowest. Body showed a negative R^2 of −0.257 despite a relatively low MAE of 0.051, suggesting that the model predictions for Body are slightly miscalibrated in their distribution even though they are numerically close to the target values on average.

Table 5.1: Per-annotation test-set results for the expanded dataset experiment

Annotation	MAE	R^2
Body	0.051	−0.257
Effort_Weight	0.153	0.005
Effort_Time	0.113	0.030
Effort_Flow	0.133	0.025
Shape	0.068	−0.031
Space	0.178	0.015

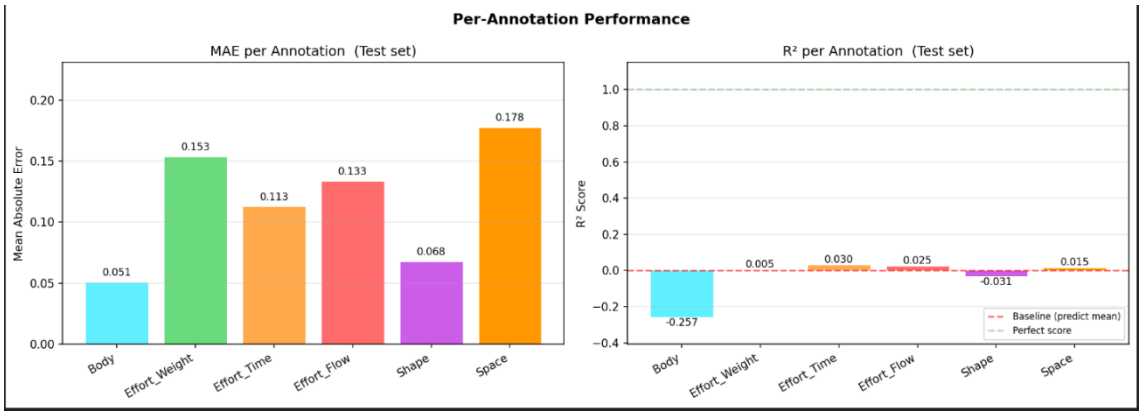

Figure 5.4: Per-Annotation MAE and R^2 on the Test set – Expanded Dataset

The predicted-vs-ground-truth time series plots confirm that the model tracks the general level of each annotation without capturing the full range of variation. The predicted signal (dashed red) tends to follow a smoothed version of the ground truth (solid blue), missing sharp peaks and valleys but staying broadly in the correct range. This is consistent with the model learning a regression towards a conditional mean rather than capturing the full temporal dynamics of each LMA descriptor.

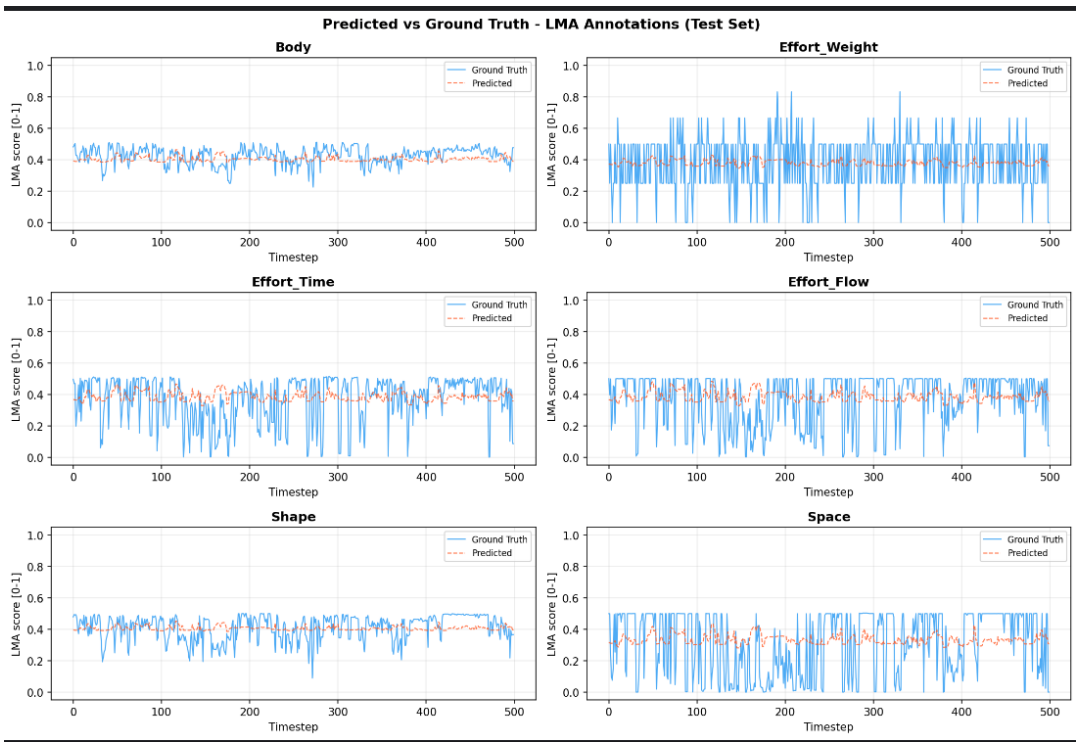

Figure 5.5: Predicted vs Ground Truth LMA Scores Over Time – Expanded Dataset

5.3 Discussion

The transition from the baseline to the expanded dataset produced a clear improvement in training stability and generalisation. The validation loss curve changed from a highly erratic pattern with frequent divergence to a smooth, stable convergence that closely tracked the training loss. This confirms that the fundamental bottleneck in the baseline was data scarcity rather than a flaw in the model architecture or training procedure.

The per-annotation results reveal an interesting pattern. Shape achieved the best overall performance (lowest MAE of 0.068), which is consistent with it being derived from volumetric body measurements that are relatively stable and geometrically direct to predict from joint positions. Space and the Effort sub-factors (Weight, Time, Flow) achieved small positive R^2 values, indicating that the model extracts some information relevant to these higher-level descriptors. Body showed the worst R^2 despite the lowest MAE, which suggests that the Body annotation has low variance in this dataset (the model predictions cluster near the mean) and that the annotation itself may have limited discriminative range across the test samples.

The predicted-vs-ground-truth plots reveal a common failure mode across all annotations: the model learns to predict a smoothed version of the ground truth that follows the general trend but misses high-frequency peaks and transient features. This is likely a consequence of using mean-squared-error as the training objective, which penalises large deviations quadratically and thus encourages conservative, near-mean predictions. A model that always predicts the mean incurs zero R^2 but minimises MSE loss in expectation — this "regression to the mean" effect is a well-known limitation of MSE-trained regressors on high-variance targets.

The Effort annotations (Weight, Time, Flow) remain the hardest targets. These descriptors depend on subtle qualitative dynamics — the onset profile of acceleration, the jerk signature of bound vs free flow — that may not be fully captured in the 16-frame (approximately 3 second) temporal window used. This is consistent with the finding of Wang et al. [11] that longer temporal windows improve performance on LMA-related tasks. Increasing the window length from 16 to 32 or 64 frames is therefore a high-priority direction for future work.

Overall, the expanded-dataset results demonstrate that MotionCNN3DV2 can learn a meaningful mapping from joint positions to LMA annotations when given sufficient

training data, and that the body-part-aware architecture generalises stably. The absolute performance values leave room for substantial improvement, which the future work directions in Chapter 6 aim to address.

6 Conclusions and Future Work

6.1 Summary of Contributions

This thesis presented an automated pipeline for predicting continuous LMA annotation scores from BVH motion capture data using supervised deep learning. The central contribution is MotionCNN3DV2, a body-part aware 3D CNN that separates joints into five anatomical groups and processes each through an independent tower with multi-scale temporal reasoning and residual refinement. A secondary contribution is a skeleton retargeting pipeline that converts 22-joint BVH files to the 38-joint format, enabling dataset expansion from the DanceDB nohands collection.

The experimental evaluation showed that training on the 38-joint-only corpus caused severe overfitting, with all six per-annotation R^2 scores remaining negative. Adding the retargeted DanceDB recordings resolved the generalisation failure: validation loss stabilised at 0.0187 MSE and the model achieved positive R^2 values for Effort_Weight, Effort_Time, Effort_Flow and Space on the held-out test set. Shape produced the lowest MAE (0.068), consistent with its derivation from geometrically direct volumetric body measurements. These results confirm that the body-part-aware architecture can learn meaningful LMA representations when provided with sufficient training data, and identify dataset size and temporal window length as the primary bottlenecks for further improvement.

6.2 Limitations

The most important limitation is the dataset size. Despite the addition of DanceDB data, the total corpus remains relatively small for a model with 1.9 million parameters. The overfitting observed in the 38-joint-only baseline — where validation loss diverged while training loss fell near zero — and the small absolute R^2 values achieved after dataset expansion both point to the same root cause: the model has insufficient diversity in the training distribution to learn robust representations of the full range of LMA dynamics. Larger and more diverse motion capture corpora, covering a broader range of movement

styles, performers, and recording conditions, would likely yield substantially improved generalisation.

The model size is also a practical limitation. With approximately 1.9 million trainable parameters, MotionCNN3DV2 is large relative to the amount of training data available and relative to the complexity of its six-dimensional regression target. The parameter count results in slow inference on devices without GPU acceleration and increases memory requirements during both training and deployment, which may limit applicability in real-time or resource-constrained settings.

The skeleton retargeting introduces approximation errors in the four finger joints (LeftFingerBase, LThumb, RightFingerBase, RThumb), which are synthesized by extending past the wrist in the forearm direction rather than measured directly. These joints contribute to hand distance calculations in the Body and Shape annotations, introducing a small but systematic bias in the retargeted annotations compared to the original recordings.

The 16-frame temporal window (approximately 3 seconds at 5 Hz effective rate) appears insufficient to capture the full temporal dynamics of the Effort annotations. The predicted-vs-ground-truth plots for Effort_Weight, Effort_Time and Effort_Flow all show that the model predicts a smoothed, near-mean signal while the ground truth exhibits large, transient spikes. These transients correspond to changes in dynamic quality — sudden weight shifts, bursts of speed, bound-to-free flow transitions — that may require context spanning several seconds to detect reliably. The positive but small R^2 values for the Effort sub-factors (0.005–0.030) confirm that the model captures some signal but falls well short of explaining the full variance in these targets.

Finally, Effort:Space – the seventh LMA annotation which requires head orientation computation from the Neck1 -> Head vector relative to the root joint velocity – was not implemented in this thesis and remains deferred to future work.

6.3 Future Work

Several directions are identified for future work, ordered broadly by the strength of evidence from the experimental results. First, increasing the temporal window length

from 16 to 32 or 64 frames is the single highest-priority change. The results showed that all three Effort sub-factors (Weight, Time, Flow) exhibit large transient spikes that the model consistently fails to track, and the smoothed, near-mean predictions visible in the time-series plots are consistent with a window too short to detect these qualitative transitions. This is directly motivated by the finding of Wang et al. [11] that a frame interval of 60 frames produced the best results in their LMA classification experiments.

Second, replacing MSE with a loss function that penalises distributional collapse may help address the regression-to-the-mean failure mode. Because MSE minimisation in expectation is achieved by predicting the conditional mean, models trained with MSE tend to produce oversmoothed outputs. A combination of MSE with a variance penalty, or replacing MSE entirely with a quantile loss or a normalised ranking loss, could encourage the model to produce outputs that match the full range of the target distribution rather than collapsing to the mean. The negative R^2 for Body (−0.257) and the near-zero R^2 for Shape (−0.031) despite their low MAE values are the clearest symptom of this problem in the current results.

Third, data augmentation techniques such as random temporal subsampling, spatial mirroring of the skeleton, and additive noise on joint positions could increase effective dataset size without requiring new recordings. Given that the 38-joint-only baseline showed clear overfitting with a model of 1.9 million parameters, augmentation is likely to be beneficial even after dataset expansion and could defer the need for additional data collection.

Fourth, the retargeting pipeline could be extended to support skeleton formats that include full finger chains. This would provide real joint data for the four synthesised finger joints (LeftFingerBase, LThumb, RightFingerBase, RThumb) currently approximated by forearm extension, improving the accuracy of hand distance calculations in the Body and Shape annotations and reducing the systematic bias introduced by the approximation.

Fifth, incorporating a recurrent component such as an LSTM or Transformer temporal encoder after the body-part towers may improve the model's ability to capture long-range temporal dependencies relevant to the Effort annotations, complementing the longer-window direction above.

Sixth, reducing the model size through architecture compression is a natural direction given the large parameter count identified as a limitation. Techniques such as channel

width reduction, depthwise separable convolutions, or knowledge distillation from the current model into a smaller student network could substantially reduce inference cost with minimal loss in predictive performance. A lighter model would also be less prone to overfitting on small datasets, potentially improving generalisation without requiring additional training data.

Finally, the implementation of Effort:Space as a seventh annotation target would complete the Effort component. The head orientation feature can be computed as the angle between the Neck1 → Head direction vector and the trajectory of the root joint, as described by Aristidou et al. [2]. The style word index for this feature would be approximately 88, requiring a modification to create_style_words.py and an update to the model's output dimension from 6 to 7.

References

[1] R. Laban and L. Ullman, The Mastery of Movement, 4 ed., Binsted: Dance Books Ltd, 2011.

[2] A. Aristidou, E. Stavrakis, P. Charalambous, Y. Chrysanthou and S. Loizidou Himona, "Folk dance evaluation using Laban Movement Analysis," *ACM Journal on Computing and Cultural Heritage,* vol. 8, no. 4, pp. 1-19, 2015.

[3] A. Aristidou, P. Charalambous and Y. Chrysanthou, "Emotion analysis and classification: Understanding the performers," *Computer Graphics Forum,* vol. 34, no. 6, pp. 262-276, 2015.

[4] U. Bernardet, S. F. Alaoui, K. Studd, K. Bradley, P. Pasquier and T. Schiphorst, "Assessing the reliability of the Laban Movement Analysis system," *PLoS ONE,* vol. 14, no. 6, p. e0218179, 2019.

[5] L. Santos and J. Dias, "Laban Movement Analysis towards behaviour patterns," in *Emerging Trends in Technological Innovation (DoCEIS 2010)*, Berlin, 2010.

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] W. Falcon, "PyTorch Lightning," GitHub, 2019.

[8] O. Yadan, "Hydra — A framework for elegantly configuring complex applications," GitHub, 2019.

[9] L. Biewald, "Experiment tracking with Weights and Biases," Weights & Biases, 2020.

[10] M. Meredith and S. Maddock, "Motion capture file formats explained," 2001.

[11] S. Wang, J. Li, T. Cao, H. Wang, P. Tu and Y. Li, "Dance emotion recognition based on Laban motion analysis using convolutional neural network and long sort-term memory," *IEEE Access,* vol. 8, p. 124928–124938, 2020.

[12] W. Guo, O. Craig, T. Difato, J. Oliverio, M. Santoso, J. Sonke and A. Barmpoutis, “AI-driven human motion classification and analysis using Laban Movement System,” in *HCI International 2022 (LNCS vol. 13319)*, Cham, 2022.

[13] J. Du, J. Wang and J. Li, “Skeleton-based motion recognition for Labanotation generation based on the fusion of neural networks,” *Computer Animation and Virtual Worlds,* 2025.

[14] X. Liu, K. Wang, F. Liu, W. Zhao and J. Liu, “3D convolution neural network with multiscale spatial and temporal cues for motor imagery EEG classification,” *Cognitive Neurodynamics,* vol. 17, no. 5, p. 1357–1380, 2023.

[15] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] University of Cyprus, “Dance Motion Capture Database,” [Online]. Available: https://dancedb.cs.ucy.ac.cy.