ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Characterization of Goalkeeper Actions from Skeletal Tracking Data

Practical Work (263-0650-00L)

Afonso Ferreira da Silva Domingues, Kaushik Karthikeyan

Friday 27th February, 2026

Advisors: Prof. Dr. Ulrik Brandes, Hugo Fabrègues
Department of Computer Science, ETH Zürich

Abstract

Modern player-tracking systems provide detailed 3D skeletal trajectories, yet technical analysis in soccer remains constrained by limited labeled datasets and reliance on predefined action taxonomies. We present a fully unsupervised framework for discovering goalkeeper behaviors directly from large-scale skeletal data collected across all 51 matches of UEFA Euro 2024 using Hawk-Eye SkeleTRACK. Our preprocessing synchronizes ball and skeleton data streams and extracts continuous goalkeeper trajectories when the ball is in the defensive final third. Goalkeeper trajectories are further filtered to retain high-energy segments, which serve as a proxy for interesting or salient movements, using an adaptive motion-energy signal. These sequences are then represented in a canonical goal-centered coordinate frame to enable analysis that is invariant to the goalkeeper's absolute position on the pitch

To learn motion representations without annotations, we use the CrosSCLR contrastive learning framework with an ST-GCN backbone and multi-view encoders operating on joint, motion, and bone representations. The model is trained with domain-specific augmentations tailored to goalkeeping dynamics. The resulting 128-dimensional embeddings are projected using UMAP and clustered with HDBSCAN to identify recurring behaviors in a fully data-driven manner.

Training the model from scratch outperforms fine-tuning from NTU RGB+D, producing coherent and semantically interpretable clusters corresponding to ready stances, lateral shuffling, ball distribution, and diverse save techniques. Further subclustering of the save category isolates fine-grained mechanics such as lateral dives, low saves, forward smothers, and jumping punches. We assess cluster quality by visually inspecting representative clips and analyzing the kinematic characteristics of each cluster. Future work should focus on enriching representations with additional contextual features, refining clip segmentation, and systematically evaluating how preprocessing choices—such as spatial normalization and dimensionality reduction—impact clustering stability and the quality of resulting clusters.

Acknowledgments

We would like to express our sincere gratitude to our supervisors, Prof. Dr. Ulrik Brandes and Hugo Fabrègues, for their invaluable guidance, insightful feedback, and continuous support throughout this project. We are grateful for their mentorship, which significantly enhanced the direction and quality of our research.

We also wish to express our gratitude to the Social Networks Lab for fostering a highly collaborative research environment. The regular discussions with lab members and fellow students provided diverse perspectives and constructive feedback, which contributed significantly to the development and refinement of our work.

Contents

Chapter 1

Introduction

1.1 Background

In association football (soccer), advances in data acquisition technology have enabled increasingly detailed representations of player movement. In particular, emerging skeletal and mesh tracking data make it possible to move beyond traditional center-of-mass trajectories and incorporate richer pose- and motion-level information into the analysis of player technique, performance, and decision-making.

Despite recent technological developments, much of the detailed information contained in new movement data remains underutilized. To date, research on body movement in football has largely focused on health-related questions or technical performance, with limited connection to action-level analysis. While studies of goalkeeper behavior have begun to demonstrate the value of detailed movement data, systematic methods for uncovering meaningful structures within large volumes of unlabeled skeletal trajectories are still lacking. By learning the structure of goalkeeper actions, we can identify and retrieve specific action types, providing targeted feedback to improve movement technique. When combined with contextual game information, this analysis can also support tactical decision-making and be integrated into a goalkeeper coach's workflow, enabling data-driven training interventions tailored to individual players.

1.2 Contributions

Building on this emerging line of work, this project contributes an unsupervised framework for the characterization of goalkeeper actions from large-scale skeletal tracking data. In contrast to prior approaches that rely on manually defined labels or predefined action taxonomies, we adopt a fully *unsupervised* methodology based on contrastive learning to learn latent representations of skeletal trajectories directly from industry-grade tracking data. These representations capture structural and temporal regularities in goalkeeper movement without requiring annotated training data.

On top of the learned latent space, we employ clustering techniques to group structurally similar motion patterns, enabling a data-driven exploration of goalkeeper behavior. Rather than presupposing a fixed taxonomy, the objective is to uncover which distinct action types emerge directly from the data and to analyze what characterizes them in terms of pose and movement dynamics. In this way, the project moves toward a systematic understanding of goalkeeper behavior grounded in large-scale skeletal tracking data, allowing action categories and their defining features to be inferred rather than imposed.

1.3 Report Outline

This report is divided into 5 chapters. In Chapter 1, we introduce the problem setting, outline the core challenges addressed in this project, and summarize our main contributions within the broader context of soccer analytics. Chapter 2 provides a structured review of prior research in soccer action recognition and related domains, critically examining existing methodologies and highlighting the specific gaps our approach seeks to address. In Chapter 3, we describe the methodology used to transform skeletal trajectories into goalkeeper action embeddings, including model architecture choices, data preprocessing, and training procedures. In Chapter 4, we compare finetuning a pretrained model with training from scratch and analyze the resulting embedding space using both quantitative and qualitative evaluation methods. Finally, in Chapter 5, we summarize our findings, highlight the strengths and limitations of our approach, and discuss potential directions for future work.

Chapter 2

Related Work

2.1 Skeleton-Based Approaches in Soccer Analytics

Traditional soccer analytics has largely relied on center-of-mass tracking data to analyze player positioning and tactical structures. While effective for modeling macroscopic movement patterns, such representations are insufficient for capturing fine-grained technical behavior and individual mechanics. This limitation is also reflected in the broader literature on body movement in football, which, perhaps with the exception of research on visual scanning behavior [1], has focused predominantly on (in vitro) analyses of technical performance and health-related issues [2]. The increasing availability of high-resolution skeletal tracking data enables a more detailed representation of player posture and motion, opening new possibilities for event detection, pose and movement analysis, and more comprehensive technical evaluation beyond traditional center-of-mass-based approaches.

One of the earliest works to leverage skeletal representations for goalkeeper analysis is that of Wear et al. [3]. They extract 3D body poses from broadcast images of 1v1 situations and penalty saves, construct a five-dimensional handcrafted feature representation, and apply unsupervised k-means clustering to identify interpretable posture categories such as *spread* and *smother*. While demonstrating that unsupervised grouping of goalkeeper poses can recover meaningful structure, their approach operates on static single-frame skeletons and therefore does not model the temporal dynamics of continuous goalkeeping actions. Similarly, based on large-scale skeletal tracking data from TRACAB [4], Gotthardt [5] investigates K-means-based and deep clustering methods to analyze the distribution of single-frame football poses, with particular attention to rare or challenging poses and their relation to tracking performance, rather than modeling actions at the trajectory level.

Other works move beyond static pose analysis toward temporal skeleton modeling, but methodological choices remain divided across learning paradigms. For instance, Yeung et al. [6] adopt an unsupervised Graph Recurrent Autoencoder to learn sequence-level pose representations for shot posture and technique analysis. While this ap-

proach avoids reliance on manual labels, its application is still primarily oriented toward specific technical interpretations rather than learning action taxonomy and dynamics. In contrast, Bian [7] builds a large-scale skeleton extraction and validation pipeline but operates within a predefined event taxonomy for classification tasks such as duel-like interactions and goalkeeping outcomes. Similarly, Schepers et al. [8] use supervised learning with handcrafted pose-based features, including balance and relative orientation measures, to predict dribble success.

Collectively, these works highlight the promise of skeletal tracking data for detailed movement and performance analysis but also reveal a methodological gap in representation learning for player actions in soccer. Current approaches tend to either focus on unsupervised modeling of limited posture semantics, rely on fixed event taxonomies, or depend on supervised objectives and manually engineered features. As a result, the continuous and highly dynamic nature of soccer motion remains underexplored, particularly in terms of learning label-free representations that can capture technical behavior across diverse playing situations.

2.2 Skeleton-Based Action Recognition beyond Soccer

Beyond soccer-specific applications, 3D skeleton-based action recognition has been extensively studied in computer vision. Spatial-Temporal Graph Convolutional Networks (ST-GCN) [9] model skeleton sequences as graphs with spatial edges representing anatomical connections and temporal edges linking joints across time. This formulation enables joint spatiotemporal feature extraction and has become a standard approach for supervised action classification. Subsequent variants [10, 11, 12, 13] improve long-range dependency modeling and feature expressivity through adaptive adjacency matrices, multi-scale aggregation, or attention mechanisms.

While these methods achieve strong benchmark performance, they are inherently supervised and require labeled data for training. In professional sports settings, however, large-scale skeletal tracking data is typically unlabeled, and action categories are not always clearly defined. This limits the direct applicability of action recognition frameworks to real-world goalkeeper analysis.

2.3 Self-Supervised Representation Learning

To reduce dependence on manual annotation, recent work has explored self-supervised and contrastive learning for generic skeletal motion data. For example, Li et al. [14] propose a cross-view contrastive framework that learns consistent representations across complementary skeletal views such as joints, bones, and motion. Such approaches demonstrate that meaningful spatiotemporal embeddings can be learned without explicit action labels, often serving as pre-training for downstream supervised tasks.

Despite these advances, the application of self-supervised representation learning to large-scale skeletal tracking data in professional soccer remains limited. In particular, little work has addressed the problem of discovering and characterizing player action taxonomy directly from unlabeled, temporally continuous skeletal trajectories.

2.4 Research Gap and Positioning

Overall, prior work either (i) clusters static goalkeeper poses, (ii) learns sequence representations for narrow technical analyses, or (iii) applies supervised models within predefined event taxonomies. Even though self-supervised contrastive learning has shown strong potential for generic skeleton sequences, it has not been studied on the temporally continuous, unlabeled skeletal trajectories available in professional soccer, where action boundaries and categories are inherently ambiguous. This leaves a gap in learning representations that are both label-free and explicitly sensitive to the spatiotemporal structure of real match motion.

To address these limitations, we adopt a fully unsupervised contrastive learning framework that operates directly on skeletal trajectories, eliminating the need for handcrafted features or predefined action taxonomies. By learning from industry-grade tracking data without manual annotations, our model aims to capture the structural and temporal patterns underlying goalkeeper actions and to provide embeddings suitable for downstream discovery and analysis.

Chapter 3

Methodology

3.1 Dataset

For this project, we used tracking data from all 51 matches of the UEFA Euro 2024 tournament. The dataset was generated from broadcast video footage using the Hawk-Eye SkeleTRACK system [15], which employs ultra-high-speed camera networks and AI-based reconstruction techniques to estimate player motion in three-dimensional space [16]. The system provides positional information for the ball and player centroids, as well as a skeletal representation composed of 29 anatomical keypoints. These keypoints include detailed limb and extremity landmarks such as heels, toes, and fingers, along with major joints including the shoulders, hips, knees, and ankles, as illustrated in Figure 3.2.

The tracking data is organized into modality-specific sets of minute-level JSON files for each match, where the ball, centroid, and skeletal data are stored in separate file groups. In the skeletal modality, each frame records the 3D coordinates (in meters) of all 29 skeletal keypoints for every player and referee on the pitch, capturing the full-body pose configuration at a given timestamp.

The coordinate system (see Fig. 3.1) is right-handed and pitch-centric, with the origin $(0, 0, 0)$ located at the center of the field. The X-axis runs along the length of the pitch (touchline direction), the Y-axis spans the width of the pitch (goal-line direction), and the Z-axis represents the vertical (upward) direction. All recordings were sampled at 25 frames per second.

Prior to analysis, the raw data were preprocessed to extract only goalkeeper-related actions of potential relevance. The detailed preprocessing procedure is described in the following section.

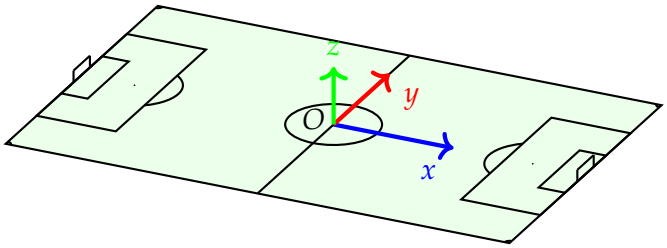
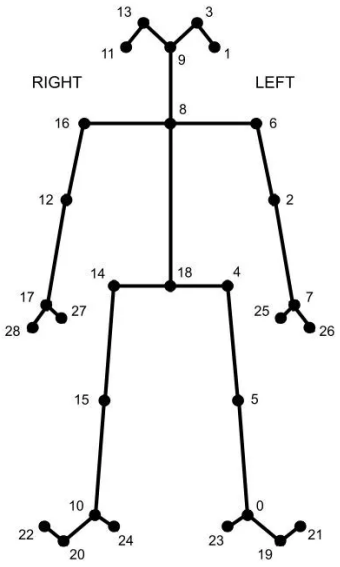


Figure 3.1: Pitch-centric 3D coordinate system. The origin is located at the center of the field. The x-axis (blue) runs longitudinally along the length of the pitch, the y-axis (red) spans the width of the pitch, and the z-axis (green) represents vertical elevation orthogonal to the pitch plane.



0 - Left ankle
1 - Left ear
2 - Left elbow
3 - Left eye
4 - Left hip
5 - Left knee
6 - Left shoulder
7 - Left wrist
8 - Neck
9 - Nose
10 - Right ankle
11 - Right ear
12 - Right elbow
13 - Right eye
14 - Right hip
15 - Right knee
16 - Right shoulder
17 - Right wrist
18 - Mid hip
19 - Left big-toe
20 - Right big-toe
21 - Left small-toe
22 - Right small-toe
23 - Left heel
24 - Right heel
25 - Left thumb
26 - Left pinky
27 - Right thumb
28 - Right pinky

Figure 3.2: The 29-joint skeletal representation used in the UEFA Euro 2024 tracking dataset.

3.2 Data Preprocessing

The raw skeletal tracking data goes through a multi-stage preprocessing pipeline to ensure suitability for ST-GCNs [9]. The pipeline consists of four main phases: filtering, spatial normalization, action segmentation, and tensor transformation.

3.2.1 Data Filtering and Synchronization

A standard soccer match generates large volumes of continuous tracking data, the majority of which correspond to players and particularly the goalkeeper remaining in passive or low-activity states that are not directly related to the defensive actions of interest in this study. To efficiently filter and isolate relevant actions, we leverage the player and ball positions as contextual proxies for relevant events. This requires a

multi-stage synchronization and aggregation process:

- **Role-based extraction:** The raw tracking data contains skeletal information for all players and referees on the pitch. As a preliminary step, we filter the input stream to strictly retain frames where the identified role is "Goalkeeper".
- **Stream synchronization:** Since skeletal and ball tracking data are provided as independent streams, we synchronize the goalkeeper's pose with the ball trajectory by aligning timestamps chronologically. To account for ball occlusions or tracking dropouts, we implement a "sample-and-hold" strategy: if ball data is missing, the last known position is retained for up to 5 seconds. This ensures we do not discard valid defensive sequences due to momentary ball-tracking failures.
- **Spatial filtering:** After aligning the streams, we filter the data, retaining *only* goalkeeper pose frames corresponding to moments when the ball is within the goalkeeper's defending final third.

  We define the *defending final third* as the 35m portion of the pitch, measured along its length, that is closest to the goal being defended. Hence, each team (and goalkeeper) has its own defending final third, defined relative to the goal it is defending. On a 105m pitch with the origin at the center spot and the x-axis along the length of the field (see Fig. 3.1), the defending final thirds of *both* teams are defined by the region $|x| \geq 17.5$m.

  This criterion serves as a high-level heuristic to discard the large volume of uninteresting data where the goalkeeper is idle, focusing the model exclusively on potential defensive scenarios.
- **Sequence aggregation:** The spatial filtering process yields a stream of potentially relevant poses (frames), but meaningful defensive actions occur over time as continuous sequences of frames. We aggregate these frames into multiple *contiguous* temporal sequences, grouping them by the goalkeeper's unique ID. Whenever the ball leaves the third (i.e., the spatial filtering conditions are no longer satisfied), the current sequence is effectively terminated and treated as a complete segment. To ensure sufficient temporal context for motion analysis, we strictly enforce a minimum duration: any resulting continuous sequence shorter than 50 frames (2 seconds at 25 FPS) is discarded. The final output maps each goalkeeper to a list of discrete, uninterrupted frame sequences of potentially relevant actions.

3.2.2 Spatial and Skeletal Normalization

To ensure the model learns generalized movement patterns rather than overfitting to specific pitch locations or player physical characteristics, we perform two critical normalization steps:

- **Goal alignment:** In the raw global tracking data, goalkeeper movements are expressed in a pitch-centric coordinate system (see Fig. 3.1). Consequently, goalkeepers defending opposite ends of the pitch face opposite directions along the touchline direction axis. From a learning perspective, this means that biomechanically identical actions may appear as mirrored or inverted patterns in the data. In addition, this pitch-centric coordinate system is not ideal for characterizing goalkeeper movements, as it encodes positions relative to the entire field rather than relative to the defended goal, which is the primary spatial reference for goalkeeper actions. Hence, movements expressed in global field coordinates do not directly reflect their functional meaning.

  To obtain a representation that more naturally reflects goalkeeper-specific movements, and to ensure that identical actions share a consistent geometric interpretation, we transform all raw skeletal coordinates into a new single *canonical local reference frame* defined relative to the defended goal. In this standardized representation, every goalkeeper is positioned in the same spatial configuration and always faces the same forward direction.

  a. Target coordinate definition: For each sequence, we define a new local coordinate system attached to the goal being defended, as illustrated in Fig. 3.3:

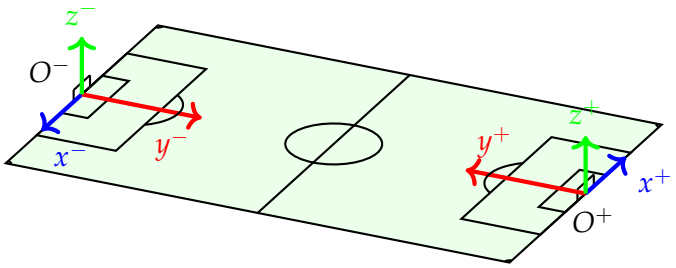


Figure 3.3: Goal-centered 3D local reference frame (x', y', z'). The origin of the coordinate system is located at the midpoint of the defended goal line. O^- and O^+ represent the origins of the local coordinate systems at the negative and positive end goals, respectively. The x-axis (blue) runs laterally along the goal line (positive toward the goalkeeper's right when facing into the field), the y-axis (red) points forward into the pitch, and the z-axis (green) represents vertical elevation orthogonal to the pitch plane.

  - **Origin** $(0, 0, 0)$**:** The midpoint of the defended goal line, i.e., the point on the ground equidistant from both goalposts.
  - **X-Axis (Lateral):** The horizontal direction parallel to the goal line. The positive X-direction corresponds to lateral movement to the goalkeeper's right when facing into the field (positive Y).
  - **Y-Axis (Forward):** The direction perpendicular to the goal line on the pitch plane, pointing from the goal line into the field of play.
  - **Z-Axis (Vertical):** The axis orthogonal to the pitch plane, pointing upward, identical to the vertical axis in the original (raw) global coordinate system.

b. Transformation logic: To transform global pitch coordinates (x, y, z) into the negative- and positive-end canonical local frames (x^-, y^-, z^-) and (x^+, y^+, z^+) respectively, we apply a rigid-body transformation composed of a rotation and a translation, conditioned on which goal is being defended. Note that the vertical coordinate remains unchanged, i.e., $z^- = z^+ = z$, while the horizontal and forward coordinates are reoriented and shifted to align with the local goal-centered reference system:

- *Defending the negative-end goal ($x \approx -52.5m$, facing $+x$):* The local coordinate-system forward direction aligns with the global coordinate-system touchline direction. After translating the origin to the goal line center, we rotate the system so that forward motion is consistently mapped to the positive local y-axis:
$$x^- = -y, \quad y^- = 52.5 + x.$$
- *Defending the positive-end goal ($x \approx 52.5m$, facing $-x$):* The global coordinate-system touchline direction axis points toward the goal rather than into the field. We therefore invert this axis and translate the origin to the goal line center:
$$x^+ = y, \quad y^+ = 52.5 - x.$$

This procedure ensures that every sample encodes a goalkeeper located at the same virtual goal and facing the same positive forward direction ($+y^+$ or $+y^-$). As a result, equivalent movements share the same geometric representation, eliminating directional bias and reducing unnecessary variance in the training data.

- **Body size normalization:** To account for substantial physique variation among goalkeepers (e.g., differences in height and arm span), we construct a *canonical skeleton* by computing average bone lengths across the entire dataset. Each skeleton in each frame is then rescaled to match this standardized skeletal structure. This normalization reduces spurious correlations between body shape and motion patterns caused by differences in physique rather than biomechanical behavior.
- **Input normalization** Finally, after body size normalization, no further global spatial rescaling (e.g., mapping skeletal coordinates into a unit cube) is applied before feeding data into the network. Instead, the raw metric structure of the pitch dimensions is preserved. The ST-GCN encoders include an input-level batch normalization layer, which stabilizes feature distributions during training and reduces the need for explicit global rescaling.

3.2.3 Adaptive Action Segmentation

Even after restricting the data to movement sequences where the ball is located in the goalkeeper's defending final third, many segments still contain low-activity behav-

ior such as walking or standing. Since the primary interest is in characterizing and clustering high-intensity actions such as saves and diving movements, we further perform automated sequence segmentation to reduce the number of uninteresting action clips. To achieve this, we employ an energy-based segmentation approach analogous to outlier detection, using a robust Z-score method:

- **Motion Energy calculation:** For each continuous skeletal sequence extracted in 3.2.1, we compute a temporal energy profile $E = \{E_1, \ldots, E_T\}$. The motion energy at each frame t, E_t, represents the total kinetic displacement of the skeletal structure, defined as the sum of Euclidean distances traveled by all V joints relative to the previous frame:

$$E_t = \sum_{v=1}^{V} \|\mathbf{p}_{t,v} - \mathbf{p}_{t-1,v}\|_2$$

  where $\mathbf{p}_{t,v} \in \mathbb{R}^3$ is the normalized 3D position of joint v at time t. To prevent high-frequency tracking noise (jitter) from triggering false positives, this raw energy signal is smoothed using a temporal sliding window (window size ≈ 0.4s).

- **Robust adaptive thresholding:** To distinguish high-intensity actions (saves, dives) from baseline idle behavior (walking, shuffling), we employ a robust statistical outlier detection method. Standard deviation is sensitive to extreme values, so a few high-energy saves could inflate the threshold and lead to missed detections. Instead, we estimate the background noise level using the Median Absolute Deviation (MAD) of the energy signal:

$$\text{MAD}(E) = \text{Median}_{t=1,\ldots,T}\left(|E_t - \text{Median}_{t=1,\ldots,T}(E_t)|\right)$$

  We define the baseline "non-action" energy level as the median of the sequence's energy profile, $\text{Median}_{t=1,\ldots,T}(E_t)$. We then establish a dynamic action threshold:

$$\text{Threshold}_{\text{adaptive}} = \text{Median}_{t=1,\ldots,T}(E_t) + \lambda \cdot \max(\sigma_{\text{MAD}}, \varepsilon_\sigma)$$

  where $\sigma_{\text{MAD}} = 1.4826 \cdot \text{MAD}$ is a robust estimator of standard deviation (scaled for consistency with a Gaussian energy distribution assumption). The parameter λ controls sensitivity, determining how many robust deviations above the baseline are required to classify a frame as active. To prevent excessive sensitivity in extremely still sequences (where MAD ≈ 0), we enforce a minimum noise floor ($\varepsilon_\sigma = 0.2$). Finally, to ensure that valid actions possess a minimum absolute kinetic intensity, the effective threshold is clamped to a hard lower bound τ_{min}:

$$\text{Threshold}_{\text{final}} = \max(\text{Threshold}_{\text{adaptive}}, \tau_{\text{min}})$$

  We empirically set $\tau_{\text{min}} = 1.35$, a value corresponding approximately to the 75th percentile of energy in our dataset analysis.

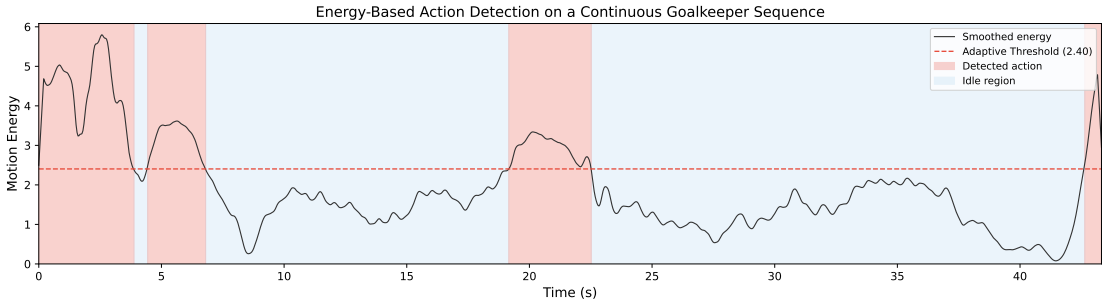


Figure 3.4: Example of the motion energy profile for a continuous goalkeeper sequence after applying the spatial filtering stage. Frames exceeding the adaptive energy threshold are marked as active (candidate actions), while frames below the threshold are classified as idle. The highlighted peaks correspond to high-intensity movements. The corresponding animation of this sequence is available online: https://polybox.ethz.ch/index.php/s/HHpawYaQ6wjBMRF

Then, frames where the motion energy exceeds this threshold ($E_t > \text{Threshold}_{\text{final}}$) are flagged as active candidates, as illustrated in Figure 3.4. Note that this method is not foolproof, as behaviors such as walking or shuffling, if performed with sufficient intensity, may still be classified as actions. Nevertheless, it effectively filters out a large number of clearly uninformative sequences.

Subsequently, we transform these raw candidate frames into cohesive action sequences using a "temporal gap filling" step (morphological closing). Because complex actions like diving often contain brief moments of lower velocity, a simple frame-by-frame threshold triggers fragmentation. To resolve this, any active frames or short segments separated by a gap of less than 0.2 seconds are merged into a single continuous block (i.e., if the frames are close enough, the current frame is considered part of the ongoing action). This ensures that a complete movement is captured as one unified sequence rather than being split into multiple disjoint clips.

- **Clip extraction and centering:** Once the *high-energy action segments* have been identified by the adaptive thresholding algorithm (defined by their start and end frames), we proceed to extract action and idle clips from the continuous sequences from 3.2.1.
  - *Action Extraction:* To capture the full motion context, including the preparatory set position and the recovery (follow-through) phase, we extend each identified segment by adding a temporal buffer of 1.0 seconds before the start and 1.5 seconds after the end. A safety limit of 300 frames is enforced to ensure consistent tensor sizes. This upper bound follows the preprocessing convention established in [14] for the NTU RGB+D data [17]. If an extended segment exceeds this limit, we create a 300-frame window around the frame of peak motion energy within the action. This ensures that for exceptionally long sequences, the primary action remains the focal point

while additional context is trimmed.

- *Idle Extraction:* In order to have a representative distribution of movement intensities, we also extract clips from the *low-energy segments* where the adaptive threshold was not met. We filter these idle periods to find stable windows lasting at least 4.0 seconds. From the center of each valid idle period, we extract a single 4.0-second sample. These clips represent "non-action" states (e.g., walking back to the line, standing still) and are crucial for training the model to distinguish between relevant goalkeeping actions and background movement.

Finally, to standardize the temporal dimension for the neural network (which expects fixed-length sequences), every resulting clip (action or idle) is temporally resampled via linear interpolation to a fixed tensor length of $T = 50$ frames, regardless of its original physical duration. Because temporal resampling alters the effective execution speed of the motion, we first fixed the duration of each idle clip to 4.0 seconds and only then applied linear interpolation, thereby minimizing speed distortions introduced by the resampling process. This duration provides sufficient temporal context to represent stable non-action behavior while maintaining comparable temporal scaling across action and idle clips.

3.2.4 Tensor Representation

The final processed data is structured a 5D tensor of shape (N, C, T, V, M), where N denotes the number of sequences, $C = 3$ corresponds to the spatial coordinates in 3D (x, y, z), $T = 50$ is the fixed temporal length (number of frames) per sequence, V represents the number of skeletal joints, and M denotes the number of subjects included in the sample (i.e., the number of players analyzed per sequence).

Data Splits

To ensure robust evaluation, we partitioned the dataset by goalkeeper identity rather than random shuffling, preventing data leakage where a model might memorize a specific player's movement style. The *validation set* is composed of all sequences from goalkeepers whose teams were eliminated in the group stages (approx. 20% of the data), while the *training set* contains all sequences (including group stage matches) from goalkeepers who advanced to the knockout stages. This allows us to test the model on unseen subjects.

Additionally, to accommodate different modeling strategies, we construct two distinct versions of the dataset:

Native representation

This format preserves the maximum amount of original data detail for training architectures from scratch (full training).

- **Topology:** Retains all 29 original skeletal joints.
- **Temporal resolution:** Maintains the original capture rate of 25 FPS.
- **Coordinate system:** Uses the standard metric system with the Z-axis representing height (up).
- **Alignment:** To remove variations caused by the goalkeeper's facing direction on the pitch, we perform a canonical alignment. We compute the vector connecting the left shoulder to the right shoulder for every frame in a sequence and calculate its yaw angle in the horizontal plane (X-Y). By taking the median angle across the sequence, we determine the goalkeeper's dominant orientation. We then rotate the entire sequence around the vertical (Z) axis by the inverse of this angle, effectively aligning the median shoulder plane with the X-axis. Finally, the sequence is translated so that the root joint (mid-hip) is at the origin in every frame, removing absolute pitch coordinates while preserving relative limb motion. This root-centering transformation follows the preprocessing convention adopted in [14] for the NTU RGB+D dataset [17], using the mid-hip joint as the reference point instead of the spine joint, which is not defined in the 29-joint Hawkeye skeletal model.
- **Subject dimension:** $M = 1$, as we isolate single-player performance.

NTU-aligned representation

To leverage transfer learning from models pre-trained on the NTU RGB+D dataset [17] (a standard benchmark in skeletal action recognition), we mapped our data to better match its specifications.

- **Topology:** The original 29-joint skeletal representation is mapped onto the standard 25-joint NTU RGB+D skeleton topology. The corresponding NTU RGB+D keypoints are illustrated in Figure 3.6, and the detailed joint-mapping specification between the Hawk-Eye and NTU RGB+D formats is provided in Table A.1.
- **Temporal resolution:** Sequences are linearly resampled to 30 FPS to match the temporal dynamics of the NTU RGB+D dataset.
- **Coordinate system:** The local coordinate system defined in 3.2.2 follows a standard Euclidean convention where the Z-axis represents height. To align with the input requirements of the pre-trained ST-GCN model on NTU RGB+D, we transform the coordinates to the Kinect V2 camera reference frame. This involves mapping the source coordinates (x, y, z) to the target Kinect coordinates (x', y', z') such that (see Figure 3.5):

  $$x' = -x, \quad y' = z \quad (\text{Height}), \quad z' = y \quad (\text{Depth})$$

  This ensures that the *vertical axis corresponds to Y* and the *depth axis corresponds to Z*, matching the skeletal representation the network was pre-trained on.

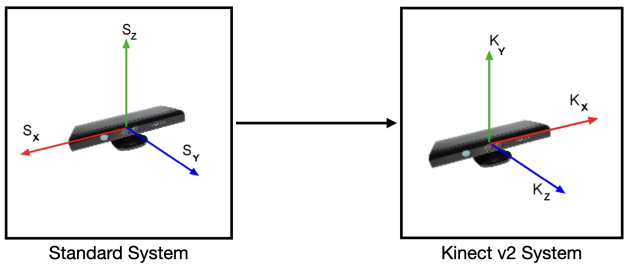


Figure 3.5: Coordinate axis remapping to align the original dataset representation with the Kinect V2 camera reference frame used in NTU RGB+D.

- **Alignment:** We apply the same view-invariant normalization logic as the native format, but adjusted for the Kinect coordinate system. We compute the median yaw angle of the shoulders and rotate the skeleton around the vertical axis (now Y-axis) so the subject faces the positive Z-direction (depth). Finally, the entire sequence is centered by subtracting the *spine shoulder* joint (i.e., joint 21 in Figure 3.6) from all coordinates, aligning the data relative to the upper torso as per standard NTU RGB+D convention.

  Note that, unlike the strict normalization strategy used in [14], which rigidly aligns the body orientation to the X-axis at every single frame (effectively removing all rotation), we only perform a sequence-level alignment. By rotating the entire sequence based on the median shoulder angle, we remove arbitrary global variations (e.g., facing left vs. right) while preserving the intrinsic rotational dynamics of the action itself (e.g., twisting the torso during a dive), which provides critical motion cues for embedding learning.

- **Subject Dimension:** $M = 2$, effectively zero-padding the second person channel to satisfy the input layer dimensions of standard pre-trained ST-GCN weights.

The finalized dataset for the *Native* representation contains a total of 28,285 samples, composed of 17,103 active clips and 11,182 idle clips. These are split into 21,204 training samples (75%) and 7,081 validation samples (25%). The *NTU-aligned* version resulted in a total of 24,369 samples (18,307 training, 6,062 validation), containing 13,593 active and 10,776 idle clips. This reduction primarily affects the active class, as we enforce a fixed minimum energy floor ($\tau_{min} = 1.35$). The reduction in joints (29 to 25) and increase in frame rate (25 to 30 fps) lowers the total computed motion energy per frame, causing more low-intensity action clips to fall below this absolute cutoff.

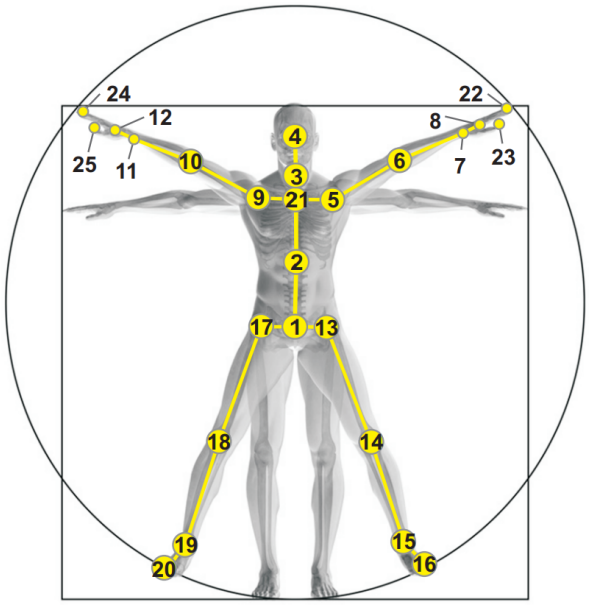


Figure 3.6: Illustration from [17]. The 25-joint skeletal representation used in the NTU RGB+D dataset. The labels of the joints are: 1-base of the spine, 2-middle of the spine, 3-neck, 4-head, 5-left shoulder, 6-left elbow, 7-left wrist, 8-left hand, 9-right shoulder, 10-right elbow, 11-right wrist, 12-right hand, 13-left hip, 14-left knee, 15-left ankle, 16-left foot, 17-right hip, 18-right knee, 19-right ankle, 20-right foot, 21-spine, 22-tip of the left hand, 23-left thumb, 24-tip of the right hand, 25-right thumb.

3.3 Model Architecture

To learn distinct motion representations from the skeletal sequences, we adopt the CrosSCLR architecture proposed in [14]. CrosSCLR is an unsupervised, contrastive learning framework designed specifically for skeleton-based action representation learning.

The model utilizes the Spatio-Temporal Graph Convolutional Network (ST-GCN) [9] as its encoder backbone. CrosSCLR consists of up to three parallel encoders, each operating on a different modality derived from the same underlying skeleton sequence: joint, motion, and bone representations. The *joint* modality uses the raw 3D joint coordinates; the *motion* modality captures temporal differences between consecutive frames (velocity); and the *bone* modality encodes relative geometric relationships vectors between connected joints. By learning from these complementary views, the model captures both the spatial structure and temporal dynamics of human movement.

A key motivation behind this multi-view design is to improve the construction of positive and negative pairs in the contrastive learning objective. Each sample in a typical single-view contrastive framework usually has only a few positive pairs (e.g., augmented versions of itself), while all other samples are treated as negatives. These samples are taken from a large, first-in-first-out memory bank to ensure a sufficient variety of negatives. Importantly, in the unsupervised setting, no ground-truth action

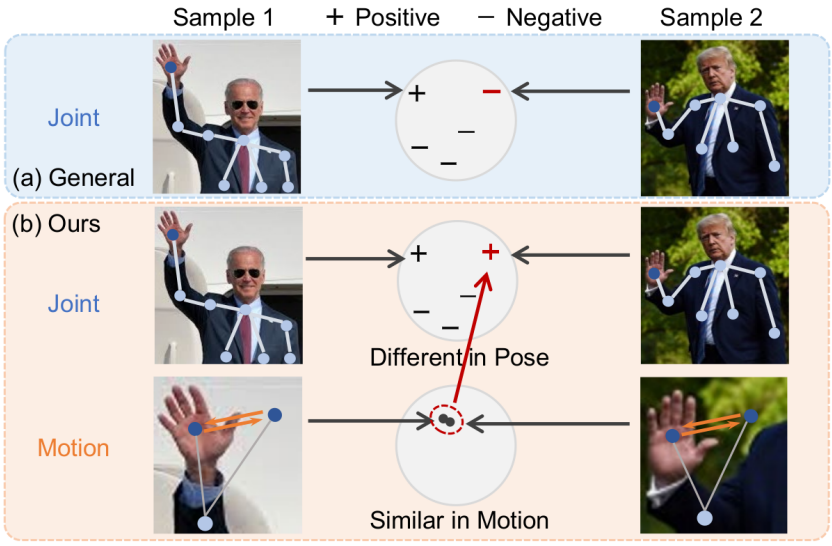


Figure 3.7: Illustration from [14] demonstrating the limitation of standard contrastive learning and the advantage of the CrosSCLR framework. In a conventional contrastive setup, two semantically similar actions (e.g., hand-waving) may be treated as negative pairs and therefore pushed apart in the embedding space. CrosSCLR mitigates this issue through cross-view information exchange: if two samples are identified as similar in one modality (e.g., motion), this similarity is propagated to the other modalities, encouraging consistent and semantically meaningful representations across views.

labels are available to determine semantic similarity. As a result, the notion of a "negative" is defined purely by sample identity: any sequence that does not originate from the same instance as the anchor is automatically assigned as a negative. This means that two motions depicting the same underlying action but performed by different subjects are still treated as negatives and are therefore pushed apart in the embedding space (see Figure 3.7). Such label-agnostic negative assignment can prevent the model from organizing the embedding space according to meaningful action-level structure.

CrosSCLR addresses this false-negative limitation through a high-confidence knowledge mining mechanism. During the initial training phase (warm-up), each modality encoder is trained independently using standard contrastive learning to establish a base representation. Once this phase ends, the cross-view mining begins.

The mechanism operates by leveraging agreement across different views. The approach is based on the observation that, after single-view contrastive learning, embeddings that are close in the representation space are likely to belong to the same semantic category, while distant embeddings are unlikely to correspond to the same action class. Accordingly, the model enhances contrastive learning by identifying highly similar embeddings and treating them as soft positive pairs to encourage tighter semantic clustering.

For a given anchor sample, similarity scores are computed between the anchor and samples stored in the memory bank across modalities. If a sample is identified as

highly similar to the anchor in one high-confidence view, this similarity information is transferred to the other views. Crucially, the model does not simply treat these discovered samples as binary ”yes/no” matches. Instead, it uses the actual similarity scores from a high-confidence view to *weight* how the other views should represent that sample. By transferring this soft knowledge, one modality can guide another to recognize semantically similar movements that it might have otherwise missed, thus expanding the set of positive pairs.

The selection of CrosSCLR as our representational backbone is motivated by three main factors. First, its unsupervised nature allows us to leverage large-scale skeletal data without requiring manual annotation, which would be impractical due to the scale of the dataset. Second, its multi-view design jointly models spatial pose structure and temporal motion dynamics, which is well suited for capturing goalkeeper movement patterns. Finally, the framework is compatible with data augmentation strategies that encourage embedding invariance to symmetries in goalkeeper actions, such as left–right dives, as well as to minor temporal misalignment within the clip window. These augmentation strategies are described in detail below.

Data Augmentations

Since contrastive learning relies heavily on the quality and diversity of data views to learn invariant features, a set of skeletal augmentations is applied during training. To adapt the framework to the specific challenges of goalkeeper motion analysis, we extend the standard augmentation suite proposed in [14] with domain-specific transformations.

Standard CrosSCLR augmentations: The original framework employs two primary augmentations to ensure temporal and spatial invariance:

- **Temporal crop:** A temporal cropping augmentation is applied to each sequence to introduce temporal shift invariance. The sequence is first symmetrically padded along the temporal dimension using reflection, after which a contiguous window of the original length is randomly cropped. This operation introduces temporal shift invariance by exposing the model to sequences that are slightly displaced in time, thus reducing sensitivity to the exact temporal alignment of the action within the clip window.
- **Shear:** A random linear shear transformation is applied to the 3D joint coordinates. This operation inclines the 3D pose by a small random angle through a shear matrix, simulating mild viewpoint variations and perspective distortions. By perturbing the global spatial configuration while preserving the underlying motion dynamics, the augmentation encourages the learning of features that are robust to viewpoint changes and minor geometric deformation.

Additional domain-specific augmentations: We introduce two additional augmentations to address the symmetries and noise characteristics inherent to goalkeeper skeletal data:

- **Mirroring (Left-right flip):** With a fixed probability, the skeleton is mirrored along the lateral movement axis, and the corresponding left and right body joints are swapped (e.g., left hand $\leftrightarrow$ right hand). Because goalkeeping actions are largely symmetric with respect to lateral motion, left–right mirroring encourages the model to treat opposite directional movements as semantically equivalent.
- **Gaussian noise (jitter):** Small zero-mean Gaussian noise is added to the joint coordinates. This simulates minor tracking inaccuracies commonly present in pose estimation systems and helps the model become more robust to noisy skeletal inputs.

The four augmentations are applied sequentially to each input sample, with mirroring performed randomly with a probability of 0.5.

3.4 Training Details

Our implementation adapts the original CrosSCLR framework configuration [14], as provided in the official model implementation. All models are trained using Stochastic Gradient Descent with a standard momentum of 0.999 and a weight decay of 10^{-4}. We set the contrastive loss temperature $\tau = 0.07$ and the feature dimension to 128. To prevent overfitting, we maintain the reduced model capacity of the original implementation, setting the hidden channels to 16 ($1/4\times$ standard ST-GCN). During training, we apply data augmentation including random shear (amplitude 0.5), temporal padding (ratio 6), Gaussian noise ($\sigma = 0.01$), and mirroring (probability 0.5).

Hyperparameter Adjustments: We introduce some modifications to the original settings to suit our dataset constraints and hardware:

- **Queue size:** We reduce the contrastive memory queue (i.e., memory bank) size from $K = 32,768$ to $K = 16,384$ to ensure sufficient negative sample coverage relative to our dataset cardinality.
- **Batch scaling:** Due to hardware constraints, we reduce the batch size from 128 to 64. Accordingly, we linearly scale the base learning rate (from 0.1 to 0.05).

We evaluate two distinct training protocols corresponding to the data representations described in Section 3.2.4:

Training from scratch (Native): Using the native 29-joint graph layout, we train for 300 epochs with an initial learning rate of 0.05, decayed by a factor of 10 at epoch 250. The cross-view consistency loss is enabled after epoch 150, mirroring the schedule of the original implementation to allow for initial feature stability.

Fine-tuning (NTU-Mapped): We initialize the encoders with weights pre-trained on the NTU-RGB+D 60 dataset (cross-view protocol). For fine-tuning, we lower the base learning rate to 0.01 to preserve learned features and train for 200 epochs. The cross-view loss is reintroduced at epoch 100 of the fine-tuning phase. Weights correspond-

ing to the contrastive memory queue are explicitly excluded from loading to prevent distribution contamination from the pre-training domain.

All experiments were performed on a single NVIDIA GeForce RTX 4070 Laptop GPU.

3.5 Unsupervised Action Discovery

To discover latent action classes from the learned representations, we employ a two-stage pipeline consisting of dimensionality reduction and density-based clustering. This approach mirrors the methodology of BERTopic [18], adapting it for skeletal action analysis.

3.5.1 Dimensionality Reduction

The raw output of our CrosSCLR backbone consists of 128-dimensional feature vectors. In such high-dimensional spaces, distance metrics (like Euclidean or Cosine) suffer from the "curse of dimensionality," where the contrast between the nearest and farthest neighbors diminishes, making direct clustering unreliable.

To mitigate this, we employ Uniform Manifold Approximation and Projection (UMAP) [19]. While commonly used for visualization, we leverage UMAP for its efficacy as a non-linear manifold learning algorithm. Unlike linear techniques such as PCA [20], which we found insufficient for separating the complex, non-linear action manifolds in our data, UMAP excels at preserving the intricate local neighborhood structure. Intuitively, UMAP adapts to the local density of the data: it tightens local neighborhoods in the projection, effectively making dense regions denser and sparser regions sparser. This density exaggeration creates clear separation between action manifolds, transforming subtle high-dimensional structures into distinct "islands" suitable for density-based clustering.

We therefore employ UMAP to project the 128-dimensional embeddings into a lower-dimensional space. To remain consistent with the angular similarity objective of the contrastive loss, we use cosine distance as the metric in the embedding space. Preliminary experiments showed that the choice of projection dimensionality (evaluated between 5 and 100 dimensions) had little effect on clustering stability, provided all other pipeline parameters were held constant.

Robustness Checks: A limitation of UMAP for dimensionality reduction is the potential to introduce apparent cluster structure even when the underlying data are uniformly distributed noise. To rule out this possibility, we assessed clustering stability using two control analyses. First, we validated the robustness of our pipeline by applying it to unstructured Gaussian noise with matched first and second-order statistics. By comparing the silhouette scores of the resulting noise clusters against those of our learned embeddings, we confirmed that the pipeline does not hallucinate spurious structure in random data. Second, we compared silhouette scores against valid

baselines (including K-Means [21] and spectral clustering [22] on raw features), confirming that the UMAP-derived clusters reflect genuine, non-linear structural density rather than projection artifacts.

3.5.2 Clustering

A key requirement for our analysis is the ability to discover actions without a priori assumptions about the number of clusters (K). Traditional approaches like K-Means [21] or spectral clustering [22] force every data point into a partition and require K to be specified, which we found unsuitable for exploring the unknown distribution of goalkeeper actions.

Instead, we utilize HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [23]. HDBSCAN extends standard density-based clustering (DBSCAN) [24] by converting it into a hierarchical framework. The intuition behind this choice is twofold:

- **Adaptive density:** It can detect clusters with varying densities and shapes, which is important for action classes exhibiting different levels of intra-class variance (e.g., a dynamic dive tends to have higher variance than a static ready pose). This flexibility is not shared by DBSCAN.
- **Noise handling:** Unlike partition-based methods, HDBSCAN explicitly models "noise." Points falling in low-density regions (ambiguous poses or transitions) are labeled as outliers (-1) rather than being forced into a cluster. This ensures that the resulting clusters represent high-confidence action prototypes.

  It is important to note that there is a distinction between inherently noisy samples in the dataset and samples that are labeled as noise by the clustering algorithm. The former refers to capture errors present in the data itself, whereas the latter is a modeling outcome produced by the clustering procedure when points are assigned to low-density regions outside stable clusters.

Chapter 4

Results and Discussion

4.1 Comparison Between Fine-Tuning a Pretrained Model and Training from Scratch

We first examined, through qualitative analysis, the cluster structure obtained from embeddings of the pretrained model fine-tuned on goalkeeper data. Despite the fine-tuning, we observed that the clusters still lacked clear semantic separation, with different goalkeeping actions often grouped together in overlapping regions of the representation space. This indicated that the pretrained features, even after fine-tuning, were not sufficiently specialized for the goalkeeping domain. We hypothesize that the pretraining dataset (NTU RGB+D), while large and diverse, does not adequately capture the motion patterns specific to goalkeeper actions, creating a domain mismatch between datasets. These findings motivated training the model from scratch to learn representations more closely aligned with the target task.

4.2 Analysis of Clusters in the Embedding Space Learned from Scratch

To investigate the representation space learned by our model trained from scratch, we computed embedding vectors for all active sequences extracted using the adaptive motion energy filter (see Section 3.2.3). Since our objective is to analyze the structure of semantically meaningful goalkeeper actions, we restrict the clustering and subsequent analysis to these active segments and exclude idle sequences that do not contain informative motion patterns. We first conducted the analysis on the same sequences that were employed during training, as our goal was not to evaluate performance, but to analyze the structure of the data that the model has internalized. Importantly, complementary validation experiments showed that similar structural patterns were observed on held-out data, indicating that the findings were not merely artifacts of overfitting.

4.2.1 Semantic Structure of the Learned Embedding Space

We reduced the embeddings to 20 dimensions using UMAP and performed clustering with HDBSCAN, following the procedure described in Section 3.5. The hyperparameters used are summarized in Table A.2. Each resulting cluster was qualitatively analyzed by randomly sampling frames and visually inspecting them. The overall cluster structure, visualized in 2D using UMAP, is shown in Figure 4.1. The labels derived upon visual inspection are given in Table 4.1.

We computed biomechanical metrics to validate the confidence of the manually assigned cluster labels. Specifically, we measured the minimum wrist position and maximum foot position over each sequence, as well as the peak velocities of the hip, ankle, and wrist. Additionally, we calculated the ratio of peak wrist and ankle velocities relative to the hip to assess whether movements involved the whole body or were isolated to specific joints. The results are summarized in Table A.4.

From the metrics, we observe that Cluster 2 (kick the ball) and Cluster 3 (pass) exhibit high ankle-to-hip velocity ratios, indicating isolated joint movements. Cluster 7 (saves) shows high wrist velocity together with maximum foot height, reflecting jumps and wrist movements consistent with diving or saving actions. Finally, Cluster 12 (sprinting) is characterized by large displacement, high hip velocity, and high ankle velocity, capturing the dynamics of sprinting.

Table 4.1: Cluster description and per-cluster statistics for action clips.

Cluster	Cluster Description	N	Duration (s)	Peak Energy
-1	Noise	5274	3.89 ± 1.16	7.45 ± 4.15
0	Walking Around	78	3.74 ± 0.27	4.04 ± 0.64
1	Walking Around	1179	2.94 ± 0.66	3.24 ± 1.06
2	Kick the ball	761	3.05 ± 0.62	8.73 ± 2.42
3	Pass or throw the ball	98	3.18 ± 0.29	5.50 ± 2.24
4	Bending down	199	3.15 ± 0.37	4.97 ± 3.04
5	Walking around and grabbing ball	196	7.53 ± 0.87	10.59 ± 2.99
6	Ready pose for shot	118	4.04 ± 0.59	10.08 ± 2.76
7	**Saves and Dives**	490	4.52 ± 1.30	12.74 ± 5.47
8	Receive pass	197	3.34 ± 0.53	6.74 ± 1.59
9	Communication with hands	489	4.93 ± 0.63	6.33 ± 1.79
10	Passing and moving around	368	3.99 ± 0.86	9.54 ± 2.42
11	Ready pose and shuffling	259	3.47 ± 0.42	5.17 ± 1.44
12	Sprint and grab ball	541	6.41 ± 1.35	18.60 ± 4.61
13	Passing and moving around	112	5.05 ± 0.63	11.13 ± 1.98
14	Moving back	855	4.34 ± 0.96	9.02 ± 3.06
15	Shuffling around with ready pose	109	3.71 ± 0.56	5.13 ± 1.02
16	Hand signals / stretching / grabbing	730	3.35 ± 0.83	3.89 ± 1.46
17	Ready pose	743	3.22 ± 0.42	4.64 ± 1.88

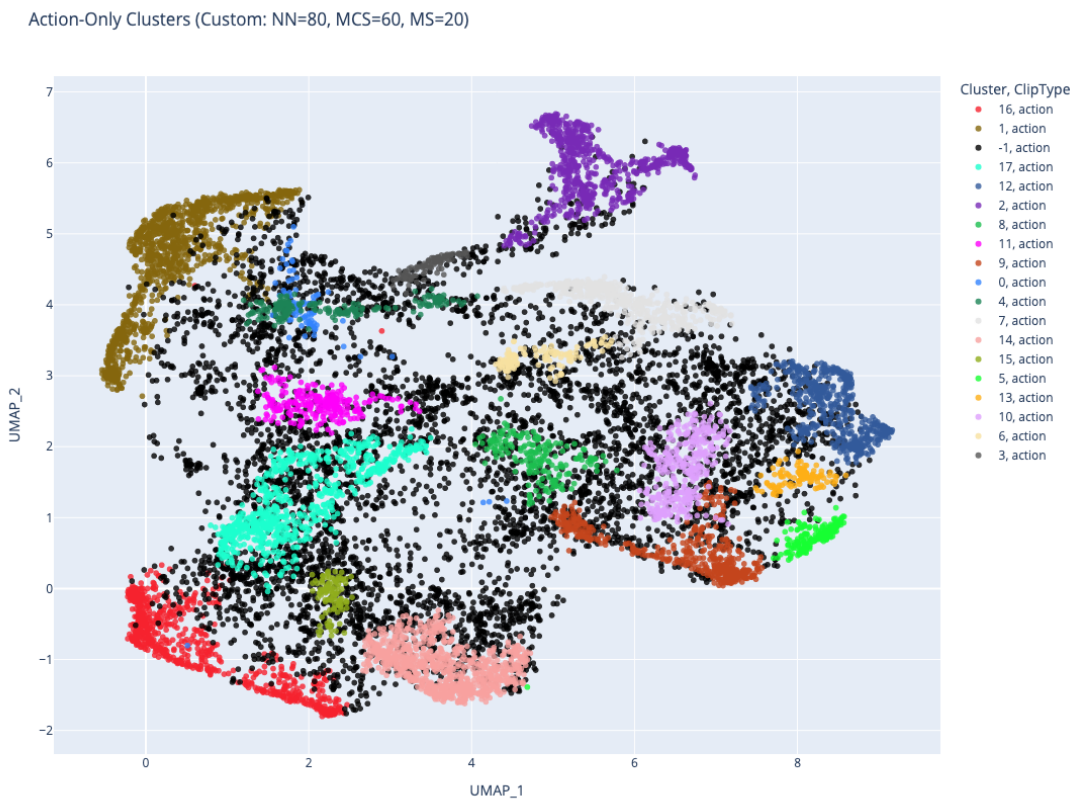


Figure 4.1: Two-dimensional UMAP projection of the learned motion embeddings for goalkeeper sequences. Each point represents a temporally normalized clip, and colors indicate cluster assignments obtained via HDBSCAN. The visualization illustrates the separation of distinct keeper motions in the learned representation space. Black points denote noise detected by HDBSCAN.

4.2.2 Quantitative Evaluation of Learned Embedding Space

To quantitatively assess cluster compactness and separability in the original embedding space, we computed the mean intra-cluster and inter-cluster cosine similarity using the raw (pre-UMAP) embeddings. We deliberately performed this analysis in the original representation space because the embeddings were trained using a contrastive objective based on cosine similarity. This objective explicitly encourages semantically similar actions to have high cosine similarity, while pushing dissimilar actions apart in angular space. Therefore, cosine similarity in the raw embedding space provides a direct measure of whether the learned representation itself meaningfully captures action similarity, independent of any downstream dimensionality reduction.

The overall mean intra-cluster cosine similarity was **0.4845**, indicating that samples within the same cluster are moderately aligned in the learned representation space. In contrast, the overall mean inter-cluster cosine similarity was substantially lower

at **0.2077**, suggesting clear separation between clusters and confirming that the contrastive objective successfully structured the embedding space.

Table 4.2: Cluster pairs with high inter-cluster cosine similarity (≥ 0.60) in the raw embedding space.

Cluster A	Cluster B	Cosine Similarity
3	4	0.6490
5	12	0.6008
5	13	**0.7388**
6	7	0.6826
10	13	0.6655
15	17	0.6445

While the overall mean inter-cluster similarity remains low (0.2077), a small number of cluster pairs exhibit elevated similarity (Table 4.2). The strongest overlap is observed between Cluster 5 (Walking around and grabbing the ball) and Cluster 13 (Passing and moving around) (0.7388). Both clusters involve ball possession combined with player movement, which explains their close proximity in the embedding space. Similarly, Cluster 6 (Ready pose for shot) and Cluster 7 (Saves and Dives) (0.6826) show high similarity, reflecting that many save sequences begin or end in a ready stance, leading to shared motion characteristics.

Clusters 3 (Pass or throw the ball) and 4 (Bending down) (0.6490) both involve forward upper-body motion toward the ball, while Clusters 10 and 13 (both passing and moving around) (0.6655) share overlapping movement patterns that combine ball distribution with repositioning. Finally, Clusters 15 (Shuffling around with ready pose) and 17 (Ready pose) (0.6445) naturally exhibit high similarity, as both are dominated by preparatory stance and small lateral adjustments.

Importantly, these overlaps occur between semantically related action types, suggesting that the embedding space preserves meaningful structural relationships rather than collapsing distinct behaviors. The substantial gap between intra-cluster (0.4845) and inter-cluster similarity (0.2077) therefore supports the presence of coherent yet hierarchically organized motion groups.

4.2.3 Validation of the Learned Representation Space

We further evaluated the learned representation space using validation data that was not seen during training. Specifically, we randomly selected 20 active validation samples and, for each sample, retrieved its five nearest neighbors in the embedding space from the training set. The cluster assignment was then determined via majority vote over the neighbors' cluster labels.

We subsequently performed a manual visual inspection of each validation sample and compared the assigned cluster label to the observed action type. Of the 20 samples,

12 were assigned to a valid cluster, while 8 were labeled as noise. Among the 12 clustered samples, 11 were judged to be correctly assigned based on visual inspection (see Table A.3).

These findings suggest that the learned embedding space generalizes beyond the training data and does not exhibit clear signs of overfitting, as validation samples are consistently mapped to semantically coherent regions of the representation space.

4.2.4 Fine-Grained Subcluster Analysis of Goalkeeper Saves

Cluster 7 primarily captures saves and diving actions. To investigate whether further clustering reveals more granular structure corresponding to distinct save types, we performed an additional clustering step. This analysis uncovered several semantically meaningful subclusters, as illustrated in Figure 4.2. Tables 4.3 and 4.4 summarize these subclusters along with their associated motion patterns and quantitative, joint height and body extension statistics.

Most sub-clusters align well with the motion semantics inferred from manual inspection. For example, sub-1 (Jumps and Punches) exhibits the highest maximum vertical neck height and large body angles relative to the horizontal plane, reflecting actions where the goalkeeper extends upward. Sub-2 (Drop Low Saves) shows a low minimum hip height, consistent with crouching, diving close to the ground, or dropping the hip low for a spread save. Sub-4 (Dive Forward to Grab Ball) is characterized by a low minimum neck height, reflecting forward-leaning or falling motions to intercept the ball. Sub-5 (Lateral Dives) displays the largest maximal body extension and relatively low body angles, indicating fully stretched horizontal dives.

Sub-0, in contrast, primarily contains noisy samples that were not semantically interesting, yet consistently consist of noisy low-lying actions, typically on the ground, as reflected by the low minimum hip and maximum neck values. Sub-3 did not exhibit a clear semantic separation, representing a mixed set of actions without a dominant motion pattern.

These quantitative metrics, including vertical neck and hip positions, body angle, and maximal extension, provide clear and interpretable characteristics for differentiating defensive action types within Cluster 7. They support the semantic labels obtained through visual inspection and help identify sub-clusters that remain noisy or semantically ambiguous.

Table 4.3: Height-related statistics for Cluster 7 (noise excluded). **N**: number of action clips in the sub-cluster; **Min/Max Neck**: minimum and maximum neck height (z-coordinate, in meters) within the sequence, averaged across clips; **Min Hip**: minimum mid-hip height (z-coordinate, in meters) across frames. All values are reported as mean ± standard deviation across clips in each sub-cluster. Minimum value per column are underlined, while maximum values per column are in **bold**.

Sub-Cluster	Description	N	Min Neck	Max Neck	Min Hip
sub-0	Noisy Samples	35	0.42 ± 0.43	$\underline{1.03} \pm 0.45$	0.18 ± 0.26
sub-1	Jumps and Punches	59	$\mathbf{0.48} \pm 0.34$	$\mathbf{1.70} \pm 0.22$	$\mathbf{0.21} \pm 0.18$
sub-2	Drop Low Saves	32	0.35 ± 0.26	1.49 ± 0.16	0.16 ± 0.13
sub-3	Mixed	39	0.32 ± 0.25	1.58 ± 0.15	0.15 ± 0.11
sub-4	Dive Forward to Grab Ball	39	$\underline{0.21} \pm 0.19$	1.42 ± 0.15	0.20 ± 0.15
sub-5	Lateral Dives	89	0.23 ± 0.15	1.51 ± 0.17	$\underline{0.14} \pm 0.06$

Table 4.4: Extension-related statistics for Cluster 7 (noise excluded). **N**: number of action clips in the sub-cluster; **Body Angle (deg)**: angle formed by the mid-hip–to–neck vector with respect to the ground plane at the frame of maximal extension; **Maximal Extension**: maximum Euclidean distance (in meters) between a wrist and an ankle joint at the frame of peak extension. All values are reported as mean ± standard deviation across clips in each sub-cluster. Minimum values per column are underlined, while maximum values per column are in **bold**.

Sub-Cluster	Description	N	Body Angle (deg)	Maximal Extension (m)
sub-0	Noisy Samples	35	37.69 ± 30.43	$\underline{1.53} \pm 0.27$
sub-1	Jumps and Punches	59	$\mathbf{60.27} \pm 21.49$	1.76 ± 0.16
sub-2	Drop Low Saves	32	46.29 ± 20.98	1.65 ± 0.14
sub-3	Mixed	39	39.78 ± 27.81	1.81 ± 0.16
sub-4	Dive Forward to Grab Ball	39	$\underline{33.77} \pm 25.22$	1.60 ± 0.23
sub-5	Lateral Dives	89	33.85 ± 22.10	$\mathbf{1.86} \pm 0.13$

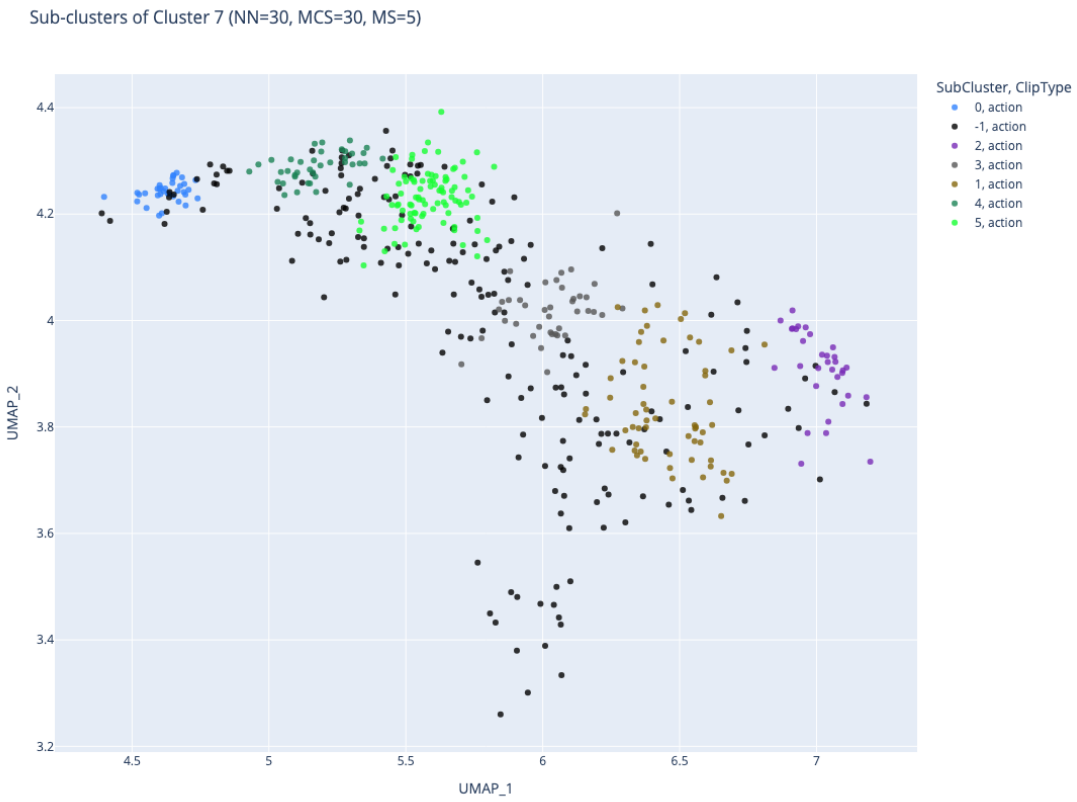


Figure 4.2: Visualization of the subclusters identified within Cluster 7. Each point represents an action clip in the learned embedding space, and the layout highlights how these subclusters separate based on motion characteristics. Black points denote noise detected by HDBSCAN.

Chapter 5

Conclusion

5.1 Key Takeaways

Our work demonstrates the potential of contrastive learning for capturing meaningful representations of goalkeeper skeletal sequences. Through visual inspection, we identified clusters that correspond to semantically distinct actions, although the quality of this semantic separation could be further improved. We validated these observations using quantitative cluster statistics, such as average joint positions, which supported the semantic distinctions. Overall, this approach highlights the promise of representation learning for skeletal sequences in football and can be extended to analyze other types of player motions beyond goalkeepers. Moreover, the proposed framework can potentially be integrated into goalkeeping coaches' existing workflows for both tactical and technical analysis, enabling data-driven insights and supporting individualized training interventions tailored to each goalkeeper's movement patterns and decision-making profiles.

5.2 Limitations and Future Work

While our approach demonstrates the potential of contrastive learning for goalkeeper action recognition, several limitations remain which outline the path for future research.

5.2.1 Feature Enrichment and Representation Alternatives

Future iterations of this work should aim to incorporate broader context into the feature set. Currently, our model relies solely on skeletal coordinates. Integrating external event data, such as ball possession and trajectory, shot outcome, or game-phase context, could provide critical information that is absent from skeletal data alone. Combining these symbolic features with learned embeddings may improve the semantic separation of clusters.

Additionally, we observed that specific joint movements introduced noise into the clustering process. For instance, a ball throw movement with a trailing leg was occasionally misclassified as a goal kick due to the similarity in lower-body movement. To mitigate this, future work should explore separate processing streams for upper and lower body joints, allowing the model to weigh relevant body parts differently based on the action type.

Furthermore, the influence of tensor representation alignment on downstream performance requires deeper investigation. Techniques such as root centering and shoulder alignment are standard; however, they may obscure critical directional and positional cues specific to goalkeeping actions (e.g., the position of an action relative to the goal line). Future work should perform an ablation study to determine whether these rigid alignment transformations aid recognition or if preserving the raw global orientation yields superior results.

Finally, while we employed dimensionality reduction to manage the high-dimensional skeletal data, future work should explore alternative reduction techniques to determine whether they yield improved clustering performance or more stable representations. Future research should also further test whether the cluster structures observed after UMAP projection reflect genuine patterns present in the original data or are partially induced by the embedding process itself. Developing diagnostic methods to detect potential cluster hallucination effects, as well as validating clustering consistency across multiple embedding techniques and the original feature space, would help ensure the reliability of the discovered structure.

5.2.2 Preprocessing and Augmentation Guidelines

The preprocessing pipeline significantly influences model performance. A possible next step is to measure the impact of the temporal dimension T on the input tensors, determining the optimal trade-off between temporal resolution and computational efficiency.

A critical limitation lies in the temporal and spatial segmentation of actions. While our adaptive segmentation improves upon fixed sliding windows, it occasionally truncates actions when motion energy drops prematurely. Similarly, the strict spatial filtering, which isolates sequences based solely on the ball's presence in the defending third, can result in abrupt cuts if the ball exits the zone mid-action. Exploring alternative strategies for reliably identifying relevant action segments within long skeletal trajectory sequences would represent an important direction for future research. Additionally, the energy-based segmentation could be refined by increasing the context window around extracted clips, allowing different sequence lengths for different movement types, or tuning the hyperparameters, ensuring that complex, multi-stage movements are captured in their entirety without being artificially severed.

We also propose expanding the augmentation strategy. While some augmentations are deployed, we recommend experimenting with more augmentations, specifically those

inducing invariance to action execution speed. Quantifying the impact of individual augmentation functions will help tailor a strategy specifically for the high-variance nature of goalkeeping movements.

5.2.3 Model Training and Extensions

Our experiments revealed a performance gap between fine-tuning and full training strategies. Future research should investigate the root cause of this discrepancy: whether it stems from aggressive preprocessing, domain shift between datasets, or the limitations of transfer learning when applied to skeletal action recognition. Investigating less aggressive data adaptation techniques might facilitate better transfer capabilities.

With sufficiently reliable labeled data, the framework could be extended by adding a classification head on top of the learned embeddings to automatically label goalkeeper actions from skeletal data sequences. However, as discussed earlier in Chapter 4, the current labels derived from clustering are not yet fully reliable. Training a supervised model on sub-optimal labels would likely produce poor results and therefore, improving the quality and consistency of the unsupervised clustering constitutes a necessary prerequisite for this extension.

Furthermore, hyperparameter tuning of the CrosSCLR model was constrained by available computational resources. A more comprehensive search over architectural and optimization parameters may yield additional performance improvements.

5.2.4 Evaluation and Cluster Quality

The most significant challenge remains the validation of cluster quality. Our current reliance on visual inspection is time-consuming and subjective. Moreover, we observed that HDBSCAN's samples labeled as noise were not actually noisy skeletal data, but rather outliers in terms of density. This implies that HDBSCAN can label interesting or meaningful motion sequences as noise simply because they reside in less dense regions of the embedding space after UMAP. Future work must establish objective, quantitative metrics for cluster quality, such as intra-cluster motion energy variance. Developing an automated optimization objective for clustering would remove the brittleness of manual hyperparameter tuning and provide a more rigorous standard for evaluating unsupervised motion learning.

Appendix A

Appendix

1. Mapping of Joints between NTU RGB+D and Hawkeye

Table A.1: Mapping between NTU RGB+D joints and Hawkeye joints.

NTU RGB+D index	NTU RGB+D joint name	Construction (using Hawkeye joints)	Logic behind construction
1	Base of Spine	Mid Hip	Direct match.
2	Middle of Spine	Midpoint(Mid Hip, Neck)	Midpoint between Mid Hip and Neck.
3	Neck	Neck	Direct match.
4	Head	Nose	Nose used as proxy for head center.
5	Left Shoulder	Left Shoulder	Direct match.
6	Left Elbow	Left Elbow	Direct match.
7	Left Wrist	Left Wrist	Direct match.
8	Left Hand	Circumcenter(Wrist, Thumb, Pinky)	Geometric center of the triangle formed by Wrist, Thumb, and Pinky joints.
9	Right Shoulder	Right Shoulder	Direct match.
10	Right Elbow	Right Elbow	Direct match.
11	Right Wrist	Right Wrist	Direct match.
12	Right Hand	Circumcenter(Wrist, Thumb, Pinky)	Geometric center of the triangle formed by Wrist, Thumb, and Pinky joints.
13	Left Hip	Left Hip	Direct match.
14	Left Knee	Left Knee	Direct match.
15	Left Ankle	Left Ankle	Direct match.
16	Left Foot	Left Big Toe	Big toe used as proxy for front of foot.
17	Right Hip	Right Hip	Direct match.
18	Right Knee	Right Knee	Direct match.
19	Right Ankle	Right Ankle	Direct match.
20	Right Foot	Right Big Toe	Big toe used as proxy for front of foot.
21	Spine Shoulder	Midpoint(L.Shoulder, R.Shoulder)	Midpoint between Left Shoulder and Right Shoulder.
22	Tip of Left Hand	Extrapolated Vector	Extrapolation from Wrist through the midpoint of fingers: Tip = Mid(Thumb, Pinky) + 0.9× (Mid(Thumb, Pinky) - Wrist).
23	Left Thumb	Left Thumb	Direct match.
24	Tip of Right Hand	Extrapolated Vector	Extrapolation from Wrist through the midpoint of fingers: Tip = Mid(Thumb, Pinky) + 0.9× (Mid(Thumb, Pinky) - Wrist).

NTU RGB+D index	NTU RGB+D joint name	Construction (using Hawkeye joints)	Logic behind construction
25	Right Thumb	Right Thumb	Direct match.

2. Hyperparameters for UMAP and HDBSCAN

Table A.2: Hyperparameters for UMAP dimensionality reduction and HDBSCAN clustering.

Component	Parameter	Value
UMAP	`n_neighbors`	80
	`min_dist`	0.0
	`n_components`	20
	`metric`	cosine
HDBSCAN	`min_cluster_size`	60
	`min_samples`	20
	`metric`	euclidean
	`cluster_selection_method`	leaf
	`cluster_selection_epsilon`	0.22

3. Validation Experiments

Table A.3: Qualitative nearest-neighbour evaluation on validation samples.

Validation Sample ID	Majority Vote Cluster	Visual Inspection Match
6497	-1	N/A
6634	1	Yes
4655	-1	N/A
6847	-1	N/A
5282	-1	N/A
5430	-1	N/A
5599	8	No
4556	11	Yes
3216	17	Yes
641	1	Yes
4229	-1	N/A
349	5	Yes
736	10	Yes
4200	14	Yes
1980	7	Yes
6734	1	Yes
2934	16	Yes
4736	1	Yes
2701	-1	N/A
5437	-1	N/A

4. Definition of Cluster Statistics

Table A.4: Cluster description and per-cluster biomechanical statistics for action clips. Maxima are in **bold**, minima are underlined. **Variable Definitions: Displacement**: Euclidean distance (meters) between the mid-hip position at the first and last frame of the clip. **Wrist Min**: minimum height (z-coordinate) reached by either wrist over the clip. **Foot Max**: maximum height (z-coordinate) reached by either ankle over the clip. **Hip Vel**: maximum mid-hip speed (m/s) over the clip. **Ankle Vel**: maximum speed (m/s) of the faster ankle over the clip. **Wrist Vel**: maximum speed (m/s) of the faster wrist over the clip. **W/H Ratio**: ratio of wrist velocity to hip velocity at the frame of peak wrist velocity. **A/H Ratio**: ratio of ankle velocity to hip velocity at the frame of peak ankle velocity. All values are reported as mean ± standard deviation across clips in each sub-cluster. Minimum value per column are underlined, while maximum values per column are in **bold**.

Cl.	Description	N	Disp (m)	Wrist Min (m)	Foot Max (m)	Hip Vel (m/s)	Ank Vel (m/s)	Wrist Vel (m/s)	W/H Ratio	A/H Ratio
-1	Noise	5274	4.52 ± 2.98	0.736 ± 0.214	0.343 ± 0.201	5.775 ± 3.262	11.499 ± 5.771	8.997 ± 5.280	2.546 ± 3.111	2.803 ± 2.257
0	Walking Around	78	3.139 ± 0.812	0.871 ± 0.135	0.243 ± 0.058	3.235 ± 0.501	7.166 ± 1.046	5.064 ± 1.158	2.619 ± 1.666	3.050 ± 0.954
1	Walking Around	1179	3.212 ± 1.117	**0.900 ± 0.123**	0.272 ± 0.101	2.512 ± 0.810	5.629 ± 1.614	4.323 ± 1.706	2.476 ± 1.518	3.193 ± 0.980
2	Kick the ball	761	3.652 ± 1.785	0.732 ± 0.114	0.644 ± 0.179	7.189 ± 2.150	17.897 ± 4.353	12.146 ± 3.735	3.540 ± 1.596	5.262 ± 2.846
3	Pass / throw	98	1.596 ± 0.929	0.762 ± 0.190	0.556 ± 0.261	3.964 ± 1.764	12.149 ± 3.891	8.195 ± 4.390	4.223 ± 2.749	**6.640 ± 4.988**
4	Bending down	199	1.529 ± 0.943	0.562 ± 0.303	0.379 ± 0.274	3.319 ± 2.420	8.148 ± 5.188	6.564 ± 3.616	**4.768 ± 4.305**	5.512 ± 6.417
5	Walk + grab	196	8.947 ± 1.990	0.714 ± 0.249	0.318 ± 0.138	8.161 ± 2.377	16.094 ± 4.218	13.473 ± 4.512	2.656 ± 4.692	2.672 ± 1.295
6	Ready for shot	118	3.675 ± 1.459	0.584 ± 0.153	0.331 ± 0.101	7.470 ± 2.112	14.751 ± 3.432	11.762 ± 3.506	2.282 ± 1.288	2.186 ± 0.397
7	Saves / Dives	490	3.653 ± 2.593	0.058 ± 0.267	**0.723 ± 0.315**	9.386 ± 4.745	16.492 ± 7.216	16.755 ± 6.391	3.357 ± 3.995	3.709 ± 5.787
8	Receive pass	197	4.548 ± 1.863	0.797 ± 0.127	0.391 ± 0.128	5.370 ± 1.331	11.183 ± 2.683	7.669 ± 2.023	1.912 ± 1.170	2.953 ± 2.342
9	Hand comm.	489	5.178 ± 1.456	0.785 ± 0.225	0.285 ± 0.111	4.872 ± 1.317	10.385 ± 2.469	8.334 ± 2.798	3.366 ± 5.818	2.975 ± 1.339
10	Pass + move	368	7.777 ± 1.801	0.804 ± 0.157	0.486 ± 0.122	7.503 ± 1.938	13.807 ± 3.813	10.267 ± 2.918	1.673 ± 0.786	2.404 ± 1.753
11	Ready + shuffle	259	1.821 ± 0.991	0.686 ± 0.127	0.236 ± 0.068	3.966 ± 1.087	8.510 ± 2.137	5.836 ± 1.894	1.990 ± 1.025	2.538 ± 0.504
12	Sprint + grab	541	**14.879 ± 4.798**	0.613 ± 0.256	0.591 ± 0.162	**14.757 ± 3.816**	**24.235 ± 6.335**	**20.561 ± 5.988**	1.765 ± 0.953	1.944 ± 1.967
13	Pass + move	112	10.457 ± 1.792	0.777 ± 0.202	0.472 ± 0.111	8.706 ± 1.578	15.608 ± 3.593	11.882 ± 2.592	1.818 ± 2.090	2.783 ± 3.357
14	Moving back	855	7.772 ± 2.575	0.788 ± 0.157	0.270 ± 0.106	7.324 ± 2.398	12.924 ± 3.991	9.894 ± 3.684	1.766 ± 2.616	2.027 ± 0.880
15	Shuffle + ready	109	4.054 ± 0.793	0.772 ± 0.116	0.182 ± 0.065	4.291 ± 0.804	7.928 ± 1.375	5.663 ± 1.391	1.850 ± 1.350	2.235 ± 0.388
16	Hand signals / stretch	730	3.485 ± 1.414	0.885 ± 0.119	0.237 ± 0.091	3.108 ± 1.196	6.916 ± 2.464	5.337 ± 2.310	2.835 ± 2.469	3.338 ± 2.888
17	Ready pose	743	2.051 ± 1.057	0.681 ± 0.199	0.201 ± 0.137	3.573 ± 1.367	7.346 ± 2.612	5.293 ± 2.814	2.086 ± 2.117	2.514 ± 0.642

Individual Contributions

Afonso:

- Execution of model training, hyperparameter tuning, and comparative experiments (training from scratch vs. fine-tuning).
- Implementation of canonical local reference frame conversion, bone-length normalisation and data augmentation.
- Implementation of robust adaptive thresholding and automated action/idle clip extraction.
- Development of native and NTU-aligned tensor representations, including 29-joint to 25-joint mapping and conversion to pretraining-compatible formats.
- Development of training-validation data split logic and dimensionality reduction robustness checks.
- Design of the Hawk-Eye skeletal model diagram.

Kaushik:

- Implementation of tracking data parsing, skeletal and ball tracking data synchronization, spatial filtering, and sequence aggregation logic.
- Hyperparameter tuning for the clustering pipeline and in-depth statistical analysis of results.
- Qualitative nearest-neighbor evaluation and detailed subcluster analysis.
- Coordinate system mapping to Kinect imagery
- Analysis of kinematic features of clusters
- Generation of interactive 2D/3D embedding visualizations, skeletal movement animations, pitch coordinate system diagrams, and motion energy profile plots.

Equal Contribution:

- Litereature review
- Technical report writing.
- Visual inspection and interpretation of clustering results.

Bibliography

[1] G. Jordet, K. M. Aksum, D. N. Pedersen, A. Walvekar, A. Trivedi, A. McCall, A. Ivarsson, and D. Priestley, "Scanning, contextual factors, and association with performance in english premier league footballers: An investigation across a season," 2020. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.553813

[2] S. Plakias, T. Tsatalas, M. A. Mina, C. Kokkotis, E. Kellis, and G. Giakas, "A bibliometric analysis of soccer biomechanics," 2024. [Online]. Available: https://www.mdpi.com/2076-3417/14/15/6430

[3] M. Wear, R. Beal, T. Matthews, T. Norman, and S. Ramchurn, "Learning from the pros: Extracting professional goalkeeper technique from broadcast footage," 2022. [Online]. Available: https://arxiv.org/abs/2202.12259

[4] TRACAB, "TRACAB — Unlocking the DNA of Sports," Redwood City, CA, USA, 2026, accessed: 2026-02-25. [Online]. Available: https://www.ea.com/tracab

[5] M. U. Gotthardt, "Clustering large-scale 3d football player skeleton data: Investigating differences in player pose distributions and their correspondence to tracking performance levels," Stockholm, Sweden, 2024, master's thesis. [Online]. Available: https://kth.diva-portal.org/smash/get/diva2:1935813/FULLTEXT01.pdf

[6] C. Yeung, K. Ide, and K. Fujii, "Autosoccerpose: Automated 3d posture analysis of soccer shot movements," 2024. [Online]. Available: https://arxiv.org/abs/2405.12070

[7] G. C. Bian, "Soccer last touch and automatic event detection with skeletal tracking data," 2023, master's thesis. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/156773/bian-gbian-meng-eecs-2024-thesis.pdf?sequence=1&isAllowed=y

[8] M. Schepers, P. Robberechts, J. V. Haaren, and J. Davis, "What makes a dribble successful? insights from 3d pose tracking data," 2025. [Online]. Available: https://arxiv.org/abs/2506.22503

[9] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018. [Online]. Available: https://arxiv.org/abs/1801.07455

[10] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," pp. 3590–3598, 2019. [Online]. Available: https://arxiv.org/pdf/1904.12659

[11] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," 2019. [Online]. Available: https://arxiv.org/abs/1805.07694

[12] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," 2020. [Online]. Available: https://arxiv.org/abs/2003.14111

[13] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," p. 103219, Jul. 2021. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2021.103219

[14] L. Li, M. Wang, B. Ni, H. Wang, J. Yang, and W. Zhang, "3d human action representation learning via cross-view consistency pursuit," 2021. [Online]. Available: https://arxiv.org/abs/2104.14466

[15] M. Michaelides, "Skeletrack: A new era of data in tennis," Apr. 2025, accessed: 2026-02-22. [Online]. Available: https://www.hawkeyeinnovations.com/news/4243365/skeletrack-a-new-era-of-data-in-tennis

[16] UEFA, "UEFA EURO 2024 Physical Analysis Report," Switzerland, 2025, accessed: 2026-02-23. [Online]. Available: https://editorial.uefa.com/resources/0297-1d4e3592fbf1-f11d4e1c826a-1000/uefa_euro_2024_physical_analysis_report_20250318094958.pdf

[17] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," 2016. [Online]. Available: https://arxiv.org/abs/1604.02808

[18] M. Grootendorst, "Bertopic: Neural topic modeling with a class-based tf-idf procedure," 2022. [Online]. Available: https://arxiv.org/abs/2203.05794

[19] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2020. [Online]. Available: https://arxiv.org/abs/1802.03426

[20] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," pp. 559–572, 1901. [Online]. Available: https://doi.org/10.1080/14786440109462720

[21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," Berkeley, CA, USA, pp. 281–297, 1967. [Online]. Available: https://projecteuclid.org/euclid.bsmsp/1200512992

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," 2002. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf

[23] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," New York, NY, USA, Jul. 2015. [Online]. Available: https://doi.org/10.1145/2733381

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," AAAI Press, pp. 226–231, 1996. [Online]. Available: https://dl.acm.org/doi/10.5555/3001460.3001507

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.