Real-Time Feasibility and Related Temporal Design Choices of Human Motion Prediction Models

by

Andrew C. G. Hutani

In partial fulfilment of the requirements for the degree of
Master of Science
at Delft University of Technology,
to be defended publicly on 19/12/2025.

Faculty: Mechanical Engineering
Department: Cognitive Robotics
Programme: Robotics

Mentors / Supervisors: Prof.dr.ir. J.C.F. De Winter
Ronald de Leeuw van Weenen (Sioux Technologies)
Graduation committee: Prof.dr.ir. J.C.F. De Winter
Dr. D. Dodou
Ronald de Leeuw van Weenen (Sioux Technologies)

An electronic version of this thesis is available at http://repository.tudelft.nl

TU Delft

Keywords:
Human Motion Prediction
Real-time
Temporal Design

Abstract

This thesis investigates how temporal design choices affect the real-time feasibility of human motion prediction models. Two state-of-the-art models were evaluated: GCNext, a data-driven graph convolutional model, and PhysMoP, a hybrid model combining a physics-based and data-driven branch. Controlled experiments showed the influence of input history length, temporal resolution, and the model architecture on prediction accuracy and latency. Results showed that longer observation windows do not necessarily improve accuracy, while increasing the latency. Both models were sensitive to changes in temporal resolution, as they implicitly assumed a fixed sampling rate. Real-time performance analysis indicated that single-pass architectures were favoured, while autoregressive models suffered from compounding delay. Retraining GCNext with shorter input histories and optimising autoregressive passes achieved substantial latency reduction with minimal accuracy loss. These results show that temporal configurations are critical design choices for achieving real-time feasibility of human motion prediction models. The code for this paper is available at https://github.com/AndrewHutani/HMP.

Real-Time Feasibility and Related Temporal Design Choices of Human Motion Prediction Models

Andrew C. G. Hutani - 4913116

University Supervisor: Prof.dr.ir. J.C.F. De Winter
Company Supervisor: Ronald de Leeuw van Weenen
Thesis Duration: April, 2025 – December, 2025
Faculty: Faculty of Mechanical Engineering, Delft

***Abstract*—This thesis investigates how temporal design choices affect the real-time feasibility of human motion prediction models. Two state-of-the-art models were evaluated: GCNext, a data-driven graph convolutional model, and PhysMoP, a hybrid model combining a physics-based and data-driven branch. Controlled experiments showed the influence of input history length, temporal resolution, and the model architecture on prediction accuracy and latency. Results showed that longer observation windows do not necessarily improve accuracy, while increasing the latency. Both models were sensitive to changes in temporal resolution, as they implicitly assumed a fixed sampling rate. Real-time performance analysis indicated that single-pass architectures were favoured, while autoregressive models suffered from compounding delay. Retraining GCNext with shorter input histories and optimising autoregressive passes achieved substantial latency reduction with minimal accuracy loss. These results show that temporal configurations are critical design choices for achieving real-time feasibility of human motion prediction models. The code for this paper is available at https://github.com/AndrewHutani/HMP.**

I. Introduction

As robotic manipulators become increasingly common in a wide range of applications, such as industrial assembly lines and assistive technology in households, so does the frequency and complexity of human-robot interactions [1]. This increase comes paired with a more demanding need for accurate predictions of human movements in these exact same workspaces, where spatial deviations of a few centimetres can already impact the safety and efficiency of humans and robots [2].

Current state-of-the-art human motion prediction models capture the relationships between past and possible future movements, making accurate predictions of 3D joint positions up to one second ahead. While some models demonstrate longer horizons [3], their accuracy typically degrades rapidly beyond 1 second into the future. Other models [4] report prediction metrics for several seconds into the future, but evaluate full-body trajectories, rather than joint positions.

These prediction models can roughly be divided into three main categories: physics-based models, data-driven models, and hybrid/fusion models combining both into one. Physics-based models aim to predict human motion using rules grounded in physics [5]; this is mainly done by modelling the human motion using equations like Euler-Lagrange. Data-driven models, on the other hand, immediately predict the future motion from previously observed frames of motion [6]. This process is iteratively learned on large datasets, which allows the model to learn temporal and spatial dependencies between joints and motion patterns.

Human motion prediction models have become more advanced and accurate. Most models are currently trained and evaluated on large motion-capture datasets such as Human3.6M and AMASS, which contain recordings of different types of human motion [7]–[9]. These evaluations are mainly focused on the accuracy of the models, and largely ignore possible constraints of real-time usage, as many existing models are benchmarked offline on their respective datasets, rather than deploying them in real-time environments [6], [10]–[12]. This creates a critical gap: a model may be able to perform well in terms of accuracy, yet not be suitable for real-time human motion prediction application, if it cannot make predictions in a timely manner, consistently, while balancing the trade-off between longer historical context and increased prediction delay.

In this thesis, the temporal aspects of human motion prediction models have been assessed. More specifically, the temporal context was considered in: i) the observation window, how far back in time the model can look; and ii) the temporal resolution, the spacing of observations within that observation window.

This thesis addressed three main research questions:

- How does the length of the past motion used by the model influence the prediction accuracy of the models? This captures the trade-off between accuracy gains and possible increases in processing time by using longer input sequences.
- How does the temporal resolution of the observations affect the models' prediction performance? This allows us to test the robustness of the models to changes in observation frequencies.
- How do the models perform on real-time performance metrics beyond accuracy, such as prediction time, latency and jitter? These end-to-end metrics decide whether a model meets the strict temporal constraints of real-time deployment, where predictions must be made faster than the frequency at which new data become available.

To answer these questions, this thesis evaluated two state-of-the-art models: one data-driven (GCNext), and one hybrid model (PhysMoP). These models have been evaluated on varying input lengths and sampling rates, while assessing their accuracy, robustness and real-time feasibility. Note that this work does not aim to reproduce the exact performance numbers from the original models. Instead, its main focus lies on the trends that arise when temporal parameters are varied in a controlled manner. By focussing on trends instead of absolute performance, this thesis aims to isolate how temporal factors influence a model's behaviour and real-time feasibility.

II. Related Work

A. Types of models

Human motion prediction models can broadly be divided into data-driven, physics-based and hybrid/fusion approaches.

Data-driven models rely on motion datasets to learn spatial and temporal dependencies immediately from recorded motion. Earlier approaches used Recurrent Neural Networks (RNNs) or encoder-decoder structures [4], [6], [11] to model this behaviour, while more recent sources have shifted towards graph-based methods [13]. Graph Convolutional Networks (GCNs) aim to model the human skeleton as a graph with nodes and edges representing joints and bones, respectively. This graph representation allows explicit encoding of spatial dependencies [14]–[16]. Among the GCN models, GCNext [17] is a state-of-the-art example that extends this idea using dynamic graph combinations. This allows the model to generalize better to complex motion sequences by capturing a wider range of spatio-temporal dependencies. Despite the recent shift towards the usage of GCNs, alternative approaches have also shown strong performances in recent studies (see [13] for a review). Transformer-based approaches model spatial and temporal relationships between joints across a motion sequence using an attention-based mechanism. This mechanism assigns the learnt weights to each previous frame, corresponding to how relevant it is for the prediction of future motion [10]. Additionally, Multilayer Perceptron (MLP) models approach motion prediction without recurrence or attention. Instead, these models treat the sequence of joint positions as a fixed-length vector and learn patterns in how the joint positions change over time through a series of transformations, which are the layers in the network [18].

Physics-based approaches incorporate biomechanical or dynamical constraints. Common model architectures include Unscented Kalman Filters (UKF) with a constant-velocity or constant-acceleration model [19], [20], or using a form of the Euler-Lagrange equations to predict joint accelerations [5], [12]. Overall, these types of models perform well for short-term prediction horizon, but tend to suffer from error propagation for longer-term predictions, as inaccuracies in the velocity or acceleration estimation can accumulate through recursive prediction [12].

Hybrid/fusion models combine data-driven with physics-based models, to take advantage of the strengths of both model types. A common strategy is to pair a neural network with explicit dynamic models, like fusing recurrent networks with Lagrangian mechanics for arm-motion prediction [20]. Another example is the PhysMoP model [12], which incorporates a physics-based motion prediction alongside a data-driven MLP. These two branches are fused using a dynamically weighted average to create predictions. While these hybrid methods generally outperform purely data-driven or physics-based models, these model architectures typically are more complex, making them computationally heavier, and thus less suitable for real-time applications.

B. Performance metrics

The two most commonly used performance metrics used for human motion prediction models are the Mean Angle Error (MAE) and the Mean Per Joint Position Error (MPJPE) [13]. MAE returns the average error between the predicted and expected joint angles of all predicted joints over time. This metric has been used in early research, and can be ambiguous; distinct poses of the human body can be represented by the same joint angle configuration, and vice versa [1]. As a result, this can affect the reported MAE, and lead to a misleading reported performance [13].

As a result, over time MPJPE has become the more commonly used metric in literature. It calculates the average error in Euclidean distance between the predicted and expected joint positions in Cartesian coordinates. By comparing actual spatial distances between the joints, MPJPE eliminates the ambiguity present in the MAE, and therefore is a more reliable assessment of prediction accuracy [13]. It is defined as:

$$\text{MPJPE} = \frac{1}{T}\sum_{i=1}^{T}\frac{1}{J}\sum_{j=1}^{J}||x_{i,j}^{GT} - x_{i,j}^{\text{pred}}||_2 \tag{1}$$

In the equation for the MPJPE, T is the number of frames, J is the number of joints, $x_{i,j}^{GT}$ is the 3D Cartesian position of the ground truth j-th joint at time step i, and $x_{i,j}^{\text{pred}}$ is the 3D Cartesian position of the predicted j-th joint at time step i. The subscript $_2$ denotes the L_2-norm, which is the Euclidean distance between the predicted and ground truth joints.

C. Datasets

Datasets play an important role in the training and evaluation of the human motion prediction models. While many datasets exist for human motion purposes, this work focuses on two datasets that are commonly used in modern research, each with a different scope and scale.

The Human3.6M dataset [7], [8] contains motion capture data from 11 different subjects performing 15 different actions. These actions range from simple walking, to more complex actions like talking on the phone. In total, this dataset contains 3.6 million human poses. Each pose is stored in both 3D joint positions in Cartesian coordinates and joint rotations in Euler angles, recorded at 50 Hz. Since all data points are collected in a laboratory experiment, the dataset offers high precision data,

[1] e.g. the Euler angles $(0, \pi, 0)$ and $(\pi, 0, \pi)$ result in the same configuration

but is quite limited in diversity and realism. A comprehensive overview of the data distribution within the evaluation set, for the joints used by the models, is provided in Appendix A.

The AMASS dataset [9] is a database of human motion, combining multiple different motion capture sources into a single unified format. This results in a dataset containing over 300 subjects and more than 11,000 motions. The data is converted to the SMPL (Skinned Multi-Person Linear) body model [21], where poses are represented as joint rotations in axis-angle format. This parameterisation ensures consistency across all the data from the different datasets, which were recorded at varying frame rates. As a result, all motion is represented in a single format at a resampled frequency of 60 Hz. A comprehensive overview of the data distribution within the evaluation set, for the joints used by the models, is provided in Appendix B. Note that for this visualisation the data has been converted to 3D joint positions for visual clarity.

III. Methodology

A. Models

Two state-of-the-art models were considered for this work: the GCNext model [17] and the PhysMoP model [12]. These two models were chosen because they cover the three different types of models introduced in subsection II-A. More specifically, the GCNext model covers a fully data-driven approach using a GCN, and the PhysMoP model implements a physics-based model alongside a data-driven MLP, which are fused together. Additionally, the original papers for both models report some of the strongest results on their respective datasets, achieving lower MPJPE values amongst comparable approaches. This makes them suitable candidates to see whether their performance carries over when real-time applicability is taken into consideration. A comprehensive overview of both models and their original datasets is included in Appendix C, Table C-I.

GCNext [17] builds on the recent popularity of GCNs in human motion prediction. Traditional GCNs include a fixed choice of graph convolution as a design choice, which may fail to capture certain types of motion dependencies. GCNext addresses this by introducing a unified graph convolution framework (UniGC) that can dynamically choose the most appropriate type(s) of graph convolution. This selection is done by a learnable module, which learns to choose between different types based on the input features at each layer. This adaptability allows the model to capture complex relationships between the data. The model is trained and evaluated primarily on the Human3.6M dataset, using 50 historical frames, to make predictions up to 1 second into the future.[2] This is all done at a subsampled frequency of 25 Hz. In total, the evaluation set for this model consists of 3840 samples of 75 (50 historical, 25 target) frames of consecutive motion.

[2]The GCNext model is trained for 85,000 iterations, with a starting learning rate of 0.0006, which drops to 0.000005 after 75,000 iterations

The PhysMoP model [12] implements a hybrid approach that combines both a physics and data branch into one. The physics branch implements the Euler-Lagrange equations of motion, which in this case, link the predicted joint accelerations to inertia, positions, velocities and forces. Thus, rather than directly predicting the future movement of each joint, the model learns to estimate the parameters of the Euler-Lagrange equations of motion (Equation 2a). Specifically the model estimates the mass matrix (M); the generalized bias force vector ($\hat{C}$), which includes Coriolis, centrifugal, and gravitational forces; and the joint actuation forces (τ), in Equation 2b. Equation 2b is derived from Equation 2a by grouping all nonlinear and gravitational effects into a single term $\hat{C}(q, \dot{q})$. This learned term approximates $C(q, \dot{q})\dot{q} + g(q)$ from the full equation, simplifying the equation while keeping the underlying principle the same. Here, q is the joint angles, $\dot{q}$ the joint velocities, and $\ddot{q}$ the joint accelerations. The model then uses the calculated joint acceleration $\ddot{q}$ to solve for future joint angles using Verlet integration (Equation 3). This process is done autorecursively until the final prediction horizon is reached, where the newly predicted joint angles are used to predict further into the future.

$$\underbrace{M(q)}_{\text{Mass matrix}}\ddot{q} + \underbrace{C(q,\dot{q})\dot{q} + g(q)}_{\text{Generalized bias force vector}} = \underbrace{\tau}_{\text{Joint actuation forces}} \tag{2a}$$

$$M(q)\ddot{q} + \hat{C}(q,\dot{q}) = \tau \tag{2b}$$

$$q(t+\Delta t) = 2q(t) - q(t-\Delta t) + \ddot{q}(t)\Delta t^2 \tag{3}$$

Alongside that, this model implements an MLP as a data-driven branch, which learns temporal dependencies directly from the past observed motion. These two branches are dynamically fused using a weighted average, which scales depending on the prediction horizon; for shorter prediction horizons, the physics-branch dominates the predictions, while the data-driven approach is weighted more for longer-term predictions (Equation 4). Here w is the learned weights vector, that takes a weighted average of the two predictions, based on their prediction horizon.

$$q_{\text{fusion}} = w q_{\text{data}} + (1-w) q_{\text{physics}} \tag{4}$$

The authors claim that the Human3.6M dataset is sampled at 25 Hz, whereas the implementation for the AMASS dataset uses 30 Hz. However looking at their implementation, their training and evaluating appeared to be using a conflicting Δt . This leads to uncertainty about the actual sampling frequency used in practice. To verify whether these discrepancies affect the model's trends, all branches were retrained on the AMASS dataset using the correct corresponding $\Delta t = \frac{1}{30}$ and frequency. This retrained model was then compared to the original model, shown in Appendix D, Figure D-1. The plots showed that there was a difference in the exact MPJPE, but also indicated that the overall trend with respect to the number of observed frames remained similar. This confirmed

that the inconsistent temporal settings in the original model played a negligible role on how the trends changed for this model. Therefore, for all subsequent experiments, the original PhysMoP model was used, evaluated at a frequency of 25 Hz for consistency. Ultimately, the model was evaluated on the AMASS evaluation set, which consists of 15467 samples of 50 (25 historical, 25 target) frames of consecutive motion, subsampled to 25 Hz. [3]

In the following experiments, only the physics and data-driven branch of the PhysMoP model were evaluated. The rationale is that the fusion branch is essentially a weighted average of the two, and therefore will not introduce fundamentally new behaviour. Since the focus was on how different models and input history affect prediction accuracy and real-time feasibility, evaluating the two branches separately isolated the effects on their performance. Moreover, since the fusion branch directly combines the outputs from the physics and data branches, any trends observed in either branch are reflected in the fusion branch.

By analysing the branches individually, we were able to compare different data-driven implementations (GCNext versus the PhysMoP data branch) and the contrast between data-driven and physics-based approaches (the two branches in the PhysMoP model).

A final difference between the models lies in the length of the historical context they require, which is ultimately a design choice of the original authors. GCNext requires 50 past frames, whereas the PhysMoP model uses at most 25 frames. Within the PhysMoP model, the physics branch only relies on the last 3 frames, as this is sufficient to make a prediction based on velocity and acceleration of the past observations. Note that both models predict up to 1 second into the future, showing that similar horizons can be reached with different input histories. While their reported accuracies are different per dataset, both models achieve comparable performance in terms of MPJPE, indicating that this difference in input length does not inherently limit the achievable prediction accuracy.

B. Evaluation Framework

The evaluation framework implemented the aforementioned models in such a way that allowed historical frames to be added to the model step by step. To this extent, each model was evaluated on their respective evaluation set, which consists of fixed-length motion sequences (75 frames for GCNext, 50 frames for PhysMoP). These evaluation sequences were constructed back-to-back from the longer motion capture recordings: when a 75-frame (or 50-frame for PhysMoP) sequence was extracted from the full motion, the next sequence started on the following frame, creating consecutive, non-overlapping samples across the entire motion. This sequence was divided into two parts: the input frames, which were fed to the model; and the prediction target or ground truth, representing the motion the model aims to predict. A visual example of this is shown in Figure 1, where the red skeletons are the ground truth and the green skeletons are the input motion used by the models to create the predictions.

[3]The PhysMoP model is trained in stages: first the data branch is trained for 5 epochs; after that the physics branch for 2 epochs; and lastly, the fusion branch for 2. All stages use a learning rate of 0.0003.

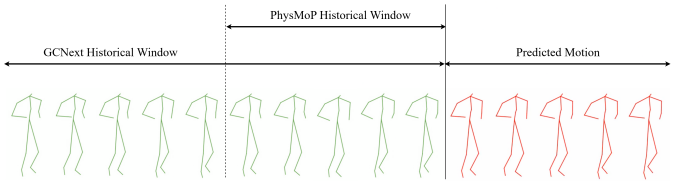


Fig. 1: Full motion, with distinction between observed (green) and predicted (red) motion for each model. The skeletons are visualised in 5-frame intervals.

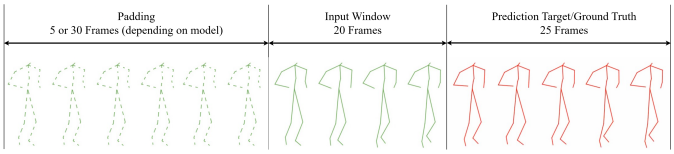


Fig. 2: A visual example of how the padding mechanism works to ensure a consistent number of input frames. The dotted skeletons are the first frame repeated. The skeletons are visualised in 5-frame intervals.

For real-time applications, the number of input frames is not consistent for the first few time steps of a new recording, thus requiring a mechanism to ensure a constant number of frames is fed to the model. For this, a padding mechanism was added, which repeated the first frame of observed motion, as shown in Figure 2. This type of padding avoids adding artificial temporal inconsistencies in the motion, as verified empirically during implementation. Alternative padding strategies such as: i) repeating the most recent/last frame, ii) repeating frames in the middle of the sequence, or iii) stretching the sequence by repeating every frame X times, were considered, but ultimately rejected. Repeating the most recent frame would introduce an artificial still skeleton at the end of the sequence, right before the moment of prediction. Inserting duplicate frames somewhere in the middle of the sequence disrupts the temporal flow of the sequence. And stretching the motion by repeating every frame a set number of times produced a slowed-down, choppy sequence of the motion. In contrast, repeating the first frame of the sequence only delays the start of the motion, while maintaining temporal consistency.

This padding ensures that the model can already create predictions from the very first frames of an input sequence, even when the full input history length has not been reached yet. Although this situation is most relevant at the start of a sequence, it also allowed the evaluation of how a model performed with fewer input frames.

To address the research questions posed in the introduction, the models were fed samples of data. This could be done in two separate ways over a fixed-length evaluation sequence (e.g., 75 frames for the GCNext model). This sequence could be divided into three distinct parts: the input frames used by

the model; the prediction target/ground truth motion; and the remaining unused frames of the sample.

1) Back-to-front: In this version, the data "anchored" itself on the end of the input movement and aimed to predict the same motion segment every time. Over time, more "historical" data became available to the model, gradually providing more data step by step, until the full motion was fed. This is visually shown in Figure 3a, without the padding. For example, if 10 frames were given as an input (frames 41-50), the model predicted the subsequent motion (frames 51-75). If 11 frames were used, the input shifted one frame earlier (frames 40-50), while the prediction target remained fixed (frames 51-75).

This way of evaluating the data was more in line with the literature, because the model aims to predict the same target segment each time. By only varying the number of observed input frames, it was possible to evaluate how each additional frame influenced the prediction accuracy without obfuscating it with changes in the target sequence.

2) Front-to-back: In this version, the data "anchored" itself on the start of the input movement and aimed to predict the subsequent motion. Over time, more new data was fed to the model. This means that the model aimed to predict a target segment that was shifted over exactly one frame every time a new frame of data was added. This is visually shown in Figure 3b, without the padding. For example, if 10 frames were given as an input (frames 1-10), the model predicted the subsequent motion (frames 11-35). The rest of the sequence was unused (frames 36-75). If instead 11 frames were given to the model, the input motion increased by 1 (frames 1-11), and subsequently the prediction target too (frames 12-36).

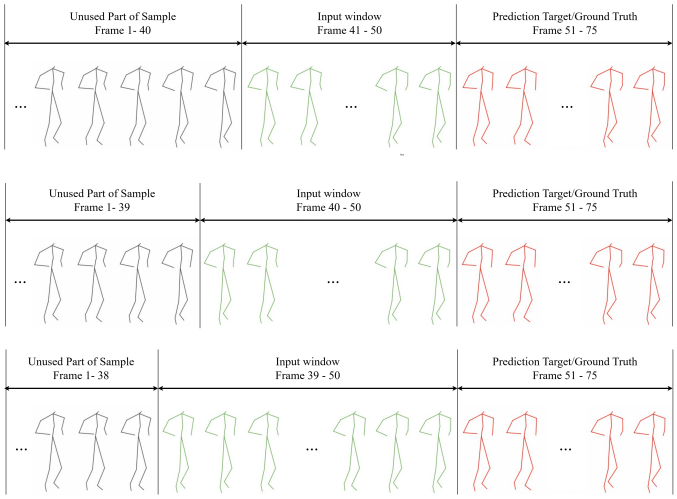
This way of evaluating the data was more in line with how real-time data acquisition will take place.

C. Experiment setup

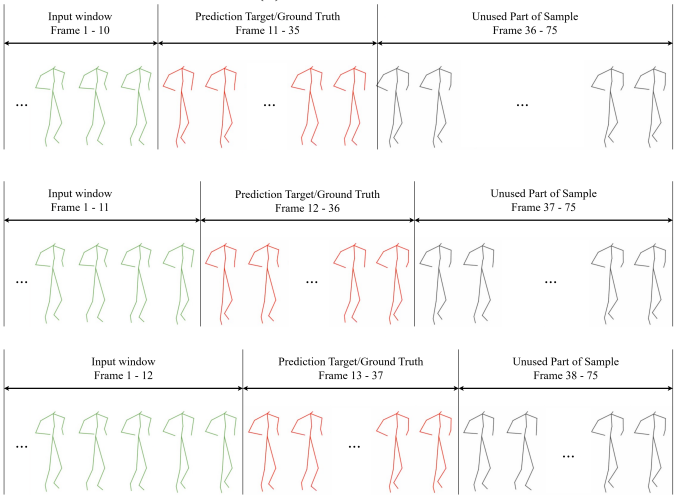
Three experiments were conducted to answer each research question.

1) Accuracy versus Number of Observed Frames: All models were evaluated on their respective datasets, as described in subsection III-A. For each of these samples, the number of input frames available to the model was incrementally increased, starting at 1 and increasing to each model's expected maximum (50 for the GCNext model, and 25 for the PhysMoP model). The reported accuracy is the average MPJPE across all samples in the evaluation set, at four distinct time horizons (80 ms, 400 ms, 560 ms, 1000 ms). [4] These horizons follow the reported evaluation metrics of the original authors, while spanning short-, medium-, and long-term predictions [12], [17]. The MPJPE metric was computed as described previously, with each joint contributing equally to this mean. Note that in this experiment, the total input size for each model remained fixed by using the padding mechanism described in subsection III-B. Therefore, this experiment aimed to isolate the effect of gradually increasing the proportion of informative frames, while keeping the model identical otherwise.

[4]Corresponding to 2, 10, 14, 25 frames into the future.



(a) Back-to-front



(b) Front-to-back

Fig. 3: Illustration of back-to-front (a) and front-to-back (b) feeding methods. Green indicates the input frames, red the prediction target (ground truth), and grey the unused part of the sample. Increasing the input window does not shift the prediction target in the back-to-front method, whereas it does in the front-to-back method.

To provide additional insight into model behaviour, the prediction accuracy was separated by upper- and lower-body joints. This division is not commonly seen in previous work, but was included here to explore whether models or datasets biased their performance on certain body regions. Lower-body motions, such as standard gait cycles, often involve more regular and cyclic patterns [22], [23], whereas upper-body motions are typically more erratic. By separating the two, it was possible to identify whether certain models or datasets favoured one region over the other, rather than evaluating their performance as a single combined value (see Appendix E, Figure E-1 for the corresponding skeletal structures and upper/lower-body division).

Additionally, to synthetically evaluate both models on the

same data, the evaluation set of the AMASS dataset was retroactively converted to the same form as the Human3.6M dataset. Conversion was done by transforming the original AMASS data in its axis-angle form to 3D Cartesian coordinates. This converted dataset was then used to evaluate the GCNext model in the same way as the PhysMoP model, to determine whether performance differences were caused by inherent model limitations or dataset-specific artifacts. While this is not a one-to-one conversion, it did give a rough comparison of the two models on the same motion. The mapping is shown in Appendix E, Table E-I

2) Temporal resolution: During training, each model is only exposed to data at a specific frame rate. To evaluate how these models perform on data that has different temporal resolutions, a specific subset of their respective evaluation set was resampled at different frame intervals. This experiment examined how changes in the spacing between observations affected the prediction accuracy. More specifically, the GCNext model was evaluated on the "walking" subset, and the PhysMoP model on the "treadmill_norm" subset of their respective datasets; brief overviews of these subsets are provided in Appendix F (Human3.6M) and Appendix G (AMASS), showing representative joint distributions that show clear patterns over a gait cycle.

When the resample rate is larger than 1, frames are skipped, increasing the time interval between data points. The input window was downsampled by creating a uniformly spaced subset of indices with spacing α, anchored at the boundary between input window and prediction target.

Concretely, say that the original sequence was $x_0, x_1, \cdots, x_{L-1}$, which represented the entire walking sample. To obtain a smaller, downsampled, evaluation sequence, the original sequence needed to be spliced, which was divided into two distinct parts:

- an input window with length T_h, which needed to be downsampled to N frames,
- the prediction target with length $P = L - T_h$, starting at x_{T_h}

The value of T_h is determined by the desired input length N and the chosen resample rate α: $T_h = \lceil N\alpha \rceil$. This ensures that the input slice is long enough to contain exactly N uniformly spaced samples at the specified resample rate.

To downsample this input window, the set of selected non-integer indices was defined as:

$$I_\alpha = \{i_k = (T_h - 1) - k\alpha, \mid k = 0, \ldots, N-1\} \quad (5)$$

Since this definition created indices in descending order, the resulting set of indices was reversed before being used. For example, using a resample rate of $\alpha = 1.8$ with $N = 25$ input frames (and thus $T_h = 45$), we obtained the indices:

$$I_{1.8} = \{0.8, 2.6, \ldots, 35.0, 36.8, 38.6, 40.4, 42.2, 44.0\} \quad (6)$$

Because the usage of non-integer resampling rates resulted in non-integer indices, the corresponding frames did not align with the discrete samples of the original sequence. To obtain the motion data at these intermediate frames, linear interpolation was used between the two neighbouring frames (e.g., for index $i_k = 42.2$, the linear interpolation used frames 42 and 43).

Conversely, when the resample rate was smaller than 1, additional frames needed to be inserted through linear interpolation, decreasing the time interval between data points. Linear interpolation was used here because the time window between consecutive frames was already very small (less than 40 ms). Prior work shows that 40 ms typically corresponds to 4 to 5% of a typical gait cycle, and finds that linear interpolation yields accuracy comparable to spline interpolation [22]. Combined with the fact that human motion often shows local linear behaviour [23], linear interpolation is sufficient for generating intermediate frames without introducing significant errors. Empirically, this was confirmed by visually comparing interpolated motion sequences to the original, which showed no noticeable jitter or discontinuities.

3) Real-time performance metrics: In addition to model accuracy, real-time deployment requires these models to meet strict temporal constraints.

To evaluate both models' ability to be used in a practical real-time settings, three performance metrics were defined. These metrics were gathered by evaluating the models on continuous walking sequences. These were the "walking" subset for the GCNext model, the "treadmill_norm" subset for the PhysMoP model.

To ensure a consistent number of input frames for the models, this experiment used a sliding window principle. This window size is specific for each of the models (i.e., 50 input frames for the GCNext model, and 25 for the PhysMoP model).

First was the prediction time. This was the time spent purely inside the model's forward pass. It measured how much time the model required to produce an output once the input tensors were already prepared and initialized. This excluded any preprocessing steps (e.g., window updates, data formatting).

Second was the latency. This was the full end-to-end time from the moment the complete input window became available to the model, until the final prediction was ready to use. This included preprocessing, the forward pass, any autoregressive loops and any storage of data. Latency therefore represented the actual delay a real-time system would experience. Prediction time is only the computation of the model, whereas latency includes the entire pipeline from receiving the input window to producing a usable prediction. Latency is therefore always greater than or equal to prediction time.

Lastly was the jitter defined as the standard deviation of the latency. This represented the variability in prediction latency, denoting the consistency of the model.

For each model, the prediction time, latency and jitter were measured every time a prediction was made over the continuous sequence, and then averaged to obtain the final values. All real-time metrics were obtained on a laptop CPU (Intel® Xeon® W-11855M @ 3.20 GHz, 32 GB RAM). Since

only a single sample was processed at once, no parallelization/batching is required, and CPU execution is generally the most efficient in this setting.

A model is considered feasible if the end-to-end latency does not exceed its frame interval. Since the implementations were not designed with real-time deployment in mind, some leeway was allowed for the end-to-end latency: values up to 20% of the total latency may still be considered acceptable, as they could be reduced through design optimizations. While no hard threshold is imposed on jitter, a high amount of jitter would limit the real-time usability.

IV. Experimental Results

The following sections address the results of the three aforementioned experiments. First the different types of feeding strategies will be evaluated in subsection IV-A. subsection IV-B reports how the number of observed frame influences prediction accuracy, and the general performance of the models. subsection IV-C covers the effect of temporal spacing between frames. Lastly subsection IV-D provides a short overview of the real-time feasibility by quantifying certain real-time performance metrics for the models. An overview of these experiments and their sub-experiments is also provided in Appendix H, Table H-I.

A. Back-to-front versus Front-to-back

The two feeding strategies were evaluated on both models. As shown in Figure 4, both approaches resulted in similar performance across all prediction horizons and input lengths. The difference in MPJPE between the two types of input methods was small, particularly after five frames of input data. The mean absolute difference in MPJPE between the two different feeding strategies is shown in Table I. Alongside that, the percentage difference is also shown. This metric is symmetric and therefore appropriate when we want to compare two values that lack a clear reference or direction [24]. Here it became clear that the largest relative difference occurred at the 1000 ms prediction horizon for the GCNext model ($\approx 4.6\%$), which corresponds to an absolute difference of 5.19 mm. This absolute difference was negligible compared to the overall MPJPE of this model at this prediction horizon. Prior to this horizon, the relative difference stayed below 5%, and for the PhysMoP model, the percentage difference stayed below 2.2%. This showed that the choice of feeding strategy had a relatively small impact on the prediction accuracy of the models, which is expected since both feeding directions contain the same temporal information. Since the front-to-back method more closely resembled real-time applications, this method was used for the remainder of the experiments.

B. Accuracy versus Number of Observed Frames

This section covers the accuracy of the models, with the number of observed frames being varied. For illustrative purposes, supplementary video material with predictions of the upcoming experiments is available at https://github.com/AndrewHutani/HMP/tree/main/motion_examples.

TABLE I: Mean Absolute Difference (MAD) in MPJPE [mm] and percentage difference between back-to-front and front-to-back feeding strategies across prediction horizons.

Model / Branch	80 ms	400 ms	560 ms	1000 ms
GCNext	0.13 (0.66%)	2.29 (3.60%)	3.83 (4.52%)	5.19 (4.64%)
PhysMoP Physics	0.01 (1.79%)	0.53 (1.82%)	1.12 (1.90%)	2.85 (1.81%)
PhysMoP Data	0.23 (2.13%)	0.69 (1.68%)	0.99 (1.43%)	1.22 (1.13%)

TABLE II: Comparison of the model's MPJPE (mm) between the original implementation and this work's reimplementation.

Model / Branch	80 ms	400 ms	560 ms	1000 ms
GCNext (Original)	9.3	56.4	74.6	108.7
This work	9.7	56.2	77.4	105.0
PhysMoP - Physics	0.6	21.0	47.6	140
This work	0.1	17.4	43.1	136
PhysMoP - Data	1.4	11.6	21.0	53.8
This work	1.1	9.8	20.0	50.5

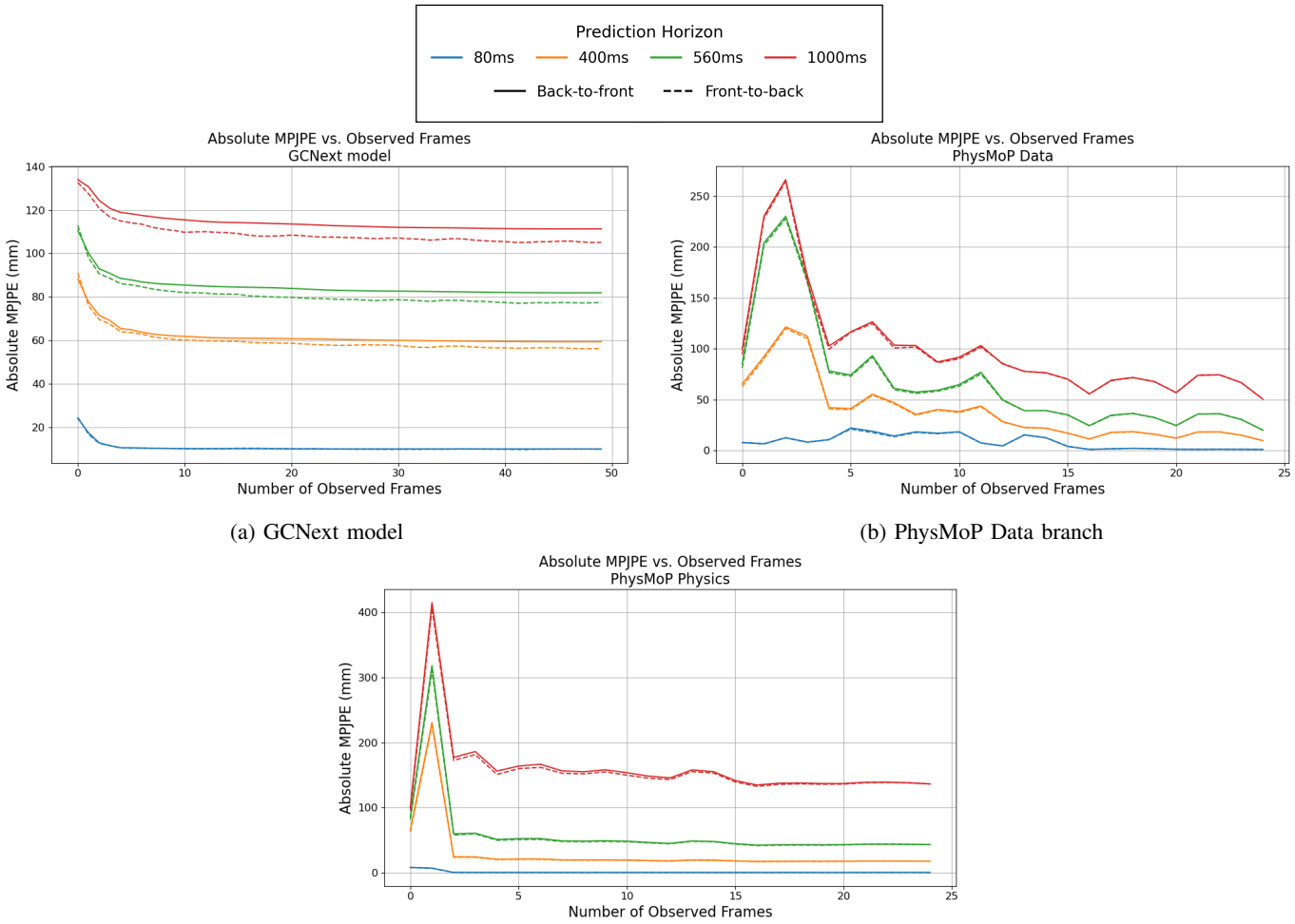
Over the two models, at the maximum number of input frames, the shown MPJPE values were within the same range as the original implementations, as reported in Table II. This showed that the current implementation performed comparably under standard evaluation conditions, and that subsequent analysis of temporal trends could be considered representative. Figure 5 summarizes the effect of the number of observed frames on the overall prediction accuracy. The corresponding raw MPJPE values with 95% confidence intervals are reported in Appendix I.

While Figure 5 showed that overall the average MPJPE decreased as more observed frames became available, the absolute errors did not converge to a single value for each horizon across the models. This is to be expected, since the three implementations differ in both architecture and underlying modelling approach (data-driven versus physics-based), which produces different error ranges. Moreover, convergence to a specific error value did not happen for all models: for instance, the PhysMoP data branch in Figure 5d showed a downward trend without settling into a clear plateau.

1) GCNext: The performance is shown in Figure 5a. Here it was visible that overall, the model's performance improved as more historical data became available to the model. Additionally, the performance increase seemed to plateau between 5-10 observed frames, with diminishing returns in the performance. The gain in MPJPE dropped below 1% after 6 frames for all prediction horizons.

Additionally, it was visible that the GCNext model was able to make more accurate predictions for the lower body for larger prediction horizons. At first glance, this could be explained by the cyclic and more structured nature of the movement of the lower body, which introduces temporal regularities. [23] However, this interpretation is preliminary and is revisited in subsubsection IV-B4.

A qualitative example of the GCNext predictions is shown in Figure 6, where the ground-truth and predicted poses for a



(a) GCNext model

(b) PhysMoP Data branch

(c) PhysMoP Physics branch

Fig. 4: The performance of the models using the two different feeding methods

single Human3.6M walking sample is shown. It is visible that the model was able to predict poses nearly identical to the ground truth at 80 ms, while for longer prediction horizons the predictions started to deviate gradually.

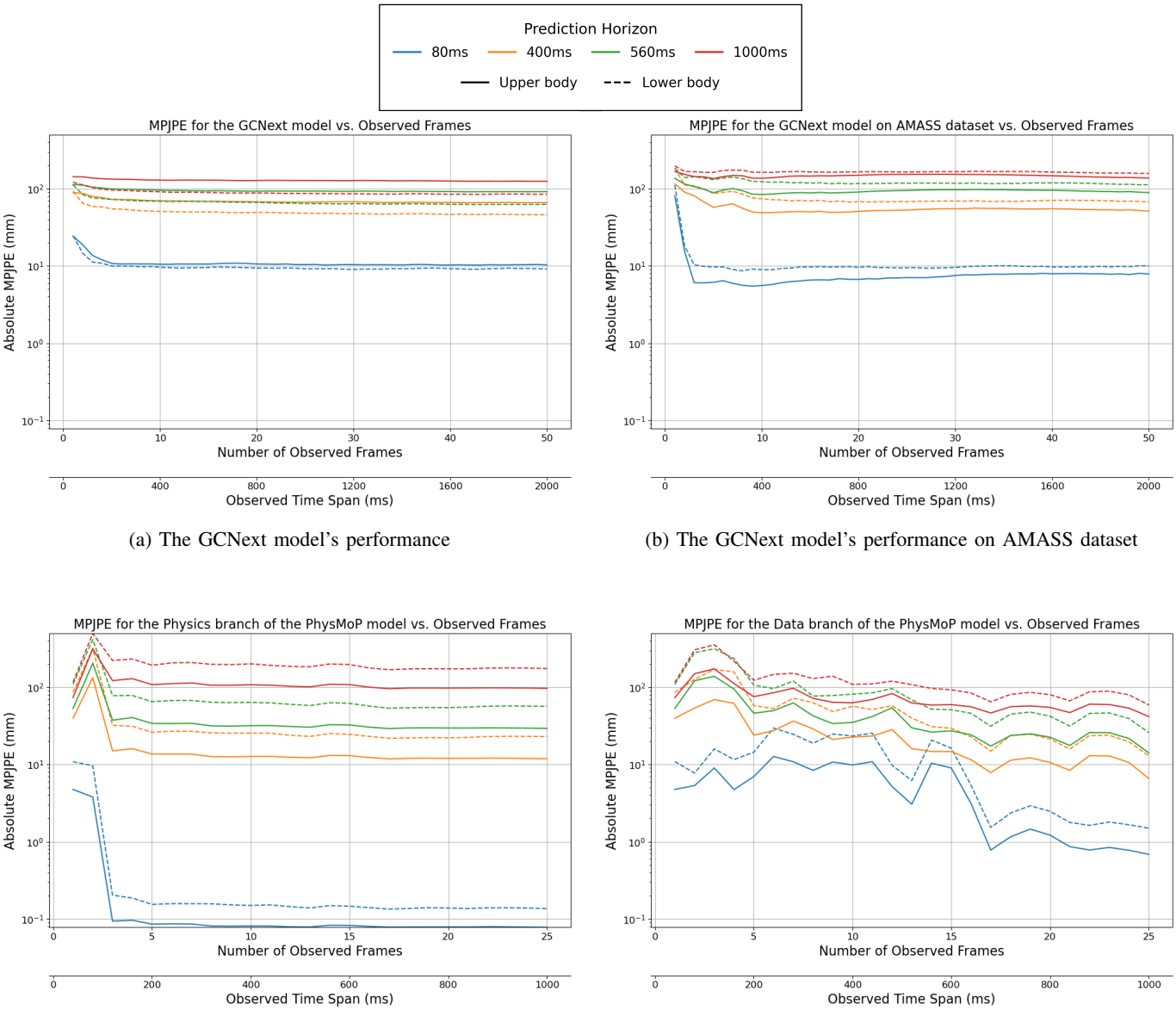
2) PhysMoP - Physics Branch: When looking at the performance for the physics model in Figure 5c, it was notable that the performance for the shortest prediction horizon was better than that of the GCNext model (Figure 5a). For the physics branch, the MPJPE went to 0.1 mm for the 80 ms time horizon. Additionally, it was visible that the predictions for the upper body were better than the predictions for the lower body.

Moreover, the performance was worse when using fewer than 3 frames, and stabilized once 3 frames became available to the model. This is because this physics model only ever uses 3 frames, meaning that when the model has access to more than 3 frames, only the most recent 3 frames are used to predict the future movement. This reliance on 3 frames is not arbitrary, but reflects the underlying physics. Acceleration is a second-order derivative of position, and requires at least 3 successive frames to approximate in discrete form.

A notable outlier occurred at 2 observed frames, where performance degraded for the longer prediction horizons. With only 2 frames available, the model had insufficient temporal information to infer meaningful accelerations, which required at least 3 consecutive frames. This resulted in incorrect velocity and acceleration estimations, suggesting that predictions obtained from fewer than 3 input frames provided insufficient temporal context, and should not be considered usable.

Because the physics branch grounds its predictions in only the three most recent frames, it effectively assumes that the movement trend in those three frames continues. In other words, the branch does not infer new motions, but propagates the most recent dynamics forward. This works well for short horizons, where the motion remains consistent, but becomes unrealistic when the trajectory of certain joints changes or stops. For example, if a leg is about to reverse its direction during a gait cycle, the physics branch will continue to predict the upwards motion of the leg.

A qualitative example of the PhysMoP physics branch



(a) The GCNext model's performance

(b) The GCNext model's performance on AMASS dataset

(c) The Physics branch of the PhysMoP model's performance

(d) The Data branch of the PhysMoP model's performance

Fig. 5: The performance of the models on their respective datasets with varying number of observed frames, separated by upper and lower body. The raw MPJPE and 95% confidence intervals are shown in Appendix I

prediction is shown in Figure 7. Here, the continuation of the recent dynamics is visible in the backwards tilt of the torso, illustrating how the physics branch propagate the motion trend from the last observed frames.

3) PhysMoP - Data Branch: For the performance of the data-driven branch (Figure 5d), we observed the same phenomenon where the lower body errors were noticeably higher than the upper body errors (where this was the other way around for the GCNext model). As with the GCNext model, the prediction error decreased as the observation length increased. In this case however, the improvement was more gradual and substantial. More specifically, the error decreased notably up to around 10-15 frames, after which the gains became smaller. Although the improvement beyond this point was smaller, a slight downwards trend was still visible.

Additionally, comparing the data-driven and physics-based branches, it was visible that for longer prediction horizons, the data-driven model outperformed the physics-based branch; for the 1000 ms prediction horizon, the data-driven branch stabilized around a MPJPE of 50 mm for the upper body and 80 mm for the lower body, where the physics-based branch stabilized around 100 mm and 180 mm respectively. This is most likely the case due to the error propagation in the physics branch, where small inaccuracies in physical parameters grow over time.

Overall, the PhysMoP data branch showed more fluctuation

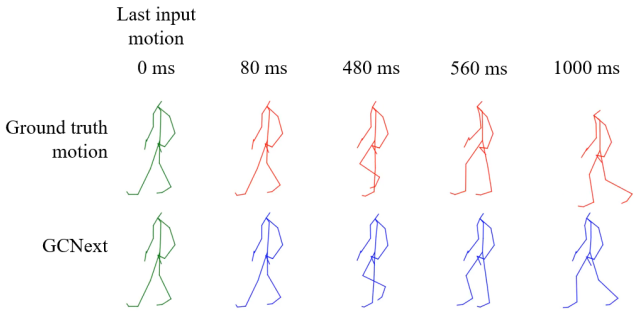


Fig. 6: Qualitative comparison between the ground-truth motion (top, red) and the GCNext model's predicted motion (bottom, blue) for one Human3.6M sample. The green skeleton shows the last observed input frame.

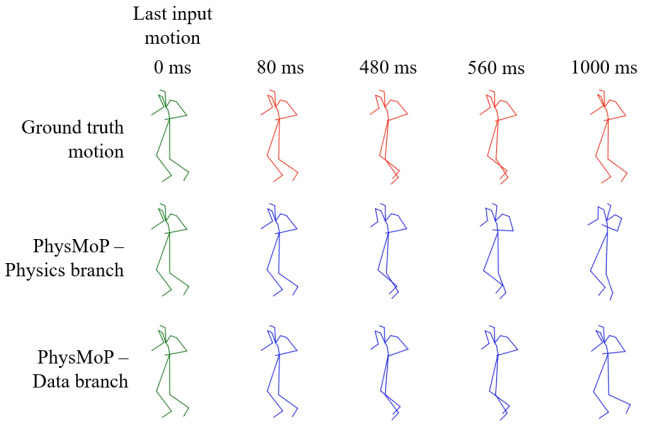


Fig. 7: Qualitative comparison between the ground-truth motion (top, red) and the PhysMoP's predicted motion (bottom, blue) for one AMASS sample, separated by physics and data branch. The green skeleton shows the last observed input frame.

in performance across different prediction horizons compared to the GCNext model. This variability could be attributed to its model architecture: the MLP treats all input features independently and has no explicit spatial or temporal structure. As a result, additional frames simply add more input features without any explicit structure, which could result in the model forming misleading associations between features.

The same Figure 7 also shows the predictions of the data branch. Compared to the physics branch, the predicted poses remained more similar to the ground-truth over longer horizons.

4) GCNext on AMASS (cross-dataset test): To test whether certain model performance characteristics are model-specific or dataset-specific, the evaluation set of the AMASS dataset was converted into the Human3.6M format and used to re-evaluate the GCNext model. The results are shown in Figure 5b. Note that the complementary experiment (PhysMoP evaluated on Human3.6M dataset) could not be conducted, because of the difficulty of transforming the Cartesian coordinates into Euler angles without a body model. As a result, this analysis should be considered as suggestive, rather than conclusive.

Overall, we observed that the performance was worse compared to the GCNext model on its original dataset, Human3.6M. This is to be expected, since the model was not trained on the AMASS dataset. Additionally, the conversion from the AMASS to the Human3.6M format, is not fully one-to-one; there are joints in the spine, feet and hands that are omitted in the AMASS format, which are available and required in the Human3.6M format (see Table E-I). This mismatch can therefore be an explanation for the reduced performance, since the model is using observations and making predictions in a representation it was not trained on.

For the shortest prediction horizons, the overall change in performance was similar to the performance on the Human3.6M dataset (Figure 5a); the MPJPE decreased as more frames of observed data became available to the model, and plateaued around 10 to 15 frames.

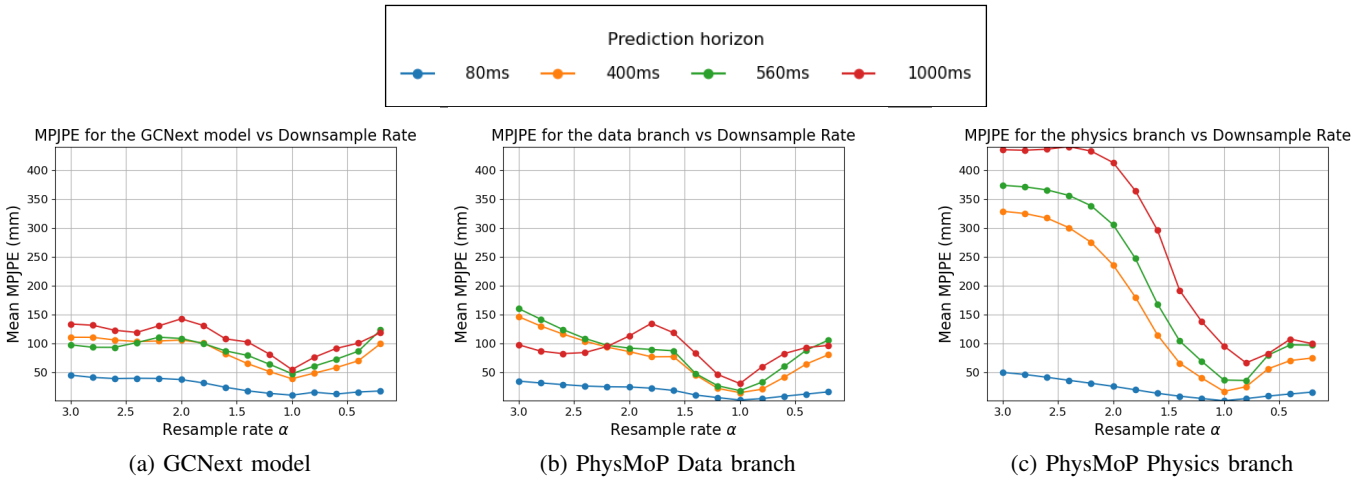
Lastly, we saw that for all prediction horizons, the upper body performed better than the lower body. This was the complete opposite of the initial performance of the GCNext model (Figure 5a), but was in line with the performance of the PhysMoP model (Figure 5c, Figure 5d). This suggests that the performance difference between the upper and lower body is not inherent to the model architectures, but is likely influenced by specific characteristics of the dataset. For example, differences in the diversity of the included motion, or the presence of robust cyclic lower-body actions, could all bias the models toward better performance for a particular body region.

These results directly addressed the first question. Both models benefited from longer historical inputs, but to different extents. Beyond a certain number of frames, adding additional input frames no longer yielded meaningful improvements in prediction accuracy. However, the threshold at which this occurred seemed to be specific to each model. Furthermore, the AMASS experiment suggested that dataset characteristics also played a key role, which could be seen in the reversed upper-/lower-body accuracy pattern of GCNext between the two datasets

For real-time human motion prediction, this means that using a longer historical window does not always translate into better performance; beyond a model-specific cut-off point (e.g., $\sim$ 6-10 frames for GCNext), the possible additional latency of processing longer input windows may not be justified by the small gains in accuracy.

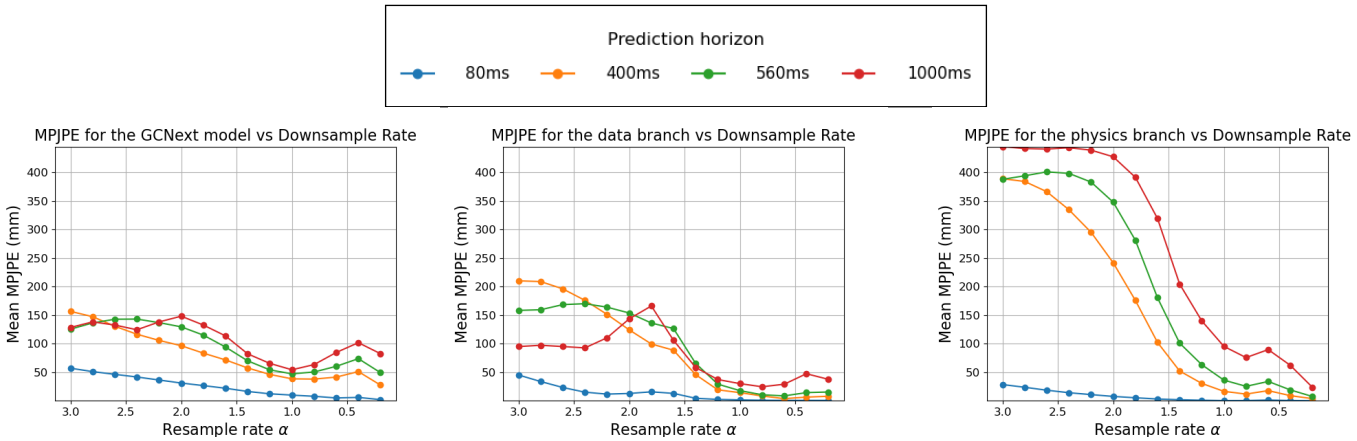
C. Temporal resolution

Figure 8 shows the performance of the models when downsampling ($\alpha > 1$) or upsampling ($\alpha < 1$) the input sequence.



(a) GCNext model (b) PhysMoP Data branch (c) PhysMoP Physics branch

Fig. 8: MPJPE across prediction horizons for different resample rates (α), evaluated while keeping the ground truth frequency fixed. The raw MPJPE and 95% confidence intervals are shown in Appendix I



(a) GCNext model (b) PhysMoP Data branch (c) PhysMoP Physics branch

Fig. 9: MPJPE across prediction horizons for different resample rates (α), evaluated while matching the ground truth frequency. The raw MPJPE and 95% confidence intervals are shown in Appendix I

Looking at the performance of both models, we saw that the prediction error was the lowest at the training sampling rate ($\alpha = 1$), and increased as the resampling rate deviated.

Especially the physics branch of the PhysMoP model seemed to degrade completely as the resample rate increased, with the MPJPE increasing beyond 400 mm for the longest prediction horizons.

It was notable that for the PhysMoP data branch, the error decreased slightly at the 1000 ms prediction horizon when the resample rate exceeded $\alpha > 1.8$. Spatial plots of the predictions showed that at this specific resample rate, the model still attempted to predict motion patterns, but failed. At higher resample rates $\alpha > 1.8$, the predictions no longer resembled realistic motion; instead, the model reverted to joint positions that were jittery around a somewhat static configuration. This behaviour results in lower magnitudes of positional error, explaining the apparent "improvement" in MPJPE.

While both models show noticeable degradation in performance as the temporal resolution changes, it is important to take into account that the model is trained with a fixed time interval, meaning that the model learns motion patterns based on a specific frequency.

Both the GCNext model and the PhysMoP data branch are unaware of the time difference between frames; their predictions are based on the assumption that each input frame represents a fixed time step. When the time interval between these frames is changed, this assumption no longer holds.

As a result, changing the time interval between the input frames could implicitly be interpreted as changing the speed of the motion, which also affects the observed dynamics. For example, doubling the time interval results in a larger displacement between observations, which could be interpreted as higher velocities and accelerations. On the other hand, adding additional frames through interpolation reduces the apparent velocities and accelerations. As a result, the models no longer receive motion cues that are temporally consistent with the trained data, which leads to misinterpretation of the underlying dynamics. Note that this behaviour is expected for these implementations, since they do not explicitly compute velocities or accelerations and therefore do not preserve their behaviour when the underlying time-scale changes.

For the physics branch of the PhysMoP model, this is a bit more complex. As mentioned earlier, this branch makes predictions based on estimated parameters of the Euler-Lagrange equations. The predicted joint accelerations are then used in a Verlet integration step to predict the future joint positions. This integration includes a fixed $\Delta t = \frac{1}{25}$, which is pre-defined in the training process, and thus is not learned or explicitly known as a time scale. When the input sequence is resampled to a different frequency, the effective time interval is also altered, but the model still uses the pre-defined Δt. Additionally, the estimation of the Generalized bias force vector $C(q, \dot{q})$ depends on inferred joint velocities $\dot{q}$, which are estimated on a fixed time interval. Thus resampling the sequence may produce mis-scaled velocity features, which in turn warp the estimated dynamics. This combination of mis-scaled inputs and mismatched integration constants explains why the physics branch shows such noticeable performance degradation.

More generally across all three implementations, resampling could effectively be interpreted as a difference in the speed of motion, since none of the branches learn of the true time interval between frames. This was further supported by looking at the spatial predictions and ground truth motion at the specific resample rates. For resample rates smaller than 1, the predicted motion resembled a motion that was slower than the actual motion.

To further test this hypothesis, the experiment was altered to have the ground truth frequency match the altered input frequency. This changed the ground-truth motion to be the same "speed" as the input motion, which the model used to create the prediction. In Figure 9, we saw that predictions, for a resample rate lower than 1, resulted in an improved performance over all prediction horizons, compared to when the output did not match the input frequency. This supported the hypothesis that the model evaluates these resampled sequences as motions at different speeds.

However, we also saw that for predictions with a resample rate higher than 1, the performance did not improve, and for some specific resample rates the performance actually declined. Looking at the spatial predictions and ground truth motion for these downsample rates, we saw that even at the same accelerated rate, the model's predictions remained unstable.

To further examine whether the shown performance degradation for higher resample rates came from increased motion speed or a different underlying artifact, an additional comparison was performed. Here, the resample rate was chosen in such a way that the resampled sequence's cadence matched the different speeds of locomotion in the AMASS treadmill subset. Concretely, the previously used "treadmill_norm" sequences were resampled to have their cadence match that of the "treadmill_fast" sequence ($\alpha \approx 1.10$), and "treadmill_slow" sequence ($\alpha \approx 0.82$). These results are shown in Table III. Here it was visible that for artificially accelerated motions, the prediction error increased over all prediction horizons. This showed that the degradation observed for $\alpha > 1$ was not only a consequence of faster motion, but also due to the lack of dynamic cues found in natural faster walking patterns. In contrast, when the "treadmill_normal" sequences were resampled to match the cadence of the "treadmill_slow"' sequence, the models performed roughly the same. This indicates that slower motions remain consistent after resampling.

These findings provided insight into the second question. The models assume a fixed temporal spacing during training. As a result, changes in the time interval between data points are interpreted as changes in the speed of the motion. This is reflected in a better performance for resample rates $\alpha < 1$, since these slower motions are easier to approximate. For resample rates $\alpha > 1$, the performance deteriorated sharply, since artificially faster motion lacks natural cues. Matching the output frequency to the resampled input further supported this interpretation: slowing down the motion improved performance, while speeding it up did not. Together, these results indicate that the models are particularly vulnerable to artificially accelerated dynamics.

In a real-time setting, this susceptibility implies that models trained on fixed frame rates likely will struggle when input timing varies, requiring a framework that includes strict control of the sensor frequency, or explicit temporal context in the model.

D. Real-time performance metrics

The results of the real-time metrics are shown in Table IV. Since the models operate at 25 Hz, the real-time limit is a 40 ms frame interval. Latency relative to this interval is the only metric that determines feasibility.

At first glance, only the data-driven implementation of the PhysMoP model was able to complete its prediction time within the interval between two data points. However this is a bit misleading; prediction time only measures the model's internal forward pass. Real-time feasibility is determined by the end-to-end latency, which for this branch still exceeded the 40 ms frame interval.

Looking at the three other implementations, their prediction time and latency far exceeded the time between data points. This could be because these three implementations use some sort of autoregression, which is shown in Appendix C [25];

TABLE III: Comparison of MPJPE [mm] between natural and cadence-matched artificial motion. Treadmill-normal sequences were downsampled by $\alpha_{\text{fast}} = 1.10$ and interpolated by $\alpha_{\text{slow}} = 0.82$ to match treadmill-fast and treadmill-slow cadence, respectively.

	Fast Walking				**Slow Walking**			
Model / Branch	80 ms	400 ms	560 ms	1000 ms	80 ms	400 ms	560 ms	1000 ms
PhysMoP Data (natural)	**1.94**	**14.39**	**17.43**	**33.61**	**1.06**	**11.33**	**14.95**	**26.97**
PhysMoP Data (artificial)	2.38	14.64	19.83	29.47	1.05	15.51	20.61	30.99
PhysMoP Physics (natural)	**0.36**	**20.36**	**43.75**	**108.61**	**0.23**	**12.78**	28.57	86.95
PhysMoP Physics (artificial)	0.41	22.04	47.28	120.66	0.46	13.65	**28.27**	**74.91**
PhysMoP Fusion (natural)	**0.37**	**16.31**	**25.43**	**49.59**	**0.23**	**11.44**	**22.25**	**42.92**
PhysMoP Fusion (artificial)	0.43	20.71	32.14	52.29	0.46	12.64	25.07	45.31

TABLE IV: Average real-time performance metrics over the whole motion sequence

Metric	GCNext	PhysMoP		
		Data	Physics	Fusion
Prediction time [ms]	76.02	6.26	77.4	78.2
Latency [ms]	76.51	92.4	105	205
Jitter in latency [ms]	6.85	11.8	11.0	23.7

the prediction is made by predicting for a shorter time horizon and using that prediction to predict further, until the final time horizon has been reached. This looping nature of these implementation is time-intensive and likely causes this slower performance. In contrast, the data branch of the PhysMoP model directly predicts the entire horizon in a single pass, resulting in a faster prediction.

In Table IV, we saw that the physics and fusion branch of the PhysMoP model had the largest difference between their latency and prediction time. For the physics branch, this is likely because the implementation is more memory intensive. The model initializes several tensors that for the data branches do not need to be initialized [12]. For the fusion branch, this even larger increase is explained by the exact implementation of this branch; the fusion branch takes both the data and physics branch's prediction, and takes a weighted average of them. This means that the fusion branch has to wait for both branches before it can make its own prediction, which on its own is also done recursively. This additional wait time explains the largest difference between prediction time and latency.

Lastly we saw that the GCNext model had the lowest jitter, reflecting the most stable runtime. For the PhysMoP model, we saw a similar jitter for the data and physics branches, while the fusion branch had more variability. This can be attributed to the fusion branch combining both, and performing additional steps, which results in compounding variance in its latency. In general, the observed jitter remains relatively small compared to the total latency, indicating that timing variability is negligible for real-time deployment.

Overall, suitability is determined strictly by the latency relative to the frame interval. Under this criterion, the GCNext model was the closest candidate, but still exceeded the time interval between frames. The other branches fell even further out of the feasible range. This means that structural changes are necessary before any of these models can be deployed in a real-time system.

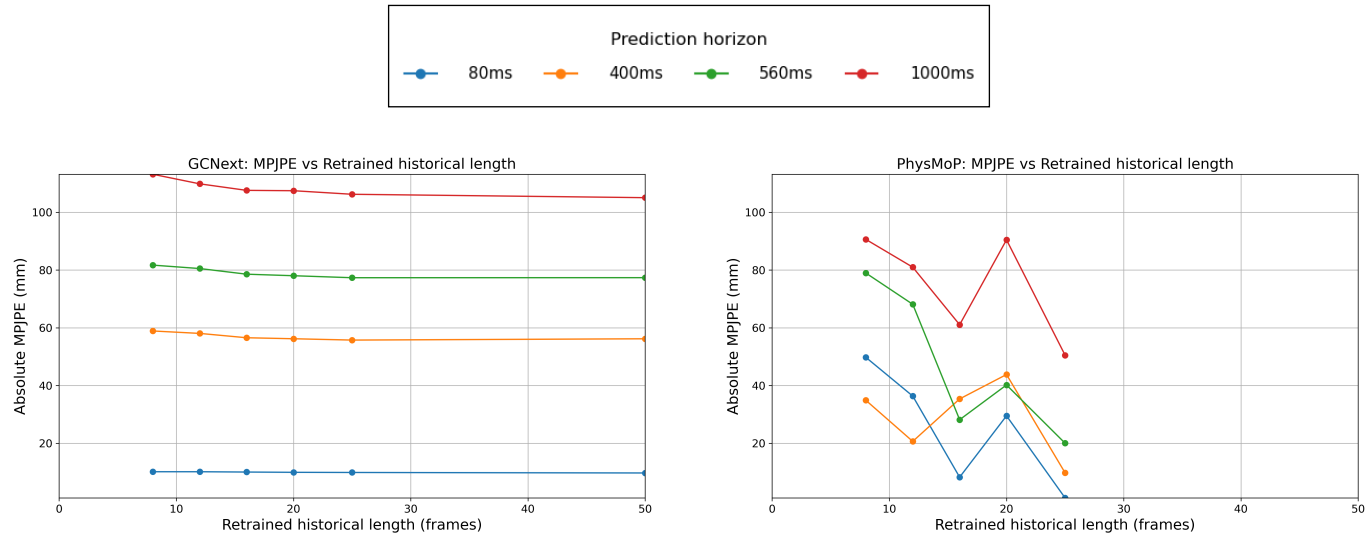
V. Reducing input history for real-time feasibility

The experiments shown in section IV evaluated existing models in terms of accuracy, temporal robustness and real-time performance. These analyses highlighted important trade-offs between accuracy and computational costs, while revealing limitations as well. For these models, their prediction accuracy improved only marginally beyond a certain input length, while these longer observed sequences could lead to additional prediction time and latency in the model. Because the accuracy gains did not compensate for possible additional prediction time, it suggested that the current models are not necessarily designed with real-time usage in mind, and that the current configurations are not automatically optimal for these applications.

In this section, we therefore go beyond the baseline evaluations and investigate whether reducing the input length of these models can improve responsiveness while keeping most of their prediction accuracy. To achieve this, the GCNext model and the PhysMoP data branch were retrained with input lengths of 8, 12, 16, 20, and 25, while keeping all other settings unchanged. The goal was to evaluate how their prediction accuracy changed with shorter historical windows, which was measured as the average MPJPE in the same manner as the previous experiments. This change in accuracy was then evaluated against the corresponding effect on prediction time and latency, to determine whether shorter input lengths offered a beneficial trade-off for real-time applicability.

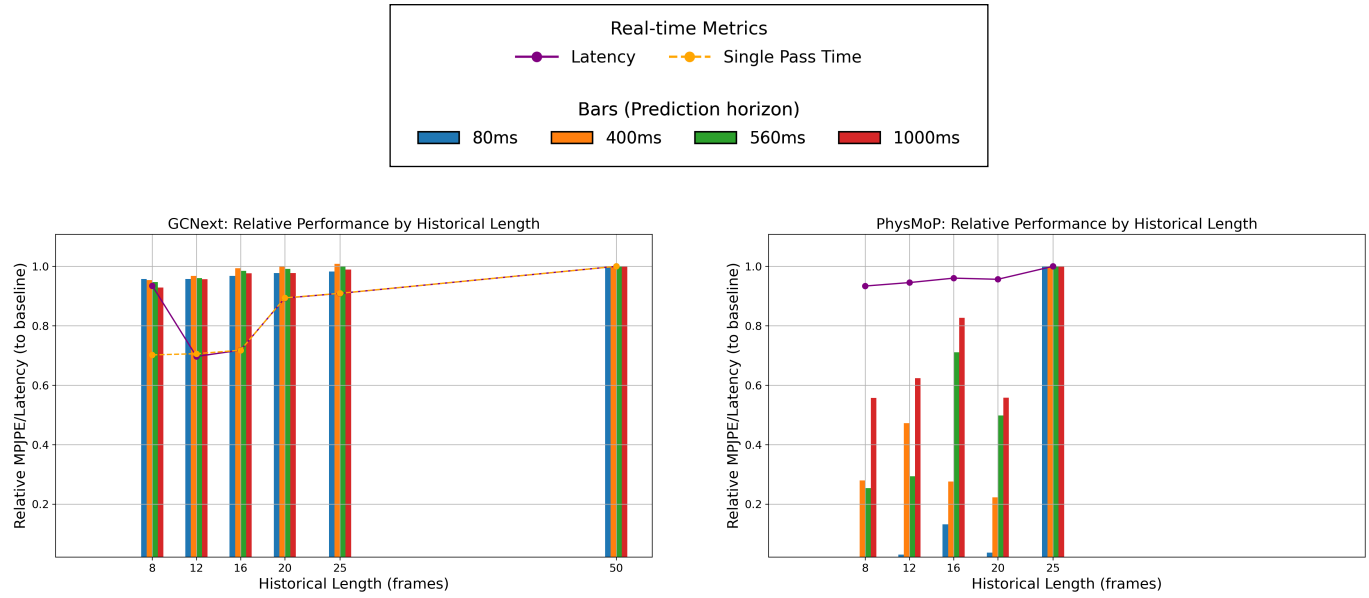
The retraining of the GCNext model showed that reducing the input history length led to a noticeable decrease in latency, while the prediction performance only marginally degraded (Figure 10a, 10c). When reducing the input length from the original 50 frames to 12, the MPJPE across all horizons increased only slightly, with a relative performance loss of less than 5%. In contrast, the overall latency decreased considerably, reaching a decrease of around 30% in latency, relative to the baseline. This can be attributed to a shorter historical window requiring less input frames to be processed by the forward pass. This results in smaller tensors and fewer convolution layers.

An exception to this behaviour was observed at the 8-frame configuration, where the latency increased again. This



(a) Absolute MPJPE of the GCNext model, retrained with different historical input lengths, evaluated at 4 prediction horizons.

(b) Absolute MPJPE of the PhysMoP model, retrained with different historical input lengths, evaluated at 4 prediction horizons.



(c) Relative performance changes of the GCNext model retrained with different historical input lengths. The values shown are the accuracy and latency as a fraction of the 50-frame baseline.

(d) Relative performance changes of the PhysMoP model retrained with different historical input lengths. The values shown are the accuracy and latency as a fraction of the 25-frame baseline.

Fig. 10: Prediction performance of the GCNext and PhysMoP models retrained with varying historical input lengths.

behaviour is a direct consequence of the autoregressive nature of the GCNext implementation; the model can only ever predict as many frames in the future as the number of frames provided in its input window. The original implementation of GCNext is designed to predict 10 frames into the future with each autoregressive pass, thus requiring 3 passes to reach the desired 25 total frames. However, when the model was retrained with only 8 input frames, this limit was also reduced to 8. This resulted in the model requiring 4 autoregressive passes to reach the 25 frame horizon. This is further supported by the single-pass time reduction shown in Figure 10c; this showed that at 8 historical frames, a single forward pass was still getting shorter, whereas the total latency was getting longer. This divergence confirmed that the additional autoregressive pass is responsible for the increase.

Overall, these results proved that retraining GCNext with shorter input histories produced models that had a more favourable balance between accuracy and latency. Additionally, this suggested that the original 50-frame configuration was excessive for a real-time application, and that a more

efficient design could be obtained by adjusting the history length.

For the PhysMoP model, retraining the model with shorter historical windows showed a different trend compared to GCNext (Figure 10b, 10d). While reducing the historical window of the model resulted in a lower latency, the prediction accuracy dropped much more and with greater variability. This erratic performance change was also shown in Figure 5d. More specifically, in all but one case (16 frames model at the 1000 ms horizon), the accuracy deteriorated to below 70% of the baseline, whereas the GCNext model maintained performance above 95%. This indicated that the PhysMoP data branch was more sensitive to the length of its input history, and also did not benefit from shorter historical windows as much. Additionally, this could suggest that a non-autoregressive model does not gain substantial performance efficiency from reducing the historical length.

It was noted however, that the performance of the baseline model of PhysMoP was better than the baseline model of GCNext. Across all prediction horizons, the PhysMoP data branch achieved lower absolute MPJPE values, even when the input history was reduced to only 8 frames. This showed that while PhysMoP suffered more in relative terms when retrained with shorter historical windows, the absolute accuracy still remained high. This highlighted that the strong baseline accuracy of PhysMoP continued to influence its performance, allowing it to outperform GCNext in absolute terms even when retrained with much shorter histories.

A. Reducing autoregressive passes

The analyses above showed that the GCNext model retained a high percentage of its original performance, but its latency was hindered by its autoregressive nature. To address this, we trained two modified GCNext models that optimized or reduced the autoregressive part of this model, while keeping other model architectures constant:

- Model 1 - Historical window: 16 (since that showed a lower latency). Each single pass predicts 13 frames into the future; the 25-frame prediction horizon is reached in 2 passes, while only discarding 1 frame of predictions.
- Model 2 - Historical window: 25. Each single pass predicts 25 frames, removing the autoregressive nature.

The performance of these models is shown in Table V. These results show that reducing the autoregressive passes was the most effective way to improve real-time feasibility. Both the two-pass and single pass variants achieved lower prediction time and latency compared to the baseline, while keeping a large portion of their accuracy. More specifically, the single-pass variant reduced the latency by around 70%, while maintaining the accuracy within 2 mm of the baseline across all horizons. From a broader perspective, these results highlight that real-time performance is influenced by both the size of the input history, and the number of autoregressive passes. However, this performance is dominated by the latter; minimizing the number of autoregressive passes results in the largest latency decrease. This suggests a general design guideline for human motion prediction models: use single-pass architectures where possible, and if autoregression is required, maximize the output horizon per pass to keep the total number of passes as small as possible.

VI. Discussion

The experiments shown in this thesis demonstrate that specific design choices, such as the input window length, temporal resolution of the observations, and model architecture, directly determine whether human motion prediction models are suitable for real-time deployment. While recent models, such as GCNext and PhysMoP achieve high prediction accuracy, their design and implementation have not been focused on real-time feasibility.

More specifically, this work finds that temporal configuration is not a secondary implementation detail, but an important design decision in the viability of real-time usage. Two aspects were found to be particularly important. The first concerns the trade-off between the length of the observed motion and the resulting prediction latency. Longer observation windows can provide more temporal context, but this comes at the cost of increased processing time, while not necessarily improving accuracy. In fact, this thesis has empirically shown that longer input windows add little benefit. This limited accuracy improvement suggests that human motion prediction can largely be inferred from its recent history. Human movement is locally smooth, meaning that short-term dynamics already capture most of the predictive information. As a result, extending the input window mainly adds redundant context after a model specific cut-off point, while increasing prediction time.

Second, the model architecture strongly influences the latency. Autoregressive models, such as GCNext and the physics branch of PhysMoP, require multiple sequential passes to reach the desired prediction horizon. By contrast, single-pass architectures, like the data branch of PhysMoP, show a prediction time roughly ten times lower than autoregressive models. Nevertheless, these multi-pass designs can still be optimized for real-time use by minimizing the number of autoregressive passes. This approach was shown to preserve most of the GCNext model's accuracy, while reducing the average latency substantially. This strong relationship between model structure and latency also highlights why different approaches, such as transformer- and MLP-based predictors, are still highly competitive: they can generate entire motion predictions in a single inference step. These findings suggest that both the length of the historical window and the model architecture determine its responsiveness in real-time settings.

Beyond the latency aspect, the experiments on temporal resolution show a limitation affecting the reliability of current models. It is shown that these models' performance can deteriorate when the temporal resolution of the observations differs from what they were trained on. This sensitivity shows that real-time feasibility is not only determined by whether a prediction can be made with the expected time frame, but also by the model's ability to remain accurate when the temporal spacing between the input frames changes.

TABLE V: Accuracy and timing of GCNext variants, all predicting 25 frames into the future. Hist. = input history length, SP. = single pass prediction length.

GCNext model			Avg. MPJPE (mm)				Avg. Single Pass Time (ms)	Avg. Pred. Time (ms)	Avg. Latency (ms)
	Hist. — SP.	**# passes**	**80 ms**	**400 ms**	**560 ms**	**1000 ms**			
Baseline	50 — 10	3	9.73	56.2	77.4	105.0	25.34	76.02	76.51
Best short-history	16 — 10	3	10.0	56.7	78.6	107.6	18.19	54.57	54.94
2-pass variant	16 — 13	2	10.3	57.0	78.5	106.9	18.21	36.43	36.79
Single-pass variant	25 — 25	1	11.2	57.9	78.7	106.5	23.49	23.50	23.88

Both models studied here were found to implicitly assume a fixed time interval between input frames. When this interval is changed, the prediction accuracy tends to deteriorate, especially when the time interval is increased. A contributing factor is that these models implicitly require a certain number of samples per underlying motion cycle to correctly represent the motion. When the sample rate becomes too low relative to the cycle duration, the models lose too much information crucial to reconstruct the dynamics of the motion. More generally, if too few frames are sampled, aliasing occurs and the model is unable to accurately infer the speed of the motion. Quantifying the exact sampling density for the specific cycle duration would require an extensive analysis of gait frequencies and is left for future research. Nevertheless, this sensitivity in the models highlights a practical limitation for usage in real-world robotic environments: reliable predictions require these input frames to be spaced strictly evenly. However, such consistency might be difficult to guarantee in practice. Research has shown that motion capture pipelines typically have variable latencies, resulting in differing time intervals between input frames [26]. This thesis argues that even minor variations in the sampling frequency, in the order of ten percent, can cause these models to misinterpret the motion dynamics, resulting in unstable predictions.

This limitation could be addressed with models that are explicitly aware of the associated time stamp of each input frame, rather than assuming fixed sampling intervals. Future work could explore these time-aware prediction architectures, where the true time difference between historical frames is provided as an additional input. An example implementation could be to add the true time interval as an additional feature during training, causing the model to be explicitly aware of absolute timestamps. Such designs could enable motion prediction models to be more general across various frequencies and maintain more stable performance in real-time settings where sensor timing could not be perfectly held constant.

Beyond these temporal aspects, another limitation lies in the datasets used to train the models. The GCNext and PhysMoP models were trained and evaluated on different datasets, which inevitably differ in motion diversity, scale, and skeletal parameterisation. More concretely, the AMASS dataset is larger, and includes a broader range of motion sequences. The experimental results suggest that these datasets differences bias the predictions (e.g., the model may predict upper-body movements more accurately than lower-body movements). Future work could examine to what extent the choice of dataset influences the exact model performance. This could be done by training a model on multiple dataset and evaluating the performance differences between these.

At a more global level, accurate motion prediction can improve both safety and efficiency in shared workspaces between humans and robots. It can allow robots to anticipate human trajectories and plan their movements proactively. In industrial environments, these predictions can aid in collision avoidance, while in healthcare settings they can enable early detection of instability or falls if the actual movements start to deviate too far from the predictions. The insights from this work show that temporal design choices directly influence whether a model can operate fast and consistently enough for such real-time settings.

Finally, the scope of this study is limited to two representative human motion prediction models. GCNext and PhysMoP cover data-driven and physics-based approaches, but do not reflect the entire range of existing architectures. As a result, the conclusions drawn here should be interpreted as contributing to a broader understanding of temporal effect, rather than an extensive overview of how temporal choices affect all model types. Future work could extend this analysis to a larger set of models to validate the applicability of these findings.

VII. Conclusion

This thesis investigated how temporal design choices affect the real-time feasibility of human motion prediction models. By conducting controlled experiments on two state-of-the-art models (GCNext and PhysMoP), it has been shown that both the length of the input history and the temporal resolution of the observations have a direct impact on model accuracy, latency and robustness.

The thesis showed that longer historical windows can improve prediction accuracy up to a model-specific threshold, after which additional frames yield in diminishing returns while increasing latency. Both models were also found to be highly sensitive to variations in the temporal spacing between input frames, highlighting the need for consistent sensor timing or time-aware architectures for real-time applications.

Among the evaluated models, single-pass designs proved the most suitable for real-time usage, as they generally have the lowest latency. Nonetheless, autoregressive models can still be viable if their architecture is optimized with real-time deployment in mind.

REFERENCES

[1] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," en, *Auton. Robots*, vol. 42, no. 5, pp. 957–975, Jun. 2018. DOI: 10.1007/s10514-017-9677-2.

[2] A. Sampieri, G. D'Amely, A. Avogaro, *et al.*, *Pose forecasting in industrial human-robot collaboration*, 2022. DOI: 10.48550/arXiv.2208.07308. arXiv: 2208.07308 [cs.RO].

[3] J. Jeong, D. Park, and K.-J. Yoon, "Multi-agent long-term 3d human pose forecasting via interaction-aware trajectory conditioning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024, pp. 1617–1628. DOI: 10.48550/arXiv.2404.05218.

[4] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of Human Full-Body Movements with Motion Optimization and Recurrent Neural Networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, ISSN: 2577-087X, May 2020, pp. 1792–1798. DOI: 10.1109/ICRA40945.2020.9197290.

[5] Z. Ren, M. Jin, H. Nie, J. Shen, A. Dong, and Q. Zhang, "Towards Realistic Human Motion Prediction with Latent Diffusion and Physics-Based Models," English, *Electronics*, vol. 14, no. 3, p. 605, 2025, Num Pages: 605 Place: Basel, Switzerland Publisher: MDPI AG. DOI: 10.3390/electronics14030605.

[6] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," 2017. DOI: 10.48550/arXiv.1705.02445. eprint: 1705.02445 (cs.CV).

[7] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014. DOI: 10.1109/TPAMI.2013.248.

[8] C. Ionescu, F. Li, and C. Sminchisescu, "Latent structured models for human pose estimation," in *2011 International Conference on Computer Vision*, 2011, pp. 2220–2227. DOI: 10.1109/ICCV.2011.6126500.

[9] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct. 2019. DOI: 10.48550/arXiv.1904.03278.

[10] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "A spatio-temporal transformer for 3d human motion prediction," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 565–574. DOI: 10.1109/3DV53792.2021.00066.

[11] H. Wang, J. Dong, B. Cheng, and J. Feng, "PVRED: A Position-Velocity Recurrent Encoder-Decoder for Human Motion Prediction," *IEEE Transactions on Image Processing*, vol. 30, pp. 6096–6106, 2021, Conference Name: IEEE Transactions on Image Processing. DOI: 10.1109/TIP.2021.3089380.

[12] Y. Zhang, J. Kephart, and Q. Ji, "Incorporating physics principles for precise human motion prediction," Jan. 2024, pp. 6152–6162. DOI: 10.1109/WACV57701.2024.00605.

[13] T. Deng and Y. Sun, "Recent advances in deterministic human motion prediction: A review," *Image and Vision Computing*, vol. 143, p. 104 926, Mar. 2024. DOI: 10.1016/j.imavis.2024.104926.

[14] Q. Li, G. Chalvatzaki, J. Peters, and Y. Wang, "Directed Acyclic Graph Neural Network for Human Motion Prediction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, ISSN: 2577-087X, May 2021, pp. 3197–3204. DOI: 10.1109/ICRA48506.2021.9561540.

[15] J. Liu and J. Yin, *Multi-grained Trajectory Graph Convolutional Networks for Habit-unrelated Human Motion Prediction*, arXiv:2012.12558 [cs], Dec. 2020. DOI: 10.48550/arXiv.2012.12558.

[16] W. Zhang, S. Zhao, F. Meng, S. Wu, and M. Liu, "Dynamic Compositional Graph Convolutional Network for Efficient Composite Human Motion Prediction," in *Proceedings of the 31st ACM International Conference on Multimedia*, arXiv:2311.13781 [cs], Oct. 2023, pp. 2856–2864. DOI: 10.1145/3581783.3612532.

[17] X. Wang, Q. Cui, C. Chen, and M. Liu, *GCNext: Towards the Unity of Graph Convolutions for Human Motion Prediction*, arXiv:2312.11850 [cs], Dec. 2023. DOI: 10.48550/arXiv.2312.11850.

[18] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer, *Back to mlp: A simple baseline for human motion prediction*, 2022. DOI: 10.48550/arXiv.2207.01567.

[19] M. Ferrari, S. Sandrini, C. Tonola, E. Villagrossi, and M. Beschi, "Predicting Human Motion using the Unscented Kalman Filter for Safe and Efficient Human-Robot Collaboration," in *2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, ISSN: 1946-0759, Sep. 2024, pp. 1–8. DOI: 10.1109/ETFA61755.2024.10710736.

[20] W. Liu, X. Liang, and M. Zheng, "Dynamic Model Informed Human Motion Prediction Based on Unscented Kalman Filter," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5287–5295, Dec. 2022, Conference Name: IEEE/ASME Transactions on Mechatronics. DOI: 10.1109/TMECH.2022.3173167.

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A Skinned Multi-Person Linear Model," en, *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–16, Nov. 2015. DOI: 10.1145/2816795.2818013.

[22] S. J. Howarth and J. P. Callaghan, "Quantitative assessment of the accuracy for three interpolation techniques in kinematic analysis of human movement," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 13, no. 6, pp. 847–855, 2010, PMID: 21153975. DOI: 10.1080/10255841003664701.

[23] G. Liu, "A data-driven, piecewise linear approach to modeling human motions," Ph.D. dissertation, 2007.

[24] T. Cole and D. Altman, "Statistics notes: What is a percentage difference?" *BMJ*, vol. 358, j3663, Aug. 2017. DOI: 10.1136/bmj.j3663.

[25] Z. Sun, Z. Li, H. Wang, Z. Lin, D. He, and Z.-H. Deng, "Fast structured decoding for sequence models," 2019. DOI: https://doi.org/10.48550/arXiv.1910.11555.

[26] G. Welch and E. Foxlin, "Motion tracking: No silver bullet, but a respectable arsenal," *IEEE Computer Graphics and Applications*, vol. 22, no. 6, pp. 24–38, 2002. DOI: 10.1109/MCG.2002.1046626.

APPENDIX A
HUMAN3.6M DATASET OVERVIEW

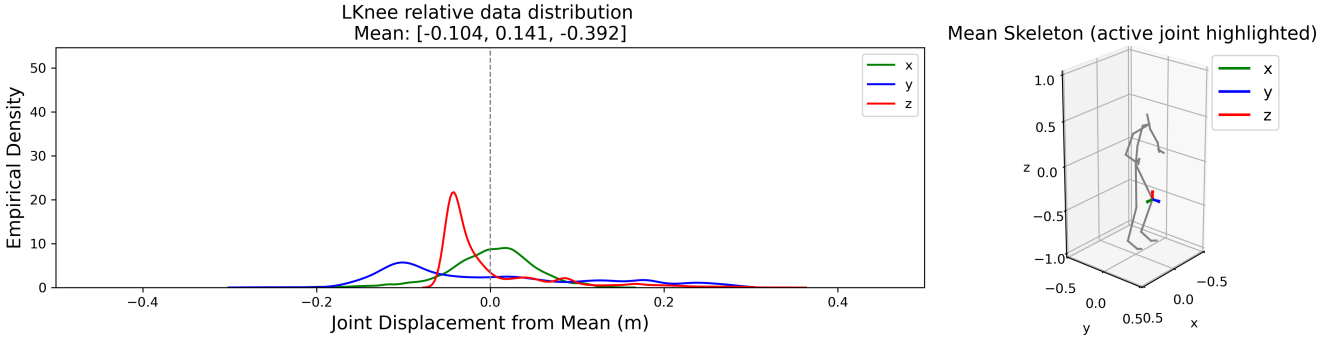


Fig. A-1: Data distribution of the LKnee joint in the Human3.6M evaluation set.

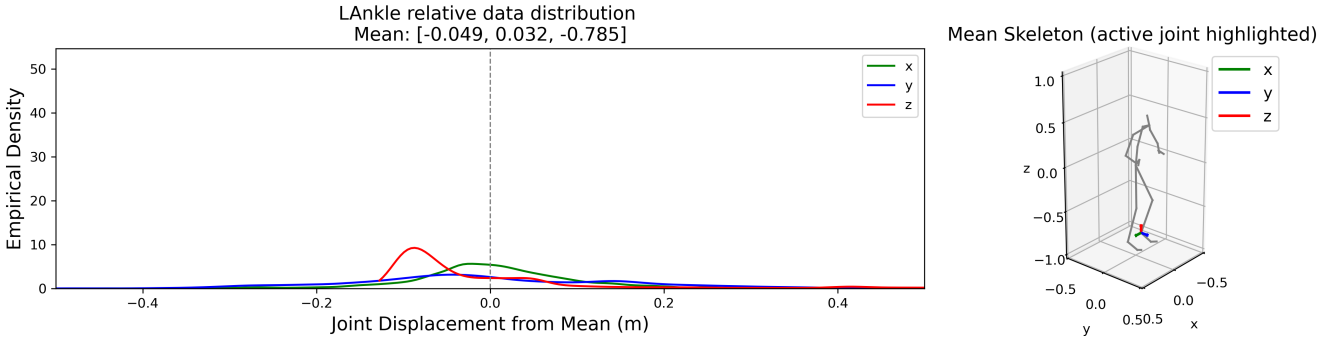


Fig. A-2: Data distribution of the LAnkle joint in the Human3.6M evaluation set.

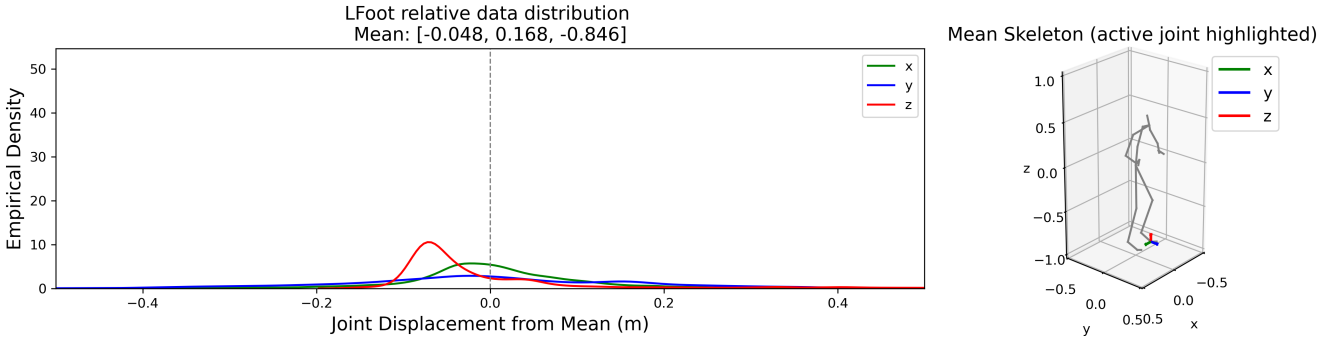


Fig. A-3: Data distribution of the LFoot joint in the Human3.6M evaluation set.

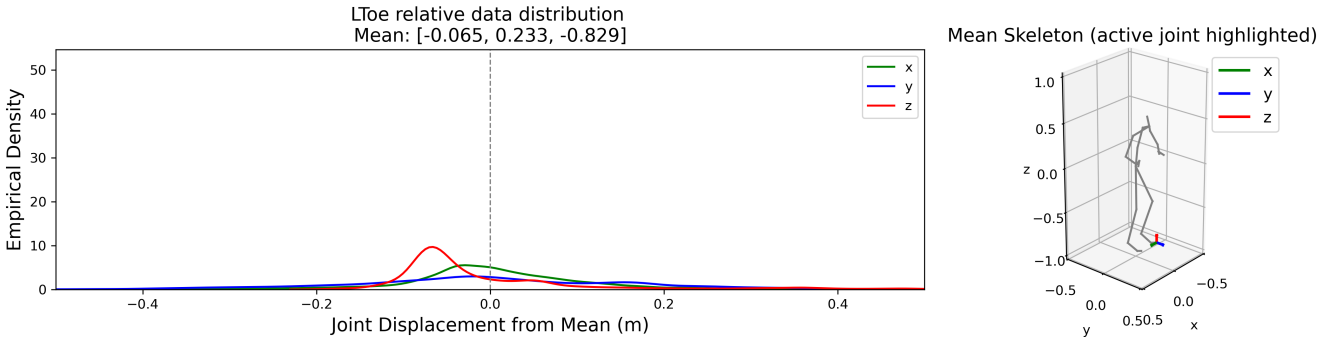


Fig. A-4: Data distribution of the LToe joint in the Human3.6M evaluation set.

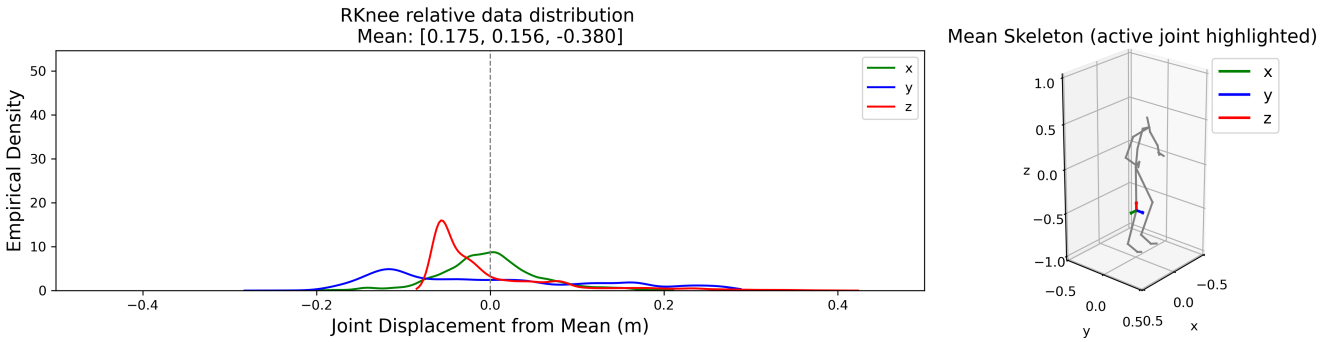


Fig. A-5: Data distribution of the RKnee joint in the Human3.6M evaluation set.

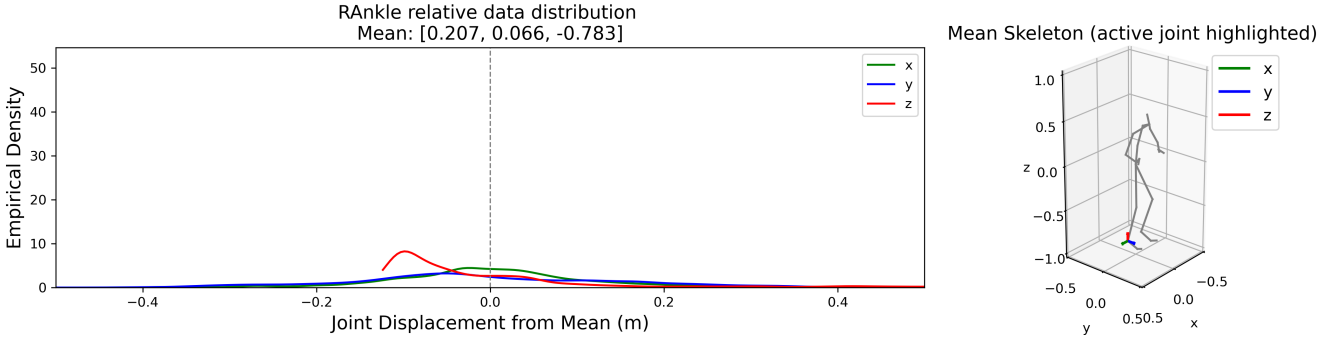


Fig. A-6: Data distribution of the RAnkle joint in the Human3.6M evaluation set.

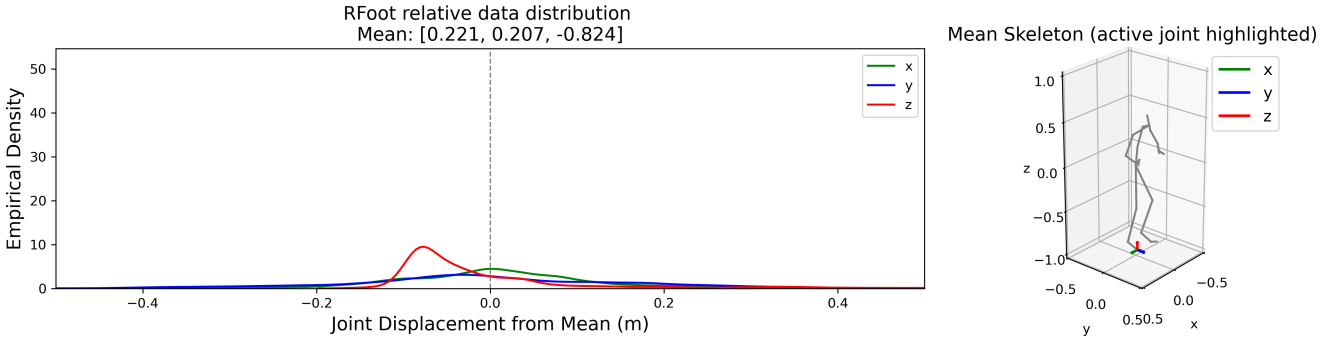


Fig. A-7: Data distribution of the RFoot joint in the Human3.6M evaluation set.

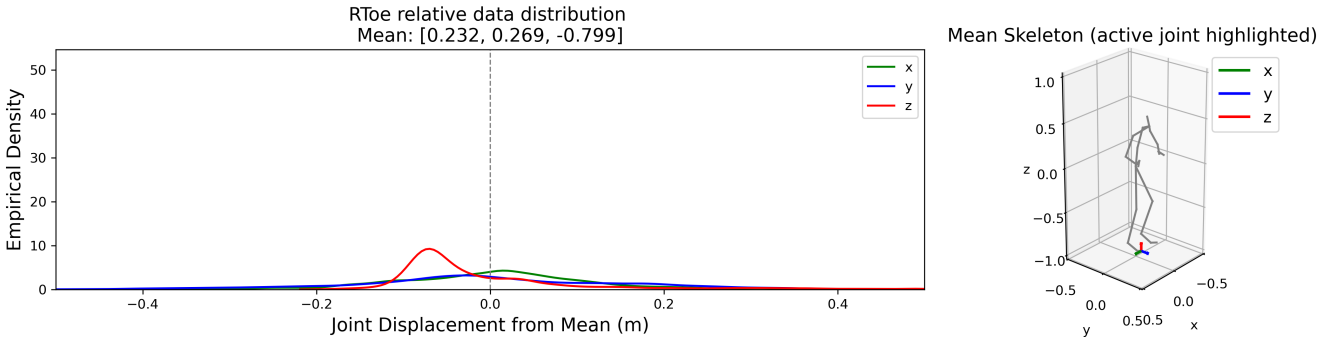


Fig. A-8: Data distribution of the RToe joint in the Human3.6M evaluation set.

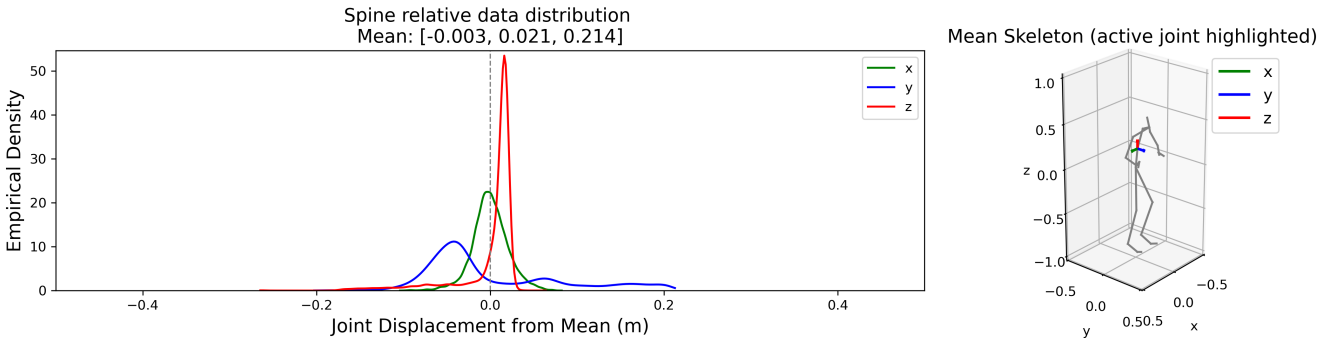


Fig. A-9: Data distribution of the Spine joint in the Human3.6M evaluation set.

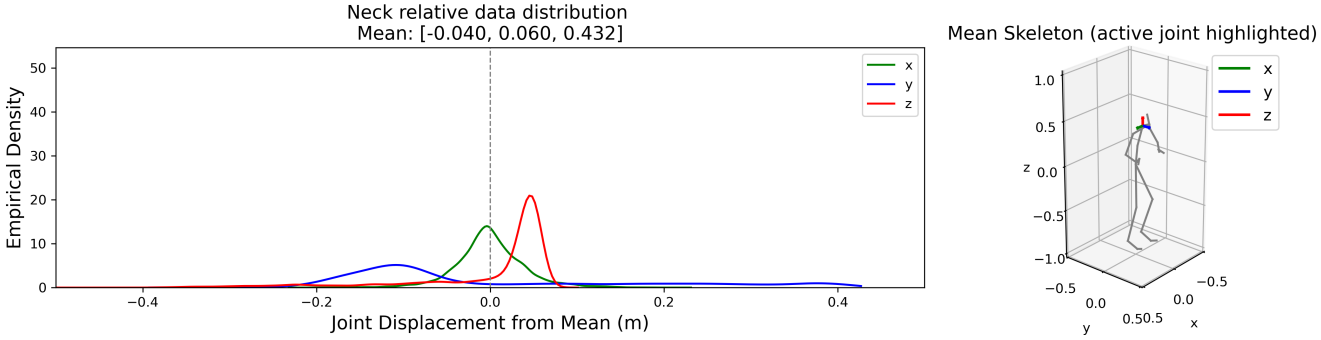


Fig. A-10: Data distribution of the Neck joint in the Human3.6M evaluation set.

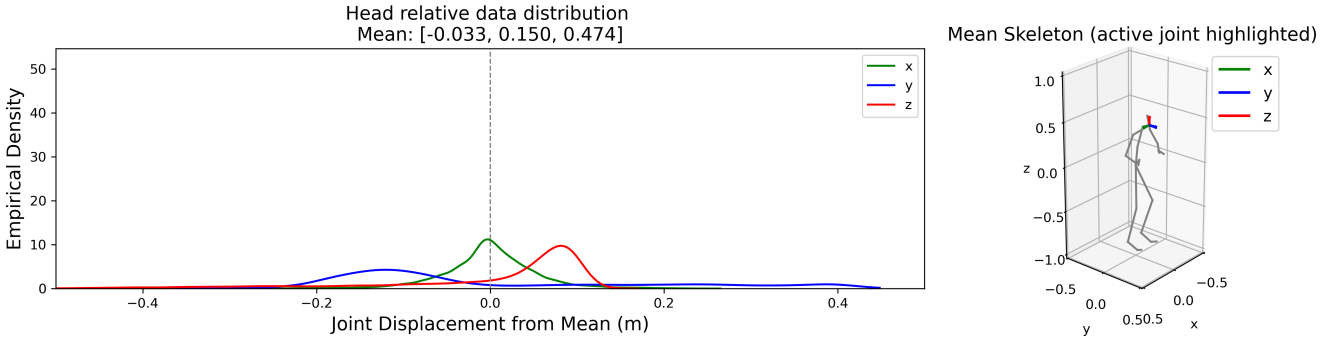


Fig. A-11: Data distribution of the Head joint in the Human3.6M evaluation set.

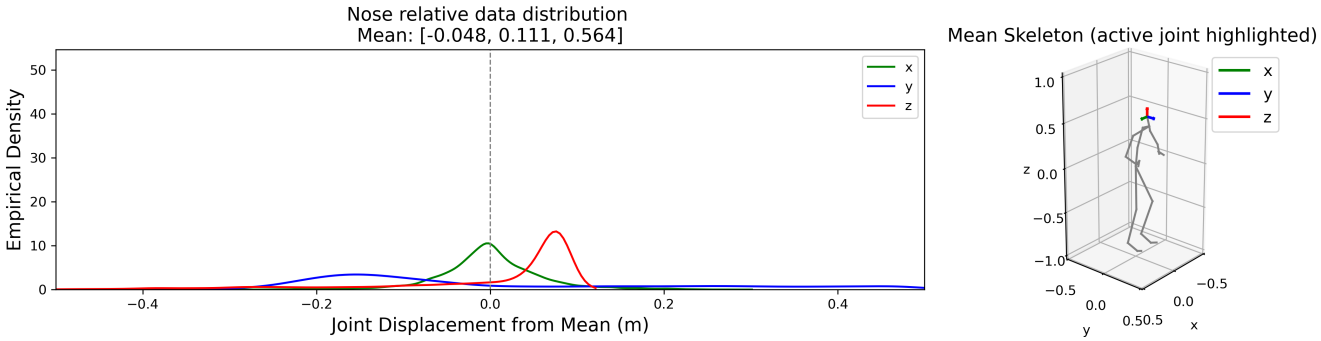


Fig. A-12: Data distribution of the Nose joint in the Human3.6M evaluation set.

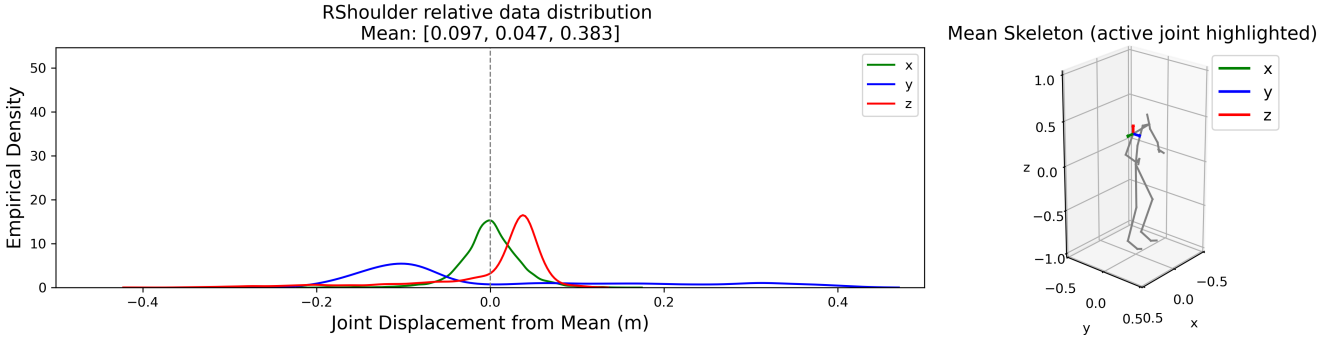


Fig. A-13: Data distribution of the RShoulder joint in the Human3.6M evaluation set.

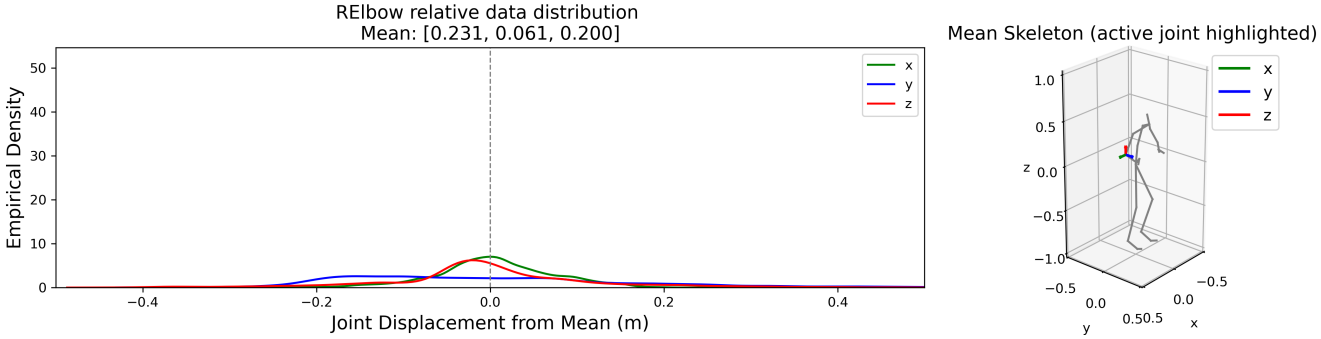


Fig. A-14: Data distribution of the RElbow joint in the Human3.6M evaluation set.

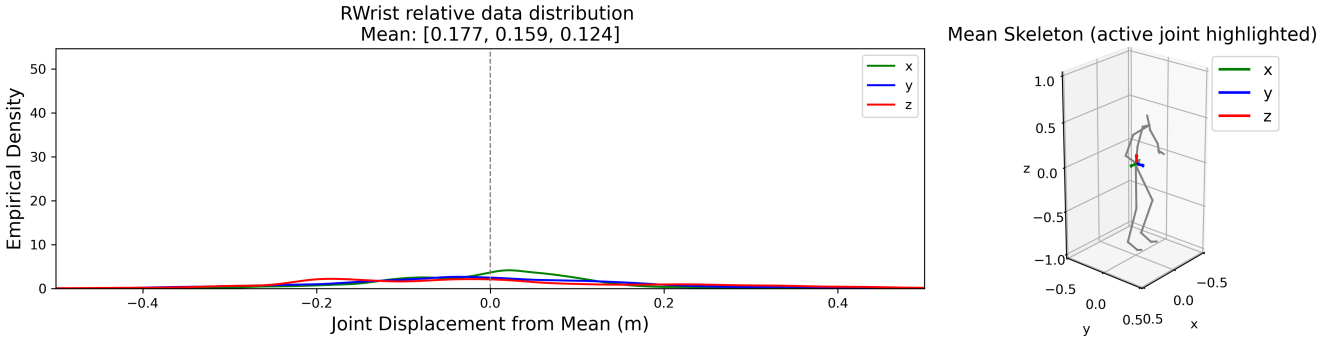


Fig. A-15: Data distribution of the RWrist joint in the Human3.6M evaluation set.

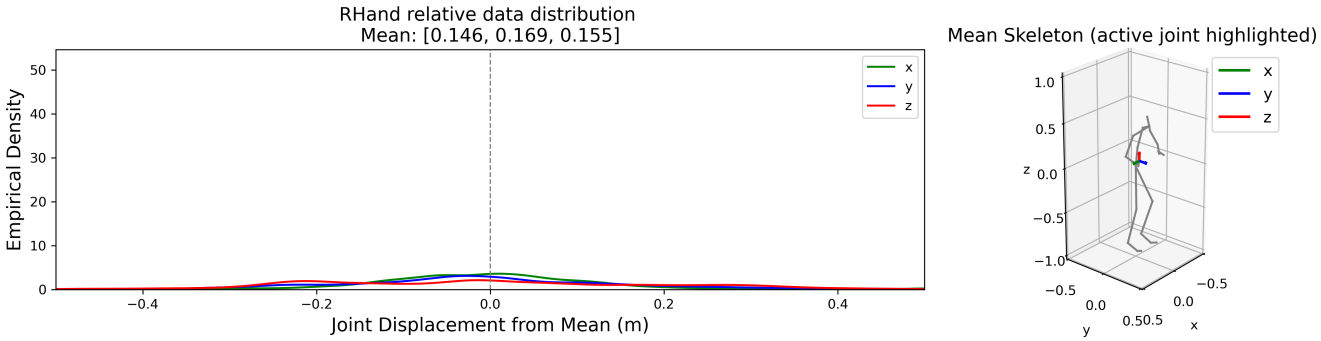


Fig. A-16: Data distribution of the RHand joint in the Human3.6M evaluation set.

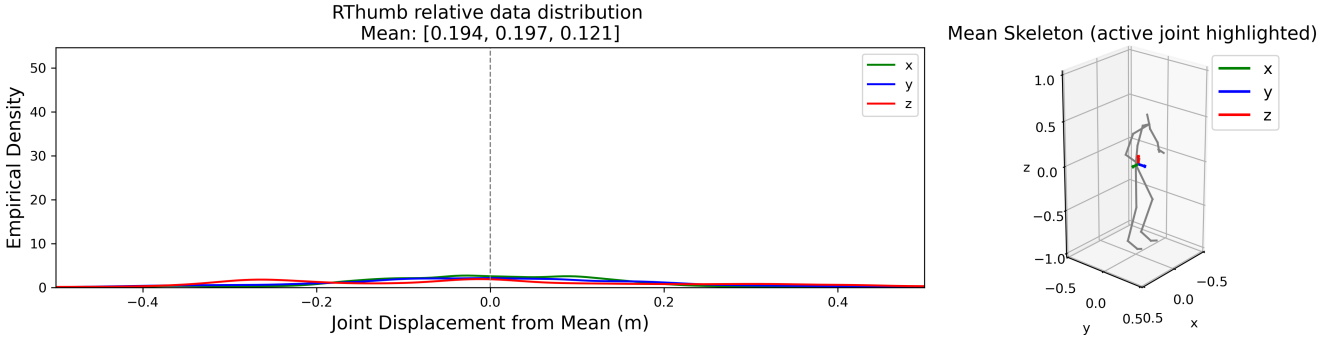


Fig. A-17: Data distribution of the RThumb joint in the Human3.6M evaluation set.

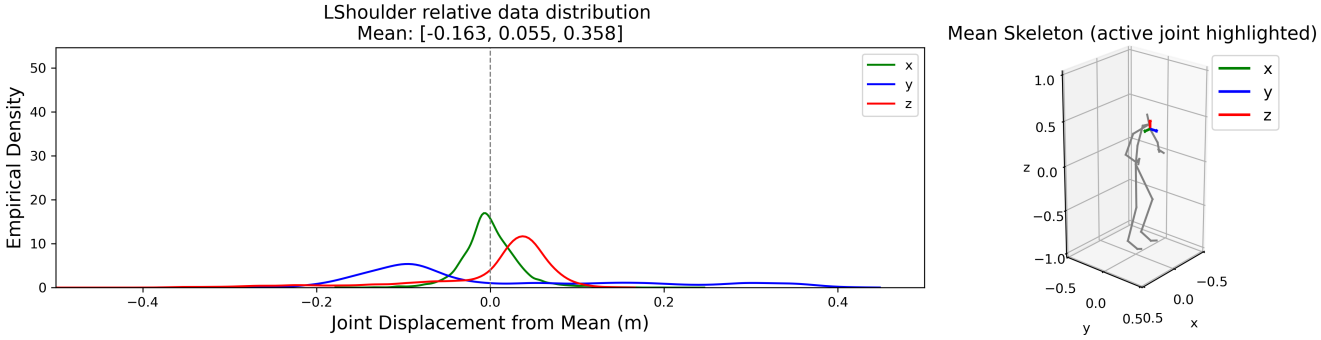


Fig. A-18: Data distribution of the LShoulder joint in the Human3.6M evaluation set.

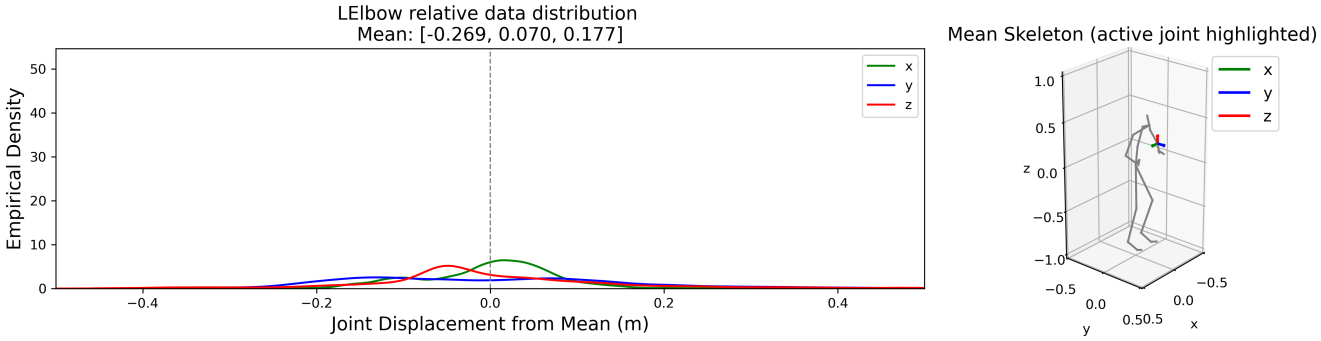


Fig. A-19: Data distribution of the LElbow joint in the Human3.6M evaluation set.

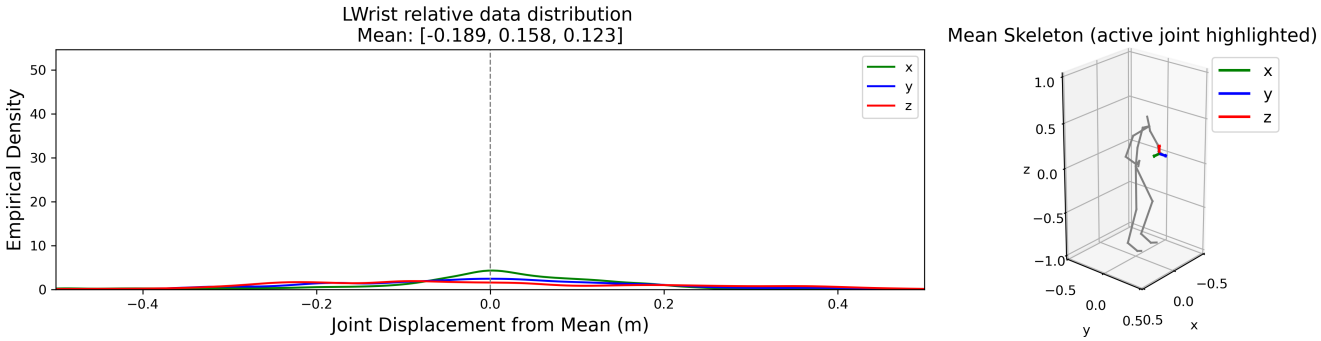


Fig. A-20: Data distribution of the LWrist joint in the Human3.6M evaluation set.

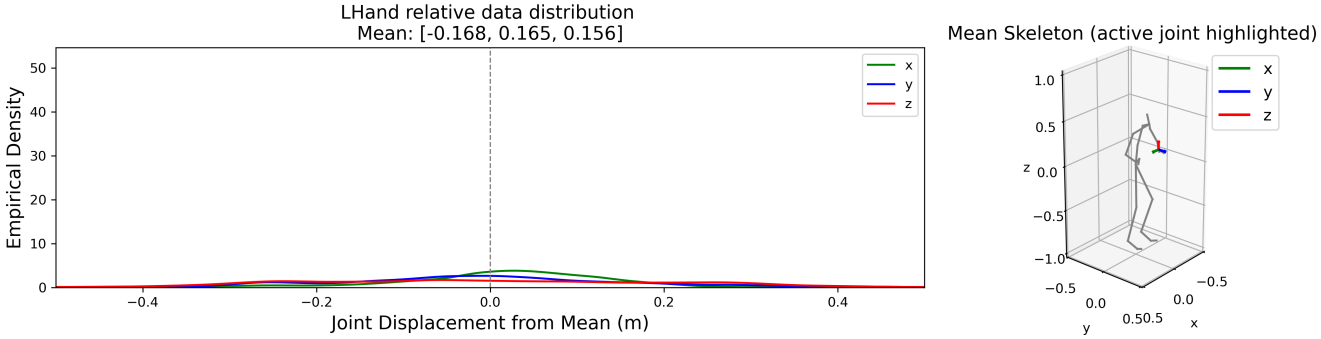


Fig. A-21: Data distribution of the LHand joint in the Human3.6M evaluation set.

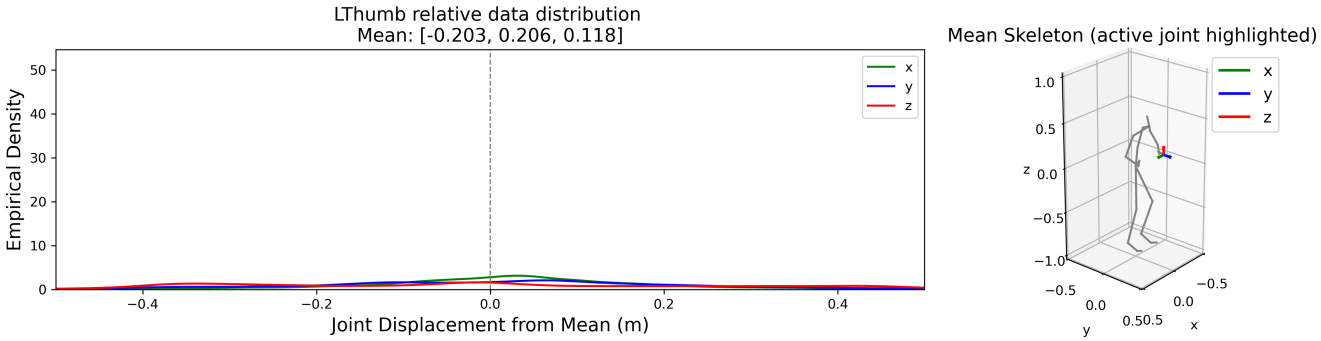


Fig. A-22: Data distribution of the LThumb joint in the Human3.6M evaluation set.

APPENDIX B
AMASS OVERVIEW

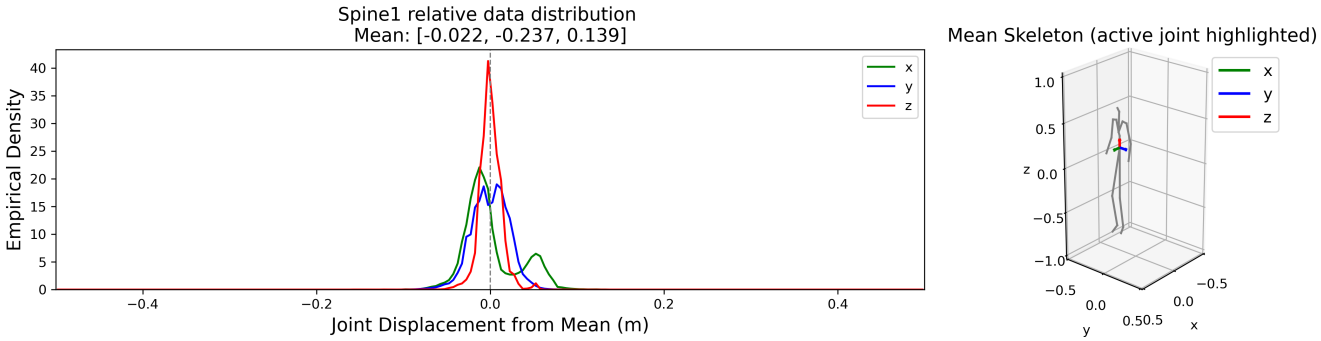


Fig. B-1: Data distribution of the Spine1 joint in the AMASS evaluation set.

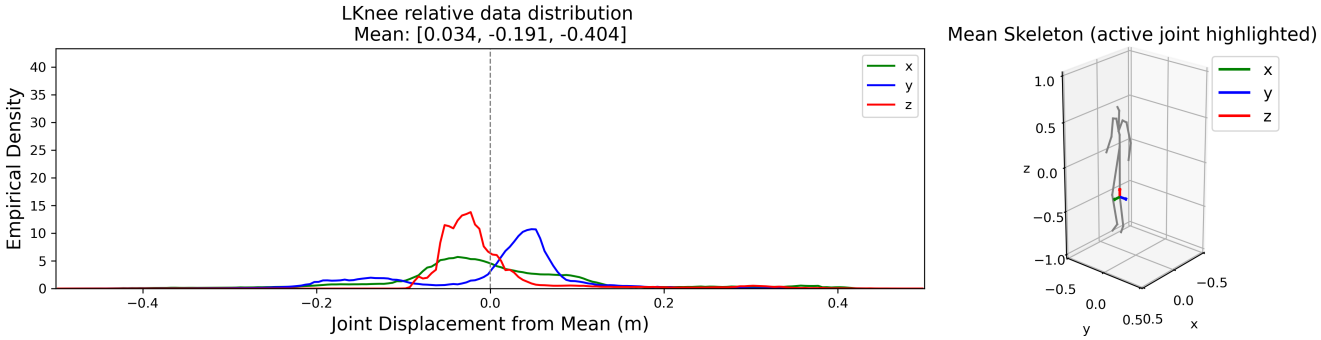


Fig. B-2: Data distribution of the LKnee joint in the AMASS evaluation set.

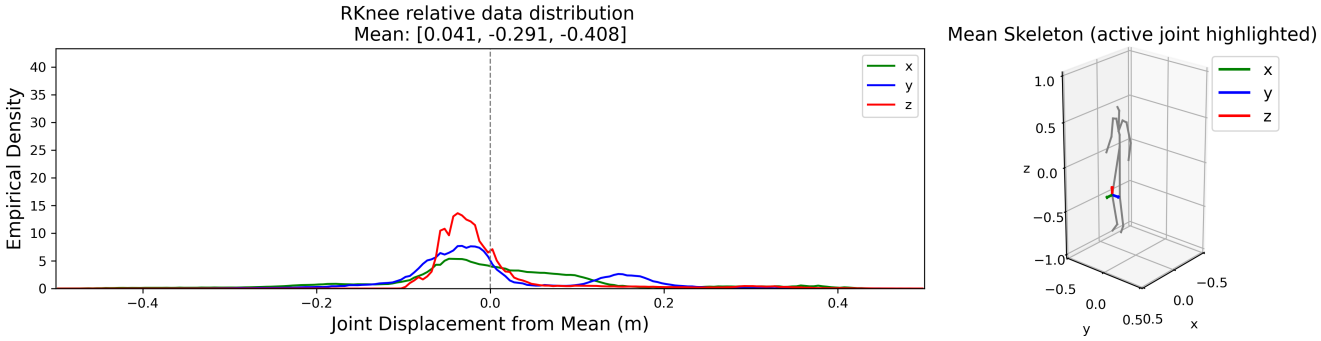


Fig. B-3: Data distribution of the RKnee joint in the AMASS evaluation set.

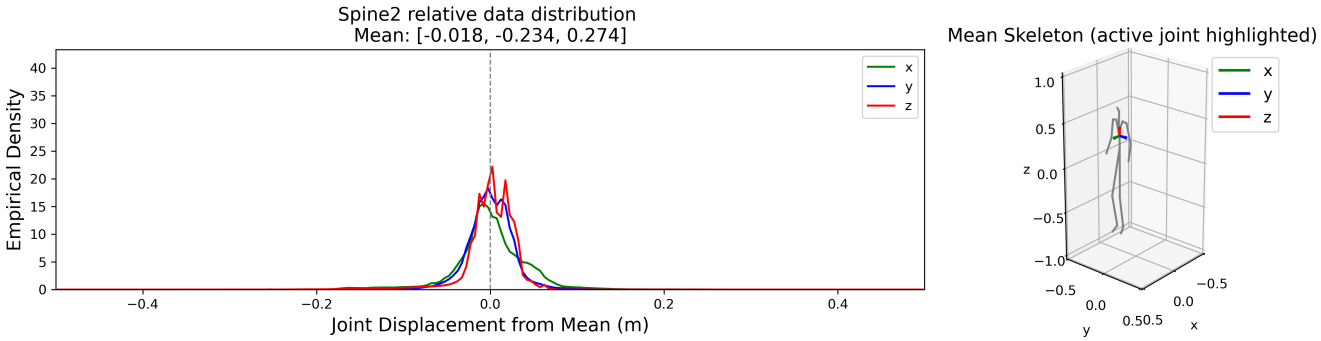


Fig. B-4: Data distribution of the Spine2 joint in the AMASS evaluation set.

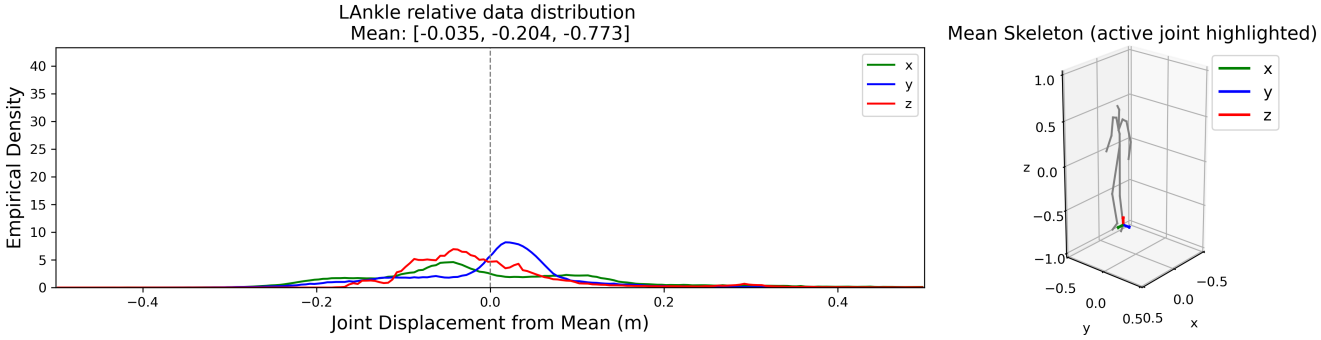


Fig. B-5: Data distribution of the LAnkle joint in the AMASS evaluation set.

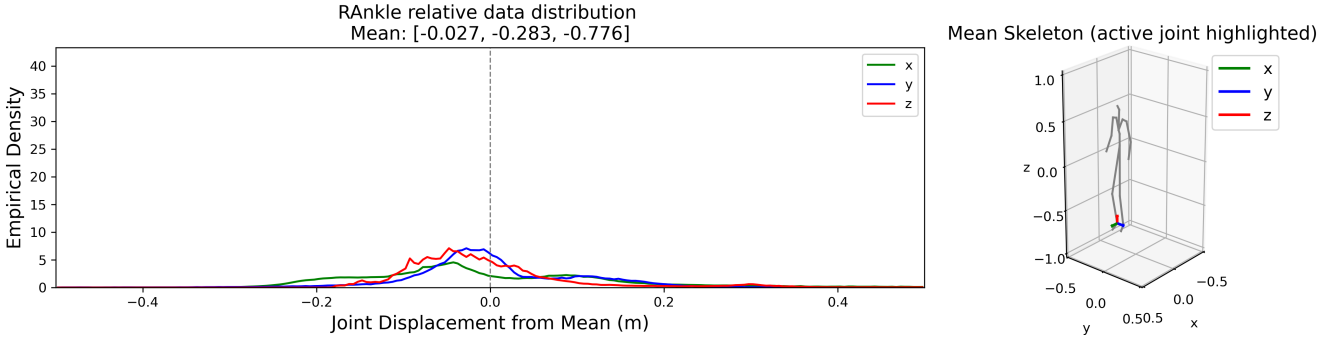


Fig. B-6: Data distribution of the RAnkle joint in the AMASS evaluation set.

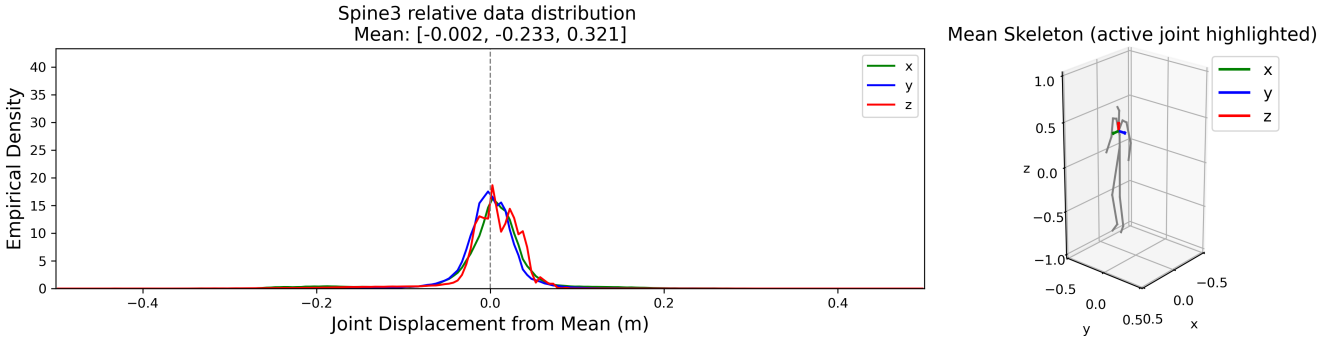


Fig. B-7: Data distribution of the Spine3 joint in the AMASS evaluation set.

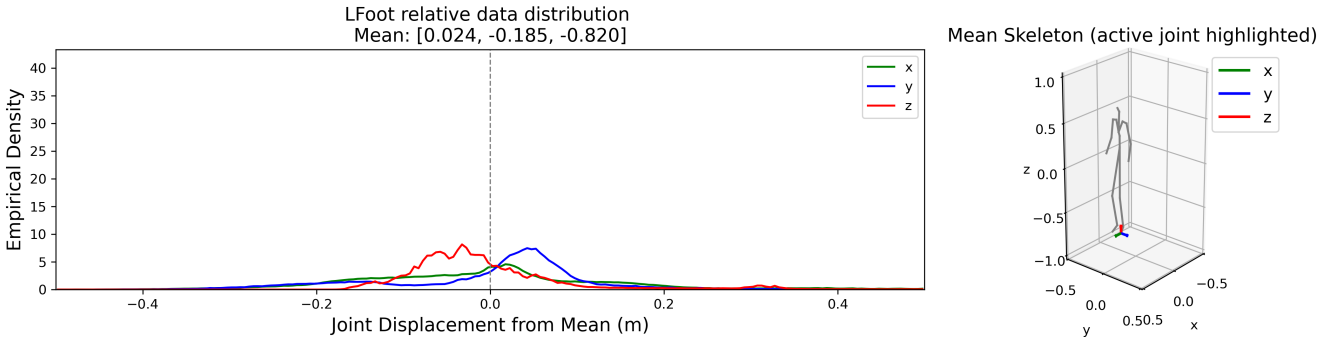


Fig. B-8: Data distribution of the LFoot joint in the AMASS evaluation set.

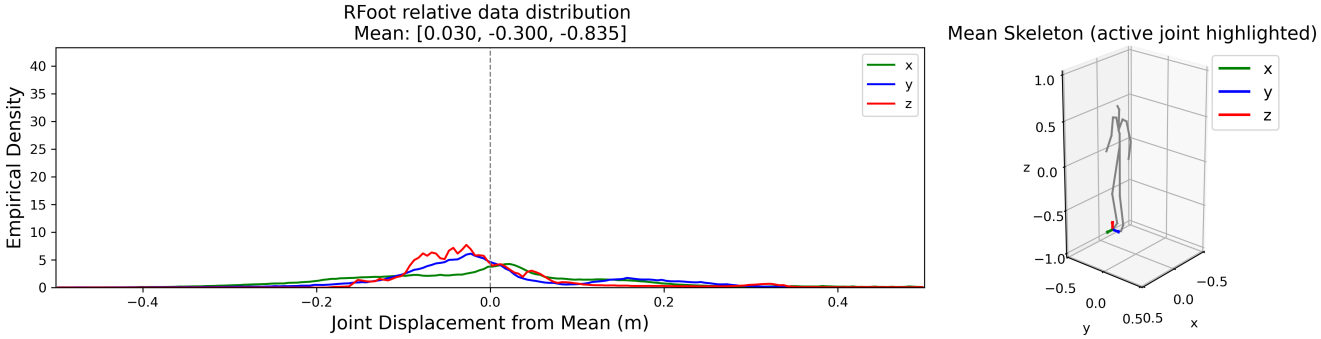


Fig. B-9: Data distribution of the RFoot joint in the AMASS evaluation set.

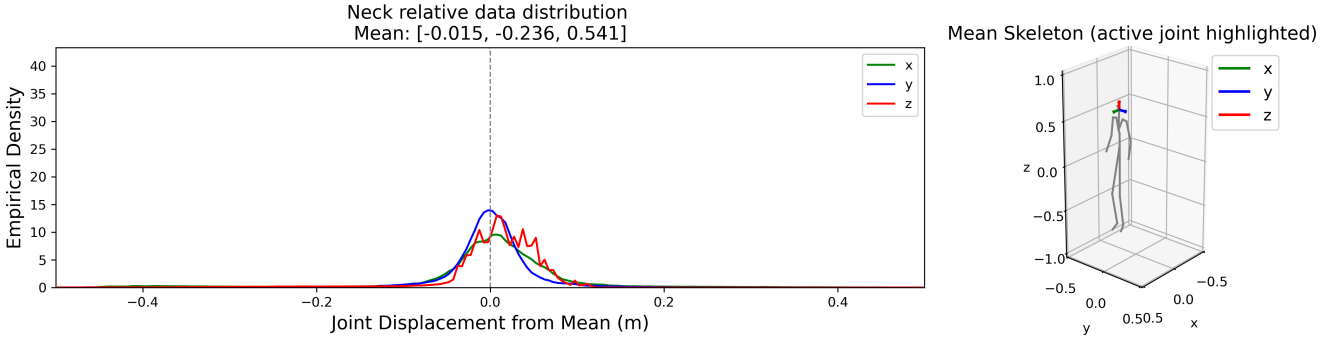


Fig. B-10: Data distribution of the Neck joint in the AMASS evaluation set.

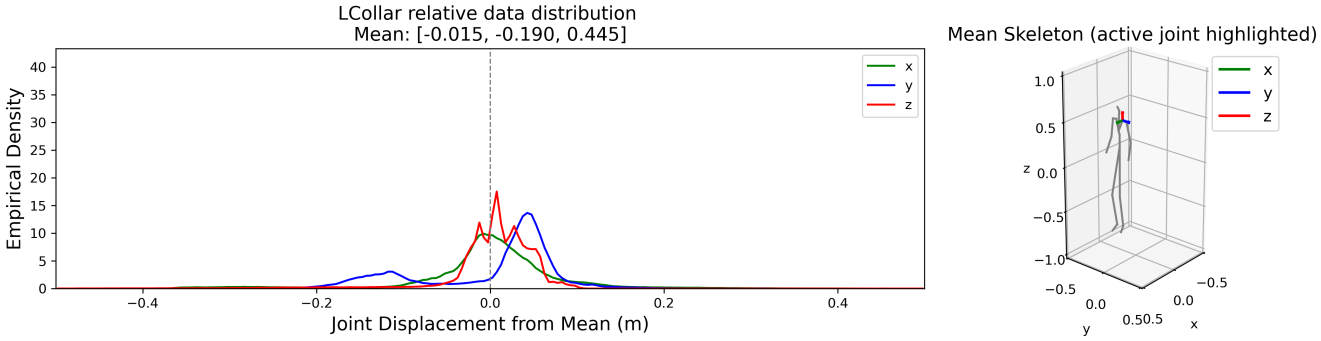


Fig. B-11: Data distribution of the LCollar joint in the AMASS evaluation set.

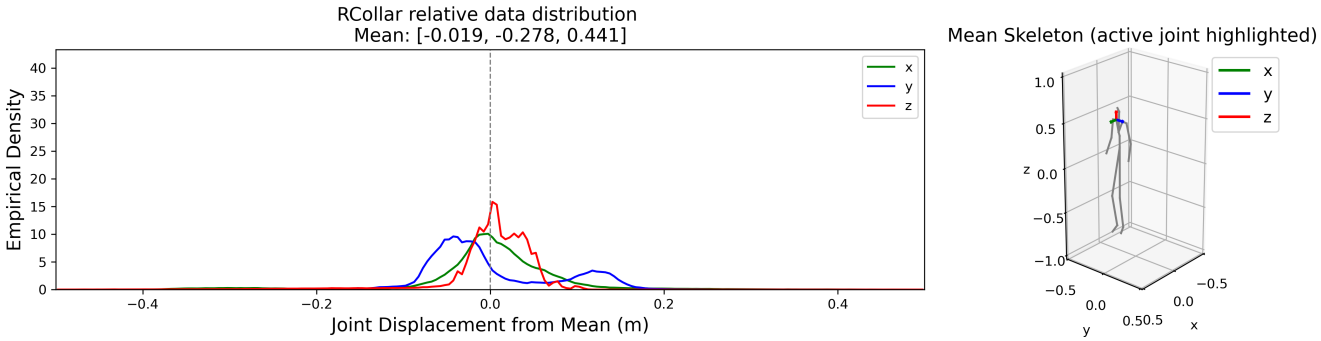


Fig. B-12: Data distribution of the RCollar joint in the AMASS evaluation set.

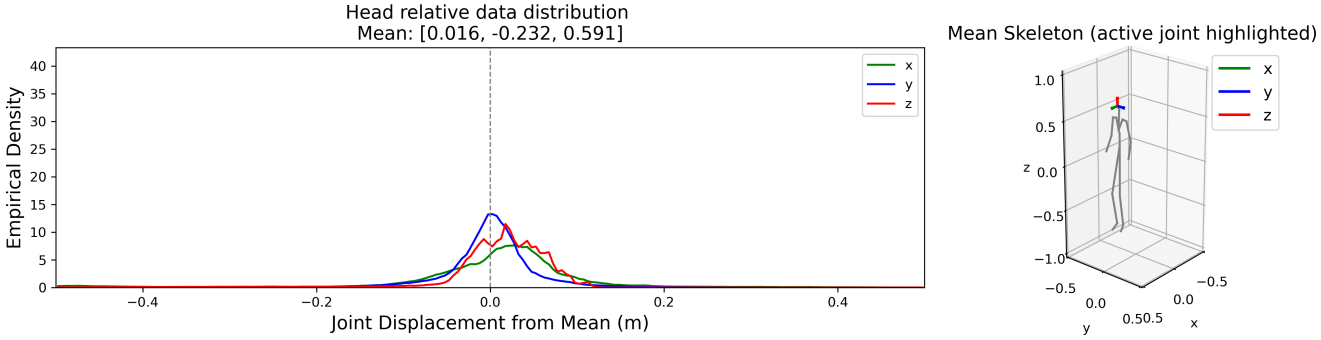


Fig. B-13: Data distribution of the Head joint in the AMASS evaluation set.

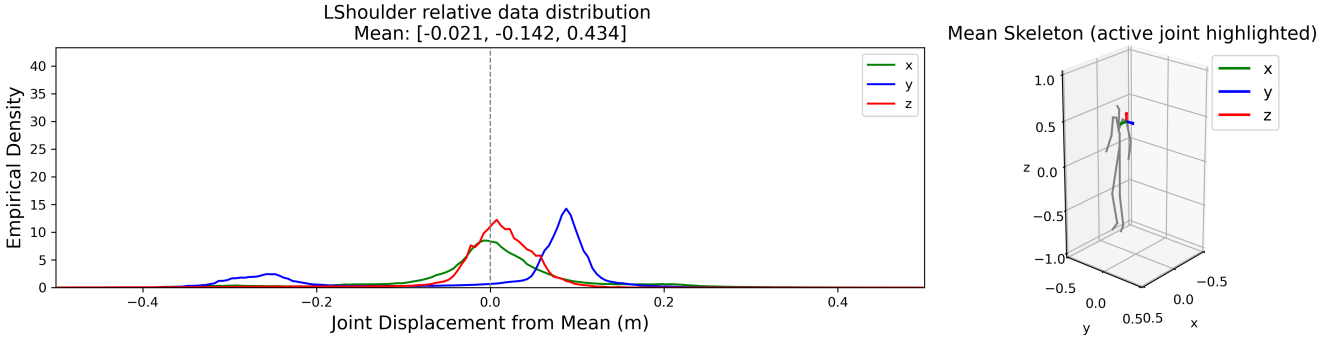


Fig. B-14: Data distribution of the LShoulder joint in the AMASS evaluation set.

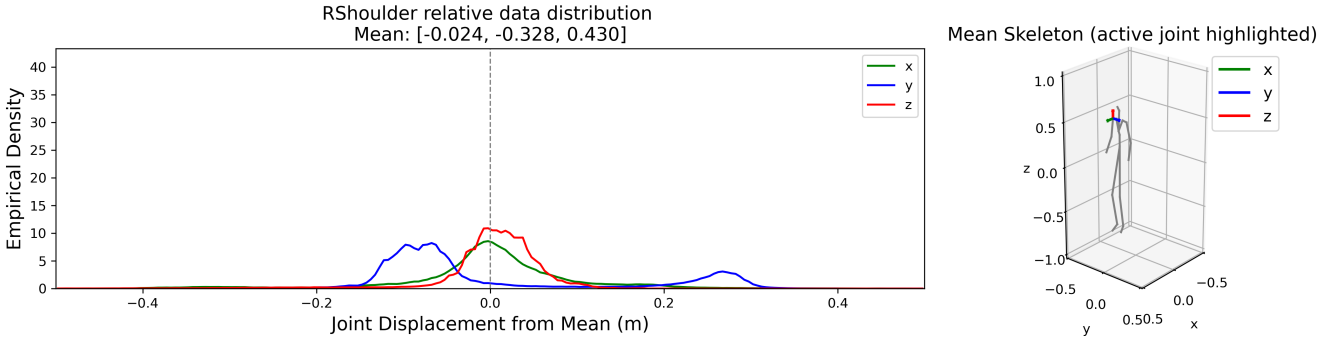


Fig. B-15: Data distribution of the RShoulder joint in the AMASS evaluation set.

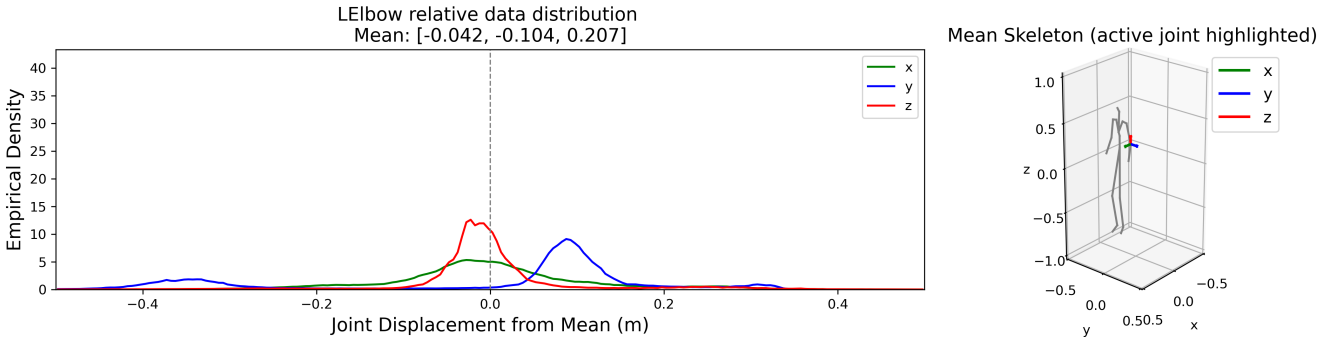


Fig. B-16: Data distribution of the LElbow joint in the AMASS evaluation set.

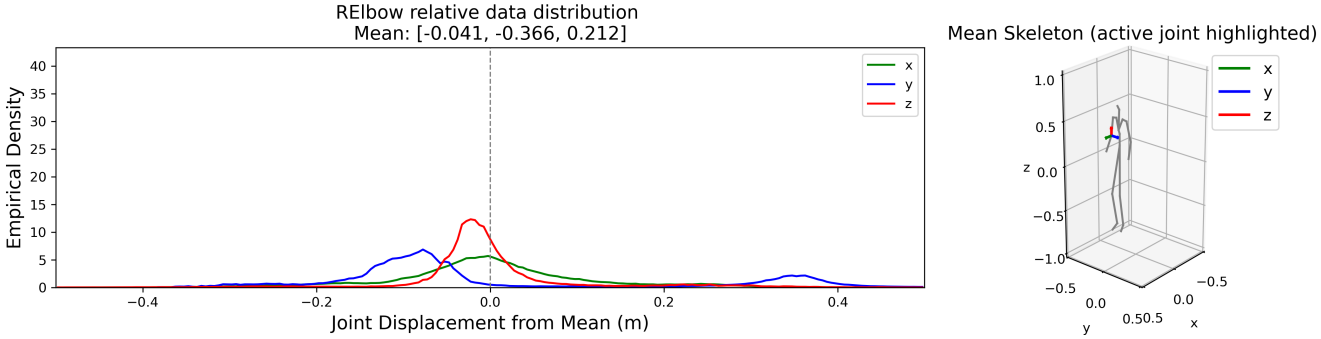


Fig. B-17: Data distribution of the RElbow joint in the AMASS evaluation set.

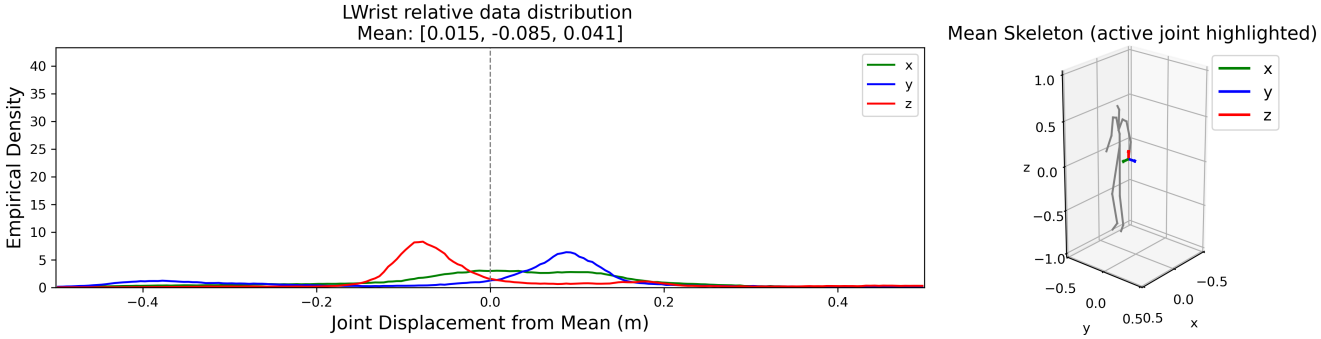


Fig. B-18: Data distribution of the LWrist joint in the AMASS evaluation set.

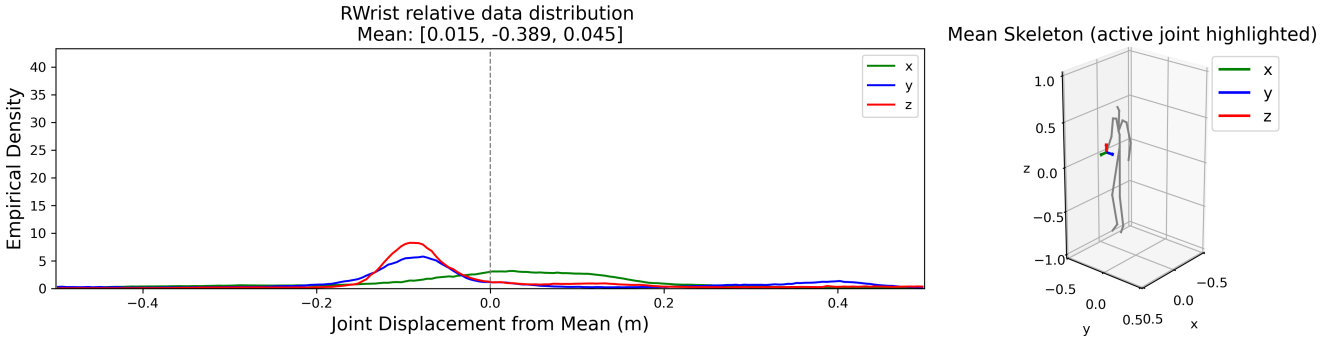


Fig. B-19: Data distribution of the RWrist joint in the AMASS evaluation set.

APPENDIX C
MODEL OVERVIEW

TABLE C-I: Overview of evaluated models, their datasets, and temporal configurations.

Model / Branch	**Type**	**Dataset**	**Input history (frames)**	**Prediction horizon (frames)**	**Sampling frequency (Hz)**
GCNext	Data-driven (Graph Convolutional Network)	Human3.6M	50	25	25
PhysMoP – Physics branch	Physics-based	AMASS	3	25	25
PhysMoP – Data branch	Data-driven (MLP)	AMASS	25	25	25
PhysMoP – Fusion branch (not evaluated)	Hybrid (weighted average)	AMASS	25	25	25

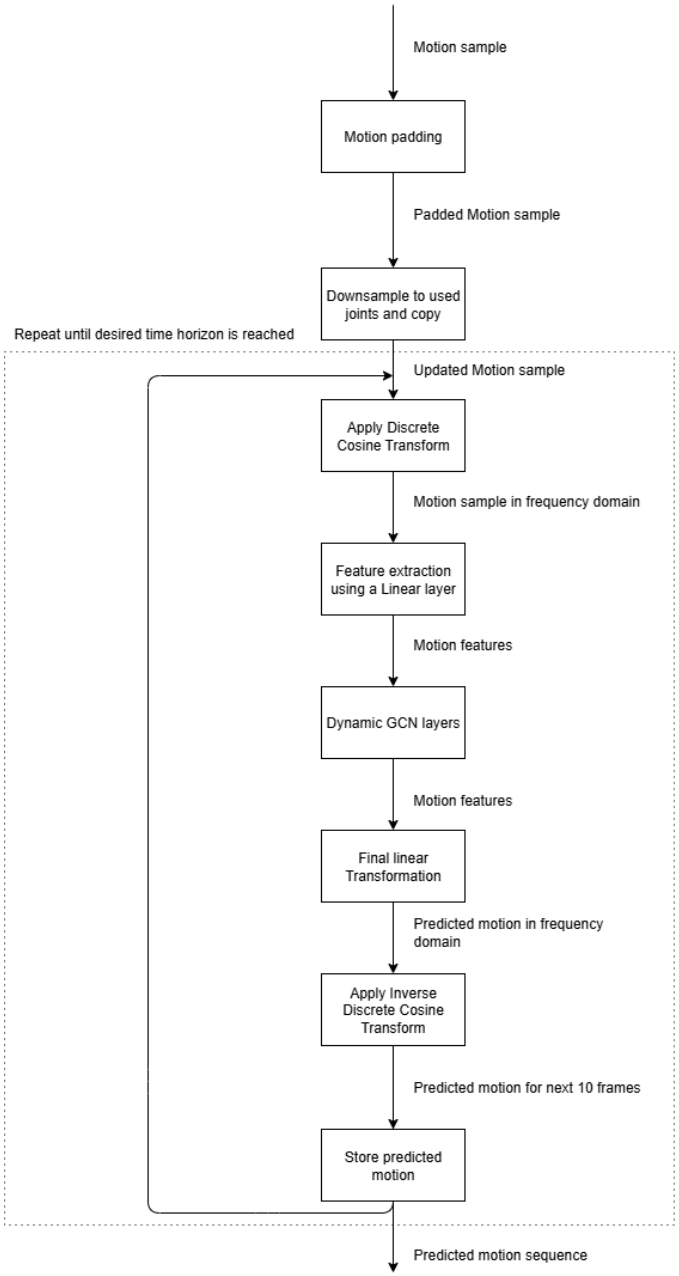


Fig. C-1: Overview of the GCNext model architecture

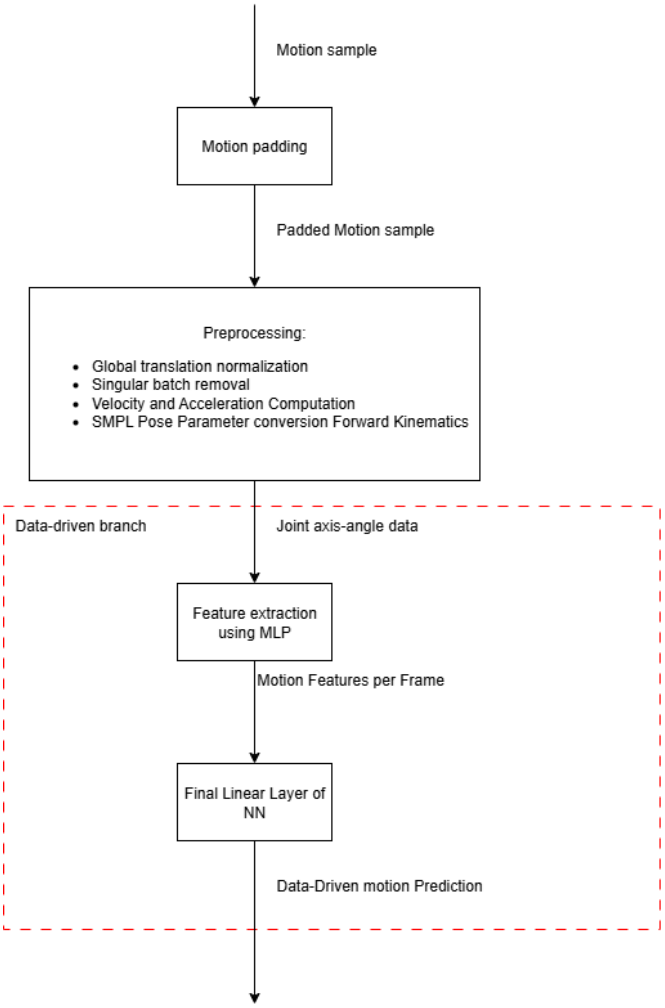


Fig. C-2: Overview of the PhysMoP data branch architecture

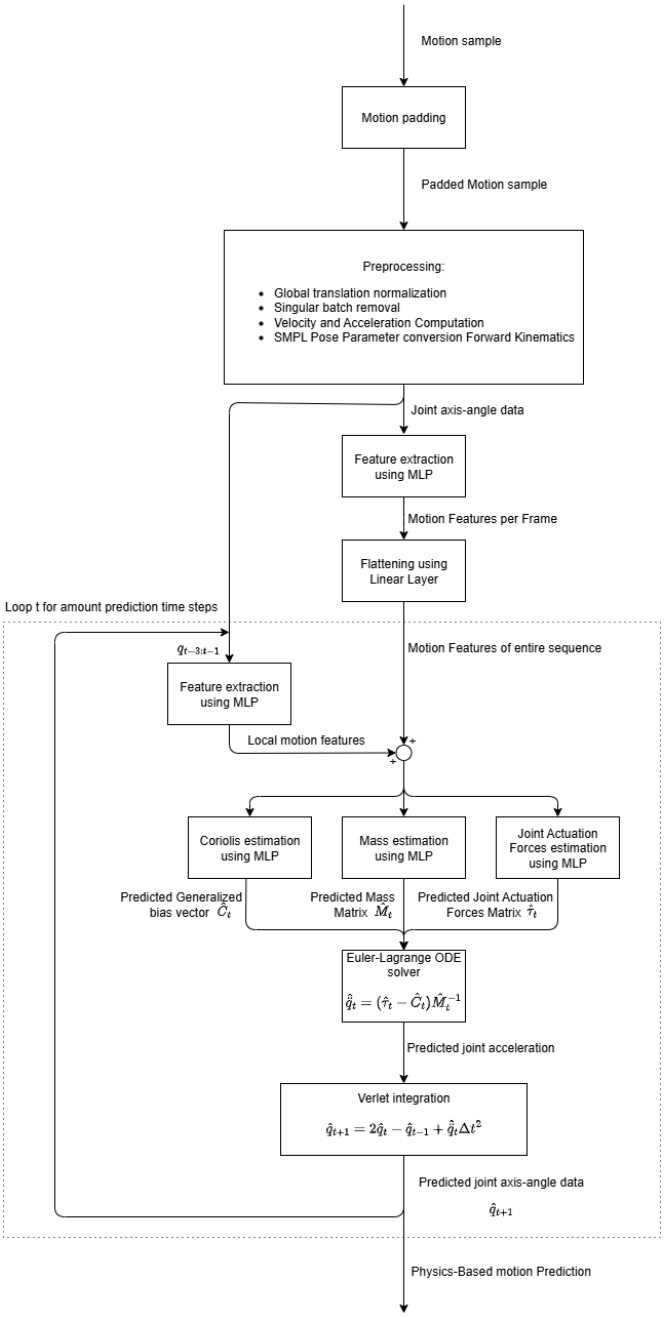


Fig. C-3: Overview of the PhysMoP physics branch architecture

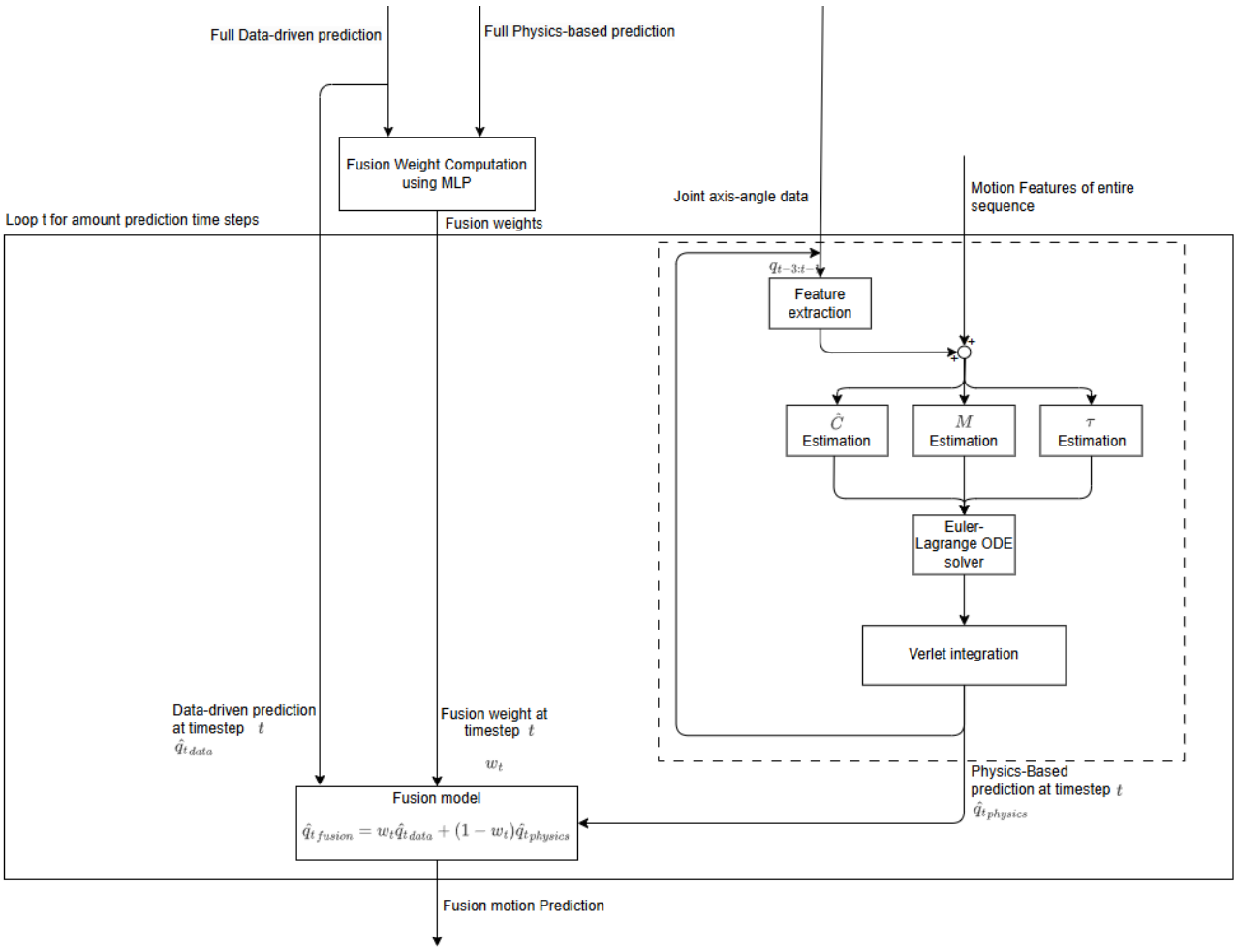
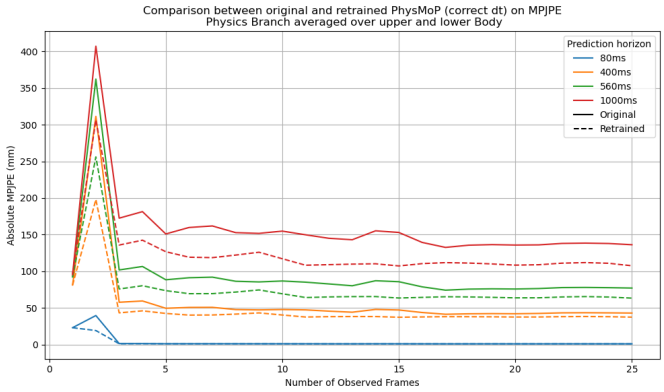
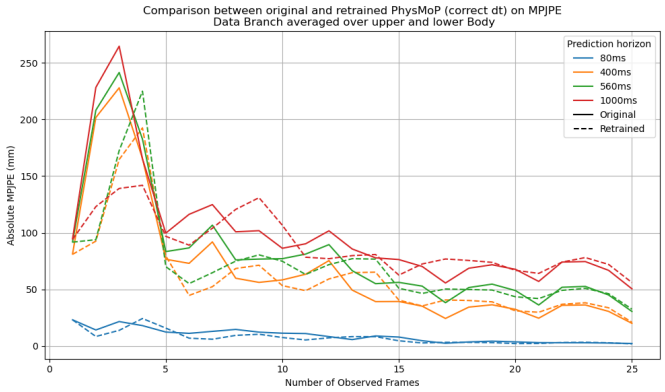


Fig. C-4: Overview of the PhysMoP fusion branch architecture

APPENDIX D
PHYSMOP CONSISTENCY CHECK



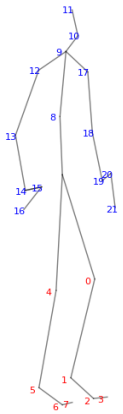
(a) PhysMoP physics branch



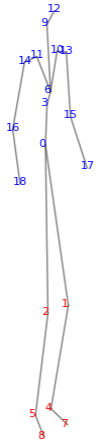
(b) PhysMoP data branch

Fig. D-1: Comparison between the original and retrained PhysMoP models using a consistent time step Δt and frequency.

APPENDIX E
SKELETON STRUCTURES AND JOINT MAPPING



(a) Human3.6M skeleton with joint indices.



(b) AMASS/SMPL skeleton with joint indices.

Lower body		Upper body	
Idx	**Joint name**	**Idx**	**Joint name**
0	LKnee	8	Spine
1	LAnkle	9	Neck
2	LFoot	10	Nose
3	LToe	11	Head
4	RKnee	12	RShoulder
5	RAnkle	13	RElbow
6	RFoot	14	RWrist
7	RToe	15	RHand
		16	RThumb
		17	LShoulder
		18	LElbow
		19	LWrist
		20	LHand
		21	LThumb

Lower body		Upper body	
Idx	**Joint name**	**Idx**	**Joint name**
1	LKnee	0	Spine1
2	RKnee	3	Spine2
4	LAnkle	6	Spine3
5	RAnkle	9	Neck
7	LFoot	10	LCollarbone
8	RFoot	11	RCollarbone
		12	Head
		13	LShoulder
		14	RShoulder
		15	LElbow
		16	RElbow
		17	LWrist
		18	RWrist

Fig. E-1: Skeletons and joint lists used in this work. Each table lists the index–name mapping and the upper/lower-body assignment used for region-wise MPJPE.

TABLE E-I: Mapping from Human3.6M to AMASS joints used to align PhysMoP outputs with GCNext input format.

H36M Idx	H36M Joint	AMASS Idx	AMASS Joint	Notes	H36M Idx	H36M Joint	AMASS Idx	AMASS Joint	Notes
0	LKnee	1	LKnee		11	Head	12	Head	
1	LAnkle	4	LAnkle		12	RShoulder	14	RShoulder	
2	LFoot	7	LFoot		13	RElbow	16	RElbow	
3	LToe	7	LFoot (reused)	No toe joint in AMASS	14	RWrist	18	RWrist	
4	RKnee	2	RKnee		15	RHand	18	RWrist (reused)	No explicit hand joint
5	RAnkle	5	RAnkle		16	RThumb	18	RWrist (reused)	
6	RFoot	8	RFoot		17	LShoulder	13	LShoulder	
7	RToe	8	RFoot (reused)	No toe joint in AMASS	18	LElbow	15	LElbow	
8	Spine	3	Spine2		19	LWrist	17	LWrist	
9	Neck	9	Neck		20	LHand	17	LWrist (reused)	
10	Nose	12	Head (approx.)	AMASS has no face joints	21	LThumb	17	LWrist (reused)	

APPENDIX F
HUMAN3.6M WALKING SUBSET

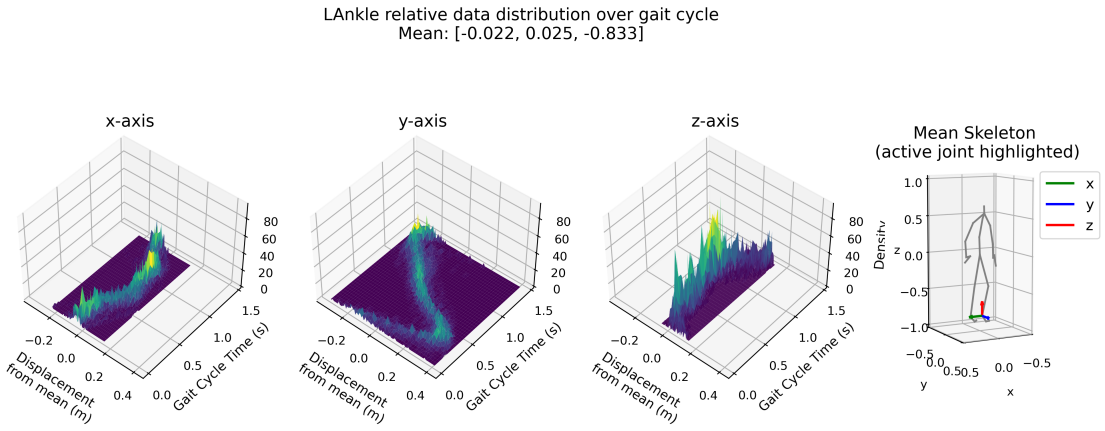


Fig. F-1: Temporal evolution of the LAnkle joint coordinate distributions over the gait cycle.

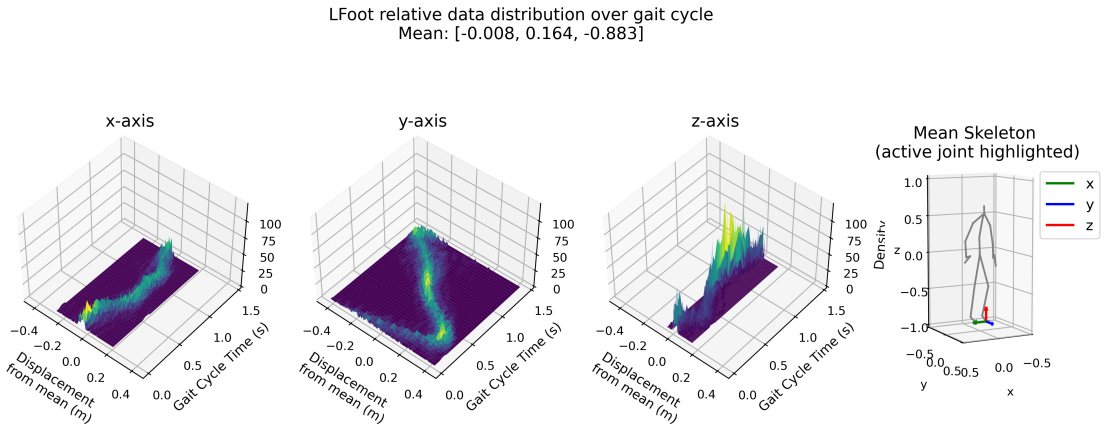


Fig. F-2: Temporal evolution of the LFoot joint coordinate distributions over the gait cycle.

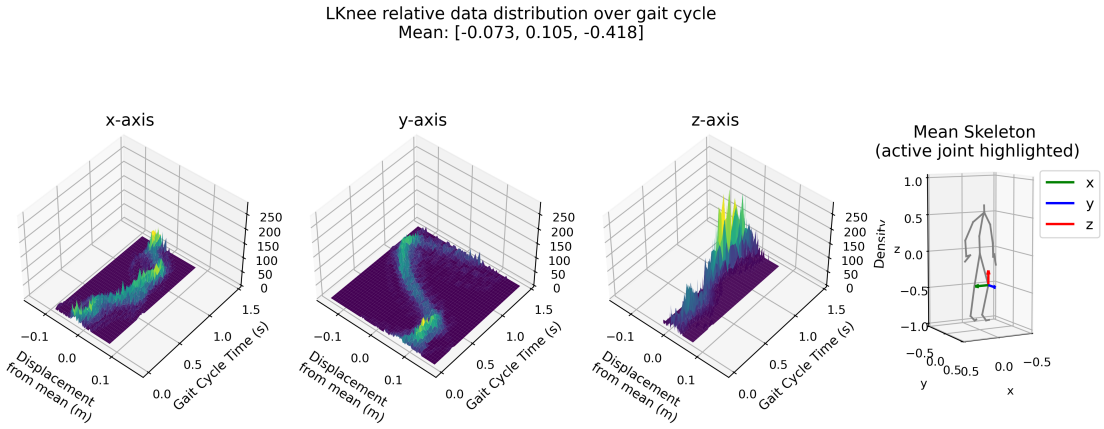


Fig. F-3: Temporal evolution of the LKnee joint coordinate distributions over the gait cycle.

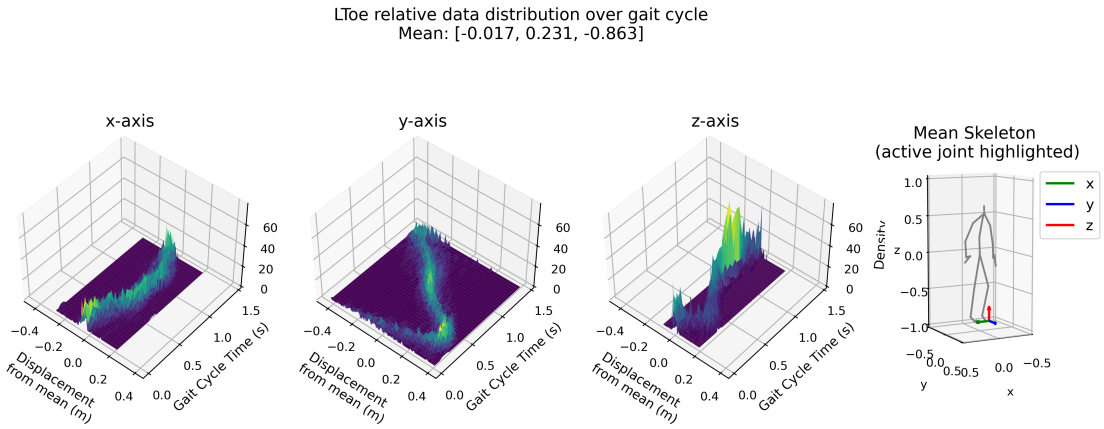


Fig. F-4: Temporal evolution of the LToe joint coordinate distributions over the gait cycle.

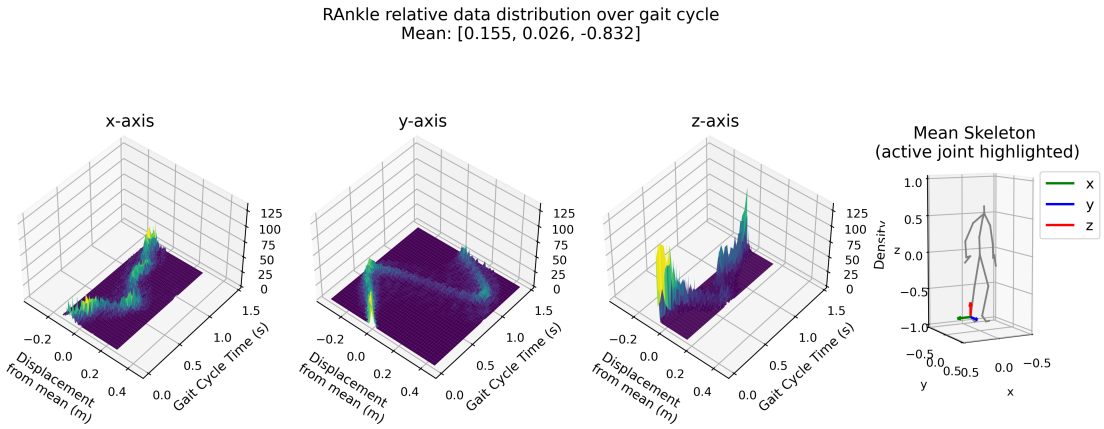


Fig. F-5: Temporal evolution of the RAnkle joint coordinate distributions over the gait cycle.

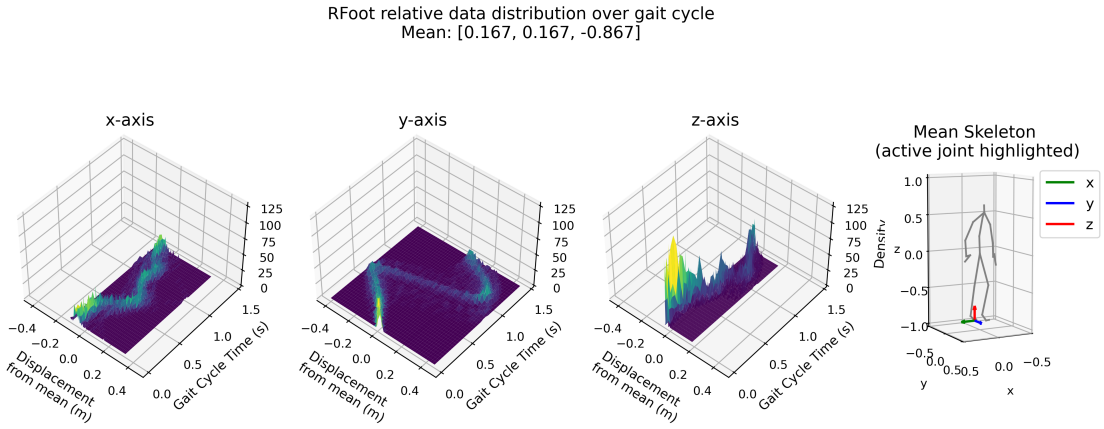


Fig. F-6: Temporal evolution of the RFoot joint coordinate distributions over the gait cycle.

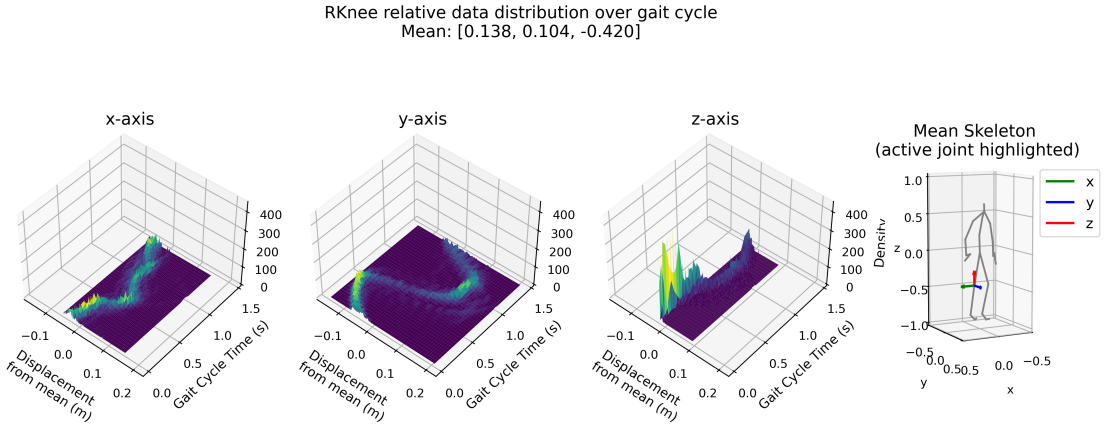


Fig. F-7: Temporal evolution of the RKnee joint coordinate distributions over the gait cycle.

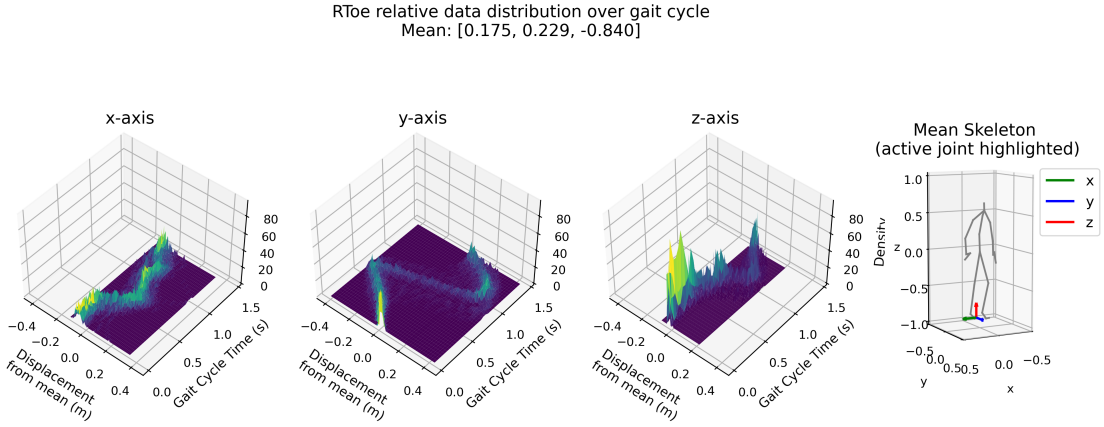


Fig. F-8: Temporal evolution of the RToe joint coordinate distributions over the gait cycle.

APPENDIX G
AMASS WALKING SUBSET

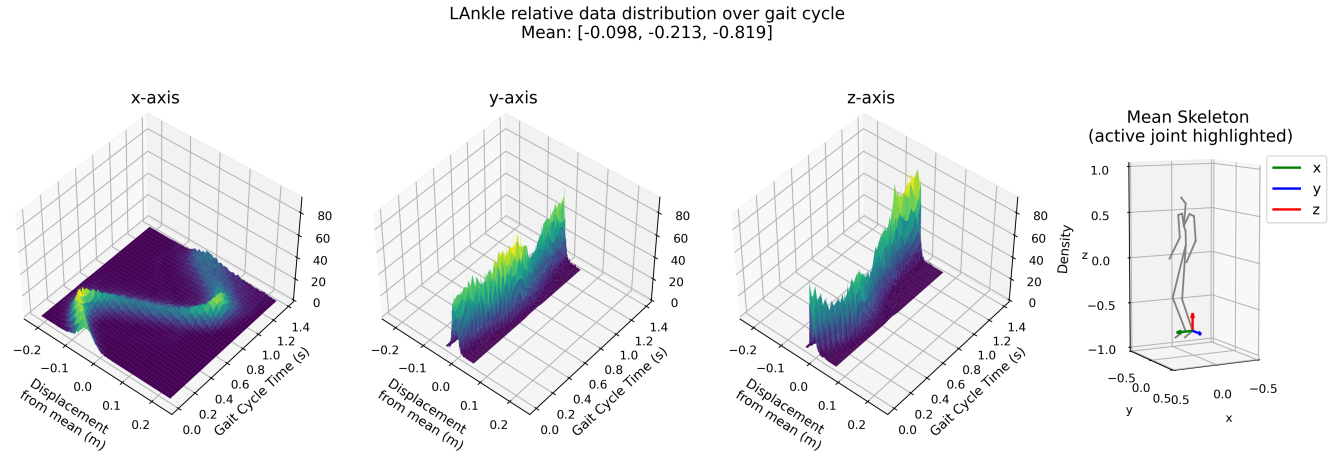


Fig. G-1: Temporal evolution of the LAnkle joint coordinate distributions over the gait cycle.

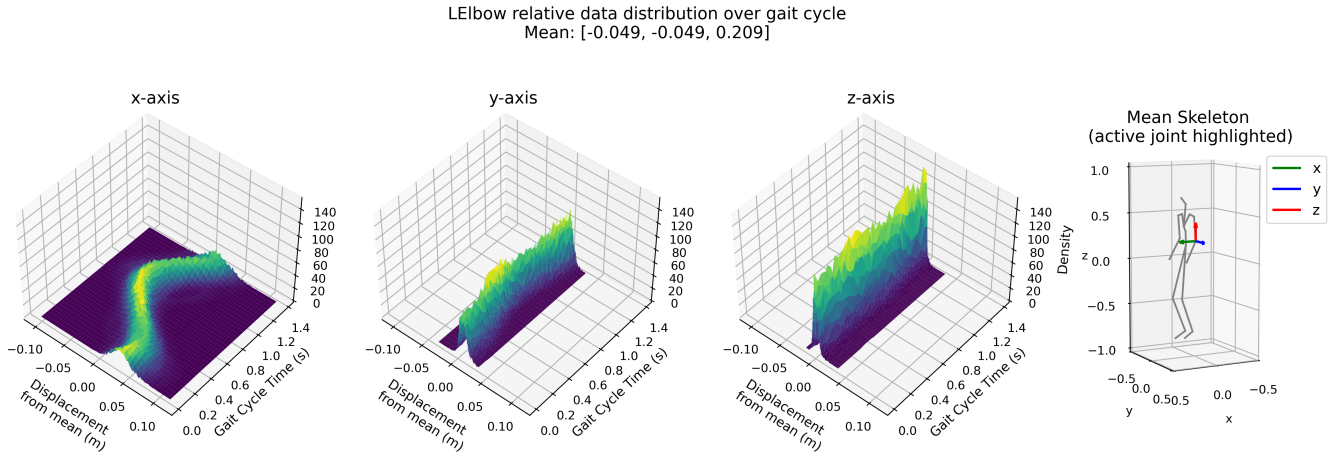


Fig. G-2: Temporal evolution of the LElbow joint coordinate distributions over the gait cycle.

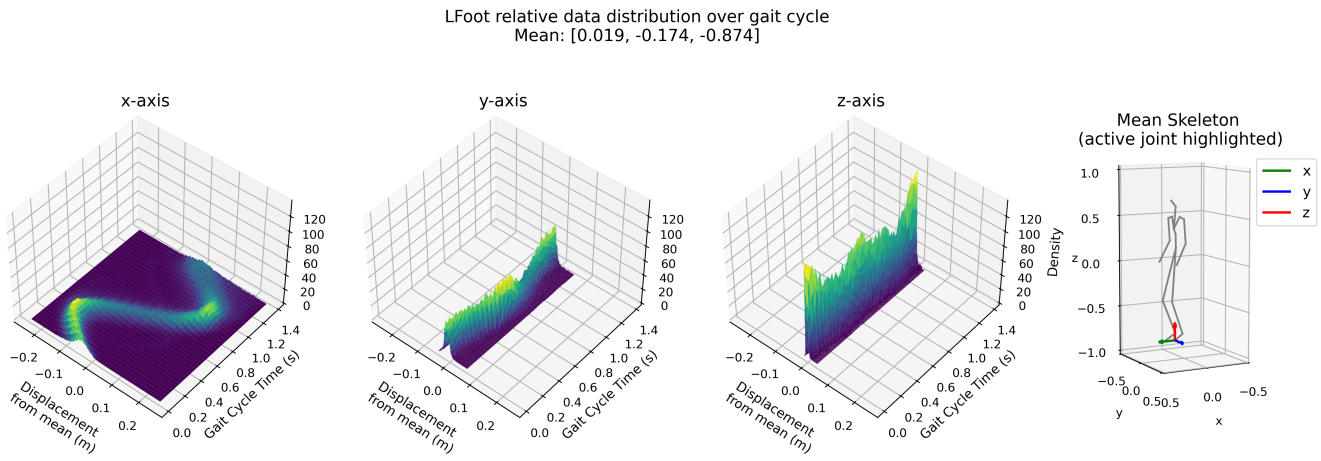


Fig. G-3: Temporal evolution of the LFoot joint coordinate distributions over the gait cycle.

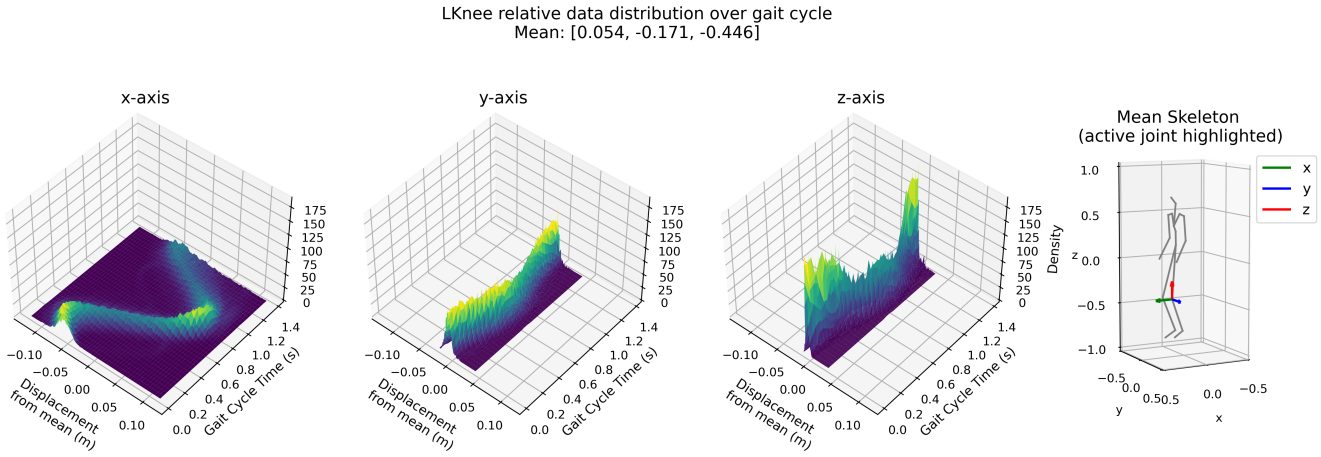


Fig. G-4: Temporal evolution of the LKnee joint coordinate distributions over the gait cycle.

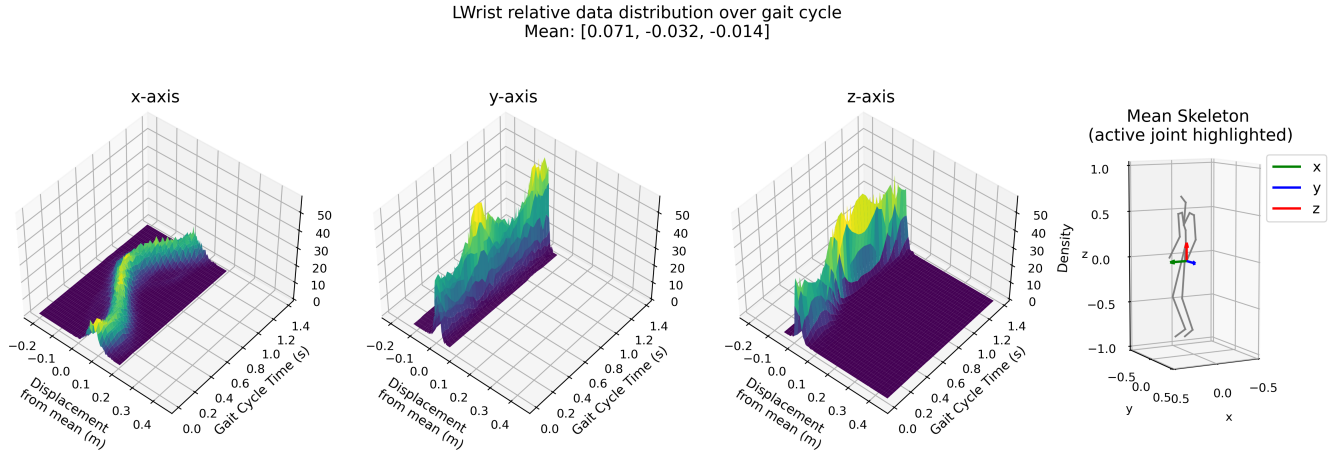


Fig. G-5: Temporal evolution of the LWrist joint coordinate distributions over the gait cycle.

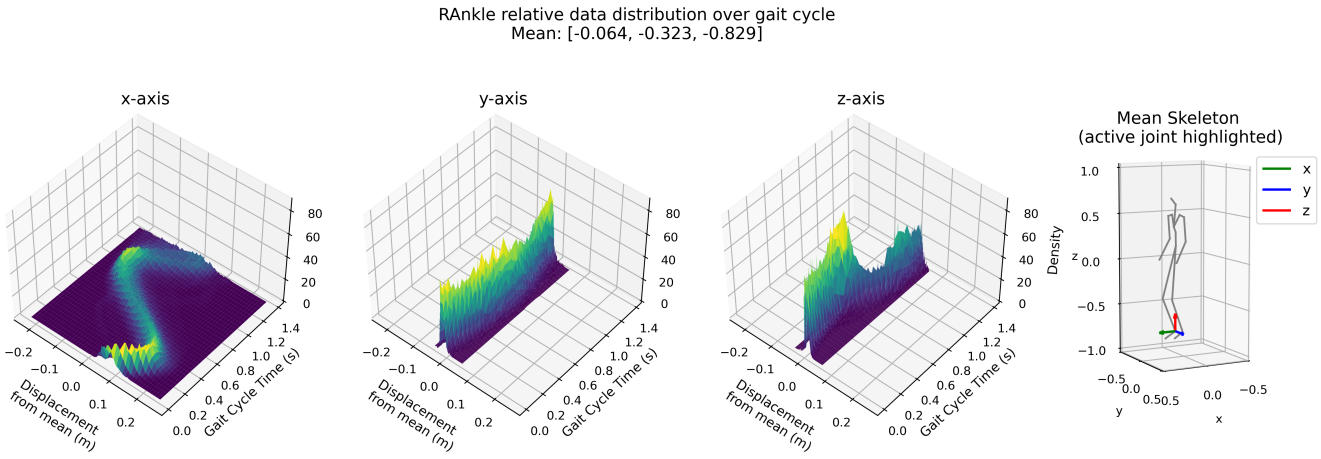


Fig. G-6: Temporal evolution of the RAnkle joint coordinate distributions over the gait cycle.

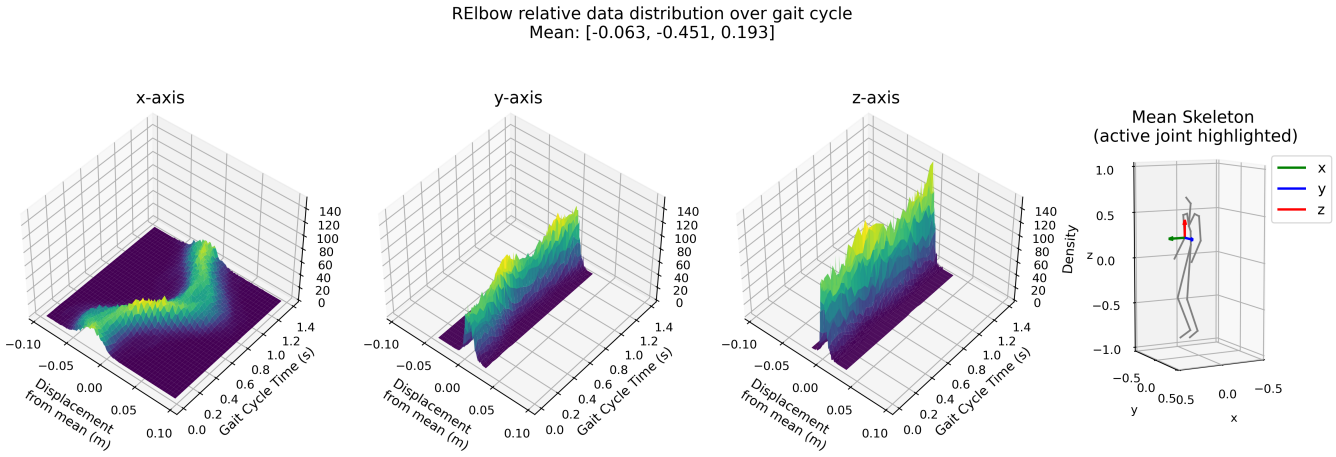


Fig. G-7: Temporal evolution of the RElbow joint coordinate distributions over the gait cycle.

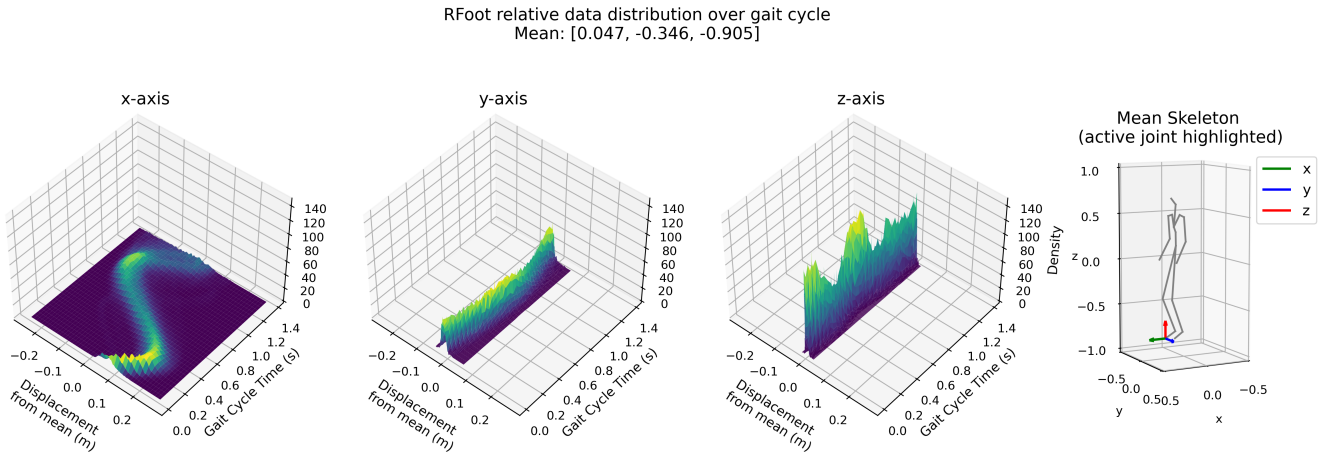


Fig. G-8: Temporal evolution of the RFoot joint coordinate distributions over the gait cycle.

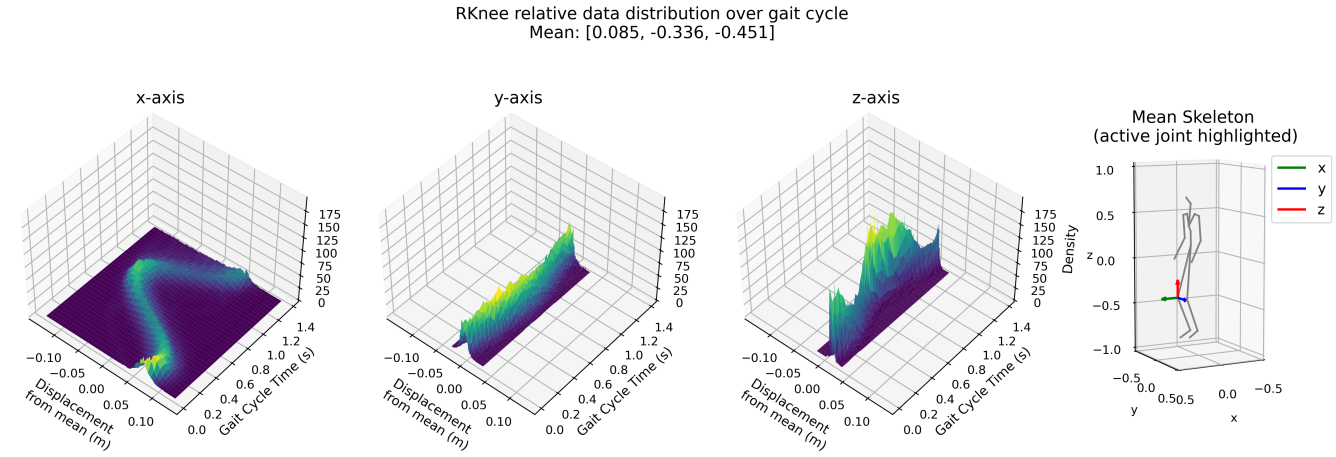


Fig. G-9: Temporal evolution of the RKnee joint coordinate distributions over the gait cycle.

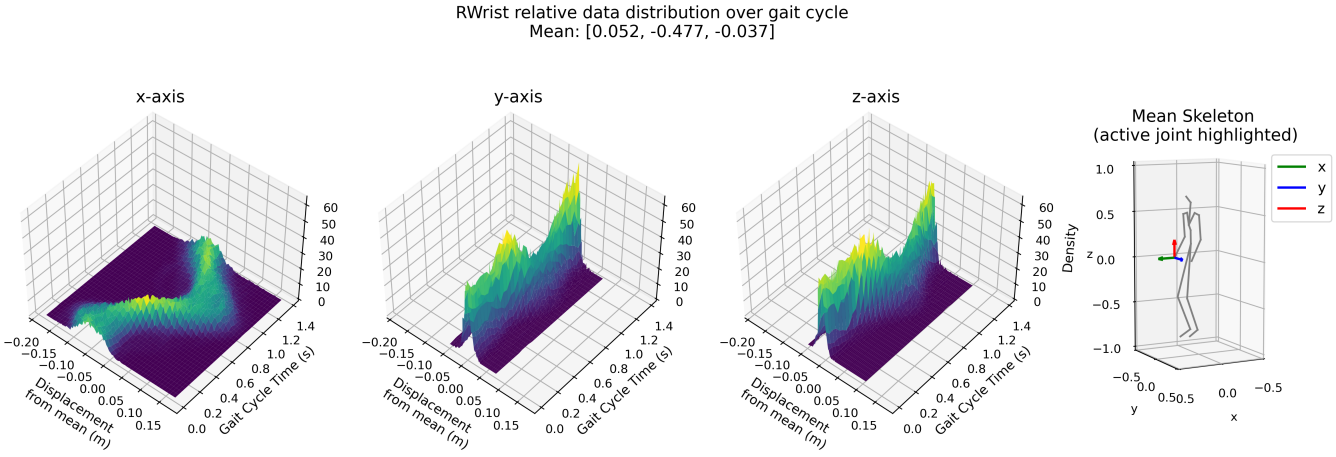


Fig. G-10: Temporal evolution of the RWrist joint coordinate distributions over the gait cycle.

APPENDIX H
EXPERIMENT OVERVIEW

TABLE H-I: Overview of the experiments conducted in this work. Experiment B is divided into four sub-experiments with the multiple model and dataset configurations.

Experiment	Main Focus	Models / Branches	Dataset(s)	Evaluation setup
A	Back-to-front versus Front-to-back	GCNext & PhysMoP (Physics, Data)	Human3.6M (GCNext), AMASS (PhysMoP)	Back-to-front vs. front-to-back input feeding, fixed targets
B	**Accuracy vs. Number of Observed Frames**			
B-1		GCNext	Human3.6M	Input window length increased from 1 to model-specific maximum (50 or 25)
B-2		PhysMoP Physics	AMASS	
B-3		PhysMoP Data	AMASS	
B-4		GCNext	AMASS (converted to Human3.6M format)	
C	Temporal resolution (sampling rate)	GCNext; PhysMoP (Physics, Data)	Walking subset of each dataset, resampled with different frequencies	Resampled sequences at varying frame intervals (α); fixed/matched output rates
D	Real-time performance	GCNext; PhysMoP (Physics, Data, Fusion)	Walking subsets of each dataset	Sliding-window on continuous walking sample on CPU

APPENDIX I
RAW MPJPE AND 95% CONFIDENCE INTERVALS

A. Accuracy versus Number of Observed Frames

TABLE I-I: Mean MPJPE [mm] $\pm$ 95% confidence-interval half-widths for the GCNext model, separated between upper and lower body regions across prediction horizons and select numbers of observed input frames. The number of total samples is 3840. This table corresponds to Figure 5a.

	Upper body (MPJPE $\pm$ 95% CI) [mm]				Lower body (MPJPE $\pm$ 95% CI) [mm]			
Input frames	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**80 ms**	**400 ms**	**560 ms**	**1000 ms**
1	24.33 ± 0.62	89.66 ± 2.24	113.32 ± 2.64	143.38 ± 3.01	24.69 ± 0.56	92.30 ± 1.97	112.41 ± 2.30	121.97 ± 2.40
3	13.71 ± 0.34	79.94 ± 1.77	105.27 ± 2.21	137.84 ± 2.66	11.26 ± 0.28	59.17 ± 1.25	75.97 ± 1.53	103.80 ± 1.96
5	10.78 ± 0.25	72.99 ± 1.60	99.50 ± 2.11	133.57 ± 2.64	10.00 ± 0.24	54.83 ± 1.13	72.65 ± 1.51	96.46 ± 1.86
10	10.52 ± 0.24	70.02 ± 1.58	96.03 ± 2.12	129.74 ± 2.59	9.63 ± 0.23	50.78 ± 1.07	68.82 ± 1.41	91.47 ± 1.80
15	10.58 ± 0.24	68.86 ± 1.59	94.17 ± 2.07	129.61 ± 2.62	9.56 ± 0.23	50.22 ± 1.06	68.40 ± 1.44	89.53 ± 1.81
20	10.61 ± 0.25	67.79 ± 1.56	93.09 ± 2.06	128.01 ± 2.61	9.40 ± 0.23	49.52 ± 1.06	66.63 ± 1.44	88.24 ± 1.81
25	10.41 ± 0.24	67.07 ± 1.51	93.35 ± 2.06	128.07 ± 2.65	9.20 ± 0.22	48.26 ± 1.08	64.49 ± 1.44	87.01 ± 1.85
30	10.42 ± 0.25	67.96 ± 1.55	93.03 ± 2.07	127.75 ± 2.67	9.02 ± 0.22	47.72 ± 1.08	64.11 ± 1.45	86.32 ± 1.85
35	10.31 ± 0.24	66.80 ± 1.52	92.76 ± 2.10	127.05 ± 2.67	9.22 ± 0.22	47.52 ± 1.04	63.96 ± 1.42	85.92 ± 1.81
40	10.29 ± 0.23	66.53 ± 1.56	92.27 ± 2.12	126.34 ± 2.70	9.23 ± 0.23	46.39 ± 1.02	62.90 ± 1.41	84.85 ± 1.81
45	10.31 ± 0.24	66.31 ± 1.57	91.64 ± 2.10	125.20 ± 2.64	9.27 ± 0.23	46.72 ± 1.03	62.77 ± 1.38	85.58 ± 1.82
50	10.31 ± 0.24	66.11 ± 1.55	91.86 ± 2.11	125.04 ± 2.60	9.16 ± 0.23	46.30 ± 1.01	62.87 ± 1.38	85.11 ± 1.78

TABLE I-II: Mean MPJPE [mm] $\pm$ 95% confidence-interval half-widths for the GCNext model on the AMASS dataset, separated between upper and lower body regions across prediction horizons and select numbers of observed input frames. The number of total samples is 9203. This table corresponds to Figure 5b.

	Upper body (MPJPE $\pm$ 95% CI) [mm]				Lower body (MPJPE $\pm$ 95% CI) [mm]			
Input frames	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**80 ms**	**400 ms**	**560 ms**	**1000 ms**
1	81.10 ± 0.63	115.72 ± 0.88	136.74 ± 1.18	167.95 ± 1.77	111.51 ± 1.13	186.04 ± 1.32	184.88 ± 1.30	197.78 ± 1.57
3	6.07 ± 0.05	81.18 ± 1.02	106.95 ± 1.32	144.18 ± 1.88	10.36 ± 0.09	109.00 ± 1.08	143.71 ± 1.37	166.71 ± 1.72
5	6.14 ± 0.08	57.46 ± 0.69	89.02 ± 1.10	135.61 ± 1.86	9.70 ± 0.12	86.85 ± 0.89	131.56 ± 1.28	163.35 ± 1.66
10	5.58 ± 0.08	49.04 ± 0.62	84.44 ± 1.09	136.91 ± 1.96	8.97 ± 0.11	74.15 ± 0.75	123.60 ± 1.25	163.69 ± 1.69
15	6.57 ± 0.09	50.14 ± 0.63	88.37 ± 1.15	145.98 ± 2.12	9.70 ± 0.11	69.49 ± 0.72	118.19 ± 1.17	165.07 ± 1.66
20	6.70 ± 0.10	51.01 ± 0.66	90.92 ± 1.24	149.75 ± 2.21	9.59 ± 0.11	68.34 ± 0.67	117.02 ± 1.12	165.98 ± 1.70
25	7.07 ± 0.10	53.04 ± 0.72	95.13 ± 1.31	153.67 ± 2.25	9.48 ± 0.11	68.14 ± 0.67	118.07 ± 1.16	164.82 ± 1.75
30	7.53 ± 0.11	55.12 ± 0.75	97.29 ± 1.34	153.89 ± 2.22	9.58 ± 0.11	69.34 ± 0.73	118.08 ± 1.21	167.20 ± 1.82
35	7.80 ± 0.12	55.73 ± 0.75	97.49 ± 1.32	152.13 ± 2.16	10.07 ± 0.11	68.81 ± 0.76	117.22 ± 1.26	167.63 ± 1.83
40	7.93 ± 0.13	55.21 ± 0.74	95.68 ± 1.29	147.58 ± 2.11	9.71 ± 0.11	70.93 ± 0.81	119.46 ± 1.33	165.02 ± 1.79
45	7.92 ± 0.12	53.41 ± 0.71	92.34 ± 1.25	141.82 ± 1.99	9.89 ± 0.11	70.42 ± 0.85	116.43 ± 1.29	160.04 ± 1.70
50	7.87 ± 0.12	51.33 ± 0.69	89.32 ± 1.22	137.70 ± 1.90	10.05 ± 0.12	67.71 ± 0.81	112.75 ± 1.22	158.54 ± 1.68

TABLE I-III: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Physics branch, separated between upper and lower body regions across prediction horizons and select numbers of observed input frames. The number of total samples is 15467. This table corresponds to Figure 5c.

	Upper body (MPJPE ± 95% CI) [mm]				**Lower body (MPJPE ± 95% CI) [mm]**			
Input frames	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**80 ms**	**400 ms**	**560 ms**	**1000 ms**
1	4.75 ± 0.08	39.67 ± 0.66	53.19 ± 0.90	73.81 ± 1.34	10.87 ± 0.14	86.08 ± 1.12	109.61 ± 1.42	116.93 ± 1.66
2	3.79 ± 0.05	133.44 ± 1.39	206.20 ± 1.88	316.07 ± 1.93	9.63 ± 0.12	319.68 ± 3.20	416.35 ± 3.48	497.89 ± 3.03
3	0.09 ± 0.00	15.01 ± 0.20	37.53 ± 0.48	122.16 ± 1.39	0.20 ± 0.00	32.17 ± 0.41	78.10 ± 0.96	222.86 ± 2.33
4	0.10 ± 0.00	16.06 ± 0.25	40.64 ± 0.63	129.67 ± 1.75	0.19 ± 0.00	31.32 ± 0.47	78.55 ± 1.15	233.11 ± 2.70
5	0.09 ± 0.00	13.69 ± 0.21	34.06 ± 0.52	108.48 ± 1.51	0.15 ± 0.00	26.16 ± 0.36	65.19 ± 0.90	193.46 ± 2.35
10	0.08 ± 0.00	12.72 ± 0.18	31.91 ± 0.45	108.16 ± 1.42	0.15 ± 0.00	25.52 ± 0.34	63.87 ± 0.84	201.54 ± 2.28
15	0.08 ± 0.00	13.03 ± 0.20	32.57 ± 0.50	108.45 ± 1.50	0.15 ± 0.00	24.63 ± 0.34	61.98 ± 0.85	197.56 ± 2.39
20	0.08 ± 0.00	12.00 ± 0.18	29.69 ± 0.45	97.95 ± 1.40	0.14 ± 0.00	22.17 ± 0.31	54.59 ± 0.76	173.71 ± 2.19
25	0.08 ± 0.00	11.86 ± 0.18	29.34 ± 0.44	97.17 ± 1.42	0.14 ± 0.00	23.03 ± 0.32	56.86 ± 0.80	175.34 ± 2.19

TABLE I-IV: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Data branch, separated between upper and lower body regions across prediction horizons and select numbers of observed input frames. The number of total samples is 15467. This table corresponds to Figure 5d.

	Upper body (MPJPE ± 95% CI) [mm]				**Lower body (MPJPE ± 95% CI) [mm]**			
Input frames	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**80 ms**	**400 ms**	**560 ms**	**1000 ms**
1	4.76 ± 0.08	39.67 ± 0.66	53.17 ± 0.90	73.54 ± 1.34	10.91 ± 0.14	85.93 ± 1.12	109.39 ± 1.42	115.72 ± 1.65
2	5.33 ± 0.28	53.94 ± 0.57	121.81 ± 1.12	150.29 ± 1.46	7.75 ± 0.31	126.94 ± 1.56	281.69 ± 3.30	306.38 ± 3.29
3	9.02 ± 0.34	69.34 ± 0.65	139.30 ± 1.16	173.76 ± 1.48	15.95 ± 0.41	170.80 ± 2.13	316.57 ± 3.69	355.56 ± 3.77
4	4.73 ± 0.10	61.95 ± 0.74	94.92 ± 1.06	111.91 ± 1.56	11.58 ± 0.17	158.14 ± 1.90	236.18 ± 2.79	218.48 ± 2.87
5	6.97 ± 0.25	23.97 ± 0.32	46.20 ± 0.59	75.75 ± 1.10	14.44 ± 0.48	58.11 ± 0.73	106.98 ± 1.41	123.57 ± 1.62
10	9.86 ± 0.22	22.68 ± 0.29	35.14 ± 0.49	63.08 ± 1.05	23.40 ± 0.56	56.86 ± 0.79	81.58 ± 1.02	109.50 ± 1.36
15	9.02 ± 0.26	14.72 ± 0.24	27.33 ± 0.38	59.75 ± 0.93	16.28 ± 0.46	29.47 ± 0.43	51.33 ± 0.66	92.90 ± 1.17
20	1.22 ± 0.02	10.63 ± 0.18	22.60 ± 0.35	55.06 ± 0.94	2.48 ± 0.03	21.43 ± 0.29	42.21 ± 0.56	80.37 ± 1.14
25	0.69 ± 0.02	6.58 ± 0.13	14.15 ± 0.25	41.90 ± 0.77	1.49 ± 0.03	12.96 ± 0.18	25.92 ± 0.35	59.11 ± 0.92

B. Temporal Resolution (Output Fixed)

TABLE I-V: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the GCNext model on different resample rates, while keeping the ground truth frequency fixed. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 8a.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	44.68 ± 0.01	110.51 ± 0.11	97.36 ± 0.10	133.55 ± 0.13	2413
2.80	40.79 ± 0.00	110.37 ± 0.07	93.07 ± 0.06	131.35 ± 0.09	2433
2.60	38.77 ± 0.03	105.66 ± 0.05	93.01 ± 0.06	122.57 ± 0.09	2453
2.40	39.28 ± 0.01	102.96 ± 0.08	101.26 ± 0.12	118.88 ± 0.12	2473
2.20	39.00 ± 0.03	104.01 ± 0.07	110.49 ± 0.11	130.42 ± 0.15	2493
2.00	37.01 ± 0.02	105.61 ± 0.09	108.37 ± 0.09	142.50 ± 0.18	2513
1.80	31.19 ± 0.02	100.54 ± 0.05	99.53 ± 0.05	130.94 ± 0.11	2533
1.60	23.56 ± 0.01	81.46 ± 0.03	86.77 ± 0.01	107.67 ± 0.12	2553
1.40	17.47 ± 0.01	64.57 ± 0.02	78.93 ± 0.04	102.25 ± 0.08	2573
1.20	12.98 ± 0.00	50.72 ± 0.02	63.36 ± 0.00	80.78 ± 0.02	2593
1.00	9.88 ± 0.00	38.42 ± 0.04	47.19 ± 0.04	54.31 ± 0.06	2613
0.80	14.95 ± 0.01	48.07 ± 0.03	60.23 ± 0.03	75.57 ± 0.02	2613
0.60	11.91 ± 0.01	57.75 ± 0.03	72.30 ± 0.03	91.08 ± 0.01	2613
0.40	15.41 ± 0.01	69.64 ± 0.01	86.25 ± 0.01	100.24 ± 0.00	2613
0.20	17.23 ± 0.01	99.26 ± 0.07	123.09 ± 0.07	118.43 ± 0.03	2613

TABLE I-VI: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Data branch model on different resample rates, while keeping the ground truth frequency fixed. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 8b.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	34.24 ± 0.41	146.25 ± 2.46	160.41 ± 3.10	97.47 ± 2.38	2420
2.80	31.17 ± 0.37	130.06 ± 2.15	141.84 ± 2.71	86.37 ± 1.90	2479
2.60	28.27 ± 0.36	116.13 ± 1.87	123.95 ± 2.29	82.05 ± 1.59	2527
2.40	25.76 ± 0.36	103.73 ± 1.70	108.42 ± 1.98	83.90 ± 1.46	2586
2.20	24.50 ± 0.38	93.39 ± 1.60	96.47 ± 1.82	94.42 ± 1.59	2634
2.00	24.14 ± 0.40	85.56 ± 1.59	91.67 ± 1.91	112.69 ± 1.98	2693
1.80	22.07 ± 0.43	76.53 ± 1.50	89.28 ± 1.93	134.63 ± 2.50	2741
1.60	18.14 ± 0.35	76.77 ± 1.12	86.97 ± 1.35	118.27 ± 1.88	2800
1.40	10.15 ± 0.11	45.01 ± 0.60	47.33 ± 0.76	82.47 ± 1.19	2848
1.20	5.66 ± 0.06	21.71 ± 0.27	26.17 ± 0.32	45.78 ± 0.65	2907
1.00	1.51 ± 0.02	14.22 ± 0.18	17.74 ± 0.28	30.02 ± 0.48	2955
0.80	3.92 ± 0.05	20.29 ± 0.26	32.57 ± 0.47	59.17 ± 0.80	2955
0.60	8.12 ± 0.11	41.05 ± 0.53	59.73 ± 0.82	82.05 ± 1.13	2955
0.40	11.85 ± 0.15	64.03 ± 0.83	88.09 ± 1.13	92.66 ± 1.43	2955
0.20	15.64 ± 0.21	80.24 ± 1.06	105.59 ± 1.40	96.34 ± 1.41	2955

TABLE I-VII: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Physics branch model on different resample rates, while keeping the ground truth frequency fixed. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 8c.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	49.48 ± 0.72	329.32 ± 2.32	374.36 ± 2.38	436.19 ± 2.91	2420
2.80	45.87 ± 0.65	325.39 ± 2.29	371.72 ± 2.35	435.27 ± 2.90	2479
2.60	41.01 ± 0.58	317.54 ± 2.32	366.42 ± 2.26	437.06 ± 2.82	2527
2.40	35.70 ± 0.51	300.76 ± 2.44	356.78 ± 2.33	441.58 ± 2.88	2586
2.20	30.61 ± 0.43	275.47 ± 2.54	338.89 ± 2.40	433.80 ± 3.00	2634
2.00	25.23 ± 0.35	235.87 ± 2.60	305.74 ± 2.69	413.80 ± 3.25	2693
1.80	19.25 ± 0.27	180.17 ± 2.27	247.41 ± 2.78	364.90 ± 3.38	2741
1.60	13.09 ± 0.19	114.49 ± 1.58	168.24 ± 2.29	296.31 ± 3.41	2800
1.40	8.14 ± 0.11	65.42 ± 0.90	104.05 ± 1.45	191.95 ± 2.70	2848
1.20	4.16 ± 0.06	39.64 ± 0.54	68.62 ± 0.95	137.43 ± 2.08	2907
1.00	0.29 ± 0.00	16.47 ± 0.26	36.15 ± 0.58	95.25 ± 1.56	2955
0.80	3.96 ± 0.05	24.60 ± 0.25	35.48 ± 0.41	65.97 ± 0.92	2955
0.60	8.37 ± 0.11	55.73 ± 0.73	79.52 ± 1.05	82.12 ± 1.10	2955
0.40	11.97 ± 0.16	70.00 ± 0.93	97.27 ± 1.29	107.27 ± 1.59	2955
0.20	15.24 ± 0.20	74.51 ± 0.98	96.70 ± 1.28	99.26 ± 1.46	2955

C. Temporal Resolution (Output Matched)

TABLE I-VIII: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the GCNext model on different resample rates, while matching the ground-truth frequency. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 9a.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	57.07 ± 0.03	156.71 ± 0.15	125.49 ± 0.16	128.49 ± 0.13	2313
2.80	51.07 ± 0.03	147.33 ± 0.05	136.41 ± 0.10	138.26 ± 0.06	2343
2.60	46.29 ± 0.04	130.61 ± 0.09	142.89 ± 0.11	132.71 ± 0.08	2375
2.40	41.66 ± 0.03	116.59 ± 0.06	143.15 ± 0.11	124.41 ± 0.14	2405
2.20	36.35 ± 0.02	105.78 ± 0.06	136.96 ± 0.11	138.19 ± 0.12	2435
2.00	31.06 ± 0.02	96.49 ± 0.03	129.55 ± 0.09	148.22 ± 0.15	2465
1.80	26.57 ± 0.01	83.44 ± 0.00	115.05 ± 0.04	132.70 ± 0.12	2495
1.60	21.93 ± 0.01	71.57 ± 0.03	94.17 ± 0.02	113.66 ± 0.03	2525
1.40	16.30 ± 0.00	57.46 ± 0.04	70.00 ± 0.05	82.23 ± 0.07	2555
1.20	12.19 ± 0.00	46.17 ± 0.04	54.13 ± 0.04	65.50 ± 0.06	2585
1.00	9.88 ± 0.00	38.42 ± 0.04	47.18 ± 0.04	54.31 ± 0.06	2615
0.80	7.84 ± 0.00	37.99 ± 0.03	50.42 ± 0.04	63.35 ± 0.03	2613
0.60	4.91 ± 0.00	41.45 ± 0.01	60.33 ± 0.01	84.60 ± 0.04	2613
0.40	5.88 ± 0.01	51.11 ± 0.05	73.79 ± 0.07	101.78 ± 0.09	2613
0.20	1.98 ± 0.00	27.46 ± 0.03	49.24 ± 0.05	82.47 ± 0.06	2613

TABLE I-IX: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Physics branch on different resample rates, while matching the ground-truth frequency. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 9c.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	45.17 ± 0.86	210.21 ± 3.34	158.37 ± 2.92	94.96 ± 1.71	2420
2.80	33.71 ± 0.69	208.94 ± 3.22	159.53 ± 2.86	97.28 ± 1.69	2479
2.60	23.28 ± 0.49	196.13 ± 2.95	168.49 ± 2.90	95.03 ± 1.70	2527
2.40	15.01 ± 0.29	175.96 ± 2.54	170.14 ± 2.94	92.70 ± 1.78	2586
2.20	11.66 ± 0.18	151.36 ± 2.00	164.02 ± 2.92	110.41 ± 2.33	2634
2.00	12.80 ± 0.23	124.27 ± 1.52	153.82 ± 2.78	143.55 ± 2.90	2693
1.80	15.64 ± 0.31	99.50 ± 1.21	136.23 ± 2.36	166.16 ± 3.07	2741
1.60	12.91 ± 0.31	88.58 ± 1.01	126.62 ± 1.60	105.98 ± 2.15	2800
1.40	4.34 ± 0.08	45.76 ± 0.66	65.47 ± 0.99	58.38 ± 1.02	2848
1.20	2.40 ± 0.03	19.17 ± 0.27	29.13 ± 0.43	37.23 ± 0.71	2907
1.00	1.51 ± 0.02	14.22 ± 0.18	17.74 ± 0.28	30.02 ± 0.48	2955
0.80	0.89 ± 0.01	8.39 ± 0.13	10.51 ± 0.18	24.53 ± 0.37	2955
0.60	0.65 ± 0.01	3.70 ± 0.05	8.55 ± 0.11	28.99 ± 0.47	2955
0.40	0.66 ± 0.00	6.35 ± 0.09	14.27 ± 0.20	47.37 ± 0.68	2955
0.20	0.60 ± 0.01	7.90 ± 0.11	15.37 ± 0.21	37.53 ± 0.50	2955

TABLE I-X: Mean MPJPE [mm] ± 95% confidence-interval half-widths for the PhysMoP Physics branch on different resample rates, while matching the ground-truth frequency. The number of test samples varies across conditions due to the sliding-window segmentation after resampling. This table corresponds with Figure 9c.

	MPJPE ± 95% CI [mm]				
Resample rate α	**80 ms**	**400 ms**	**560 ms**	**1000 ms**	**Number of samples**
3.00	28.39 ± 0.45	389.15 ± 3.40	388.02 ± 2.46	445.46 ± 2.93	2420
2.80	23.65 ± 0.37	384.42 ± 3.42	394.31 ± 2.60	442.04 ± 2.93	2479
2.60	18.49 ± 0.29	366.42 ± 3.49	401.19 ± 2.69	441.10 ± 2.82	2527
2.40	14.11 ± 0.22	334.81 ± 3.51	398.54 ± 2.93	443.52 ± 2.87	2586
2.20	10.79 ± 0.17	295.28 ± 3.38	383.29 ± 3.23	439.26 ± 2.99	2634
2.00	7.60 ± 0.12	241.89 ± 3.01	348.07 ± 3.46	427.64 ± 3.15	2693
1.80	5.23 ± 0.08	175.95 ± 2.46	281.38 ± 3.39	391.68 ± 3.49	2741
1.60	3.08 ± 0.05	102.31 ± 1.72	180.51 ± 2.74	319.49 ± 3.94	2800
1.40	1.81 ± 0.02	52.13 ± 0.86	101.00 ± 1.56	204.54 ± 3.12	2848
1.20	0.91 ± 0.01	30.40 ± 0.47	63.20 ± 0.91	139.95 ± 2.16	2907
1.00	0.29 ± 0.00	16.47 ± 0.26	36.15 ± 0.58	95.25 ± 1.56	2955
0.80	0.35 ± 0.00	11.71 ± 0.14	25.27 ± 0.33	75.68 ± 1.22	2955
0.60	1.23 ± 0.02	17.77 ± 0.22	33.83 ± 0.42	89.68 ± 1.16	2955
0.40	0.39 ± 0.01	8.91 ± 0.12	19.00 ± 0.26	62.00 ± 0.82	2955
0.20	0.10 ± 0.00	3.71 ± 0.04	7.40 ± 0.09	23.42 ± 0.29	2955