Array 15 (2022) 100212

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Spatio-temporal aggregation of skeletal motion features for human motion prediction

Itsuki Ueda *, Hidehiko Shishido, Itaru Kitahara

University of Tsukuba, 1-1-1, Tennodai, Tsukuba, Ibaraki 305-8577, Japan

ARTICLE INFO

Keywords:
Human motion
Lie-algebra
Temporal aggregation
Spatial aggregation
Attention

ABSTRACT

This study proposes a human body motion prediction model that can adapt to the disorders in various human motion patterns and represent the kinematic constraints. In human motion prediction, the acquisition of features that capture inter-motion and inter-joint linkages is considered effective. To generate links that are adaptive to the reference time of dominant features and their crossing-over joints, we construct an attention-based network that aggregates motion sequences temporally and spatially. We evaluated the motion prediction results using the Human3.6M dataset with the indices of mean angle error and mean per joint position error and showed that our method outperforms other state-of-the-art methods.

1. Introduction

3D human motion prediction is a process that takes a numeric human body pose sequence as input and predicts the future pose sequence as shown in Fig. 1. Motion prediction is an essential technology with many applications such as human–robot interaction [1], automated driving [2], pedestrian tracking [3]. Various input formats are possible for person motion prediction, including video [4], scene mesh [5], object information [6,7]. Especially motion capture information about a single person is widely used due to its high practicality in terms of sensing cost and accuracy. These motion predictions deal with numerical values of skeletal information obtained from motion captures such as joint positions and angles, as shown in Fig. 2. Since human motion involves many interlocking joints, the description of its characteristics requires high-dimensional information in both time and space. In addition, the prediction task is highly dynamic and nonlinear and inherently involves high-dimensional uncertainty as time passes. Therefore, there is a limit to the analytical acquisition of features required for motion prediction [8,9]. Recently, data-driven approaches such as [10–16] has become the mainstream. For example, in the case of walking, both hands and feet dominates in the same motion cycle, while the left and right hands and feet are linked in opposite phases. Thus, there is a strong bias in the continuity and regularity of natural human motion. Data-driven approaches have improved motion prediction performance by separating pose information into temporal and spatial axes and modeling feature aggregation with appropriate network structures, respectively [17]. In this paper, we propose a method to classify the features that have been generalized by the conventional networks by temporal and spatial aggregation and to obtain high generalization performance by incorporating each aggregation model by adding appropriate information to the framework of attention [18]. We propose a method to achieve a high generalization performance within each aggregation model by adding appropriate information to the framework.

Temporal aggregation involves "translation generalization" and "time scale generalization" as shown in Fig. 3. The generalization of translation means acquiring features that do not depend on at which time in the input sequence a given action occurs. The generalization of time scales means acquiring features that do not depend on how long a behavior occurs and how fast the same behavior is performed. Previous methods adopt mainly four approaches, including convolutional neural network (CNN) [11], recurrent neural network (RNN) [10,13], discrete cosine transform (DCT) [12] and attention mechanism (Attention) [15,19]. CNN has excellent generalizability of temporal translation, but it is not easy to generalize the time scale because the kernel size fixes the reference time time-length. RNNs have the generalizability of time-scale in terms of the reference time-length stretch modeling. However, it belongs to the Markovian models, which is not suitable for translational generalization, due to the accumulation of errors when the dominant features of the motion appear in the forward part of the sequence. DCT can acquire features referring to the overall input time by transferring them to frequency space before processing them. The pose information transferred to the frequency space can be generalized for both translation and scale because the time shift and frequency are separated. However, DCT has difficulties

* Corresponding author.
E-mail address: ueda.itsuki@image.iit.tsukuba.ac.jp (I. Ueda).

https://doi.org/10.1016/j.array.2022.100212
Received 26 January 2022; Received in revised form 18 June 2022; Accepted 21 June 2022
Available online 23 June 2022

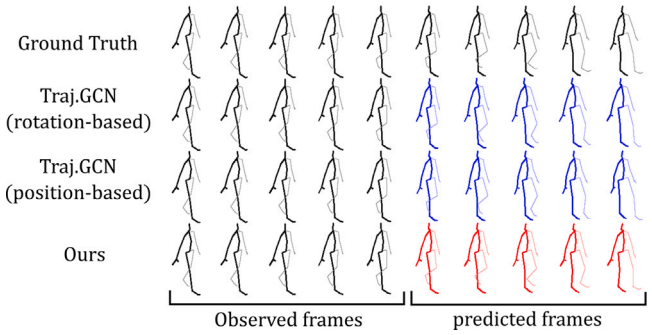

Fig. 1. 3D motion prediction: The model takes the observation sequence on the left as input and outputs the prediction result sequence on the right. From top to bottom, the ground-truth value, the position-based and angle-based models of Traj-GCN [12], and the output of the proposed method are shown. The proposed method produces a pose sequence closer to the actual value.

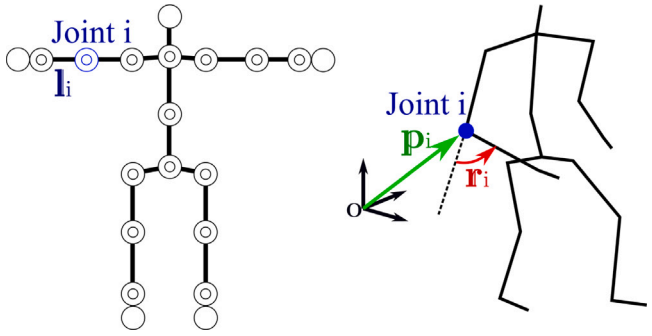

Fig. 2. Examples of how to describe pose information. The human skeleton is modeled with links and rotational joints. (Left) The offset $\mathbf{l}_i$ of each joint is set based on the T pose. (Right) There are two ways to quantify the pose: a position-based description, such as the 3D position of joint i, and an angle-based description, such as the rotation vector of joint angles, $\mathbf{r}_i \in \mathbb{R}^3$. There are two types of angle-based descriptions.

generalizing the translation because the time shift remains absolute due to the boundary condition in acyclic operation. Attention can provide feature descriptions referring to the entire time region by explaining the proximity between time points with Positional Encoding. Although Positional Encoding provides generalization for both translation and scale, it may be inferior to DCT for acquiring periodic motion features because it gives the proximity discontinuously for each time pair. In this method, we propose a new positional encoding in the attention architecture, which reproduces the operations in frequency space such as in DCT. This method provides frequency and time-shift information to Attention's Query in the framework of relative position embedding (RPE). The model can acquire motion features with separate translation and time scales, which means it can interpret the same motion events with the same parameters when they appear at different times and speeds of occurrence.

Spatial aggregation critically involves generalizing motion features per joint and spatial dependencies between joints. Generalization of motion features per joint means acquiring independent features regardless of any joint. For example, the motion features observed in the right knee are also applicable to the left knee. Generalization of spatial dependence among joints means the acquisition of reference relationships among joints to provide each motion, for example, the rotation linkage of the shoulder, elbow, and wrist to produce a straight-line trajectory of the fingertips. Previous methods have focused on the expression of spatial dependency using network structures such as Graph Convolution and Graph Attention. Specifically, they generate a joint graph with each joint as a node, then detach the joint-wise features and the spatial dependencies across joints. For joint-wise features, they consider the feature dimensions retained by each node as

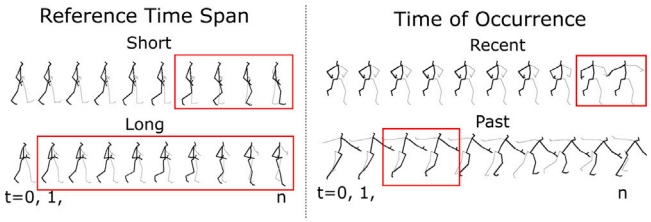

Fig. 3. Generalization of temporal features: (Left) The foot during walking is dominated by short-time features, while the center of gravity during a change of direction is dominated by long-time features. (Right) Features to be watched, such as a change in motion, are not always located behind the input.

identical and share the rotation matrix applied between joints. They also provide the importance that each node attaches to other nodes during aggregation, as adjacency matrices, to represent the spatial dependencies between joints.

The methods that use motion trees as adjacency matrices can describe spatial proximity in link connection relationships, but they make it difficult to reference distant joints. Wei et al. [12] enables referencing of distant joints by learning the adjacency matrix. Due to the lack of information on the original connection relations of the motion tree, it was not sufficient for acquiring global features such as capturing the entire near joints of arms and legs. Li et al. [13] gives the closeness of joints to each other by constructing a multi-scale graph that integrates close joints. While wide-area features are easier to acquire, the choice of which joints to integrate relies on heuristics. The generalization of spatial dependence is still an unsolved problem.

We focus on the fact that the generalization issues are different between the two data description methods, position-based and angle-based, which have been evaluated separately as independent problems. Concretely, the position-based description can capture the features that represent a wide range of motions, such as the interlocking of distant joints in the kinematic tree due to the relative positional relationships of the joints. On the other hand, angle-based description has an advantage in capturing features that represent local actions, such as local actions at a terminal joint, regardless of the posture of the root side in the kinematic tree. Therefore, we employ an approach in which the angle-based description gives the features held by each node of the joint graph, and the position-based description provides the spatial dependence during feature aggregation.

As shown in Fig. 6, we attempt to generalize both local and global motion features by building a predictor that handles angle-based features while incorporating position-based features suitable for representing spatial dependencies. Since the rotation axis of each joint can vary from moment to moment as the human pose, Inverse Kinematics (IK), an algorithm that calculates the angle of each joint from the input position, requires the asymptotic computation of recursive equations. Since the inclusion of IK in the network degrades the prediction accuracy for actions far from the reference pose, adding a location-based description to the input does not directly improve accuracy. Therefore, previous methods use only angle-based or position-based descriptions for input and output, such as Fig. 5 (see Fig. 4).

The contribution of this work is to improve the performance of motion prediction by increasing the generalizability of motion feature acquisition, both in terms of temporal and spatial aggregation. For temporal aggregation, we introduce operations in frequency space in the framework of RPE to Self-Attention to achieve generalization in both translations of occurrence time and time width scale for the reference time of dominant motion features. For spatial aggregation, we derive position-based and angle-based pose descriptions that can be linearly mapped in the vicinity of the input final pose through the parameter space of Lie algebra, and assign position-based features to angle-based features by Cross-Attention, which allows us to both generalize the motion of individual joints and generalize the spatial

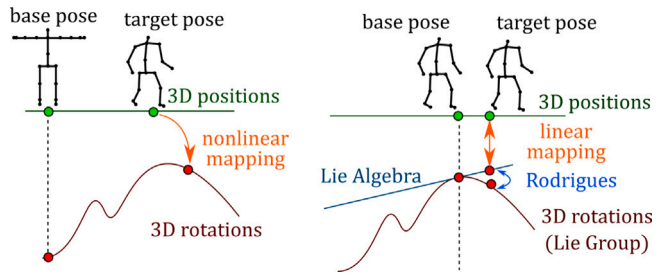

Fig. 4. (Left) IK is in general a nonlinear map. (Right) The Lie algebra associated with the tangent plane of the final pose can define a linear map between the position and the tangent plane.

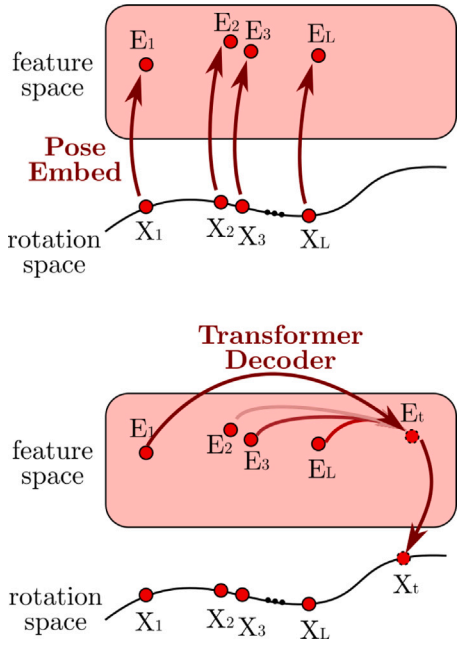

Fig. 5. Example of a transformer that takes only angles as input and output: (Top) Pose described by angle is embedded into feature space. (Bottom) Output of the future pose X_t described by angle, aggregated by the Transformer Decoder.

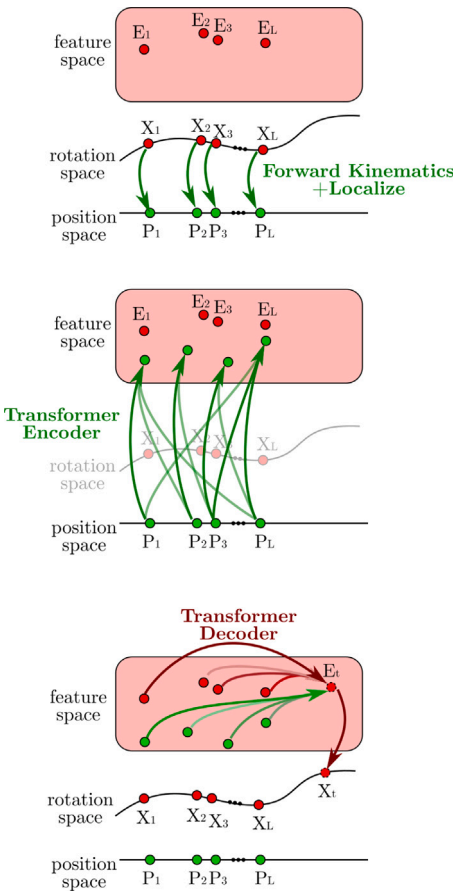

Fig. 6. Linking position and angle: (Top) Defining a mapping from joint angles to positions in the neighborhood of X_L, and generating a position-based description with the necessary elements for the inverse mapping. (Middle) Feature extraction and transformation to angle-based features are represented by Transformer. (Bottom) Future poses are output by aggregating both position- and angle-derived features with the Transformer Decoder.

dependence between joints. We performed benchmark tests for motion prediction using the Human3.6M [20] and CMU-Mocap [21] datasets and showed that our model has superior performance compared to conventional methods.

2. Related works

2.1. Temporal aggregation

Modeling human motion prediction is difficult due to its high dimensionality, nonlinear dynamics, and the uncertainty of human motion. Analytical methods for short time forecasting have been proposed such as Gaussian process latent variable model [8], hidden Markov model [9], and random forest [9]. However, these methods have short applicability due to the models' limited complexity. Thus, the mainstream methods in recent years have shifted to using deep learning.

Previous research has mainly used four models, CNN, RNN, DCT, and Attention, as efficient ways to acquire motion features in time-series data processing by deep learning. Li et al. [11] built a Convolutional Sequence-to-Sequence model using time series convolution. Since the range of spatial and temporal dependencies captured by this model is statically determined by the size of the convolutional filter, it cannot handle behaviors with different periods. Hence, regression-based methods, such as LSTM, are widely used because they can train a time-dependent range [13,22,23]. However, regression-based methods assume Markovianity, which reduces the robustness in the temporal direction. DCT transforms a time-discrete input sequence into a continuous sequence in the time axis and a discrete sequence in the frequency axis. DCT aggregates centrally in the temporal direction and directly references distant times, which provides excellent generalization performance of the feature scale. On the other hand, the transformation from discrete values requires extrapolation, such as signal folding to form a periodic function. As a result, boundary conditions remain, and generalization of translation is challenging for acyclic transition behavior. Therefore, several works proposed a method using DCT to predict frequency space instead of pose space [12,14]. DCT aggregates centrally in the temporal direction and directly references distant times, which provides excellent generalization performance of the feature scale. On the other hand, the transformation from discrete values requires extrapolation, such as signal folding to form a periodic function. As a result, boundary conditions remain, and generalization of translation is challenging for acyclic transition behavior. Aksan et al. [15] proposed another approach that uses Attention to refer to features at distant times without distinction. In Attention, Positional Encoding assigns time proximity. The generalization performance for time translation of features is high, and the generalization performance to feature scales is also maintained due to the possibility of wide-area referencing. However, it is inferior to the method using DCT in periodic operation because the proximity exists for each time pair.

We attempt to reproduce the scale generalization performance of DCT with temporal attention by focusing on the frequency components dominant to the motion in the framework of Positional Encoding. There are two computational forms of Positional Encoding: absolute positional encoding (APE) and relative positional encoding (RPE). Benyou et al. [24] has reported that the performance of BERT can be improved by using APE and RPE together. Wu et al. [25] proposed a multiplicative positional encoding as contextual RPE. In this method, we propose an RPE based on the contextual RPE [25] that can reproduce operations in frequency space, i.e., operations when using DCT, as a convolution in pose space.

2.2. Spatial aggregation

There are two essential aspects of spatial aggregation: generalization of the motion between individual joints, such as the typical motion of the right and left knees, and generalization of the spatial dependency between joints, such as the interlocking of the shoulder, elbow, and wrist, such as the smooth trajectory of the fingertips, which is independent of the posture of the whole body. In order to generalize these two characteristics, conventional methods focus on the representation of dependencies between joints by network structures. Graph Convolution and Graph Attention consider the motion tree a graph with each joint as a node and represent spatial dependencies by adjacency matrices. Li et al. [11] use the connection relation of the motion tree as the adjacency matrix of Graph Convolution. Graph Convolution can express spatial monotonicity such that the closer joints give more importance because it aggregates only the information of adjacent joints in each layer. On the other hand, it is difficult to refer to distant or strongly synchronized joints in the motion tree, such as the left and right wrists during hand-holding, despite their high spatial dependency. Wei et al. [12] introduce the adjacency matrix as a trainable parameter. They consider the connections between joints as complete graphs and describe the spatial dependence in edge weights, which enables distance joint referencing. On the other hand, the original information of the connection relation is lost, such as which joints are adjacent to each other. Li et al. [13] constructs a multi-scale graph that integrates close joints to obtain global features such as focusing on the movement of the entire foot instead of independent joints such as the knee and ankle. While this method makes it easier to obtain global features by omitting similar joints, it relies on heuristics to select which joints to integrate. Emre et al. [15] uses Graph Attention to feed the adjacency matrix dynamically. In common with each method, the problem is generalizing the spatial dependence between joints.

We focus on generalizing spatial dependency by utilizing data structures instead of network structures. In general, networks utilize position-based or angle-based descriptions for data structures describing input/output poses. The position-based description directly uses the 3D position of each joint acquired by motion capture [26], etc. The angle-based description determines a skeletal reference shape, such as T-pose or A-pose. It uses Euler angles and rotation vectors to describe the rotation of each joint from the reference shape.

In the position-based description, it is easy to obtain spatial constraints on the end joints, such as the mutation of the ankle position into a zero vector when the axial foot stops during ground contact. The position-based description is also suitable for describing the relationship between joints and selecting the joints to be gazed at from a global perspective. It can explicitly describe the positional relationship between joints distant from each other in the motion tree by using relative positions. On the other hand, the link length strongly affects the movement of the joint position, and the scale is different between the behavior of the hip joint and that of the toe. For example, even if the motion from the elbow to the toe is the same, the dynamic coordinate transformation depends on the shape of the root side, making it difficult to identify the motion features. Therefore, it is more difficult to obtain generalized features of individual joints and features of local joints in comparison with angle-based descriptions. In addition, it is difficult to apply the constraint that the link length is invariant, such as the arm length does not change during the motion when the output is directly represented as a 3D position.

With angle-based description, it is easy to obtain local behavioral features such as gazing from the elbow to the tip of the elbow, regardless of trunk behavior such as prone or crouching. Also, by describing the output as angle-based, fixed link length can be applied. The disadvantage is that the link length information is lost from the data, making it difficult to describe the relationship between distant joints that are not directly connected. This makes it difficult to acquire global features and model spatial dependencies between joints dynamically. Wei et al. compared the performance of Res-Sup, ConvSeq2Seq, Traj-GCN, and Traj-GCN networks by using a position-based description for direct motion prediction and an angle-based description for transformation to each joint position after prediction. It is pointed out that the prediction performance of the angle-based description is inferior to that of the position-based description.

In the proposed method, features and spatial dependencies acquired from the position-based description are introduced into the intermediate features of the motion prediction network using the angle-based description using Cross-Attention. The transformation from positional information to angular information, called inverse kinematics, requires recursive computation because the superposition of rotations cannot be decomposed. Since even a deep-learning-based predictor has a significant error, simply adding a position-based description to the input does not improve performance. We decompose the pose sequence into differences from the final input pose to isolate rough motion information, such as standing or walking, which facilitates the coordination of position and angles. In a similar approach, Liu et al. [16] proposed a joint trajectory space that can efficiently analyze motion context while preserving information on joint trajectories by decomposing the description of joint positions into the final pose and the velocity of each frame. We assume that the input and output poses are in the vicinity of the final input pose and achieve the linkage between position and angle via the Lie algebra around the rotation matrix of the final pose.

3. Our approach

Fig. 7 shows the entire proposed method. For input and output, we use the rotation of each joint in the form of a rotation matrix. As a preprocessing step, we convert the input to position-based and angle-based representations based on the final pose. Full connection and graph convolution gives the pose information as a feature vector for Self-Attention and provide the APE information. This network comprises a Transformer Encoder, which takes the position-based description as input and outputs the feature values, and a Transformer Decoder, which takes the angle-based description as input and performs forward prediction.

The transformer contains Temporal Attention for aggregation between time points, Spatial Attention for aggregation between joints, Cross Attention for importing position-based features to the decoder side, and a Feed Forward network. The decoder outputs the data in rotation vectors in all coupling layers for each time and joint. The post-processing obtains the output pose sequence by finding the rotation matrix corresponding to the vector using the matrix exponential function and applying it to the final input pose.

3.1. Position-based and angle-based pose descriptions

In this method, position-based and angle-based features are obtained for spatial information aggregation to describe the coordination of distant joints and generalize each joint. The inverse kinematics problem is converted from joint positions to angles, and it is difficult to solve the error even with recent deep learning techniques. In this study, we define a highly accurate mapping from the position description to the

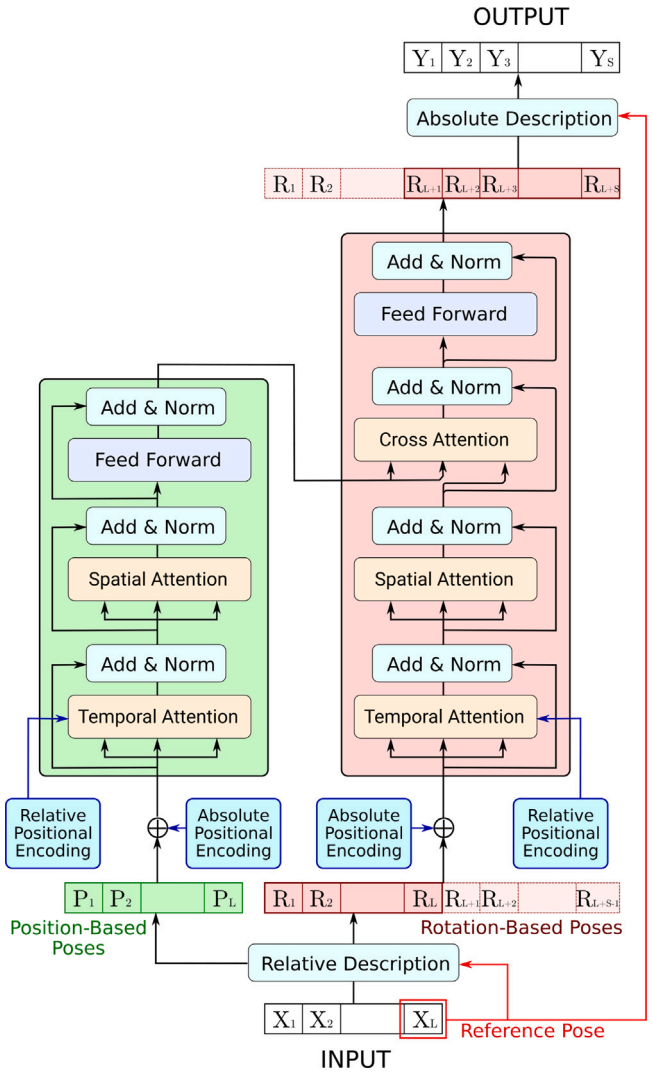

Fig. 7. Overall process diagram.

angle description and embed the necessary information in advance to avoid the accumulation of errors. Specifically, forward kinematics can provide a differentiable mapping from joint angles to positions using the offset and rotation matrices for each link in the reference pose. The rotation matrix takes a tangent plane at the reference pose and defines a Lie algebra, which can be used to create a 3-DOF parameter representation for each joint. Furthermore, this parameter space can be interconverted by defining Jacobians with position descriptions. In this section, we first introduce the parameter notation of Lie algebra as the angle notation. Then, we show that the position change and the transformation map of Lie algebra to the parameter space can be described in terms of the local coordinate system of joint positions.

We denote the angular description of the pose at time t as $X_t = [\mathbf{M}_{t,1}, \dots, \mathbf{M}_{t,N}]$. Each $\mathbf{M}_{t,i} \in \mathbb{R}^{3\times3}$ is an orthogonal matrix whose determinant is 1, which is a member of the special orthogonal group $SO(3)$. We consider the Lie algebra associated with the tangent plane in the input final frame $\mathrm{M}_{L,i}$. The matrix $W \in \mathbb{R}^{3\times3}$ on the tangent plane is multiplied by the source of the Lie algebra $[\mathbf{r}]_\wedge$, represented by the rotation vector $\mathbf{r} = [r_1, r_2, r_3]^T$, as follows.

$$\mathrm{B} = \mathrm{M}_{L,i} + [\mathbf{r}]_\wedge \tag{1}$$

$$[\mathbf{r}]_\wedge = \begin{pmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{pmatrix} \tag{2}$$

The W can be defined to map to $SO(3)$ by the matrix exponential function. This means that we can compute the rotation matrix $\mathrm{M}_{t,i}$ for frame t, joint i using the rotation vector $\mathrm{M}_{t,i}$ relative to the input final frame as follows:

$$\mathrm{M}_{t,i} = exp\left([\mathbf{r}_{t,i}]_\wedge\right)\mathrm{M}_{L,i}. \tag{3}$$

By using $R_t = [\mathbf{r}_{t,1}, \dots, \mathbf{r}_{t,N}]^T$ as the angle description, we can quickly obtain the derivative for the angle of the joint position given by the forward kinematics. The Lie algebraic parameter of rotation provides a stable property as the input–output space of the network because it avoids singularity and uniqueness problems such as gimbal lock, unlike angle descriptions such as Euler angles.

Next, we focus on the position-based description that can collaborate with the angle-based description. For simplicity, we assume that the links are connected as $1, 2, \dots, i, \dots, j-1, j, \dots, N$. The offset of link i at the reference pose is $\mathbf{l}_i \in \mathbb{R}^3$, the rotation matrix from the reference pose is $\mathrm{M}_i \in \mathbb{R}^{3\times3}$, the translation of the root is $\mathbf{l}_0$, and the rotation matrix $\mathrm{M}_0 = \mathrm{I}$. The homogeneous coordinate transformation matrix T_i by link i is shown in Eq. (4).

$$T_i = \begin{pmatrix} \mathrm{M}_i & \mathbf{l}_i \\ O & 1 \end{pmatrix} \tag{4}$$

The angle of each joint provides the position $\mathbf{p}_j^{(0)}$ in the world coordinate system of joint j by forward-kinematics in Eq. (2).

$$\begin{pmatrix} \mathbf{p}_j^{(0)} \\ 1 \end{pmatrix} = \mathrm{T}_0\mathrm{T}_1 \cdots \mathrm{T}_{j-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{5}$$

We derive the behavior when link i is further rotated by $\mathbf{r}_i$ in rotation vector notation based on $\mathrm{M}_1, \dots, \mathrm{M}_K$. The rotation matrix of joint i after applying $\mathbf{r}_i$ can be written as $\mathrm{M}_i' = exp([\mathbf{r}_i]_\wedge)\mathrm{M}_i$. The position $\mathbf{p}_j^{(0)\prime}$ of joint j after the transformation is as follows.

$$\begin{pmatrix} \mathbf{p}_j^{(0)\prime} \\ 1 \end{pmatrix} = \mathrm{T}_0\mathrm{T}_1 \cdots \begin{pmatrix} exp([\mathbf{r}_i]_\wedge) & 0 \\ 0 & 1 \end{pmatrix} \mathrm{T}_i \cdots \mathrm{T}_{j-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{6}$$

We consider a joint coordinate system i whose coordinate transformation from the world coordinate system is given by $\mathrm{T}_{i-1}^{-1}\mathrm{T}_{i-2}^{-1} \cdots \mathrm{T}_0^{-1}$ as the coordinate system based on link i. For the position $\mathbf{p}_j^{(i)}$ of link j in the joint coordinate system i, the following is established from Eq. (4).

$$\begin{pmatrix} \mathbf{p}_j^{(i)} \\ 0 \end{pmatrix} = \mathrm{T}_{i-1}^{-1}\mathrm{T}_{i-2}^{-1} \cdots \mathrm{T}_0^{-1} \begin{pmatrix} \mathbf{p}_j^{(0)} \\ 0 \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} \mathbf{p}_j^{(i)} \\ 0 \end{pmatrix} = \mathrm{T}_i\mathrm{T}_{i+1} \cdots \mathrm{T}_{j-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{8}$$

By assigning them to Eq. (6), we get the following:

$$\begin{pmatrix} \mathbf{p}_j^{(i)\prime} \\ 1 \end{pmatrix} = \begin{pmatrix} exp([\mathbf{r}_i]_\wedge) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_j^{(i)} \\ 1 \end{pmatrix} \tag{9}$$

In the neighborhood of $\mathrm{M}_1, \dots, \mathrm{M}_K \in SO(3)$, a good approximation is given by $exp([\mathbf{r}_i]_\wedge) \in so(3)$ using the source $[\mathbf{r}_i]_\wedge$ on the tangent plane. The mapping between the source and the position description on the tangent plane is as follows.

$$\begin{pmatrix} \mathbf{p}_j^{(i)\prime} \\ 1 \end{pmatrix} = \begin{pmatrix} \mathrm{I} + [\mathbf{r}_i]_\wedge & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_j^{(i)} \\ 1 \end{pmatrix} \tag{10}$$

$$\mathbf{p}_j^{(i)\prime} - \mathbf{p}_j^{(i)} = [\mathbf{r}_i]_\wedge \mathbf{p}_j^{(i)} \tag{11}$$

From the anti-commutativity of the Lie bracket product, the expression (11) is transformed as follows:

$$\mathbf{p}_j^{(i)\prime} - \mathbf{p}_j^{(i)} = [\mathbf{p}_j^{(i)}]_\wedge \mathbf{r}_i. \tag{12}$$

Similarly, when each joint is rotated by $\mathbf{r}_1, \ldots, \mathbf{r}_K$, the amount of change in position is related to the amount of change in angle as follows:

$$\mathbf{P}' - \mathbf{P} = (\mathrm{B} \otimes \mathrm{J})\mathbf{R} \tag{13}$$

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1^{(0)} \\ \vdots \\ \mathbf{p}_K^{(0)} \end{pmatrix} \tag{14}$$

$$\mathrm{J} = \begin{pmatrix} \mathrm{M}_1^{(0)}[\mathbf{p}_1^{(0)}]_\wedge & \cdots & M_K^{(0)}[\mathbf{p}_K^{(0)}]_\wedge \\ \vdots & \ddots & \vdots \\ M_1^{(0)}[\mathbf{p}_1^{(N)}]_\wedge & \cdots & M_K^{(0)}[\mathbf{p}_K^{(N)}]_\wedge \end{pmatrix} \tag{15}$$

$$\mathrm{R} = \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_K \end{pmatrix}. \tag{16}$$

Note that B is the matrix representing the parent–child relationship between the joints, which in this example is the unit lower triangular matrix. The J can be computed from P using Eq. (2). Thus, by substituting $P_L^{(0)}$ for the position description at time L in the reference pose P, $P_t^{(0)}$ for the position description at time t in the destination pose P', and $\mathbf{R}_t$ for the rotation $\mathbf{R}$, we can associate the angle description with the position description $\mathbf{R}_t \mathrm{P}_t - \mathrm{P}_L = (\mathrm{B} \otimes \mathrm{J}_L)\mathbf{R}_t$.

$$\mathrm{P}_t = \begin{pmatrix} \mathbf{p}_{t,1}^{(0)} \\ \vdots \\ \mathbf{p}_{t,1}^{(N)} \end{pmatrix}, \mathrm{P}_L = \begin{pmatrix} \mathbf{p}_{L,1}^{(0)} \\ \vdots \\ \mathbf{p}_{L,1}^{(N)} \end{pmatrix} \tag{17}$$

Since B is a value that can be set statically for the same skeletal model, and since the skeletal model is a tree structure with joint 1 as the root, we can transform B into a lower triangular matrix by exchanging columns. The inverse of the mapping by the lower triangular matrix can be easily represented in the network. By adding the information of J to the encoder's input, the transformation to the angle can be embedded in the network. This method uses Cross-Attention to combine angle-based features, so the rotation matrix is further separated at the encoder side, and the combination of joint position and joint coordinate system $\mathrm{P}_t \in \mathbb{R}^{N \times 3N}$ is used as the input position-based description.

$$\mathrm{P}_t = \begin{pmatrix} \mathbf{p}_{t,1}^{(0)} & \cdots & \mathbf{p}_{t,1}^{(N)} \\ \vdots & \ddots & \vdots \\ \mathbf{p}_{t,N}^{(0)} & \cdots & \mathbf{p}_{t,N}^{(N)} \end{pmatrix}^T \tag{18}$$

3.2. Embedding of human pose information

Pose-Embedding generates D-dimensional feature vectors of poses that can be aggregated by Self-Attention from pose sequences with position-based and angle-based descriptions transformed in Section 3.1. Let the input and output pose sequences be the tensors of $\mathrm{G} \in \mathrm{R}^{L \times N \times C}, \mathrm{E} \in \mathrm{R}^{L \times N \times D}$, respectively. C is the number of elements in each description, e.g., $3K$ for the position-based description in Section 3.1 and 3 for the angle-based description. G is the number of elements in each description, e.g., $3K$ in the position-based description and 3 in the angle-based description. G is described by a common evaluation axis for time, specifically, the feature values $\mathbf{g}_{t1,i}, \mathbf{g}_{t2,i}$ at time $t1, t2$ and joint i are directly comparable. On the other hand, for space, the feature values $\mathbf{g}_{t,i1}, \mathbf{g}_{t,i2}$ for time t and joints i, j are taken to different evaluation axes. For example, the elbow joint and the knee joint have two axes of motion: flexion–extension in the offset vertical direction and adduction–abduction in the offset direction. They have similar ranges of motion, but the directions of the axes of motion are different. Therefore, the input $G_t \in \mathbb{R}^N \times C$ at each time is rotated for each joint, and then static graph convolution is performed on the joint graph to output the feature vector $E_t \in \mathbb{R}^{N \times D}$.

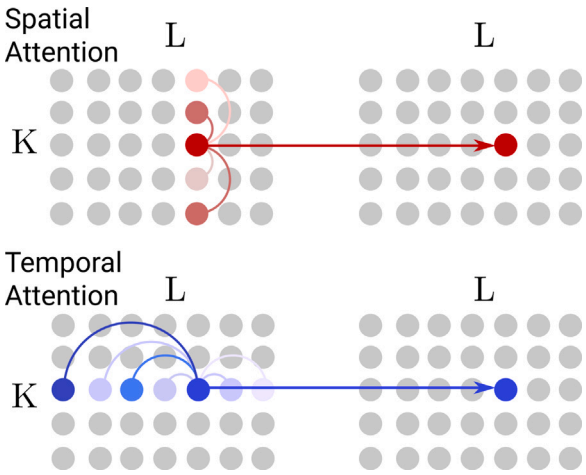

Fig. 8. (Top) Spatial Attention aggregates the information of different joints at the same time. (Bottom) Temporal Attention aggregates the information of the same joint at different times.

Using the weights to be learned for each joint $[\mathrm{W}_0^G, \ldots, \mathrm{W}_N^G], \mathrm{W}_i^G \in \mathbb{R}^{D \times C}$, the middle layer output $E_t^0 \in \mathbb{R}^{N \times D}$ with rotation applied is calculated as follows.

$$E_t^0 = \begin{pmatrix} \mathrm{W}_0^G \mathrm{G}_{t,0} \\ \vdots \\ \mathrm{W}_N^G \mathrm{G}_{t,N} \end{pmatrix} \tag{19}$$

The output E_t^l of the lth layer of the graph convolution is calculated using the adjacency matrix $A_l \in \mathbb{R}^{N \times N}$, the weights $W_l \in \mathbb{R}^{D \times D}$, and the activation function $\sigma(\cdot)$ as follows:

$$\mathrm{E}_t^l = \sigma(\mathrm{A}_l \mathrm{E}_t^{l-1} \mathrm{W}_l) \tag{20}$$

Note that we assumed that the adjacency matrix is a complete graph with edge weights instead of a kinematic tree, and A_l is the trainable parameter for the reference of distant joints.

3.3. Attention

In this method, we use the Self-Attention mechanism to efficiently aggregate information from distant joints and times. The Cross-Attention mechanism incorporates position-based features into angle-based features. As shown in Fig. 8, we use two models for Self-Attention: Spatial Attention, which aggregates the information of a different joint at each time, and Temporal Attention, which aggregates the information of a different joint at each time. Attention, which aggregates information for each joint at different times.

Spatial Attention is an independent process at each time. The input/output to the layer at time t is $E_t = [\mathbf{e}_{t,1}, \ldots, \mathbf{e}_{t,N}]$, $E'_t = [\mathbf{e}'_{t,1}, \ldots, \mathbf{e}'_{t,N}] \in \mathbb{R}^{N \times D}$. To apply the Scale Dot-product Attention [18], we calculated Query Q_t, Key K_t, and Value V_t from the linear transformation of E_t by the learnable weights $\mathrm{W}_{temporal}^Q, \mathrm{W}_{temporal}^K, \mathrm{W}_{temporal}^V \in \mathbb{R}^{D \times D}$ of the training target as follows:

$$\mathrm{Q}_t = \mathrm{E}_t \mathrm{W}_{temporal}^Q, \tag{21}$$

$$\mathrm{K}_t = \mathrm{E}_t \mathrm{W}_{temporal}^K, \tag{22}$$

$$\mathrm{V}_t = \mathrm{E}_t \mathrm{W}_{temporal}^V. \tag{23}$$

To ensure that the tight joints of Q, K are strongly referenced, we compute the Attention map $\mathrm{A}_t \in \mathbb{R}^{N \times N}$ as follows:

$$\mathrm{A}_t = \sigma(\mathrm{Q}_t \mathrm{K}_t^T). \tag{24}$$

Note that $\sigma(\cdot)$ is the activation function, which generally uses a normalization such that the sum of the directions is one after applying

the ReLU or exponential function. By aggregating V_t according to the Attention map and projecting it to the representation space with weights $W^O_{temporal} \in \mathbb{R}^{D\times D}$, the output E'_t is calculated as follows:

$$E'_t = A_t V_t W^O_{temporal}. \tag{25}$$

Eq. (25) gives the interpretation that the adjacency matrix is a dynamically acquired graph convolution.

Temporal Attention is similarly an independent process for each joint. The input/output to the layer at joint i is as follows:

$$E_i = [e_{i,1}, \dots, e_{i,L}], E'_i = [e'_{i,1}, \dots, e'_{i,L}] \in \mathbb{R}^{L\times D}. \tag{26}$$

Using the weights $W^Q_{spatial}, W^K_{spatial}, W^V_{spatial}, W^O_{spatial} \in \mathbb{R}^{D\times D}$, the temporal aggregation at joint i is calculated as follows:

$$Q_i = E_i W^Q_{spatial} \tag{27}$$

$$K_i = E_i W^K_{spatial} \tag{28}$$

$$V_i = E_i W^V_{spatial} \tag{29}$$

$$A_i = \sigma(Q_i K_i^T + M) \tag{30}$$

$$E'_i = A_i V_i W^O_{spatial} \tag{31}$$

Note that $Q_i, K_i, V_i \in \mathbb{R}^{L\times D}, M, A_i \in \mathbb{R}^{L\times L}$. The M is a mask to avoid referring to the future during inference, which is O for the encoder and $-\infty$ for the upper triangular component for the decoder.

3.4. Embedding time and joint proximity

The Attention mechanism aggregates all of the time and joint information in the input indistinctly. For this purpose, the proximity of time and joint is assigned to the feature vector by Positional Encoding. Our method uses Absolutely Positional Encoding (APE) to obtain features that depend on absolute time and specific joints and Relative Positional Encoding (RPE) to obtain generalized features that do not distinguish the timing of occurrence.

APE assigns proximity to the feature vectors generated in Section 3.2 just before the network's input. Proximity on the time axis $Z^{temporal} = [z^{temporal}_{t,c}] \in \mathbb{R}^{(L+S)\times D}$, proximity on the spatial axis $Z^{spatial} = [z^{spatial}_{i,c}] \in \mathbb{R}^{N\times D}$, the feature of the pose embedding at time t, joint i, and feature dimension c is $e^l_{t,i,c}$, then the value $e_{t,i,c}$ after APE assignment is as follows:

$$e_{t,i,c} = e^l_{t,i,c} + z^{temporal}_{t,c} + z^{spatial}_{i,c} \tag{32}$$

In this method, we use a fully learnable APE with the learnable parameters $Z^{temporal}, Z^{spatial}$ to ensure monotonicity over a wide area.

In RPE, when computing Temporal Attention, we design it in such a way that it assigns proximity only to queries, as shown in Fig. 9. For feature dimension c, if the proximity between times t_1 and t_2 is $z^{relative}_{t_1-t_2,c}$, then Eq. (24) can be replaced by the following equation

$$A_t = \sigma(Q_t K_t^T + Q_t Z^{relative}_t) \tag{33}$$

$$Z^{relative}_t = \begin{pmatrix} z_{t-1,1} & \cdots & z_{t-L,1} \\ \vdots & \ddots & \vdots \\ z_{t-1,D} & \cdots & z_{t-L,D} \end{pmatrix} \tag{34}$$

Our method uses a fixed cosine RPE in the time axis and sets the proximity $z^{relative}_{t_1-t_2,i,c}$ as follows:

$$z^{relative}_{t_1-t_2,i,c} = cos(\frac{t_1 - t_2}{10000^{2c/D}}) \tag{35}$$

3.5. Training

In the training, we use $L_{rot} + \lambda L_{pos}$ as the objective function, which is a composite of the angle error L_{rot} and the position error L_{pos} with the hyperparameter $\lambda \in \mathbb{R}^+$. The output for time $t(L+1 \le t \le L+S)$ is $Y_{t-L} = [M_{t,1}, \dots, M_{t,N}]$, and the joint position in the world coordinate

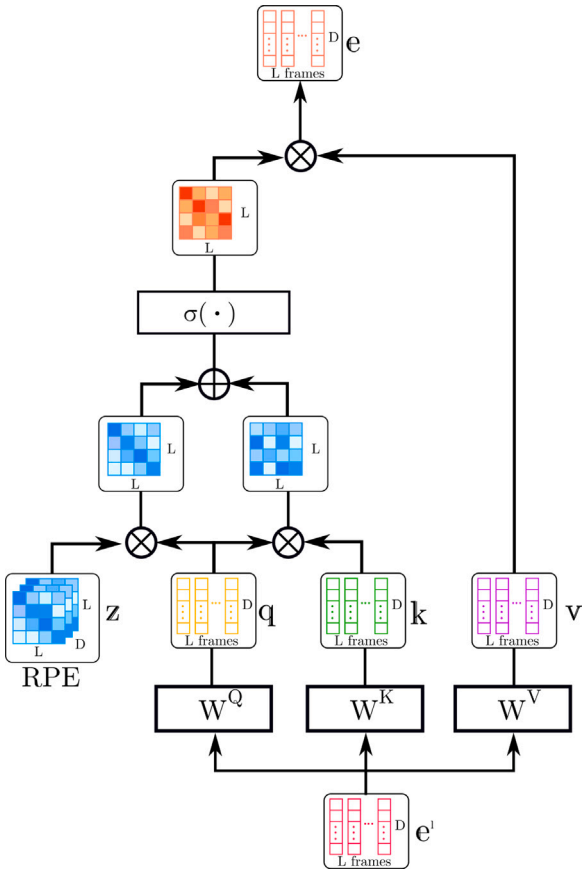

Fig. 9. RPE gives the product of query and cosine when computing Temporal Attention.

system calculated by forward kinematics is $[p^{(0)}_{t,1}, \dots, p^{(0)}_{t,N}]$, and their respective true values are $[M^{GT}_{t,1}, \dots, M^{GT}_{t,N}][p^{GT}_{t,1}, \dots, p^{GT}_{t,N}]$. The angular error $\theta_{t,i}$ for joint i is defined by the rotation angle of the rotation matrix $M^T_{t,i} M^{GT}_{t,i}$, which is the residual. The rotation angle is given by a variant of Rodriguez's formula as follows

$$\theta_{t,i} = acos\left(\frac{tr(M^T_{t,i} M^{GT}_{t,i}) - 1}{2}\right) \tag{36}$$

The overall angle error L_{rot} is as follows.

$$L_{rot} = \sum_{t=L+1}^{L+S} \sqrt{\sum_{i=1}^{N} \theta^2_{t,i}} \tag{37}$$

The position error L_{pos} summarizes the Euclidean distance of the position for each joint as follows:

$$L_{pos} = \sum_{t=L+1}^{L+S} \sum_{i=1}^{N} |p^{(0)}_{t,N} - p^{GT}_{t,N}| \tag{38}$$

4. Experiments

4.1. Datasets

We verify the performance of the model on the motion capture dataset Human3.6M [20]. The Human3.6M dataset contains 15 different classes of motion captured by seven subjects. Each sequence contains the Euler angles of each joint concerning the T pose for 32 joints at 50 fps. We applied downsampling method from 50 fps to 25 fps for all sequences for fairness. We used the data of six subjects to train the model and tested it on the movement class of another subject. The test data traditionally used eight randomly selected sequences for each

Table 1
Evaluation results of position error in H3.6M: ↓ MPJPE [mm].

Scenarios	Walking				Eating				Smoking				Discussion			
ms	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	29.36	50.82	76.03	81.51	16.84	30.60	56.92	68.65	22.96	42.64	70.14	82.68	32.94	61.18	90.92	96.19
DMGNN	17.32	30.67	54.56	65.20	10.96	21.39	36.18	43.88	8.97	17.62	32.05	40.30	17.33	34.78	61.03	69.80
Traj-GCN	12.29	23.03	39.77	46.12	8.36	16.90	33.19	40.70	7.94	16.24	31.90	38.90	12.50	27.40	58.51	71.68
MSR-GCN	12.16	22.65	38.64	45.24	8.39	17.05	33.03	40.43	8.02	16.27	31.32	38.15	11.98	26.76	57.08	**69.74**
Ours w/o Enc	12.90	23.77	39.36	46.19	8.70	16.94	33.24	40.68	7.88	16.32	31.78	38.86	12.64	27.55	58.55	71.72
Ours w/o RPE	12.65	22.24	38.28	45.39	8.03	17.43	33.02	40.27	8.16	16.32	31.15	38.32	11.96	26.84	57.22	69.63
Ours w/o Pos	10.72	20.71	36.41	42.50	7.40	15.87	33.11	41.20	7.23	15.29	31.62	39.14	11.30	27.38	61.79	75.91
Ours w/o Rot	12.69	22.81	39.63	46.66	9.27	17.70	33.69	41.32	8.80	16.58	31.10	37.92	14.12	29.39	60.87	74.55
Ours	**8.55**	**18.06**	**34.78**	**41.85**	**6.06**	**13.68**	**30.00**	**38.01**	**5.64**	**12.68**	**28.27**	**36.10**	**9.25**	**23.42**	**55.87**	70.55
Scenarios	Directions				Greeting				Phoning				Posing			
ms	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	35.36	57.27	76.30	87.67	34.46	63.36	124.60	142.50	37.96	69.32	115.00	126.73	36.10	69.12	130.46	157.08
DMGNN	13.14	24.62	64.68	81.86	23.30	50.32	107.30	132.10	12.47	25.77	48.08	58.29	15.27	29.27	71.54	96.65
Traj-GCN	8.97	19.87	43.35	53.74	18.65	38.68	77.74	93.39	10.24	21.02	42.54	52.30	13.66	29.89	66.62	84.05
MSR-GCN	8.61	19.65	43.28	53.82	16.48	36.95	77.32	93.38	10.10	20.74	41.51	51.26	12.79	29.38	66.95	85.01
Ours w/o Enc	8.88	19.30	43.67	53.46	18.36	38.60	77.50	93.96	10.32	21.20	42.73	52.41	13.96	29.27	66.89	84.32
Ours w/o RPE	8.28	19.82	43.31	53.42	16.84	36.78	77.41	93.46	10.12	20.85	41.59	51.30	12.85	29.45	66.78	85.21
Ours w/o Pos	8.12	19.98	46.89	58.59	16.51	38.45	81.90	98.69	9.28	20.71	44.85	56.02	12.59	30.95	72.14	90.70
Ours w/o Rot	10.52	22.05	46.81	58.13	19.54	40.14	80.19	96.54	11.69	23.10	45.30	55.67	16.75	34.64	72.31	90.36
Ours	**6.62**	**16.75**	**41.27**	**53.01**	**13.67**	**33.23**	**74.42**	**92.18**	**7.66**	**17.60**	**39.45**	**50.15**	**9.74**	**24.64**	**60.85**	**78.83**
Scenarios	Purchases				Sitting				SittingDown				TakingPhoto			
ms	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	36.33	60.30	86.53	95.92	42.55	81.40	134.70	151.78	47.28	85.95	145.75	168.86	26.10	47.61	81.40	94.73
DMGNN	21.35	38.71	75.67	92.74	11.92	25.11	44.59	50.20	14.95	32.88	77.06	93.00	13.61	28.95	45.99	58.76
Traj-GCN	15.60	32.78	65.72	79.25	10.62	21.90	46.33	57.91	16.14	31.12	61.47	75.46	9.88	20.89	44.95	56.58
MSR-GCN	14.75	32.39	66.13	79.64	10.53	21.99	46.26	57.80	16.10	31.63	62.45	76.84	9.89	21.01	44.56	**56.30**
Ours w/o Enc	15.78	32.25	65.62	79.90	10.91	21.14	46.12	57.47	16.64	31.39	61.13	75.70	9.53	20.99	44.26	57.12
Ours w/o RPE	14.05	32.89	66.65	79.70	10.17	19.87	46.96	56.95	13.68	28.85	60.11	76.46	9.62	21.30	44.33	56.91
Ours w/o Pos	13.99	32.45	68.59	83.03	10.02	21.92	49.39	62.45	15.57	32.03	67.27	83.24	9.51	21.46	48.81	61.47
Ours w/o Rot	17.74	35.83	69.06	82.61	12.97	25.31	50.27	62.06	18.48	35.06	67.88	83.40	12.11	24.37	49.93	62.29
Ours	**11.50**	**27.89**	**63.05**	**78.37**	**8.00**	**18.65**	**43.70**	**56.42**	**11.93**	**26.71**	**59.97**	**76.38**	**7.39**	**17.71**	**43.52**	56.80
Scenarios	Waiting				WalkingDog				WalkingTogether				Average			
ms	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	30.62	57.82	106.22	121.45	64.18	102.10	141.07	164.35	26.79	50.07	80.16	92.23	34.66	61.97	101.08	115.49
DMGNN	12.20	24.17	59.62	77.54	47.09	93.33	160.13	171.20	14.34	26.67	50.08	63.22	16.95	33.62	65.90	79.65
Traj-GCN	11.43	23.99	50.06	61.48	23.39	46.17	83.47	95.96	10.47	21.04	38.47	45.19	12.68	26.06	52.27	63.51
MSR-GCN	10.68	23.06	48.25	59.23	20.65	42.88	80.35	93.31	10.56	20.92	37.40	43.85	12.11	25.56	51.64	62.93
Ours w/o Enc	11.54	24.21	51.33	62.58	23.93	46.31	83.79	96.52	10.63	21.34	38.54	45.22	13.65	26.67	52.36	63.88
Ours w/o RPE	9.12	20.06	47.52	59.73	19.36	39.87	78.92	93.45	9.27	19.41	36.82	44.16	11.01	21.55	49.86	62.90
Ours w/o Pos	9.97	22.84	50.73	63.27	19.53	41.55	78.79	91.81	9.80	20.43	37.44	44.04	11.44	25.47	53.98	66.14
Ours w/o Rot	12.41	24.15	47.99	59.24	23.04	44.47	80.19	93.24	11.56	21.37	37.51	44.33	14.11	27.80	54.18	65.89
Ours	**7.92**	**18.88**	**44.39**	**56.95**	**16.25**	**36.99**	**75.67**	**91.42**	**7.34**	**16.31**	**34.24**	**42.45**	**9.17**	**21.55**	**48.63**	**61.30**

movement class. However, the number of evaluation sequences was small, and the chosen sequences were biased toward easily predictable ones. Lingwei et al. [14] pointed out the inaccuracy of the evaluation and evaluated the entire sequence for each action class. We inherit this benchmark and evaluate it using the entire test data.

We also evaluate the same on the CMU-Mocap dataset [21]. For the sake of fairness, we use the same data as in Res-sup [10] with the anthropometric differences removed and validate it on eight classes of test cases.

4.2. Evaluations

Following the standard evaluation metrics used in Res-sup [10], the Mean Per Joint Position Error (MPJPE), which describes the average distance from the ground-truth value of each joint position in millimeters, is used to evaluate the results. The prediction times were compared for 80, 160, 320, and 400 [ms] as in the previous study. We measured the MPJPE for the output angle converted to the position by forward-kinematics. As a baseline, we compared the results of our method with those of four recent methods: Res-sup, Traj-GCN, DMGNN, and MSR-GCN. We used the results reported in each paper for each error directly. Since DMGNN [13] did not report the error of 3D positions, only the results of angular error were compared. For Human3.6M, we conducted comparative experiments on whether it has Encoder and RPE modules and uses position-based and angle-based descriptions. We omitted the cross-attachment of the decoder and used only the self-attachment for the angle descriptions for the experiments without the encoder. For the experiments without RPE, we configured each Temporal Attention as a regular self-attention. We replaced the decoder's initial input $R_1, \ldots, R_L$ with a single zero vector for the only position-based description experiment. For the only rotation-based description experiment, we replaced the encoder's input with angle informations $R_1, \ldots, R_L$ instead of positions $P_1, \ldots, P_L$. We implemented the architecture of our method using Pytorch [27], and trained the model using AdaBound [28] for the gradient method. The learning rate is $lr = 0.0001$, $final_lr = 1.0$, $weightdecay = 0.0005$, mini-batch is trained with random extraction of batch size 256, input sequence length is 32 frames, and output sequence length is 10 frames.

4.3. Results

Table 1 shows the experimental results of the position error in Human3.6M [20]. In Human3.6M, all the motion classes outperformed the conventional method in prediction below 320 [ms]. In ablation,

Table 2
Evaluation results of position error in CMU-Mocap: ↓ MPJPE [mm].

Motion	Basketball				Basketball Signal				Directing Traffic				Jumping			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	15.45	26.88	43.51	49.23	20.17	32.98	42.75	44.65	20.52	40.58	75.38	90.36	26.85	48.07	93.50	108.90
DMGNN	15.57	28.72	59.01	73.05	5.03	9.28	20.21	26.23	10.21	20.90	41.55	52.28	31.97	54.32	96.66	119.92
Traj-GCN	11.68	21.26	40.99	50.78	3.33	6.25	13.58	17.98	6.92	13.69	30.30	39.97	17.18	32.37	60.12	72.55
MSR-GCN	10.28	18.94	37.68	47.03	3.03	5.68	12.35	16.26	5.92	12.09	28.36	38.04	14.99	28.66	55.86	69.05
Ours	**9.62**	**16.55**	**36.48**	**46.89**	**2.98**	**5.37**	**12.16**	**16.18**	**5.25**	**11.76**	**27.49**	**38.02**	**12.68**	**27.55**	**54.92**	**68.58**
Motion	Running				Soccer				Walking				WashingWindow			
Milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res-sup.	25.76	48.91	88.19	100.80	17.75	31.30	52.55	61.40	44.35	76.66	126.83	151.43	22.84	44.71	86.78	104.68
DMGNN	17.42	26.82	38.27	40.08	14.86	25.29	52.21	65.42	9.57	15.53	26.03	30.37	7.93	14.68	33.34	44.24
Traj-GCN	14.53	24.20	37.44	41.10	13.33	24.00	43.77	53.20	6.62	10.74	17.40	20.35	5.96	11.62	24.77	31.63
MSR-GCN	12.84	20.42	30.58	**34.42**	10.92	19.50	37.05	46.38	6.31	10.30	17.64	21.12	5.49	11.07	25.05	32.51
Ours	**11.54**	**16.29**	**29.21**	34.68	**9.76**	**18.45**	**35.29**	**44.65**	**6.13**	**9.67**	**16.56**	**19.22**	**5.13**	**11.56**	**23.35**	**29.43**

compared to the case without encoders or position-based input, there is a remarkable improvement in the whole-body movement classes such as Discussion, Sitting, SittingDown, and WalkingTogether. We considered that the position-based description, directly referring to the positional relationship of distant joints, worked effectively.

On the other hand, there is no significant improvement in the long-time prediction of 400 [ms]. The association between position and angle is assumed to be near the last input pose as the reference pose. This may have made it challenging to integrate the position-based features into the angle-based features when the movement is significant. In addition, the performance of short-time prediction is particularly significant for motion classes that include many periodic motions, such as Walking, Directions, Greeting, and Phoning. The considerable improvement compared to the experimental results without RPE suggests that the dominant frequency component enhancement by RPE plays a role similar to that of DCT and improves the prediction performance for periodic motions.

In the ablation of data description, the performance of long-term prediction without position-based and short-term prediction without angle-based are lower than our original. We interpret this result that the position-based description affects long-time prediction since it detects global features of the whole body. In contrast, the angle-based involves short-time prediction since it detects local features of each joint unit.

Table 2 shows the experimental results of position error in CMU-Mocap [21]. Similar to Human3.6M, we can find an improvement in the performance of CMU-Mocap for predictions below 320 [ms]. In addition, the improvement is more significant in the motion classes with complex motion patterns near the final input pose, such as basketball. We believe that this method extracts effective motion characteristics because it can refer to both position-based and angle-based features.

5. Discussion and future works

This paper proposes a method for extensive temporal and spatial information aggregation using Self-Attention. For temporal aggregation, the dominant frequencies are emphasized in the framework of relative positional encoding and used together with absolute positional encoding to achieve aggregation that incorporates the characteristics of conventional models using DCT. For spatial aggregation, we focus on the fact that position-based and angle-based descriptions are suitable for global and local feature extraction, respectively, and constructed a structure that incorporates the position-based features extracted by the encoder into the angle features by Cross-Attention. We also show that a linear mapping can be defined between the parameter space of Lie algebra and the position space of the joint coordinate system. We introduce a position- and angle-based description in the neighborhood coordinates of the final input pose. At the same time, we realized the expression of conditions such as fixation and contact of end joints by using the position as input and the expression of constraints such as connection relation and link length invariance by using angle as output. By selecting the appropriate architecture for extracting the features of position and angle, we create an inference that can predict with high accuracy even for non-periodic and difficult-to-generalize motion classes. The results show that our model outperforms the state-of-the-art methods in short-term prediction.

CRediT authorship contribution statement

Itsuki Ueda: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review and editing, Visualization. **Hidehiko Shishido:** Supervision. **Itaru Kitahara:** Conceptualization, Methodology, Resources, Writing – review and editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. Itsuki Ueda reports a relationship with Preferred Networks Inc that includes: employment.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP19H00806, JP21KK0070 and JP22H01580.

References

[1] Akhter I, Sheikh Y, Khan S, Kanade T. Nonrigid structure from motion in trajectory space. In: Proceedings of the neural information processing systems (NeurIPS); 2009.
[2] Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Trans Intell Veh 2016;1(1):33–55.
[3] Gong H, Sim J, Likhachev M, Shi J. Multi-hypothesis motion planning for visual object tracking. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV). IEEE; 2011, p. 619–26.
[4] Zhang JY, Felsen P, Kanazawa A, Malik J. Predicting 3D Human Dynamics from Video. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2019.
[5] Wang J, Xu H, Xu J, Liu S, Wang X. Synthesizing long-term 3d human motion and interaction in 3d scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR); 2021, p. 9401–9411.
[6] Adeli V, Ehsanpour M, Reid I, Niebles JC, Savarese S, Adeli E, Rezatofighi H. Tripod: Human trajectory and pose dynamics forecasting in the wild. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2021, p. 13390–13400.

[7] Corona E, Pumarola A, Alenya G, Moreno-Noguer F. Context-aware human motion prediction, In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR); 2020, p. 6992–7001.
[8] Wang JM, Fleet DJ, Hertzmann A. Gaussian process dynamical models. In: Proceedings of the neural information processing systems (NeurIPS), vol. 18. Citeseer; 2005, p. 3.
[9] Lehrmann AM, Gehler PV, Nowozin S. Efficient nonlinear markov models for human motion. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2014, p. 1314–1321.
[10] Martinez J, Black MJ, Romero J. On human motion prediction using recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2017, p. 2891–2900.
[11] Li C, Zhang Z, Lee WS, Lee GH. Convolutional sequence to sequence model for human dynamics. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2018, p. 5226–5234.
[12] Mao W, Liu M, Salzmann M, Li H. Learning trajectory dependencies for human motion prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV); 2019.
[13] Li M, Chen S, Zhao Y, Zhang Y, Wang Y, Tia Q. Dynamic Multiscale Graph Neural Networks for 3D Skeleton-Based Human Motion Prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR); 2020.
[14] Dang L, Nie Y, Long C, Zhang Q, Li G. MSR-GCN: Multi-Scale Residual Graph Convolution Networks for Human Motion Prediction. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2021, p. 11467–11476.
[15] Aksan E, Kaufmann M, Cao P, Hilliges O. A Spatio-temporal Transformer for 3D Human Motion Prediction. In: Proceedings of the international conference on 3D vision (3DV); 2021.
[16] Liu Z, Su P, Wu S, Shen X, Chen H, Hao Y, Wang M. Motion Prediction Using Trajectory Cues. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2021.
[17] Aksan E, Kaufmann M, Hilliges O. Structured Prediction Helps 3D Human Motion Modelling. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2019.
[18] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, Garnett R, editors. Proceedings of the neural information processing systems (NeurIPS), vol. 30. Curran Associates, Inc.; 2017.
[19] Wang J, Xu H, Narasimhan M, Wang X. Multi-person 3D motion prediction with multi-range transformers. In: Advances in neural information processing systems (NeurIPS), vol. 34. 2021.
[20] Ionescu C, Papava D, Olaru V, Sminchisescu C. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Trans Pattern Anal Mach Intell 2014.
[21] CMU. Carnegie-mellon mocap database. 2003, http://mocap.cs.cmu.edu.
[22] Fragkiadaki K, Levine S, Felsen P, Malik J. Recurrent network models for human dynamics. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2015, p. 4346–4354.
[23] Jain A, Zamir AR, Savarese S, Saxena A. Structural-rnn: Deep learning on spatio-temporal graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR); 2016, p. 5308–5317.
[24] Wang B, Shang L, Lioma C, Jiang X, Yang H, Liu Q, Simonsen JG. On Position Embeddings in BERT. In: Proceedings of the international conference on learning representations (ICLR); 2021.
[25] Wu K, Peng H, Chen M, Fu J, Chao H. Rethinking and Improving Relative Position Encoding for Vision Transformer. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV); 2021, p. 10033–10041.
[26] Horiuchi Y, Makino Y, Shinoda H. Computational Foresight: Forecasting Human Body Motion in Real-time for Reducing Delays in Interactive System. In: Proceedings of the 2017 ACM international conference on interactive surfaces and spaces; 2017.
[27] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S. PyTorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems (NeurIPS). Curran Associates, Inc.; 2019, p. 8024–35, URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
[28] Luo L, Xiong Y, Liu Y, Sun X. Adaptive Gradient Methods with Dynamic Bound of Learning Rate. In: Proceedings of the 7th International Conference on Learning Representations (ICLR); 2019, New Orleans, Louisiana.