[Array 15 (2022) 100212](https://doi.org/10.1016/j.array.2022.100212)

|  |  |  |
| --- | --- | --- |
|  | Contents lists available at [ScienceDirect](https://www.elsevier.com/locate/array) |  |
| Array |
| journal homepage: [www.elsevier.com/locate/array](http://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*

|  |  |  |
| --- | --- | --- |
| A R T I C L E | I N F O | A B S T R A C T |
| *Keywords:*  Human motion  Lie-algebra  Temporal aggregation Spatial aggregation  Attention |  | 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 tech-nology 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 informa-tion 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

∗ Corresponding a[uthor.](mailto:ueda.itsuki@image.iit.tsukuba.ac.jp)

*E-mail address:* [ueda.itsuki@image.iit.tsukuba.ac.jp](mailto:ueda.itsuki@image.iit.tsukuba.ac.jp) (I. Ueda).

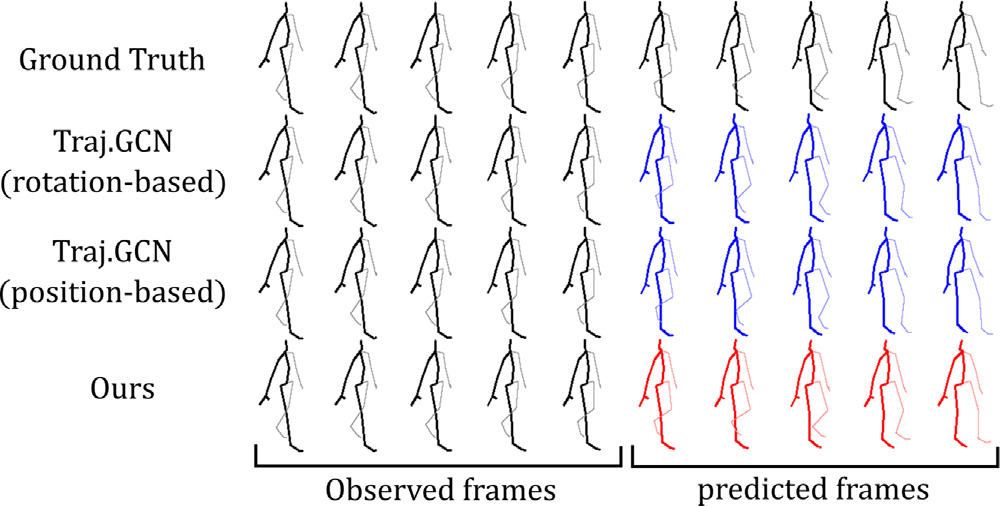
conventional networks by temporal and spatial aggregation and to obtain high generalization performance by incorporating each aggre-gation model by adding appropriate information to the framework of attention [18]. We propose a method to achieve a high generalization performance within each aggregation 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 gener-alization 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

<https://doi.org/10.1016/j.array.2022.100212>  
[Received 26 January 2022; Received in revis](https://doi.org/10.1016/j.array.2022.100212)ed form 18 June 2022; Accepted 21 June 2022   
Available online 23 June 2022   
[2590-0056](http://creativecommons.org/licenses/by-nc-nd/4.0/)/© 2022 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license ([http://creativecommons.org/licenses/by-](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

*I. Ueda et al.*  *Array 15 (2022) 100212*



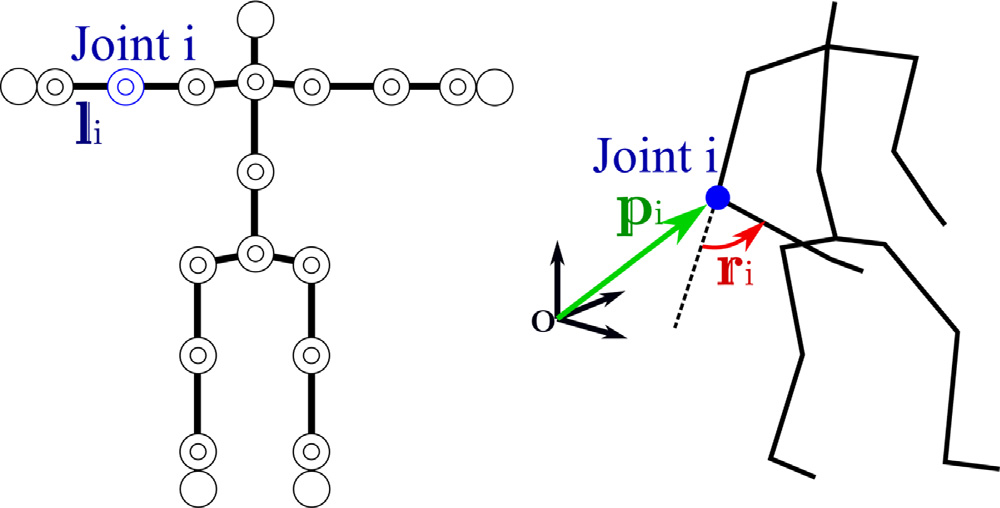
**Fig. 1.** 3D motion prediction: The model takes the observation sequence on the left

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 **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

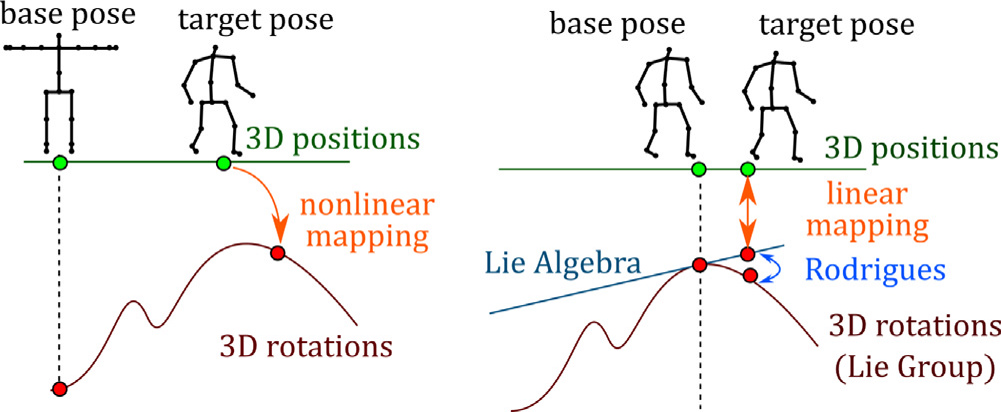
vector of joint angles, **𝐫***𝑖* ∈ R3. There are two types of angle-based descriptions.

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 re-gardless 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 fea-tures, they consider the feature dimensions retained by each node as

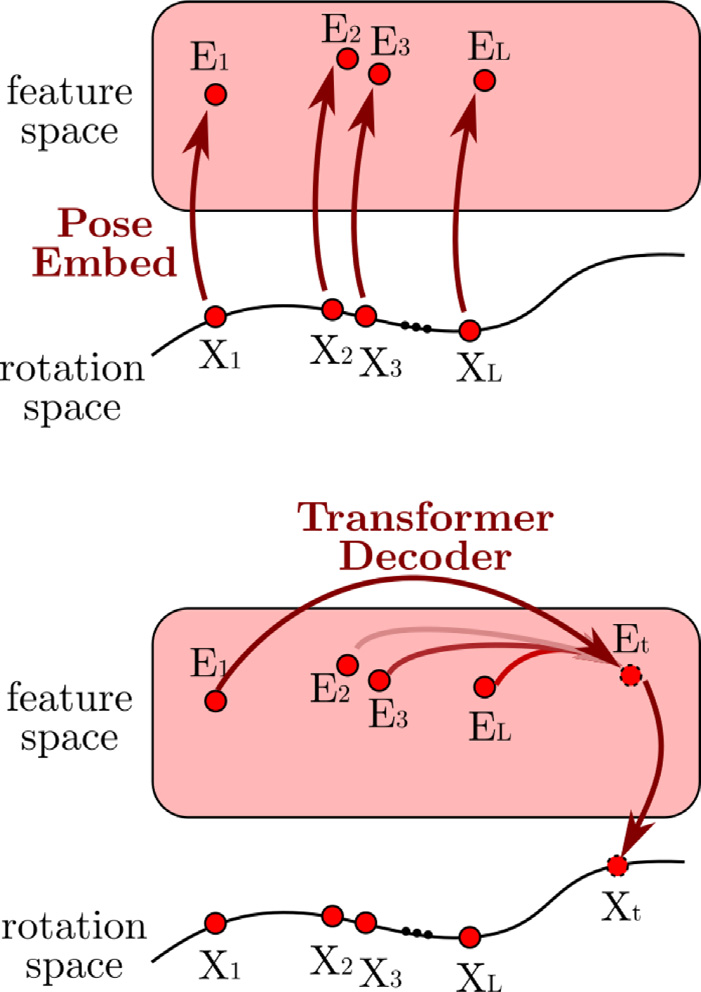
2

*I. Ueda et al.*  *Array 15 (2022) 100212*



**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 *𝑋𝑡* described by angle, aggregated by 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 pro-posed 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 Con-volutional 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,

3

*I. Ueda et al.*  *Array 15 (2022) 100212*

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 posi-tional 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 multiplica-tive 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: generaliza-tion 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 indepen-dent 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 dis-tance 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 describ-ing 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 con-straints 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 rela-tionship 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

4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *I. Ueda et al.* | | | | | | 5 | *Array 15 (2022) 100212* | | | |
|  | | | | | | The *𝑊* can be defined to map to *𝑆𝑂*(3) by the matrix exponential function. This means that we can compute the rotation matrix M*𝑡,𝑖* for frame *𝑡*, joint *𝑖* using the rotation vector M*𝑡,𝑖* relative to the input final frame as follows:  M*𝑡,𝑖* = *𝑒𝑥𝑝*([**𝐫***𝑡,𝑖* By using *𝑅𝑡* ]∧) M*𝐿,𝑖.* = [**𝐫***𝑡,*1*,* … *,* **𝐫***𝑡,𝑁*]*𝑇*as the angle description, we can (3)  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 collabo-rate with the angle-based description. For simplicity, we assume that the links are connected as 1*,* 2*,* … *, 𝑖,* … *, 𝑗* − 1*, 𝑗,* … *, 𝑁*. The offset of link *𝑖* at the reference pose is **𝐥***𝑖* ∈ R3, the rotation matrix from the reference pose is M*𝑖* ∈ R3×3, the translation of the root is **𝐥**0, and the rotation matrix M0 = I. The homogeneous coordinate transformation matrix *𝑇𝑖* by link *𝑖* is shown in Eq. (4).  *𝑇𝑖* = (M*𝑖*  **𝐥***𝑖*  1 ) (4) | | | |
| The angle of each joint provides the position **𝐩**(0) *𝑗*  in the world  coordinate system of joint *𝑗* by forward-kinematics in Eq. (2). | | | |
| We derive the behavior when link *𝑖* is further rotated by **𝐫***𝑖* in rotation  vector notation based on M1*,* … *,* M*𝐾*. The rotation matrix of joint *𝑖* after ( **𝐩**(0)  1  *𝑗*  ) = T0T1 ⋯ T*𝑗*−1⎛⎜⎜⎜⎜⎝ 0  0  1⎞⎟⎟⎟⎟⎠ (5) 0  applying **𝐫***𝑖* can be written as M′*𝑖*= *𝑒𝑥𝑝*([**𝐫***𝑖*]∧)M*𝑖*. The position **𝐩**(0)′of  joint *𝑗* after the transformation is as follows. | | | |
| We consider a joint coordinate system *𝑖* whose coordinate transforma-  tion from the world coordinate system is given by T−1 ( **𝐩**(0)′*𝑗*  1 ) = T0T1 ⋯(*𝑒𝑥𝑝*([**𝐫***𝑖*]∧) 0  1 ) T*𝑖* ⋯ T*𝑗*−1⎛⎜⎜⎜⎜⎝0  0  1⎞⎟⎟⎟⎟⎠ *𝑖*−1T−1 *𝑖*−2⋯ T−1 as the (6)  0  coordinate system based on link *𝑖*. For the position **𝐩**(*𝑖*) *𝑗*  of link *𝑗* in the  joint coordinate system *𝑖*, the following is established from Eq. (4).  ( **𝐩**(*𝑖*)  0  *𝑗* ) = T−1 *𝑖*−1T−1 *𝑖*−2⋯ T−1 ( **𝐩**(0)  0  *𝑗*  ) (7) | | | |
| **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 *𝑋𝑡* = [**𝐌***𝑡,*1*,* … *,* **𝐌***𝑡,𝑁*]. Each **𝐌***𝑡,𝑖* ∈ R3×3is 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 M*𝐿,𝑖*. The matrix *𝑊* ∈ R3×3on the tangent plane is multiplied by the source of the Lie algebra [**𝐫**]∧, represented by the rotation vector **𝐫** = [*𝑟*1*, 𝑟*2*, 𝑟*3]*𝑇*, as follows. | | | | | |
| ( **𝐩**(*𝑖*)  0  *𝑗*  By assigning them to Eq. (6), we get the following: ) = T*𝑖*T*𝑖*+1 ⋯ T*𝑗*−1⎛⎜⎜⎜⎜⎝0  0  1⎞⎟⎟⎟⎟⎠ 0  ( **𝐩**(*𝑖*)′*𝑗*  1 ) = (*𝑒𝑥𝑝*([**𝐫***𝑖*]∧) 0  1 ) ( **𝐩**(*𝑖*)  1  *𝑗* ) | | | (8)  (9) |
| In the neighborhood of M1*,* … *,* M*𝐾* ∈ *𝑆𝑂*(3), a good approximation is given by *𝑒𝑥𝑝*([**𝐫***𝑖*]∧) ∈ *𝑠𝑜*(3) using the source [**𝐫***𝑖*]∧ on the tangent plane. The mapping between the source and the position description on the tangent plane is as follows.  ( **𝐩**(*𝑖*)′ 1 *𝑗*  ) = (I + [**𝐫***𝑖*]∧ 0 1 ) ( **𝐩**(*𝑖*) 1 *𝑗* ) (10) | | | |
| **𝐩**(*𝑖*)′*𝑗* | − **𝐩**(*𝑖*) *𝑗* | = [**𝐫***𝑖*]∧**𝐩**(*𝑖*) *𝑗* | (11) |
| B = M*𝐿,𝑖* + [**𝐫**]∧ | | | (1) | | |
| From the anti-commutativity of the Lie bracket product, the expression (11) is transformed as follows: | | | |
| [**𝐫**]∧ = | 0 ⎛⎜⎜⎝−*𝑟*2 *𝑟*3 | −*𝑟*3 0 | *𝑟*2 −*𝑟*1 0 | ⎞⎟⎟⎠ | (2) |
| **𝐩**(*𝑖*)′*𝑗* | − **𝐩**(*𝑖*) *𝑗*= [**𝐩**(*𝑖*) *𝑗*]∧**𝐫***𝑖.* | | (12) |
| *𝑟*1 |

*I. Ueda et al.*  *Array 15 (2022) 100212*

Similarly, when each joint is rotated by **𝐫**1*,* … *,* **𝐫***𝐾*, the amount of

change in position is related to the amount of change in angle as

follows:

**𝐏**′− **𝐏** = (B *⊗* J)**𝐑**  (13)

R =

Note that B is the matrix representing the parent–child relationship

between the joints, which in this example is the unit lower triangular **𝐏** =

J = ⎛⎜⎜⎜⎝⎛⎜⎜⎜⎝⎛⎜⎜⎝  
 **𝐩**(0)

**𝐩**(0)

*𝑀*(0)

**𝐫***𝐾*   
**𝐫**1   
M(0)

⋮  
⋮

1

*𝐾*

1[**𝐩**(*𝑁*)

⎞⎟⎟⎠  
1[**𝐩**(0)⎞⎟⎟⎟⎠

*.*

⋮

1]∧

]∧  
 ⋯

⋱

⋯ *𝑀*(0)   
 *𝑀*(0)

*𝐾*[**𝐩**(*𝑁*)   
*𝐾*[**𝐩**(0)

⋮

*𝐾*]∧*𝐾*]∧⎞⎟⎟⎟⎠  
 (14)

(15)

(16)

matrix. The *𝐽* can be computed from *𝑃* using Eq. (2). Thus, by substi-

tuting *𝑃*(0) *𝐿*  for the position description at time *𝐿* in the reference pose

*𝑃* , *𝑃*(0) *𝑡*  for the position description at time *𝑡* in the destination pose *𝑃*′,

and **𝐑***𝑡* for the rotation **𝐑**, we can associate the angle description with

the position description **𝐑***𝑡*P*𝑡* − P*𝐿* = (B *⊗* J*𝐿*)**𝐑***𝑡*.

**𝐩**(0) *𝑡,*1 **𝐩**(0) *𝐿,*1

P*𝑡* =

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 ⎛⎜⎜⎜⎝ **𝐩**(*𝑁*) *𝑡,*1 ⋮ ⎞⎟⎟⎟⎠ *,* P*𝐿* =⎛⎜⎜⎜⎝ **𝐩**(*𝑁*) *𝐿,*1 ⋮ ⎞⎟⎟⎟⎠ (17)

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

P*𝑡* ∈ R*𝑁*×3*𝑁*is used as the input position-based description.

*𝑇*

**𝐩**(0) *𝑡,*1 ⋯ **𝐩**(*𝑁*) *𝑡,*1

P*𝑡* =

*3.2. Embedding of human pose information* ⎛⎜⎜⎜⎝ **𝐩**(0) *𝑡,𝑁* ⋮ ⋱

⋯ **𝐩**(*𝑁*) *𝑡,𝑁* ⋮ ⎞⎟⎟⎟⎠ (18)

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 G ∈

R*𝐿*×*𝑁*×*𝐶,* E ∈ R*𝐿*×*𝑁*×*𝐷*, respectively. C is the number of elements in each

description, e.g., 3*𝐾* 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., 3*𝐾* 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 **𝐠***𝑡*1*,𝑖,* **𝐠***𝑡*2*,𝑖* at time *𝑡*1*, 𝑡*2

and joint *𝑖* are directly comparable. On the other hand, for space, the

feature values **𝐠***𝑡,𝑖*1*,* **𝐠***𝑡,𝑖*2 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 *𝐺𝑡* ∈ R*𝑁*× *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 *𝐸𝑡* ∈ R*𝑁*×*𝐷*.

6

*I. Ueda et al.*  *Array 15 (2022) 100212*

the ReLU or exponential function. By aggregating *𝑉𝑡* according to the Attention map and projecting it to the representation space with weights W*𝑂 𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙*∈ R*𝐷*×*𝐷*, the output *𝐸*′*𝑡*is calculated as follows:

E′*𝑡*= A*𝑡*V*𝑡*W*𝑂 𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙.*  (25)

Eq. (25) gives the interpretation that the adjacency matrix is a dynam-ically 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*,* … *,* **𝐞***𝑖,𝐿*]*,* E′*𝑖*= [**𝐞**′*𝑖,*1*,* … *,* **𝐞**′*𝑖,𝐿*] ∈ R*𝐿*×*𝐷.*  (26)

Using the weights W*𝑄 𝑠𝑝𝑎𝑡𝑖𝑎𝑙,* W*𝐾 𝑠𝑝𝑎𝑡𝑖𝑎𝑙,* W*𝑉 𝑠𝑝𝑎𝑡𝑖𝑎𝑙,* W*𝑂 𝑠𝑝𝑎𝑡𝑖𝑎𝑙*∈ R*𝐷*×*𝐷*, the tempo-ral aggregation at joint *𝑖* is calculated as follows:

Q*𝑖* = E*𝑖*W*𝑄 𝑠𝑝𝑎𝑡𝑖𝑎𝑙*  (27)

K*𝑖* = E*𝑖*W*𝐾 𝑠𝑝𝑎𝑡𝑖𝑎𝑙*  (28)

V*𝑖* = E*𝑖*W*𝑉 𝑠𝑝𝑎𝑡𝑖𝑎𝑙*  (29)

A*𝑖* = *𝜎*(Q*𝑖*K*𝑇 𝑖*+ M) (30)

E′*𝑖*= *𝐴𝑖𝑉𝑖*W*𝑂 𝑠𝑝𝑎𝑡𝑖𝑎𝑙*  (31)

Note that Q*𝑖,* K*𝑖,* V*𝑖* ∈ R*𝐿*×*𝐷,* M*,* A*𝑖* ∈ R*𝐿*×*𝐿*. The *𝑀* is a mask to avoid referring to the future during inference, which is *𝑂* for the encoder and−∞ 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 infor-mation 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 Sec-tion 3.2 just before the network’s input. Proximity on the time axis

*𝑍𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙*= [*𝑧𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙 𝑡,𝑐*  ] ∈ R(*𝐿*+*𝑆*)×*𝐷*, proximity on the spatial axis *𝑍𝑠𝑝𝑎𝑡𝑖𝑎𝑙*=

[*𝑧𝑠𝑝𝑎𝑡𝑖𝑎𝑙 𝑖,𝑐*  ] ∈ R*𝑁*×*𝐷*, the feature of the pose embedding at time *𝑡*, joint *𝑖*, and feature dimension *𝑐* is *𝑒𝑙 𝑡,𝑖,𝑐*, then the value *𝑒𝑡,𝑖,𝑐* after APE assignment is as follows:

*𝑒𝑡,𝑖,𝑐* = *𝑒𝑙 𝑡,𝑖,𝑐*+ *𝑧𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙* + *𝑧𝑠𝑝𝑎𝑡𝑖𝑎𝑙 𝑖,𝑐*  (32)

In this method, we use a fully learnable APE with the learnable parameters *𝑍𝑡𝑒𝑚𝑝𝑜𝑟𝑎𝑙, 𝑍𝑠𝑝𝑎𝑡𝑖𝑎𝑙*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 *𝑧𝑟𝑒𝑙𝑎𝑡𝑖𝑣𝑒 𝑡*1−*𝑡*2*,𝑐*, then Eq. (24) can be replaced by the following equation

A*𝑡* = *𝜎*(Q*𝑡*K*𝑇 𝑡*+ Q*𝑡*Z*𝑟𝑒𝑙𝑎𝑡𝑖𝑣𝑒* ) (33)

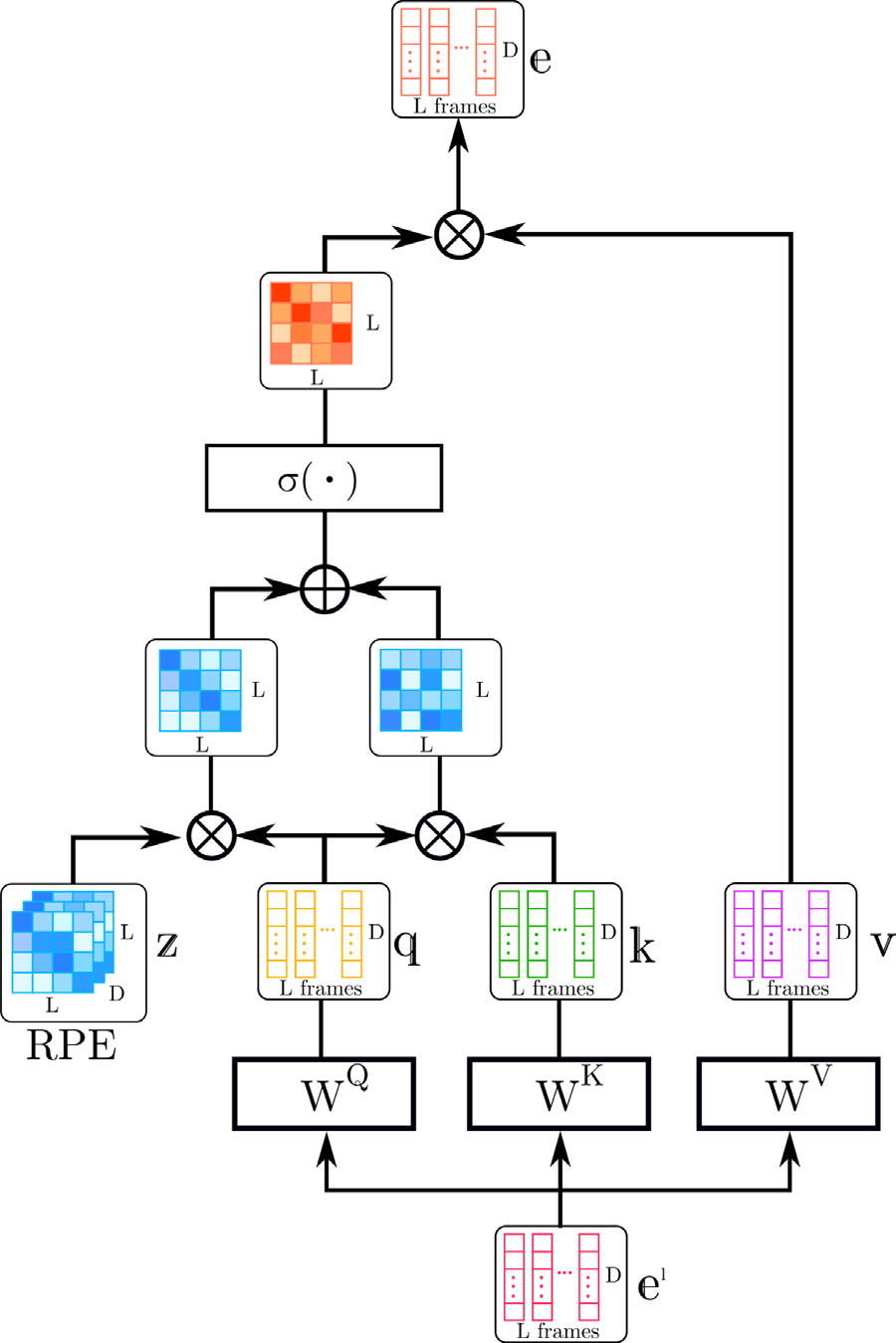
Z*𝑟𝑒𝑙𝑎𝑡𝑖𝑣𝑒*

proximity *𝑧𝑟𝑒𝑙𝑎𝑡𝑖𝑣𝑒*   
*𝑡*

Our method uses a fixed cosine RPE in the time axis and sets the =⎛⎜⎜⎝*𝑧𝑡*−1*,𝐷*

*𝑡*1−*𝑡*2*,𝑖,𝑐*as follows:  
 ⋮ ⋱

⋯ *𝑧𝑡*−*𝐿,𝐷* ⋮ ⎞⎟⎟⎠ (34) *𝑧𝑡*−1*,*1 ⋯ *𝑧𝑡*−*𝐿,*1



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

|  |  |
| --- | --- |
| system calculated by forward kinematics is [**𝐩**(0) *𝑡,*1*,* … *,* **𝐩**(0) *𝑡,𝑁*], and their  respective true values are [M*𝐺𝑇 𝑡,*1*,* … *,* M*𝐺𝑇 𝑡,𝑁*][**𝐩***𝐺𝑇 𝑡,*1*,* … *,* **𝐩***𝐺𝑇 𝑡,𝑁*]. The angular  error *𝜃𝑡,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  *𝜃𝑡,𝑖* = *𝑎𝑐𝑜𝑠* (tr(M*𝑇 𝑡,𝑖*M*𝐺𝑇*  2 *𝑡,𝑖*) − 1 ) (36)  The overall angle error *𝐿𝑟𝑜𝑡* is as follows.  L*𝑟𝑜𝑡* =  *𝑡*=*𝐿*+1  *𝐿*+*𝑆*  ∑√√√√∑*𝜃*2 *𝑡,𝑖*  (37)  The position error *𝐿𝑝𝑜𝑠* summarizes the Euclidean distance of the posi-  tion for each joint as follows: | |
| *𝐿*+*𝑆*  L*𝑝𝑜𝑠* =  **4. Experiments** *𝑡*=*𝐿*+1∑∑|**𝐩**(0) *𝑡,𝑁*− **𝐩***𝐺𝑇 𝑡,𝑁*| | (38) |

*4.1. Datasets*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *𝑧𝑟𝑒𝑙𝑎𝑡𝑖𝑣𝑒 𝑡*1−*𝑡*2*,𝑖,𝑐*= *𝑐𝑜𝑠*( | *𝑡*1 − *𝑡*2  100002*𝑐*∕*𝐷* ) | (35) | 7 | We verify the performance of the model on the motion capture dataset Human3.6M [20]. The Human3.6M dataset contains 15 dif-ferent 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 |
| *3.5. Training* | | |
| In the training, we use *𝐿𝑟𝑜𝑡* + *𝜆𝐿𝑝𝑜𝑠* as the objective function, which is a composite of the angle error *𝐿𝑟𝑜𝑡* and the position error *𝐿𝑝𝑜𝑠* with the hyperparameter *𝜆* ∈ R+. The output for time *𝑡*(*𝐿* + 1 ≤ *𝑡* ≤ *𝐿* + *𝑆*) is *𝑌𝑡*−*𝐿* =[M*𝑡,*1*,* … *,* M*𝑡,𝑁* ], and the joint position in the world coordinate | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *I. Ueda et al.* | | | | | | | | | |  |  | Smoking | | |  |  | *Array 15 (2022) 100212* | | | |
| **Table 1**  Evaluation results of position error in H3.6M: ↓ MPJPE [mm]. | | | | | | | | | |
| Scenarios | Walking | Eating | | | | | | | |  |  |  |  | Discussion | | | |
| ms | 80 | 160 | 320 | | 400 | | 80 | | 160 | 320 | 400 | 80 | 160 | | 320 | 400 | 80 | 160 | 320 | 400 |
| Res-sup.  DMGNN  Traj-GCN MSR-GCN | 29.36  17.32  12.29  12.16 | 50.82 30.67 23.03 22.65 | | 76.03 54.56 39.77 38.64 | | 81.51 65.20 46.12 45.24 | | 16.84 10.96 8.36  8.39 | 30.60  21.39  16.90  17.05 | 56.92 36.18 33.19 33.03 | 68.65 43.88 40.70 40.43 | 22.96 8.97  7.94  8.02 | 42.64  17.62  16.24  16.27 | | 70.14 32.05 31.90 31.32 | 82.68 40.30 38.90 38.15 | 32.94 17.33 12.50 11.98 | 61.18 34.78 27.40 26.76 | 90.92  61.03  58.51  57.08 | 96.19 69.80 71.68 **69.74** |
| Ours w/o Enc Ours w/o RPE Ours w/o Pos Ours w/o Rot Ours | 12.90  12.65  10.72  12.69  **8.55** | 23.77 22.24 20.71 22.81 **18.06** | | 39.36 38.28 36.41 39.63 **34.78** | | 46.19 45.39 42.50 46.66 **41.85** | | 8.70 8.03 7.40 9.27 **6.06** | 16.94  17.43  15.87  17.70  **13.68** | 33.24 33.02 33.11 33.69 **30.00** | 40.68 40.27 41.20 41.32 **38.01** | 7.88  8.16  7.23  8.80  **5.64** | 16.32  16.32  15.29  16.58  **12.68** | | 31.78 31.15 31.62 31.10 **28.27** | 38.86 38.32 39.14 37.92 **36.10** | 12.64 11.96 11.30 14.12 **9.25** | 27.55 26.84 27.38 29.39 **23.42** | 58.55  57.22  61.79  60.87  **55.87** | 71.72 69.63 75.91 74.55 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.  DMGNN  Traj-GCN MSR-GCN | 35.36  13.14  8.97  8.61 | 57.27 24.62 19.87 19.65 | | 76.30 64.68 43.35 43.28 | | 87.67 81.86 53.74 53.82 | | 34.46 23.30 18.65 16.48 | 63.36  50.32  38.68  36.95 | 124.60 107.30 77.74  77.32 | 142.50 132.10 93.39  93.38 | 37.96 12.47 10.24 10.10 | 69.32  25.77  21.02  20.74 | | 115.00 48.08  42.54  41.51 | 126.73 58.29  52.30  51.26 | 36.10 15.27 13.66 12.79 | 69.12 29.27 29.89 29.38 | 130.46 71.54  66.62  66.95 | 157.08 96.65  84.05  85.01 |
| Ours w/o Enc Ours w/o RPE Ours w/o Pos Ours w/o Rot Ours | 8.88  8.28  8.12  10.52  **6.62** | 19.30 19.82 19.98 22.05 **16.75** | | 43.67 43.31 46.89 46.81 **41.27** | | 53.46 53.42 58.59 58.13 **53.01** | | 18.36 16.84 16.51 19.54 **13.67** | 38.60  36.78  38.45  40.14  **33.23** | 77.50 77.41 81.90 80.19 **74.42** | 93.96 93.46 98.69 96.54 **92.18** | 10.32 10.12 9.28  11.69 **7.66** | 21.20  20.85  20.71  23.10  **17.60** | | 42.73 41.59 44.85 45.30 **39.45** | 52.41 51.30 56.02 55.67 **50.15** | 13.96 12.85 12.59 16.75 **9.74** | 29.27 29.45 30.95 34.64 **24.64** | 66.89  66.78  72.14  72.31  **60.85** | 84.32 85.21 90.70 90.36 **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.  DMGNN  Traj-GCN MSR-GCN | 36.33  21.35  15.60  14.75 | 60.30 38.71 32.78 32.39 | | 86.53 75.67 65.72 66.13 | | 95.92 92.74 79.25 79.64 | | 42.55 11.92 10.62 10.53 | 81.40  25.11  21.90  21.99 | 134.70 44.59  46.33  46.26 | 151.78 50.20  57.91  57.80 | 47.28 14.95 16.14 16.10 | 85.95  32.88  31.12  31.63 | | 145.75 77.06  61.47  62.45 | 168.86 93.00  75.46  76.84 | 26.10 13.61 9.88  9.89 | 47.61 28.95 20.89 21.01 | 81.40  45.99  44.95  44.56 | 94.73 58.76 56.58 **56.30** |
| Ours w/o Enc Ours w/o RPE Ours w/o Pos Ours w/o Rot Ours | 15.78  14.05  13.99  17.74  **11.50** | 32.25 32.89 32.45 35.83 **27.89** | | 65.62 66.65 68.59 69.06 **63.05** | | 79.90 79.70 83.03 82.61 **78.37** | | 10.91 10.17 10.02 12.97 **8.00** | 21.14  19.87  21.92  25.31  **18.65** | 46.12 46.96 49.39 50.27 **43.70** | 57.47 56.95 62.45 62.06 **56.42** | 16.64 13.68 15.57 18.48 **11.93** | | 31.39 28.85 32.03 35.06 **26.71** | 61.13 60.11 67.27 67.88 **59.97** | 75.70 76.46 83.24 83.40 **76.38** | 9.53  9.62  9.51  12.11 **7.39** | 20.99 21.30 21.46 24.37 **17.71** | 44.26  44.33  48.81  49.93  **43.52** | 57.12 56.91 61.47 62.29 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.  DMGNN  Traj-GCN MSR-GCN | 30.62  12.20  11.43  10.68 | 57.82 24.17 23.99 23.06 | | 106.22 59.62  50.06  48.25 | | 121.45 77.54  61.48  59.23 | | 64.18 47.09 23.39 20.65 | 102.10 93.33  46.17  42.88 | 141.07 160.13 83.47  80.35 | 164.35 171.20 95.96  93.31 | 26.79 14.34 10.47 10.56 | 50.07  26.67  21.04  20.92 | | 80.16 50.08 38.47 37.40 | 92.23 63.22 45.19 43.85 | 34.66 16.95 12.68 12.11 | 61.97 33.62 26.06 25.56 | 101.08 65.90  52.27  51.64 | 115.49 79.65  63.51  62.93 |
| Ours w/o Enc Ours w/o RPE Ours w/o Pos Ours w/o Rot Ours | 11.54  9.12  9.97  12.41  **7.92** | 24.21 20.06 22.84 24.15 **18.88** | | 51.33 47.52 50.73 47.99 **44.39** | | 62.58 59.73 63.27 59.24 **56.95** | | 23.93 19.36 19.53 23.04 **16.25** | 46.31  39.87  41.55  44.47  **36.99** | 83.79 78.92 78.79 80.19 **75.67** | 96.52 93.45 91.81 93.24 **91.42** | 10.63 9.27  9.80  11.56 **7.34** | 21.34  19.41  20.43  21.37  **16.31** | | 38.54 36.82 37.44 37.51 **34.24** | 45.22 44.16 44.04 44.33 **42.45** | 13.65 11.01 11.44 14.11 **9.17** | 26.67 21.55 25.47 27.80 **21.55** | 52.36  49.86  53.98  54.18  **48.63** | 63.88 62.90 66.14 65.89 **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 av-erage 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

8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *I. Ueda et al.* | | | | | | | | | |  |  | Directing Traffic | |  |  | *Array 15 (2022) 100212* | | | | | |
| **Table 2**  Evaluation results of position error in CMU-Mocap: ↓ MPJPE [mm]. | | | | | | | | | |
| Motion | Basketball | Basketball Signal | | | | | | | |  |  |  |  | Jumping | | | | | |
| Milliseconds | 80 | 160 | 320 | | 400 | | 80 | | 160 | 320 | 400 | 80 | 160 | 320 | 400 | 80 | 160 | 320 | | 400 | |
| Res-sup.  DMGNN  Traj-GCN MSR-GCN | 15.45  15.57  11.68  10.28 | 26.88 28.72 21.26 18.94 | | 43.51 59.01 40.99 37.68 | | 49.23 73.05 50.78 47.03 | | 20.17 5.03  3.33  3.03 | 32.98 9.28  6.25  5.68 | 42.75 20.21 13.58 12.35 | 44.65 26.23 17.98 16.26 | 20.52  10.21  6.92  5.92 | 40.58 20.90 13.69 12.09 | 75.38 41.55 30.30 28.36 | 90.36 52.28 39.97 38.04 | 26.85 31.97 17.18 14.99 | 48.07 54.32 32.37 28.66 | 93.50 96.66 60.12 55.86 | | 108.90 119.92 72.55  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.  DMGNN  Traj-GCN MSR-GCN | 25.76  17.42  14.53  12.84 | 48.91 26.82 24.20 20.42 | | 88.19 38.27 37.44 30.58 | | 100.80 40.08  41.10  **34.42** | | 17.75 14.86 13.33 10.92 | 31.30 25.29 24.00 19.50 | 52.55 52.21 43.77 37.05 | 61.40 65.42 53.20 46.38 | 44.35  9.57  6.62  6.31 | 76.66 15.53 10.74 10.30 | 126.83 26.03  17.40  17.64 | 151.43 30.37  20.35  21.12 | 22.84 7.93  5.96  5.49 | 44.71 14.68 11.62 11.07 | 86.78 33.34 24.77 25.05 | | 104.68 44.24  31.63  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 con-siderable 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 ad-dition, 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.

constraints such as connection relation and link length invariance by

using angle as output. By selecting the appropriate architecture for ex-

tracting 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, Val-

idation, 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.

**5. Discussion and future works**  **References**

This paper proposes a method for extensive temporal and spatial information aggregation using Self-Attention. For temporal aggrega-tion, 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 neigh-borhood 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

9

*I. Ueda et al.*  *Array 15 (2022) 100212*

[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.](http://refhub.elsevier.com/S2590-0056(22)00057-1/sb8)

[9] [Lehrmann AM, Gehl](http://refhub.elsevier.com/S2590-0056(22)00057-1/sb8)er 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.](http://refhub.elsevier.com/S2590-0056(22)00057-1/sb18)

10