Pattern Recognition 143 (2023) 109806

Contents lists available at ScienceDirect

Pattern Recognition

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

KD-Former: Kinematic and dynamic coupled transformer network for 3D human motion prediction

Ju Dai[a,1], Hao Li[a,b,1], Rui Zeng[a,b], Junxuan Bai[c], Feng Zhou[d], Junjun Pan[a,b,*]

[a] Peng Cheng Laboratory, Shenzhen, China
[b] Beihang University, Beijing, China
[c] Capital University of Physical Education and Sports, Beijing, China
[d] North China University of Technology, Beijing, China

ARTICLE INFO

Article history:
Received 9 October 2022
Revised 19 May 2023
Accepted 5 July 2023
Available online 6 July 2023

Keywords:
Human motion prediction
Motion kinematics
Motion dynamics
Transformer

ABSTRACT

Recent studies have made remarkable progress on 3D human motion prediction by describing motion with kinematic knowledge. However, kinematics only considers the 3D positions or rotations of human skeletons, failing to reveal the physical characteristics of human motion. Motion dynamics reflects the forces between joints, explicitly encoding the skeleton topology, whereas rarely exploited in motion prediction. In this paper, we propose the **K**inematic and **D**ynamic coupled trans**Former** (KD-Former), which incorporates dynamics with kinematics, to learn powerful features for high-fidelity motion prediction. Specifically, We first formulate a reduced-order dynamic model of human body to calculate the forces of all joints. Then we construct a non-autoregressive encoder-decoder framework based on the transformer structure. The encoder involves a kinematic encoder and a dynamic encoder, which are respectively responsible for extracting the kinematic and dynamic features for given history sequences via a spatial transformer and a temporal transformer. Future query sequences are decoded in parallel in the decoder by leveraging the encoded kinematic and dynamic information of history sequences. Experiments on Human3.6M and CMU MoCap benchmarks verify the effectiveness and superiority of our method. Code will be available at: https://github.com/wslh852/KD-Former.git.

1. Introduction

3D human motion prediction aims to predict the most possible future human postures based on past observed motion data. The task is a fundamental research topic in the computer community. It plays a critical role in a broad spectrum of applications, such as human-computer interaction [1], autonomous vehicles [2] and character animation [3]. Since human behavior is endowed with inherent uncertainty and high complexity, accurately predicting high-fidelity human motion is inherently challenging.

To address the above difficulties, traditional methods such as the hidden Markov model [4] and finite Boltzmann machine [5] are probability-based models. These models require expert-level prior knowledge and may produce non-plausible poses. Due to the progressive advances of deep learning in various vision tasks [6], it has become the dominant technology in 3D human motion prediction. Among various architectures, recurrent neural networks (RNNs) receive considerable preference because it is structured to be good at capturing video sequence temporal information [7,8]. Those methods regard motion prediction as a sequence-to-sequence (seq2seq) autoregressive prediction. The results have shown significant advantages over the convolutional neural networks (CNNs) based technique [9]. However, RNNs-based approaches are highly prone to error accumulations, especially when conducting long-term motion forecasts. To enhance the RNNs-based performance, several studies attempt to incorporate skeleton structure in the encoding and decoding via graph convolutional networks (GCNs) [10,11]. Although promising results have been achieved, those methods have suffered by large model parameters and time-consuming single-step implementation. Since the breakthrough performance of transformer in natural language processing [12], the transformer has become the new prevalence in seq2seq tasks [13]. Thanks to the global modeling ability of the attention mechanism, the transformer demonstrates outstanding performance on motion prediction [14,15]. Whereas native transformer-based models often use an autoregressive strategy to predict the next token based on pre-

* Corresponding author.
E-mail addresses: daij@pcl.ac.cn (J. Dai), haolirj@buaa.edu.cn (H. Li), zrsammy@buaa.edu.cn (R. Zeng), baijx6@163.com (J. Bai), zhoufeng@ncut.edu.cn (F. Zhou), pan_junjun@buaa.edu.cn (J. Pan).
[1] The first two authors contribute equally to this work.

https://doi.org/10.1016/j.patcog.2023.109806
0031-3203/

vious tokens to yield target sequences [12,14], which can not give full play to the parallel computing ability of neural networks and prolong the processing time.

Nevertheless, existing deep learning-based methods only consider the kinematic data of isolated skeleton joints as network input, *i.e.*, joint positions or rotations. These methods neglect the high-order interactions between skeleton segments. Although GCNs can explicitly characterize the topology of the human skeleton, the network inputs still focus on kinematic knowledge [10,16]. Besides kinematics, motion can also be modeled from the perspective of the dynamics [17]. In principle, motion dynamics are derived from kinematic clues based on inverse dynamics with mechanical models [18]. Its acquisition builds upon the human skeleton structure, involving the length of skeleton segments and the velocity and acceleration of skeleton joints. For the structured human skeleton, motion dynamic clues are capable of modeling interactive effects between joints in hierarchy hinge structure implicitly [19]. Hence, motion dynamics contain expressive higher-order information. Its powerful representation abilities have been used to distinguish different motions that are indistinguishable in the kinematic space [20]. In this paper, kinematic data denotes joint rotations, while dynamic data refers to the joint forces. Kinematic and dynamic knowledge are complimentary, characterizing motion from different aspects. However, the physical properties and joint forces of motion dynamics are rarely exploited in human motion prediction.

To utilize the complementary characteristics of motion kinematic and dynamic clues, we propose the **K**inematic and **D**ynamic coupled trans**Former** (KD-Former) network for 3D human motion prediction. To our knowledge, we are the first to introduce motion dynamics for motion prediction. To obtain dynamic data, we formulate a simplified reduced-order algorithm to calculate joint forces. The whole framework is a non-autoregressive seq2seq encoder-decoder framework based on the transformer. Specifically, the encoder module involves a kinematic encoder (K-Encoder) and a dynamic encoder (D-Encoder), which are respectively responsible for extracting strong expressive features of kinematics and dynamics for given motion sequences. The K-Encoder and D-Encoder are constructed with a spatial transformer and a temporal transformer to exploit the spatial context of skeleton joints and the temporal context of different frames. The decoder module consists of a spatial transformer and two temporal transformers. The kinematic spatial transformer is shared for the encoder and decoder, while the temporal transformers in the decoder are designed based on a cross-attention unit to establish cross-correlation between future frames and history sequences. Our model forecasts 3D human motion in a non-autoregressive manner, which can significantly enhance training and inferencing time. Experiments on Human3.6M [21] and CMU MoCap [22] prove the superiority of KD-Former.

In summary, **our main contributions** are listed as follows:

- We propose a novel non-autoregressive kinematic and dynamic coupled transformer network, which is elaborately designed to couple motion kinematic and dynamic information for 3D human motion prediction.
- To our knowledge, we are the first to introduce motion dynamics for human motion prediction. To obtain dynamic data, we formulate a simplified reduced-order algorithm of human body, which largely enhances the computation efficiency and prediction errors.
- Extensive experiments on Human 3.6M and CMU MoCap datasets demonstrate the superior performance of incorporating dynamics and the rapid reasoning ability of our non-autoregressive decoding strategy.

2. Related work

2.1. Human motion prediction

3D human motion prediction has long been a hot research topic in the computer community. Traditional methods use probability models to predict motion, such as hidden Markov model [4] and Boltzmann model [5]. However, those probabilistic modeling approaches have limited prediction accuracy, may generate implausible results, and require strong expert priors.

Recently, deep learning has become the prevalent technique. Among various architectures, RNNs are the most favored candidate. Martinez et al. [7] develop a seq2seq architecture with residual connections for posture prediction. Wang et al. [8] propose the position-velocity RNN (PVRNN) framework, which can simultaneously encode joint rotations and velocities to reinforce feature capability. Pavllo et al. [23] design an RNN architecture based on quaternions and verify its advantages over exponential maps. Dong and Xu [24] incorporate action class labels into GRU-based prediction network. Although RNN-based methods can extract expressive temporal information, they neglect the skeleton topology and suffer from error accumulation.

As the human skeleton is intrinsically a naturally connected graph with joints as nodes and bones as edges, several endeavors resort to graph neural networks (GNNs) to facilitate motion analysis. For instance, Jain et al. [25] construct spatio-temporal graphs according to the topology of skeleton sequences for motion prediction. Mao et al. [26] attempt to learn temporal information via discrete cosine transform and spatial structure with GNNs. Works in [10,11] formulate multi-scale GNNs to model the internal relations of the human body at different scales. Zhong et al. [27] formulate the spatio-temporal gating-adjacency GCN(GAGCN) to learn complex spatio-temporal dependencies over diverse action types. Besides GCNs, extensive efforts exploit the attention mechanism for modeling joint corporations within each frame. For instance, Martínez-González et al. [28] leverage transformer-based encoder-decoder architecture for fast human motion inference in a non-autoregressive manner. Nevertheless, existing GCNs or attention-based models describe human postures and movements using kinematic data, such as joint positions, rotations, and quaternions as inputs. In contrast, we introduce motion dynamics to characterize motion, which is capable of implicitly modeling human body high-order structure.

2.2. Motion dynamics applications

Learning the dynamic information of human motion is essential for mechanical research [29]. Motion dynamics has been widely applied in motion control, rehabilitation monitoring, and behavior analysis. For example, Reher and Ames [30] use an inverse dynamics approach to control the motion of a walking robot to achieve stability conditions for the robot. In the medical field, dynamic data are often regarded as health analysis data, as in [31], where patient dynamic data are analyzed and monitored by capturing human kinematic data. Zell and Rosenhahn [32] present a learning-based inverse dynamic algorithm to analyze human motion and use it as a tool to detect abnormal torque distribution in gait. Mansur et al. [20] leverage dynamic features derived by applying inverse dynamics to the human body for action recognition. Our goal is different from the above works. We aim to incorporate dynamic data into kinematic data to strengthen the representation ability of motion features, so as to benefit motion prediction performance.

3. Methodology

3.1. Overview of the proposed method

Our purpose is to make use of the complementary information characteristics of motion kinematics and dynamics for prediction performance improvements. Since existing datasets are only endowed with kinematic data, such as joint positions or rotations, we first need to obtain the dynamic data. Inspired by the reduced-order dynamic mode in super-tall complex building structures [33], we present a simplified reduced-order algorithm of human body to obtain dynamic clues of skeleton joints. Then, we formulate the **K**inematics and **D**ynamics coupled trans**Former** (KD-Former) network to leverage the benefits of motion kinematic and dynamic. Our KD-Former is a seq2seq architecture based on the transformer. Unlike the conventional autoregressive decoder of transformer networks in natural language processing [12], the proposed KD-Former possesses a non-autoregressive decoding manner, which can significantly speed up the training and testing process.

Before going into the details of the method, we first describe how the dynamic data are obtained. Then we introduce the essential components of the transformer. At last, we elaborate on the proposed KD-Former model.

3.2. Dynamics calculation

The dynamic computation requires the mass of each part of the human body and its motion information. It is normally solved by Newton Euler (N-E) equation, where the Newton equation calculates the translation process, and the Euler equation calculates the rotation process [19]. In traditional mechanical models, a human body is usually modeled as a rigid body, where the inertia tensor and mass are needed to be solved and later brought into the N-E equation to obtain dynamic data [19]. However, in the process of human body modeling, complex and delicate modeling often brings a significant computational overhead. In structural engineering, to analyze the stresses of a complex structure, the complex structure is simplified into a reduced-order dynamic model similar to "sugar gourd string" [33]. Inspired by Niu et al. [33], we establish a reduced-order model of the human skeleton, where each joint is regarded as a mass point, and a bone mass is condensed to the connected joints. Note that our dataset has no ground reaction forces (GRF), so we make a simplifying assumption and do not consider GRF. Only the gravity force, internal joint forces, and inertia forces are considered in the motion equation. As shown in Fig. 2, we divide the human body into five parts, each of which is considered a hinge structure. Assuming that the sectional density between two connected joints is uniformly distributed, a joint mass is proportional to the 3D distance between two joints. The joint mass $m_{k,j}$ of joint j in part k can be defined as:

$$m_{k,j} = \frac{\sqrt{(\mathbf{x}_{k,j} - \mathbf{x}_{k,j+1})^2}}{\sum_{k=1}^{5}\sum_{j=1}^{n_k}\sqrt{(\mathbf{x}_{k,j} - \mathbf{x}_{k,j+1})^2}}, \tag{1}$$

where $\mathbf{x}_{k,j}$ and $\mathbf{x}_{k,j+1}$ are the spatial positions for joint j in part k and its parent joint. n_k is the joint number of the part k.

To intuitively illustrate the calculation process of dynamic data, we take the end joint and its parent joint of the right leg as an example to display the details. We remove the part index k and regard the labels of the end joint and its parent joint as 1 and 2 to simplify the representation. Thus the acceleration $\mathbf{a}_1^t$ of the end joint of the right leg at frame t is formulated as:

$$\mathbf{a}_1^t = \mathbf{x}_1^{t-1} + \mathbf{x}_1^{t+1} - 2 * \mathbf{x}_1^t, \tag{2}$$

where $\mathbf{x}_1^{t-1}$, $\mathbf{x}_1^t$,$\mathbf{x}_1^{t+1}$ are spatial positions of the end joint at frame $t-1$, t and $t+1$. Then, according to the dynamic equilibrium equation, we have:

$$m_1\mathbf{a}_1^t = \mathbf{f}_1^t + m_1\mathbf{g}, \tag{3}$$

where $\mathbf{f}_1^t$ is the joint forces (dynamic information) of the end joint for right leg at frame t, and $\mathbf{g}$ is the gravity. According to Newton's third law, the reaction force of the end joint on its parent joint $\hat{\mathbf{f}}_1^t$ is:

$$\hat{\mathbf{f}}_1^t = -\mathbf{f}_1^t. \tag{4}$$

Hence, the dynamic equilibrium equation for its parent joint is:

$$m_2\mathbf{a}_2^t = \mathbf{f}_2^t + m_2\mathbf{g} + \hat{\mathbf{f}}_1^t, \tag{5}$$

where $\mathbf{f}_2^t$ is the force of the 2th joint in part k for frame t. The dynamic information of a frame $\mathbf{x}^{md}$ can be solved iteratively from the end joint to the root joint. A video with n frames is expressed as $\mathbf{X}^{md} \in \mathbb{R}^{n\times J\times 3}$.

3.3. Preliminaries

Since we leverage the benefit of Transformer for motion prediction, we first elaborate its core components, including MHSA (Multi-head Self-Attention), MHCA (Multi-head Cross-Attention), and FFN (Feed Forward Network).

MHSA. MHSA is the kernel module of the transformer structure. MHSA block first maps the token sequence $\mathbf{X} \in \mathbb{R}^{n\times d}$ into query $\mathbf{Q} \in \mathbb{R}^{n\times d}$, key $\mathbf{K} \in \mathbb{R}^{n\times d}$ and value $\mathbf{V} \in \mathbb{R}^{n\times d}$ through three linear layers $\mathbf{W}_q \in \mathbb{R}^{d\times d}$, $\mathbf{W}_k \in \mathbb{R}^{d\times d}$, $\mathbf{W}_v \in \mathbb{R}^{d\times d}$, where n is the token number and d is the embedding dimension. Then $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are split into H heads in parallel. For each head h, a softmax operator is leveraged to establish the correlation between $\mathbf{Q}_h$ and $\mathbf{K}_h$ and is scaled by a scale factor of $1/\sqrt{d}$. The result will be used to multiply $\mathbf{V}_h$. The processing of a single head h can be represented as follows:

$$\text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h) = \text{Softmax}(\mathbf{Q}_h\mathbf{K}_h^T/\sqrt{d})\mathbf{V}_h. \tag{6}$$

We process the H heads in parallel, concatenate the outputs and utilize a linear transformation $\mathbf{W}_s$ to obtain the updated representation of $\mathbf{X}$:

$$\text{MHSA}(\mathbf{X}) = \text{Concat}(\mathbf{H}_1, \mathbf{H}_2 \ldots \mathbf{H}_h)\mathbf{W}_s, \tag{7}$$

$$\mathbf{H}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h), h \in [1, 2, \ldots, H], \tag{8}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_q, \mathbf{K} = \mathbf{X}\mathbf{W}_k, \mathbf{V} = \mathbf{X}\mathbf{W}_v$.

MHCA. MHCA provides an effective manner to establish cross-correlations between different inputs. In this paper, we resort to MHCA to construct the relationships between future postures and historical postures of kinematic and dynamic so as to learn powerful representation for motion prediction. The MHCA has the same structure as the MHSA, and the only differences are that the query term $\mathbf{Q}$ comes from the projection of future motion data $\mathbf{Y} \in \mathbb{R}^{m\times d}$, while the key $\mathbf{K}$ and value $\mathbf{V}$ derive from transformations of the features of history motion frames $\mathbf{X}$, and m refers to the length of the sequence to be predicted. The process pf MHCA can be expressed as:

$$\text{MHCA}(\mathbf{Y}, \mathbf{X}) = \text{Concat}(\mathbf{H}_1, \mathbf{H}_2 \ldots \mathbf{H}_H)\mathbf{W}_c, \tag{9}$$

$$\mathbf{H}_h = \text{Attention}(\mathbf{Q}_h, \mathbf{K}_h, \mathbf{V}_h), h \in [1, 2, \ldots, H]. \tag{10}$$

Here $\mathbf{Q} = \mathbf{Y}\mathbf{W_q}$, $\mathbf{K} = \mathbf{X}\mathbf{W_k}$, $\mathbf{V} = \mathbf{X}\mathbf{W_v}$. $\mathbf{W}_c$ is the weight of a linear layer.

FFN. FFN is composed of two fully-connected (FC) layers. It is applied after MHSA or MHCA, and is leveraged for feature transformation and increases the model's non-linearity. Its processing process can be expressed as:

$$\text{FFN}(\mathbf{X}) = \sigma(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{11}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the convolution kernel parameters of the two FC layers, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are the corresponding bias terms.

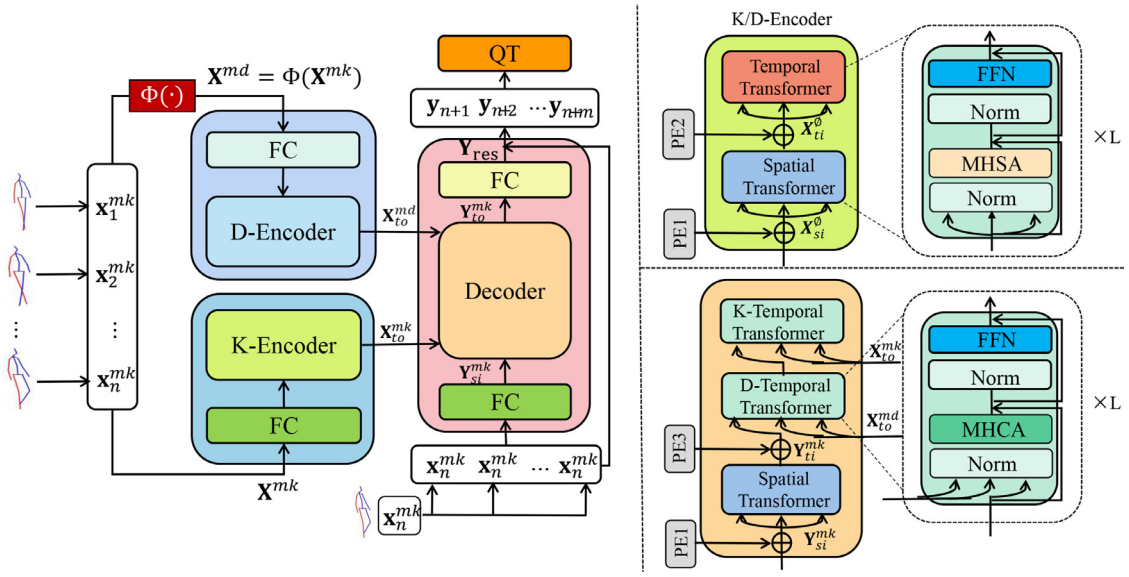

Fig. 1. The framework of our KD-Former. It receives history kinematic and dynamic data as inputs to forecast postures and is optimized in quaternion space via a quaternion transformation (QT) layer. $\Phi(\cdot)$ represents the reduced-order dynamic algorithm.

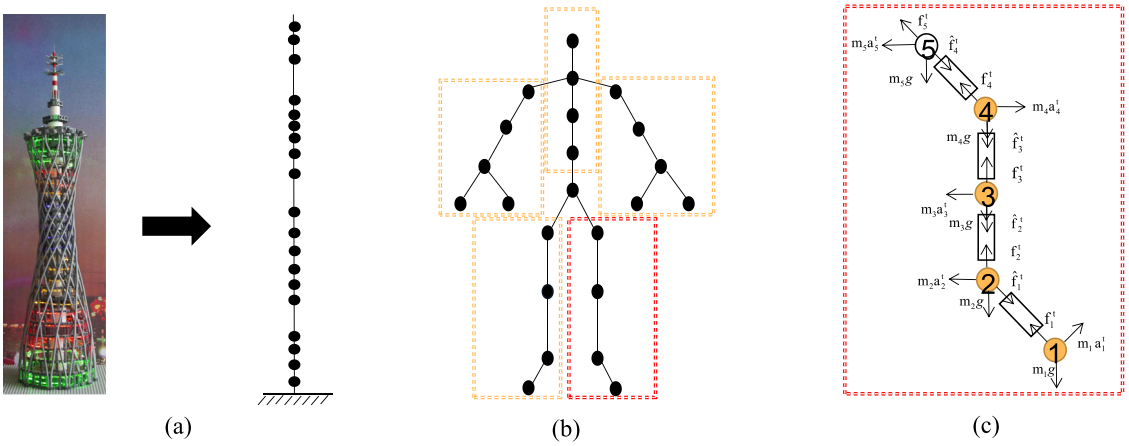

Fig. 2. Dynamics calculation models. (a) The reduced-order dynamic model in super-high building structure [33]. (b) The reduced-order dynamic model of human body. (c) The schematic diagram of stress at joints in our reduced-order algorithm.

3.4. The proposed KD-Former

Our KD-Former is a non-autoregressive seq2seq architecture based on the transformer. As shown in Fig. 1, the encoder module consists of a kinematic encoder (K-Encoder) and a dynamic encoder (D-Encoder), both of which are compromised with a spatial transformer and a temporal transformer. The decoder module is made up of a spatial transformer and two temporal transformers based on a cross-attention mechanism. Furthermore, we incorporate a quaternion transformation (QT) layer to project joint rotations into quaternion space. Since the transformer has intrinsic permutation invariance, we provide position embedding (PE) for all the transformer blocks to make our model aware of the semantic information of both skeleton joints and frame indexes. In the following, we describe the position embedding, the encoder and decoder modules, and the QT layer in detail.

Position embedding. To preserve the position information of human skeleton joints, We follow [34] to provide a learnable position embedding $\mathbf{E}_s \in \mathbb{R}^{J\times d}$ for the spatial transformer. Given an input motion sequence $\mathbf{X}^{\phi} \in \mathbb{R}^{n\times J\times 3}$, we first transform it into a high-dimensional space with a fully connected (FC) layer $\mathbf{W}^{\phi} \in \mathbb{R}^{3\times d}$, where n denotes the input sequence length, J refers to the joint number and 3 represents the joint rotation or joint force. We embed each joint into a d-dimensional space and use a learnable position embedding $\mathbf{E}_{pe1} \in \mathbb{R}^{J\times d}$ to perceive the skeleton joint information. The process of spatial position-aware embedding can be described as follows:

$$\mathbf{X}^{\phi}_{si} = \mathbf{X}^{\phi}\mathbf{W}^{\phi} + \mathbf{E}_{pe1}, \phi \in \{mk, md\}, \tag{12}$$

where mk and md represent the motion kinematic and dynamic, respectively. Thus, $\mathbf{X}^{\phi}_{si}$ refers to the kinematic embedding when $\phi = mk$, and it signifies dynamic representation when $\phi = md$. $\mathbf{X}^{\phi}_{si}$ will later be served as the spatial transformer input (si) for updating joint representations.

In light of [8], frame index information for motion prediction encourages a model to perceive time stamps and has the potential to alleviate the mean pose problem. Therefore, we also introduce position embedding for the temporal transformer. Our KD-Former involves different temporal transformers (MHSA+FFN VS. MHCA+FFN) with varied frames for the encoder and decoder. For the given history motion sequence $\mathbf{X}^{\phi}$, assuming the spatial transformer output (so) is denoted as $\mathbf{X}^{\phi}_{so} \in R^{n\times(J\cdot d)}$, similar as the spatial transformer, we provide a learnable position embedding $\mathbf{E}_{pe2} \in \mathbb{R}^{n\times(J\cdot d)}$ to distinguish different history frames. This process is for-

mulated as follows:

$$\mathbf{X}_{ti}^{\phi} = \mathbf{X}_{so}^{\phi} + \mathbf{E}_{pe2}, \tag{13}$$

where $\mathbf{X}_{ti}^{\phi}$ is the temporal position-aware embedding of kinematic or dynamic for history sequences, which is acted as the temporal transformer input (ti).

For the future motion postures to be predicted with m frames, we also utilize a leanable embedding $\mathbf{E}_{pe3} \in \mathbb{R}^{m\times(J\cdot d)}$ to perceive temporal information. We use $\mathbf{Y}_{so}^{mk} \in \mathbb{R}^{m\times(J\cdot d)}$ to represent the kinematic output of the spatial transformer in the decoder module. Then, we have:

$$\mathbf{Y}_{ti}^{mk} = \mathbf{Y}_{so}^{mk} + \mathbf{E}_{pe3}. \tag{14}$$

Encoder. The encoder concentrates on learning informative features of motion dynamic and kinematic data. Therefore, the encoder module is formulated with the K-Encoder and D-Encoder. For a given input sequence with joint rotation representation $\mathbf{X}^{mk}$, we first calculate its dynamic information $\mathbf{X}^{md}$ through the above proposed order-reduced dynamic algorithm, that is, $\mathbf{X}^{md} = \Phi(\mathbf{X}^{mk})$. Then we implement spatial-aware position embedding via Eq. (12) for $\mathbf{X}^{mk}$ and $\mathbf{X}^{md}$. The embedding results $\mathbf{X}_{si}^{mk}$ and $\mathbf{X}_{si}^{md}$ will be later sent into K-Encoder and D-Encoder. Since the two encoders have the same architecture, their learning process is exactly the same. We remove the superscript mk or md for simplified representation. Assuming that the depth of the spatial transformer is L, for the lth layer, we first leverage the MHSA to model global context dependencies among joints within frames, the result of which is added into the original input to facilitate information flow:

$$\mathbf{Z}_{si}^{(l)} = \mathrm{MHSA}(\mathrm{LN}(\mathbf{X}_{si}^{(l)})) + \mathbf{X}_{si}^{(l)}, \tag{15}$$

where $\mathbf{X}_{si}^{(l)}$ is the kinematic or dynamic input for the spatial transformer of layer l, $\mathbf{Z}_{si}^{(l)}$ is the attention features learned the MHSA, and LN stands for layer normalization. $\mathbf{Z}_{si}^{(l)}$ is later fed into an FFN block for feature transformation and add nonlinearity, and we also add the result to the input:

$$\mathbf{X}_{si}^{(l+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Z}_{si}^{(l)})) + \mathbf{Z}_{si}^{(l)}, \tag{16}$$

where $\mathbf{X}_{si}^{(l+1)}$ is the output of a single spatial transformer layer. We implement Eqs. (15) and (16) sequenctially for L times, and the final spatial encoding representations for the kinematic and dynamic data are denoted as $\mathbf{X}_{so}^{mk} \in \mathbb{R}^{n\times(J\cdot d)}$ and $\mathbf{X}_{so}^{md} \in \mathbb{R}^{n\times(J\cdot d)}$, respectively.

Similar as the spatial transformer, we endow $\mathbf{X}_{so}^{mk}$ and $\mathbf{X}_{so}^{md}$ with temporal position-aware via Eq. (13). Then temporal-aware embedding results $\mathbf{X}_{ti}^{mk}$ and $\mathbf{X}_{ti}^{md}$ will be advanced by the temporal transformer to exploit the temporal global context information. The whole learning process is similar to the spatial transformer, which can be expressed as:

$$\mathbf{Z}_{ti}^{(l)} = \mathrm{MHSA}(\mathrm{LN}(\mathbf{X}_{ti}^{(l)})) + \mathbf{X}_{ti}^{(l)}, \tag{17}$$

$$\mathbf{X}_{ti}^{(l+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Z}_{ti}^{(l)})) + \mathbf{Z}_{ti}^{(l)}, \tag{18}$$

where $\mathbf{X}_{ti}^{(l)}$, $\mathbf{Z}_{ti}^{(l)}$ and $\mathbf{X}_{ti}^{(l+1)}$ respectively refer to the input, attention features and the output of the lth layer for the temporal transformer layer. The final temporal transformer outputs for kinematic data and dynamic data of an input motion sequence are represented as $\mathbf{X}_{to}^{mk}$ and $\mathbf{X}_{to}^{md}$. The encoded kinematic and dynamic features $\mathbf{X}_{to}^{mk}$ and $\mathbf{X}_{to}^{md}$ have powerful representation abilities, which fully consider interactions between skeleton joints within the same frame and the influence of temporal structure between different frames.

Decoder. The decoder is constructed with a spatial transformer and two temporal transformers. As illustrated in Fig 1, other than the self-attention in K-Encoder and D-Encoder, the temporal transformers in decoder are built with a cross-attention mechanism, aiming to leverage $\mathbf{X}_{to}^{mk}$ and $\mathbf{X}_{to}^{md}$ for motion prediction. Furthermore, different from autoregressive-based methods PVRED [8], DMGNN [10] making predictions step-by-step, the decoder in our network forecasts all the future motion postures simultaneously with a non-autoregressive strategy. Last but not least, our model predicts the possible postures conditioned on both motion kinematic and dynamic knowledge, whereas existing methods rarely consider the dynamic information.

Since transformer-based decoder usually decodes a target sequence with a query token sequence as input, inspired by Martínez-González et al. [28], we set the query tokens as the last frame of the input kinematic data $\mathbf{x}_n^{mk}$ for m copies. The query sequence is then projected into a latent space via a FC layer and added with spatial-aware position embedding for feature learning. Given the spatial transformer in the K-Encoder and decoder playing the same role, we share the parameters of these layers and the corresponding FC layers to reduce model parameters and avoid overfitting. Thus we have:

$$\mathbf{Y}_{si}^{mk} = [\mathbf{x}_n^{mk}; \mathbf{x}_n^{mk}; \ldots; \mathbf{x}_n^{mk}]\mathbf{W}_s^{mk} + \mathbf{E}_{pe1}, \tag{19}$$

where $\mathbf{Y}_{si}^{mk} \in \mathbb{R}^{m\times J\times d}$ is the joint position-aware representation for query motion kinematic data, and m represents the length of the query sequence. After then, we promote the spatial representation ability of the query kinematic data through the spatial transformer by modeling the human skeletal joint dependencies. The process can be described as follows:

$$\mathbf{Z}_{si}^{mk(l)} = \mathrm{MHSA}(\mathrm{LN}(\mathbf{Y}_{si}^{mk(l)})) + \mathbf{Y}_{si}^{mk(l)}, \tag{20}$$

$$\mathbf{Y}_{si}^{mk(l+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Y}_{si}^{mk(l)})) + \mathbf{Z}_{si}^{mk(l)}, \tag{21}$$

where $\mathbf{Y}_{si}^{mk(l)}$, $\mathbf{Z}_{si}^{mk(l)}$ and $\mathbf{Y}_{si}^{mk(l+1)}$ refers to the input, attention feature and output of the lth spatial transformer layer in the decoder. We repeat Eqs. (20) and (21) for L times and write the result as $\mathbf{Y}_{so}^{mk} \in \mathbb{R}^{m\times(J\cdot d)}$.

The temporal information of future motion sequences is crucial for accurate prediction, therefore we incorporate the temporal position-aware embedding representation using Eq. (14), that is, $\mathbf{Y}_{ti}^{mk} = \mathbf{Y}_{so}^{mk} + \mathbf{E}_{pe3}$. We update $\mathbf{Y}_{ti}^{mk}$ with two temporal transformers constructed in a cross-attention manner, which respectively leverage the encoded history motion kinematic and dynamic information for future motion kinematic residual prediction. Specifically, we first utilize the dynamic temporal transformer to learn dynamic information-conditioned features:

$$\mathbf{Y}_{ti}^{mk(l)} = \mathrm{MHCA}(\mathrm{LN}(\mathbf{Y}_{ti}^{mk(l)}, \mathbf{X}_{to}^{md})) + \mathbf{Y}_{ti}^{mk(l)}, \tag{22}$$

$$\mathbf{Y}_{ti}^{mk(l+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Y}_{ti}^{mk(l)})) + \mathbf{Y}_{ti}^{mk(l)}. \tag{23}$$

The above steps are implemented L times. Then, we utilize the kinematic temporal transformer to build the cross-interactions between the future sequence and history sequence. The process is the same as the dynamic temporal transformer, which can be described as follows:

$$\mathbf{Y}_{ti}^{mk(l)} = \mathrm{MHCA}(\mathrm{LN}(\mathbf{Y}_{ti}^{mk(l)}, \mathbf{X}_{to}^{mk})) + \mathbf{Y}_{ti}^{mk(l)}, \tag{24}$$

$$\mathbf{Y}_{ti}^{mk(l+1)} = \mathrm{FFN}(\mathrm{LN}(\mathbf{Y}_{ti}^{mk(l)})) + \mathbf{Y}_{ti}^{mk(l)}. \tag{25}$$

The output of decoder $\mathbf{Y}_{to}^{mk}$ integrates the spatio-temporal characteristics of dynamic and kinematic of the input motion sequence, which is leveraged to predict the kinematic residuals $\mathbf{Y}_{res}$ of future postures via a FC layer. We combine $\mathbf{Y}_{res}$ and the query sequence to obtain the final results $\mathbf{Y}$:

$$\mathbf{Y} = [\mathbf{x}_n^{mk}; \mathbf{x}_n^{mk}; \ldots; \mathbf{x}_n^{mk}] + \mathbf{Y}_{res}. \tag{26}$$

Quaternion transformation Since human poses described in exponential maps may suffer from deadlock and discontinuity, quaternion space can effectively eliminate singularities and discontinuities as pointed out in [8]. To leverage the stable numerical benefits of quaternion, we follow [8] to transform the predicted pose from exponential maps to quaternion space with a Quaternion Transformation (QT) layer. Assuming that the human body has J joints, and $\mathbf{e}_{t,j}$ refers to the exponential maps of joint j. A pose at frame t can be represented as $\mathbf{x}_t = [\mathbf{e}_{t,1}; ...; \mathbf{e}_{t,j}; ...; \mathbf{e}_{t,J}]$. For each joint $j \in \{1, 2, \ldots, J\}$, we utilize the QT layer to transform its exponential maps $\mathbf{e}_{t,j}$, a three-dimensional vector, into a four-dimensional vector $\mathbf{q}_{t,j}$:

$$\mathbf{q}_{t,j}(i) = \begin{cases} \cos(0.5\|\mathbf{e}_{t,j}\|)_2 & \text{i=1,} \\ \frac{\sin((0.5)\|\mathbf{e}_{t,j}\|)_2}{\|\mathbf{e}_{t,j}\|_2} \cdot \mathbf{e}_{t,j}(i-1) & \text{i}\geq 2, \end{cases} \tag{27}$$

where $\mathbf{q}_{t,j}$ is the quaternion of joint j at frame t, $\mathbf{q}_{t,j}(i)$ is the ith element with $i \in \{1, 2, 3, 4\}$, and $\|\cdot\|_2$ is the L^2-norm.

3.5. Training and testing loss

During training, our goal is to minimize the differences between the predicted postures and ground truth (GT) in the quaternion space. Following [8], we utilize the QT layer to convert kinematic data (joint rotations) of exponential maps into quaternion space for loss calculation. The training loss $\mathcal{L}_{train}$ of an m-frames motion clip is defined as:

$$\mathcal{L}_{train} = \frac{1}{mJ} \sum_{t=1}^{m} \sum_{j=1}^{J} \|g(\mathbf{y}_{n+t,j}) - g(\mathbf{r}_{n+t,j})\|_1, \tag{28}$$

where g denotes QT operation, $\mathbf{y}_{n+t,j}$ and $\mathbf{r}_{n+t,j}$ are the GT and predicted poses of joint j at frame $n+t$. $\|\cdot\|_1$ is the $L1$-norm.

In the testing phase, we remove the QT layer and the training loss from the network and represent human poses by the original exponential map. Thus, the prediction error $\mathcal{L}_{error}$ is calculated as:

$$\mathcal{L}_{error} = \frac{1}{mJ} \sum_{t=1}^{m} \sum_{j=1}^{J} \|\mathbf{y}_{n+t,j} - \mathbf{r}_{n+t,j}\|_2. \tag{29}$$

where $\|\cdot\|_2$ is the $L2$-norm.

4. Experiments

4.1. Datasets and evaluation metrics

Human3.6M [21]. Human3.6M dataset consists of 7 subjects performing 15 activities with a total of 3.6 million 3D human poses. Following [8], we downsample motion sequences by 2 and utilize the data of subject 5 for testing with other data for training. Before experiments, the dynamic information of all samples is calculated in advance, and we filter the dynamic data with 6Hz filtering. During testing, we measure the Euclidean distance between the prediction and ground truth using Euler angles, and the test errors of 8 different seed motion clips are reported.

CMU MoCap [22]. For the CMU MoCap, we select samples of single-person action and delete multiple person interactions following [8]. In experiments, motion sequences are downsampled for 30 FPS. We use the same training and testing set partition strategy as [8]. We pre-process the data and conduct evaluation in the same way as we do on the Human3.6M. When evaluating, the average distance across 80 sampled seed clips is reported.

4.2. Implementation details

In experiments, we set the input sequence length n as 50, embedding dimension d as 8, layer number L as 4, and the prediction length m as 25. During training, batch size and maximum epochs equal 32 and 10,000, adam optimizer with a learning rate of 0.0001 is utilized to optimize our model. We obey the procedures of [10] to conduct short-term and long-term motion predictions. A sequence of fewer than 500 milliseconds (ms) is regarded as short-term prediction, and no less than 500ms is long-term prediction. Therefore, given observed 50 frames (2 s, 2s), we predict the short-term of 10 frames (0.4s) and long-term of 25 frames (1s) motion separately.

4.3. Experiments on Human3.6M

Short-term prediction. In a short period, motion is considered to be enlightening and predictable. As observed in Table 1, our method produces the most advanced short-term prediction performance at 80ms in predicting walking, greeting, phoning, posing, purchases, sitting, walking dog, and walking together. Thanks to the non-autoregressive prediction, our model is not affected by error accumulations like the autoregressive methods, such as Res GRU [7], Conv seq2seq [9] and QuaterNet [23], and the average short-term performance has been significantly enhanced. when the autoregressive model PS [24] is assisted with auxiliary class labels, its prediction result can be largely enhanced. Further, compared with the non-autoregressive POTR [28], it is not much different at 80ms. However, there are big gaps in predictions at 160ms, 320ms. The comparison results demonstrate the effectiveness and superiority of our model.

Long-term prediction. As displayed in Table 1, our model achieves slightly better to the RNNs-based PVRED [8] for long-term prediction. However, compared with the GNN-based Traj-GCN [26] and DMGNN [10], a few gaps exist. One possible reason is that our prediction decoder aims to generate the kinematic residuals of the last frame of an input sequence. As the prediction goes further, the deviations between the actual postures and the last frame of the input motion increase gradually. Although there is no error accumulation problem in our model, the difficulty of the task gradually increases with the increase of the bias to be predicted. It should be also noticed that DMGNN [10] designs multi-scale GNNs with a considerable model parameter to encode input motion and forecast future posture. In contrast, our model is much smaller with fast inference abilities, which is summarized later.

4.4. Experiments on CMU MoCap

Short-term prediction. As observed in Table 2, our model achieves the average state-of-the-art performance when performing 80ms short-term prediction. With the increase in prediction time, our results still exceed the current advanced methods except for DMGNN [10] and Traj-GCN [26]. We believe this may be because the CMU dataset is smaller than the Human3.6M, DMGNN captures multi-scale topology with complex model parameters, and Traj-GCN leverages DCT to better utilizes temporal structure. Nevertheless, we substantially surpass the CNNs-based (Conv seq2seq [9]), RNNs-based (Res GRU [7], PVRED [8] and PS [24]) and Transformer-based (POTR [28]) methods, demonstrating the superior performance of our KD-Former.

Long-term prediction. For long-term prediction on CMU MoCap, in most cases, our results are much better than most of the existing results as can be observed in Table 2. Overall, our method has the best average results on long-term motion prediction, further confirming the superiority of our model.

4.5. Depth analyses and discussions

Kinematic and dynamic input. Both kinematic and dynamic expressions can be used to describe motion variations. We inplement experiments with different inputs to validate the superior-

Table 1
Comparisons with the state-of-the-arts regarding angle error for short-term and long-term predictions on Human3.6M dataset. The best results are denoted in bold font.

Methods	Walking						Eating						Smoking						Discussion					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.28	0.49	0.72	0.81	0.93	1.03	0.23	0.39	0.62	0.76	0.95	**1.08**	0.33	0.61	1.05	1.15	1.25	1.50	0.31	0.68	1.01	1.09	1.43	1.69
Conv seq2seq [9]	0.33	0.54	0.68	0.73	-	0.92	0.22	0.36	0.58	0.71	-	1.24	0.26	0.49	0.96	0.92	-	1.62	0.32	0.67	0.94	1.01	-	1.86
QuaterNet [23]	0.28	0.49	0.76	0.83	-	-	0.22	0.47	0.76	0.88	-	-	0.25	0.47	0.93	0.90	-	-	0.48	0.74	1.20	1.37	-	-
DMGNN [10]	0.18	**0.31**	0.49	0.58	0.66	0.75	0.17	0.30	**0.49**	0.59	0.74	1.14	0.21	0.39	0.81	**0.77**	**0.83**	1.52	0.26	0.65	0.92	0.99	1.33	1.45
POTR [28]	0.16	0.40	0.62	0.73	-	-	**0.11**	0.29	0.53	0.68	-	-	**0.14**	0.39	0.84	0.82	-	-	**0.17**	0.56	0.85	0.96	-	-
PVRED [8]	0.20	0.35	0.54	0.59	**0.65**	**0.66**	0.18	0.32	0.54	0.66	0.76	1.14	0.22	0.44	0.81	0.91	0.97	**1.42**	0.24	0.60	0.83	0.93	1.29	1.77
Traj-GCN [26]	0.18	**0.31**	**0.49**	**0.56**	**0.65**	0.67	0.16	0.29	0.50	0.62	0.87	1.57	0.22	0.41	0.86	0.80	1.33	1.70	0.20	**0.51**	**0.77**	**0.85**	**0.90**	**1.27**
PS [24]	0.21	0.34	0.52	0.60	-	0.73	0.18	0.31	0.50	0.62	-	1.16	0.23	0.43	0.86	0.81	-	1.56	0.26	0.63	0.88	0.92	-	1.39
Ours	**0.15**	0.32	0.54	0.61	0.70	0.69	0.14	**0.28**	0.50	**0.51**	**0.71**	**1.08**	0.17	**0.37**	**0.76**	0.91	1.01	1.46	0.19	0.53	0.87	0.90	1.24	1.69

Methods	**Directions**						**Greeting**						**Phoning**						**Posing**					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.45	0.68	0.93	1.05	1.15	1.64	0.53	0.88	1.33	1.50	1.82	2.14	0.50	0.77	1.20	1.31	1.55	2.05	0.43	0.89	1.68	2.02	2.39	2.85
Conv seq2seq [9]	0.39	0.60	0.80	0.91	-	1.67	0.51	0.82	1.21	1.38	-	1.72	0.59	1.13	1.51	1.65	-	1.81	0.29	0.60	1.12	1.37	-	2.65
QuaterNet [23]	0.24	0.46	0.84	1.01	-	-	0.61	0.93	1.34	1.51	-	-	0.36	**0.61**	**0.98**	**1.14**	-	-	0.38	0.71	1.20	1.39	-	-
DMGNN [10]	0.25	0.44	**0.65**	**0.71**	**0.86**	1.30	0.36	0.61	**0.94**	**1.12**	1.57	1.63	0.52	0.97	1.29	1.43	**1.44**	**1.64**	0.20	0.46	1.06	1.34	**1.49**	2.17
POTR [28]	**0.20**	0.45	0.79	0.91	-	-	0.29	0.69	1.17	1.30	-	-	0.50	1.10	1.50	1.65	-	-	0.18	0.52	1.18	1.47	-	-
PVRED [8]	0.31	**0.42**	0.66	0.72	0.89	1.45	0.40	0.66	1.00	1.13	**1.36**	**1.62**	0.45	0.69	1.26	1.34	1.54	1.75	0.26	0.62	1.19	1.42	1.60	2.44
Traj-GCN [26]	0.26	0.45	0.71	0.79	-	-	0.36	**0.60**	0.95	1.13	-	-	0.53	1.02	1.35	1.48	-	-	0.19	0.44	1.01	1.24	-	-
PS [24]	0.32	0.51	**0.65**	0.72	-	**1.26**	0.43	0.68	1.01	1.19	-	1.65	0.55	0.98	1.32	1.44	-	1.50	0.20	0.45	0.93	**1.15**	-	**2.16**
Ours	0.24	0.52	0.72	0.77	0.88	1.36	**0.27**	0.72	1.11	1.25	1.53	1.89	**0.17**	0.66	1.28	1.35	1.54	1.95	**0.17**	**0.43**	**0.92**	1.18	1.53	2.29

Methods	**Purchases**						**Sitting**						**Sitting down**						**Taking photo**					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.58	0.86	1.24	1.35	1.48	2.35	0.44	0.76	1.27	1.95	1.66	1.91	0.52	0.99	1.50	1.74	1.40	2.06	0.29	0.62	1.01	1.16	0.88	1.10
Conv seq2seq [9]	0.63	0.91	1.19	1.29	-	2.52	0.39	0.61	1.02	1.18	-	1.67	0.41	0.78	1.16	1.31	-	2.06	0.23	0.49	0.88	1.06	-	1.40
QuaterNet [23]	0.54	0.92	1.36	1.47	-	-	0.34	0.59	1.00	1.15	-	-	0.47	0.81	1.31	1.50	-	-	0.23	0.39	0.69	0.81	-	-
DMGNN [10]	0.41	**0.61**	1.05	1.14	1.39	**2.13**	0.26	**0.42**	**0.76**	**0.97**	**1.12**	1.51	0.32	0.65	0.93	1.05	**1.30**	1.74	0.15	**0.34**	**0.58**	0.71	**0.83**	1.06
POTR [28]	0.33	0.63	1.04	1.09	-	-	0.25	0.47	0.92	1.09	-	-	**0.25**	0.63	1.00	1.12	-	-	**0.12**	0.41	0.71	0.86	-	-
PVRED [8]	0.47	0.71	1.05	1.10	1.48	2.35	0.30	0.47	0.84	1.56	1.66	1.91	0.58	0.70	1.03	1.19	1.40	2.06	0.17	0.40	0.66	0.79	0.88	1.10
Traj-GCN [26]	0.43	0.65	1.05	1.13	-	-	0.29	0.45	0.80	**0.97**	-	-	0.30	**0.61**	**0.90**	**1.00**	-	-	0.14	**0.34**	**0.58**	**0.70**	-	-
PS [24]	0.50	0.69	1.04	1.09	-	2.16	0.29	0.43	0.80	0.99	-	**1.50**	0.34	0.64	0.92	1.03	-	1.61	0.19	0.39	0.65	0.77	-	**1.03**
Ours	**0.26**	0.72	**0.97**	**1.07**	**1.29**	**2.13**	**0.23**	0.53	0.94	1.61	1.71	1.97	0.26	0.63	0.98	1.12	1.36	1.90	0.15	0.39	0.72	0.84	1.00	1.26

Methods	**Waiting**						**Walking dog**						**Walking together**						**Average**					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.34	0.67	1.17	1.35	1.64	2.22	0.52	0.85	1.29	1.48	1.66	1.92	0.30	0.60	0.87	0.95	1.14	1.61	0.41	0.72	1.14	1.33	1.57	2.04
Conv seq2seq [9]	0.30	0.62	1.09	1.30	-	2.50	0.59	1.00	1.32	1.44	-	1.92	0.27	0.52	0.71	0.74	-	1.28	0.38	0.68	1.01	1.13	-	1.77
QuaterNet [23]	0.32	0.54	1.00	1.15	-	-	0.48	0.78	**1.12**	**1.21**	-	-	0.28	0.45	0.69	0.79	-	-	0.37	0.62	1.00	1.14	-	-
DMGNN [10]	0.22	0.49	**0.88**	**1.10**	**1.46**	**2.12**	0.42	**0.72**	1.16	1.34	1.57	**1.75**	**0.15**	**0.33**	**0.50**	**0.57**	0.70	1.24	0.27	0.52	**0.83**	**0.95**	**1.17**	1.57
POTR [28]	**0.17**	0.56	1.14	1.37	-	-	0.35	0.79	1.21	1.33	-	-	**0.15**	0.44	0.63	0.70	-	-	0.22	0.56	0.94	1.01	1.30	1.77
PVRED [8]	0.23	0.49	0.93	1.15	1.55	2.28	0.45	0.74	1.13	1.29	1.49	**1.75**	0.17	0.38	0.59	0.64	0.75	1.26	0.31	0.53	0.87	1.03	1.22	1.66
Traj-GCN [26]	0.23	0.50	0.91	1.14	-	-	0.46	0.79	**1.12**	1.29	-	-	**0.15**	0.34	0.52	**0.57**	-	-	0.27	**0.51**	**0.83**	**0.95**	-	-
PS [24]	0.26	0.52	0.94	1.15	-	2.25	0.47	0.77	1.20	1.36	-	1.90	0.18	0.38	0.53	0.58	-	1.31	0.31	0.54	0.85	0.96	-	**1.54**
Ours	0.18	**0.47**	0.98	1.15	1.50	2.35	**0.31**	0.74	1.12	1.35	**1.48**	1.79	**0.15**	0.39	0.55	0.62	**0.68**	**1.11**	**0.20**	**0.51**	0.86	1.01	1.21	1.66

Table 2
Comparisons with the state-of-the-arts regarding angle errors for short-term and long-term predictions on CMU MoCap dataset. The best results are denoted in bold font.

Methods	Basketball						Basketball signal						Directing traffic					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.49	0.77	1.26	1.45	-	1.77	0.42	0.76	1.33	1.54	-	2.17	0.31	0.58	0.94	1.10	-	2.06
Conv seq2seq [9]	0.37	0.62	1.07	1.18	-	1.95	0.32	0.59	1.04	1.24	-	1.96	0.25	0.56	0.89	1.00	-	2.04
DMGNN [10]	0.30	**0.46**	**0.89**	1.11	-	1.66	**0.10**	**0.17**	**0.31**	**0.41**	-	1.26	**0.15**	**0.30**	0.57	0.72	-	1.98
POTR [28]	0.31	0.61	1.07	1.24	1.43	1.60	0.20	0.33	0.62	0.75	0.94	1.24	0.32	0.48	1.01	1.18	1.58	1.61
PVRED [8]	0.36	0.56	0.95	1.13	1.41	1.61	0.22	0.33	0.61	0.74	1.39	1.53	0.31	0.48	0.78	0.90	1.40	1.54
Traj-GCN [26]	0.33	0.52	**0.89**	**1.06**	-	1.71	0.11	0.20	0.41	0.53	-	**1.00**	**0.15**	0.32	**0.52**	**0.60**	-	2.00
PS [24]	0.33	0.55	1.00	1.21	-	**1.48**	0.12	0.21	0.41	0.52	-	1.07	0.22	0.47	0.73	0.84	-	2.05
Ours	**0.25**	0.56	1.04	1.20	**1.40**	1.58	0.12	0.25	0.47	0.60	**0.81**	1.26	0.28	0.47	1.04	1.18	**1.38**	**1.49**

Methods	**Jumping**						**Running**						**Soccer**					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.57	0.86	1.76	2.03	-	2.42	0.32	0.48	0.65	0.74	-	1.00	0.29	0.50	0.87	0.98	-	1.73
Conv seq2seq [9]	0.39	0.60	1.36	1.56	-	2.01	0.28	0.41	0.52	0.57	-	0.67	0.26	0.44	0.75	0.87	-	1.56
DMGNN [10]	0.37	0.65	1.49	1.71	-	1.79	**0.19**	**0.31**	0.47	**0.49**	-	0.64	0.22	0.32	0.79	0.91	-	1.54
POTR [28]	0.32	0.65	1.07	**1.30**	**1.49**	1.82	0.26	0.53	0.73	0.70	0.68	0.74	0.26	0.69	1.09	1.28	1.46	1.58
PVRED [8]	0.46	0.65	1.12	1.35	1.57	**1.75**	0.26	0.36	**0.46**	0.52	0.59	0.64	0.32	0.45	1.10	1.24	1.42	1.65
Traj-GCN [26]	0.31	0.49	1.23	1.39	-	1.80	0.33	0.55	0.73	0.74	-	0.95	**0.18**	**0.29**	**0.61**	**0.71**	-	**1.40**
PS [24]	0.38	0.66	1.46	1.64	-	1.79	0.31	0.52	0.76	0.79	-	**0.57**	0.21	0.40	0.77	0.88	-	1.48
Ours	**0.26**	**0.62**	**1.06**	**1.30**	1.51	1.90	0.23	0.44	0.47	**0.49**	**0.58**	0.62	0.19	0.40	0.86	1.16	**1.35**	1.54

Methods	**Walking**						**Washing window**						**Average**					
	Short-term				Long-term		Short-term				Long-term		Short-term				Long-term	
	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
Res GRU [7]	0.35	0.45	0.59	0.64	-	0.88	0.31	0.47	0.74	0.93	-	1.37	0.38	0.61	1.01	1.17	-	1.67
Conv seq2seq [9]	0.35	0.44	0.45	0.50	-	0.78	0.30	0.47	0.80	1.01	-	1.39	0.31	0.51	0.86	0.99	-	1.54
DMGNN [10]	0.30	0.34	**0.38**	**0.43**	-	0.60	0.20	0.27	0.62	0.81	-	1.09	0.22	**0.35**	0.69	0.82	-	1.32
POTR [28]	0.20	0.31	0.43	0.49	0.54	0.65	0.26	0.45	0.82	0.95	1.08	1.19	0.27	0.51	0.85	0.98	1.15	1.31
PVRED [8]	0.28	0.34	0.41	**0.43**	**0.47**	**0.53**	0.25	0.37	0.67	0.81	1.02	1.20	0.30	0.44	0.76	0.89	1.15	1.30
Traj-GCN [26]	0.33	0.45	0.49	0.53	-	0.61	0.22	0.33	**0.57**	0.75	-	1.20	0.25	0.39	**0.68**	**0.79**	-	1.33
PS [24]	0.31	0.39	0.39	**0.43**	-	0.59	0.25	0.37	0.61	0.81	-	**1.08**	0.27	0.45	0.77	0.89	-	**1.26**
Ours	**0.18**	**0.29**	**0.38**	**0.43**	0.50	0.59	**0.15**	**0.31**	0.62	**0.74**	**0.93**	1.09	**0.20**	0.41	0.74	0.88	**1.05**	**1.26**

Table 3
Experiments on Human3.6M with different inputs (Kinematic Vs. Dynamic), decoding manners (Autoregressive: AR, Non-autoregressive: N-AR), position embedding (PE) (Learnable Vs. Absolute) and involvement of quaternion transformation (QT).

Models	Decoding	PE	QT	Average					
				Short-term				Long-term	
				80	160	320	400	560	1000
K-Former	N-AR	Learnable	Yes	0.21	0.53	0.89	1.04	1.26	1.71
D-Former	N-AR	Learnable	Yes	0.21	0.52	0.88	1.05	1.25	1.68
KD-Former	N-AR	Absolute	Yes	0.22	0.52	0.89	1.05	1.25	1.70
KD-Former	N-AR	Learnable	No	0.30	0.59	0.95	1.10	1.28	1.77
KD-Former	AR	Learnable	Yes	0.24	0.55	0.90	1.08	1.25	1.70
KD-Former	N-AR	Learnable	Yes	**0.20**	**0.51**	**0.86**	**1.01**	**1.21**	**1.66**

ity of dynamic information. The short-term and long-term angular prediction errors of different models are summarized in Table 3, where K-Former refers to only kinematic data served as network input, and D-Former means only dynamic data used for network input. It can be observed that the dynamic data shows slightly better learnable characteristics than kinematic clue, and the prediction errors are lower than that of kinematic data. When we leverage the mutual information of kinematic and dynamic knowledge for motion prediction, our KD-Former achieves the lowest joint rotation error results. The results demonstrate the superiority of incorporating dynamic information for motion prediction.

Involvement of quaternion transformation (QT). QT is leveraged to eliminate singularities and discontinuities of rotation angles. We conduct experiments to validate the effectiveness of QT and report the results in Table 3. It can be seen that the incorporation of QT can significantly improve short-term and long-term motion prediction performance.

Autoregressive and non-autoregressive. Our model is a seq2seq framework. The decoder can decode a query sequence autoregressive (AR) as [12] or non-autoregressive (N-AR) similar to Martínez-González et al. [28]. We verify the motion prediction influence of the two decoding strategies. Table 3 reports the results

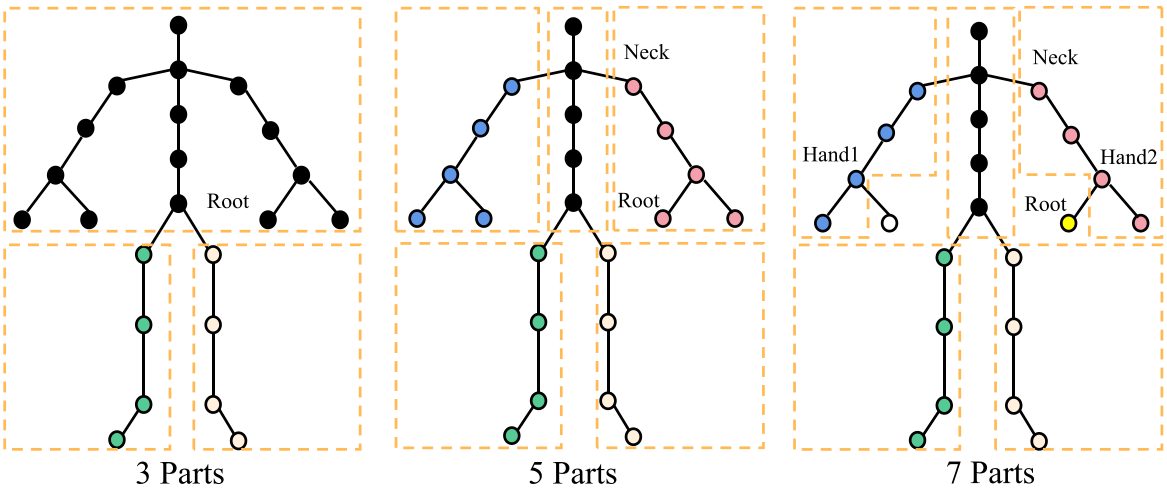

Fig. 3. Different number of body part divisions for Human3.6M dataset.

Table 4
Comparisons of different number of body parts in dynamic model on Human3.6M.

Body parts	Average					
	Short-term				Long-term	
	80	160	320	400	560	1000
3	0.22	0.54	0.90	1.07	1.28	1.72
5	**0.20**	**0.51**	**0.86**	**1.01**	**1.21**	**1.66**
7	0.21	0.53	0.89	1.05	1.27	1.70

Table 5
Experiments on Human3.6M with different fusion patterns in the decoder.

Fusion patterns	Average					
	Short-term				Long-term	
	80	160	320	400	560	1000
Serial	0.20	0.51	0.86	1.01	1.21	1.66
Parallel	0.20	0.53	0.88	1.06	1.28	1.71
Flat	0.21	0.52	0.89	1.05	1.26	1.69

of KD-Former predicting future motion in AR and N-AR manners. The N-AR predictions are found to perform better than AR results for either short-term or long-term motion predictions. The possible reason is that the N-AR technique can avoid the error accumulation problem caused by inaccurate predictions. Therefore, we leverage N-AR to simultaneously predict all frames, significantly speeding up training and testing as reported later.

Absolute and learnable. In the conventional transformer, position embedding (PE) is encoded with absolute representation [12]. In this paper, we follow [34] to implement learnable PE for spatial joints and temporal frames. To identify whether absolute or learnable PE is more suitable for our KD-Former, we conduct comparisons and report the results in Table 3. It can be safely concluded that learnable PE obtains better performance than the absolute one. Therefore, we introduce learnable PE for our framework.

Different number of body parts. To verify the effects of the number of human body parts in dynamics calculation models, as shown in Fig. 3, we divide the human body into 3, 5, and 7 parts from simple to complex structures according to natural connections between skeleton joints. Experimental results on the Human3.6M dataset have been reported in Table 4. It can be observed that a more simple or complex body division brings inferior results than dividing the body into 5 parts. Therefore, we prefer to divide the human body into 5 parts to achieve better prediction performance.

Information fusion pattern. Since kinematic and dynamic data are complementary, we simultaneously receive the encoded features of the dynamic encoder and the kinematic encoder to decode and predict. To effectively integrate the two mutually beneficial information clues, we design three fusion decoders *i.e.,* serial, parallel and flat, according to the design schemes given in [35]. The obtained results are shown in Table 5. It can be seen that the serial fusion is better than that of parallel and flat. Therefore, the proposed KD-Former leverages the serial fusion pattern to design the decoder.

Computational complexity. To further validate the superiority of our method, we make comparisons with state-of-the-arts regarding model parameters, running times in predicting 1000ms. We report the average results of short-term prediction for all actions and long-term prediction of four actions (Walking, Eating, Smoking, and Discussion) following the statistics in Traj-GCN [26]. From Table 6, we can observe that our KD-former holds an impressive advantage of short-term prediction at 80 ms. However, when the prediction goes further, the result becomes slightly inferior to the DMGNN [10] and Traj-GCN [26]. Nevertheless, compared with the RNNs-based PVRED [8] and transformer-based POTR [28] methods, our method achieves the best performance and fastest running speeds with moderate parameters.

Efficiency of the reduced-order dynamic algorithm. To verify the superiority of our simplified dynamic model, we make comparisons with the dynamic algorithm in OpenSim [19]. We utilize the C3D data to drive the mechanical model of OpenSim, which contains 17 joints with 30 degrees of freedom (DOFs). We first compute the kinematic data through inverse kinematics tool in OpenSim. Then, we use the inverse dynamic module in OpenSim to obtain the dynamic data and define the processing time as dynamics calculation time (DCT). We choose the corresponding joints and DOFs as OpenSim to make a fair comparison. The DCT of each frame and the average angle errors on CMU MoCap are illustrated in Table 7. We can observe that the DCT and prediction errors under various motion prediction lengths with 30 DOFs are lower than the algorithm in OpenSim. The results demonstrate the outstanding advantages of our simplified reduced-order algorithm.

Attention visualization. To verify whether kinematic and dynamic data are mutually beneficial, we randomly selected the walking action on the test set S5 of Human36M as an example to visualize attention results. Fig. 4(a)–(d) display the attention maps

Table 6
Model parameters, running times at 1000ms, short-term and long-term prediction performance are compared with state-of-the-art methods on Human 3.6M.

Methods	Parameters	Time	Average					
			Short-term				Long-term	
			80	160	320	400	560	1000
DMGNN [10]	61.97M	0.086	0.27	0.52	**0.83**	**0.95**	**0.89**	**1.22**
POTR [28]	13.73M	0.040	0.22	0.56	0.94	1.01	-	-
PVRED [8]	8.17M	0.034	0.31	0.53	0.87	1.03	0.92	1.25
Traj-GCN [26]	**2.27M**	**0.013**	0.27	0.51	0.83	0.95	0.90	1.27
Ours	8.53M	0.018	**0.20**	**0.51**	0.86	1.01	0.92	1.23

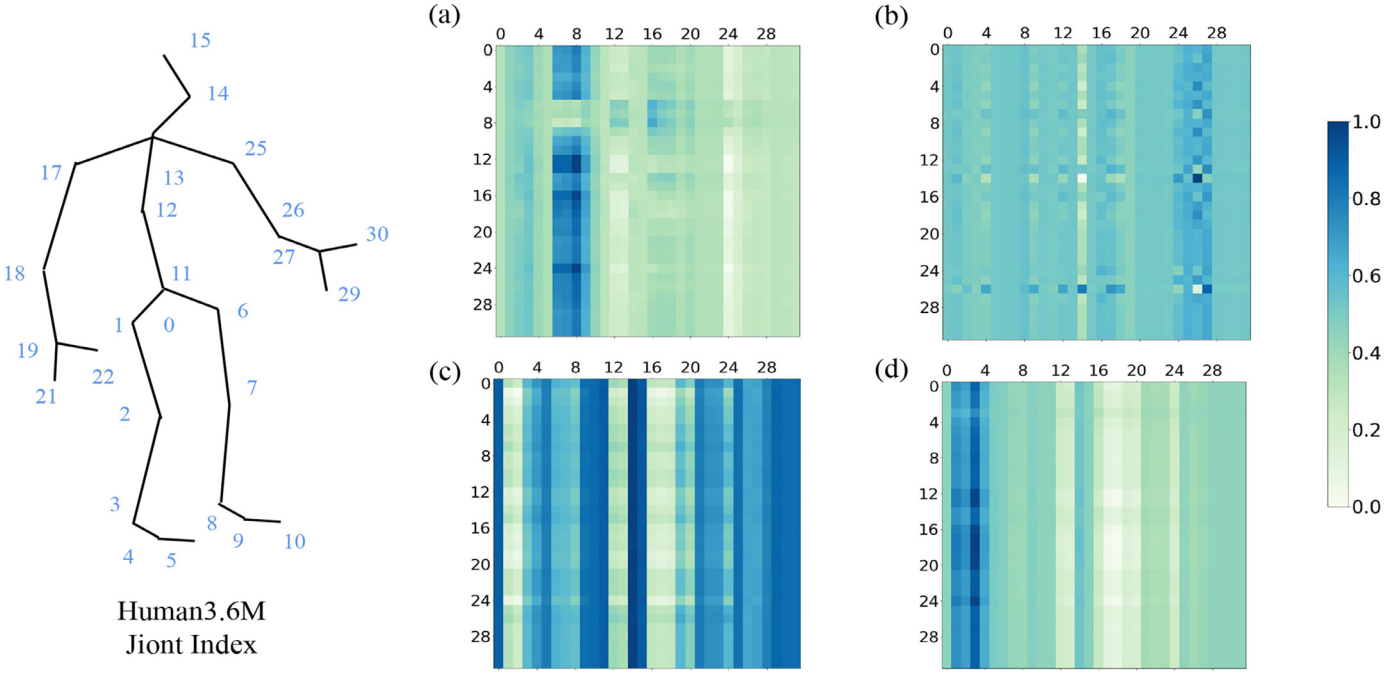

Fig. 4. Visualization of the self-attention results for the encoder module.

Table 7
Comparisons of different dynamics algorithms on CMU MoCap.

Method	DCT (ms)	Angle errors under different lengths					
		80	160	320	400	560	1000
OpenSim (30)	2.8	0.17	0.33	0.56	0.65	0.81	0.98
Ours (30)	**0.2**	**0.16**	**0.30**	**0.54**	**0.64**	**0.79**	**0.96**

of heads 5 and 7 for the kinematic and dynamic spatial encoders. We observe that head 5 in the kinematic spatial encoder focuses on joints 6, 7, 8, corresponding to the right leg (Fig. 4(a)), while head 7 concentrates on the right arm of joints 24, 26, 27 (Fig. 4(b)). For the dynamic spatial encoder, head 5 concerns the end effectors of different parts, *i.e.*, joints 5, 10, 15, 21, and 29 (Fig. 4(d)), whereas head 7 mainly notices joints 1, 2, 3, and 4, corresponding to the left leg. In our framework, the kinematic and the dynamic feature extractors have the same structure, but different representations of kinematic and dynamic information are found by visualizing the attentional maps. Hence, we believe that adding dynamic data enhances the motion expression ability.

Motion prediction. We randomly select some action samples to conduct motion prediction. The quantitative comparisons with POTR [28], PVRED [8] and DMGNN [10] on Human3.6M (Top) and CMU MoCap (Bottom) are displayed in Fig. 5. We visual prediction results of 400 ms with a time interval of 40 ms. For periodic running motion, our model generates movements closer to GT. For complex motion with rapid changes, such as jumping, our network shows better prediction results against DMGNN [10] and POTR [28]. For activities of daily living, such as purchases and smoking, we can observe that the prediction results in the hands and legs of our network are better than other methods. The qualitative visualization results further verify the excellent performance of our network for motion prediction.

Limitations. Our model currently performs well for short-term motion prediction. However, its performance is inferior for long-term prediction compared to the most advanced methods. We visualize some failure cases of long-term prediction (560ms-1000ms) in Fig. 6. It can be seen that the predicted motion movements have relatively large deviations from ground truth (GT) motion. We attribute the reason that when the prediction goes further, the deviations between actual postures and the last frame of input motion gradually increase. Therefore our future work will focus on enhancing long-term motion prediction. Besides, since we leverage the simplified order-reduced physical model to calculate motion dynamics and ignore the force acting on the ground, our algorithm can hardly estimate the truth joint forces, restricting motion prediction performance. Hence, the accurate and fast motion dynamics model is also one of the priorities for the future.

Fig. 5. Qualitative prediction results on Human3.6M (Top) and CMU (Bottom).

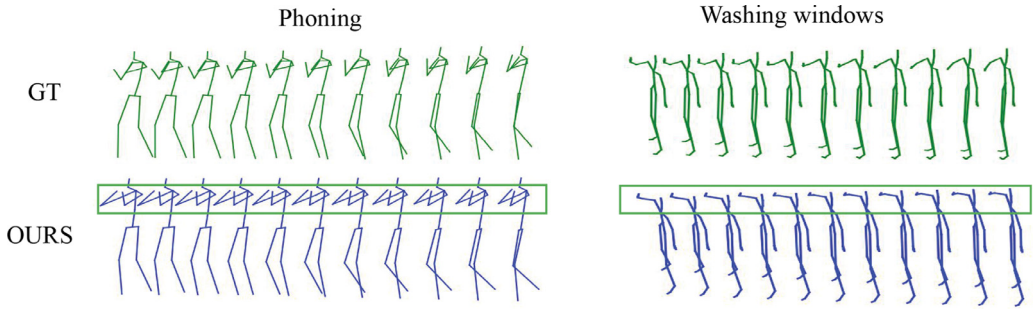

Fig. 6. Some failure cases for long-term prediction.

5. Conclusion

This paper presents a novel non-autoregressive KD-Former, a transformer-based seq2seq encoder-decoder framework, for 3D human motion prediction. Different from most existing models using only kinematic information, dynamic knowledge is introduced for motion prediction for the first time in our model and a simplified order-reduced physical model is presented to obtain dynamic data. Comprehensive experiments on Human3.6M and CMU Mocap datasets demonstrate the superiority of incorporating dynamic information for short-term motion prediction and impressive performance with fast running speed over state-of-the-art methods. In spite of that, our method perform inferior for long-term prediction over the most advanced methods. In the future, we will focus on estimating accurate dynamic information with appropriate complexity, so as to enhance long-term performance.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62102208, 61872020).

References

[1] F. Patrona, A. Chatzitofis, D. Zarpalas, P. Daras, Motion analysis: action detection, recognition and evaluation based on motion capture data, PR 76 (2018) 612–622.
[2] B. Xia, C. Wong, Q. Peng, W. Yuan, X. You, CSCNet: contextual semantic consistency network for trajectory prediction in crowded spaces, PR 126 (2022) 108552.
[3] J. Kim, T. Byun, S. Shin, J. Won, S. Choi, Conditional motion in-betweening, PR 132 (2022) 108894.
[4] M. Brand, A. Hertzmann, Style machines, in: CGIT, 2000, pp. 183–192.
[5] G.W. Taylor, G.E. Hinton, S.T. Roweis, Modeling human motion using binary latent variables, in: NIPS, 2006, pp. 1345–1352.
[6] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, PR 77 (2018) 354–377.
[7] J. Martinez, M.J. Black, J. Romero, On human motion prediction using recurrent neural networks, in: CVPR, 2017, pp. 4674–4683.

[8] H. Wang, J. Dong, B. Cheng, J. Feng, PVRED: a position-velocity recurrent encoder-decoder for human motion prediction, TIP 30 (2021) 6096–6106.
[9] C. Li, Z. Zhang, W.S. Lee, G.H. Lee, Convolutional sequence to sequence model for human dynamics, in: CVPR, 2018, pp. 5226–5234.
[10] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, Q. Tian, Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction, in: CVPR, 2020, pp. 211–220.
[11] L. Dang, Y. Nie, C. Long, Q. Zhang, G. Li, MSR-GCN: multi-scale residual graph convolution networks for human motion prediction, in: ICCV, 2021, pp. 11447–11456.
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 5998–6008.
[13] L. Li, X. Gao, J. Deng, Y. Tu, Z. Zha, Q. Huang, Long short-term relation transformer with global gating for video captioning, TIP 31 (2022) 2726–2738.
[14] E. Aksan, M. Kaufmann, P. Cao, O. Hilliges, A spatio-temporal transformer for 3D human motion prediction, in: 3DV, 2021, pp. 565–574.
[15] Y. Cai, L. Huang, Y. Wang, T. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen, D. Liu, J. Liu, N. Magnenat-Thalmann, Learning progressive joint propagation for human motion prediction, in: ECCV, 2020, pp. 226–242.
[16] Q. Cui, H. Sun, Towards accurate 3D human motion prediction from incomplete observations, in: CVPR, 2021, pp. 4801–4810.
[17] K. Xie, T. Wang, U. Iqbal, Y. Guo, S. Fidler, F. Shkurti, Physics-based human motion estimation and synthesis from videos, in: ICCV, 2021, pp. 11512–11521.
[18] V. Cahouët, M. Luc, A. David, Static optimal estimation of joint accelerations for inverse dynamics problem solution, J. Biomech. 35 (11) (2002) 1507–1513.
[19] H.-K. Kim, Y. Zhang, Estimation of lumbar spinal loading and trunk muscle forces during asymmetric lifting tasks: application of whole-body musculoskeletal modelling in opensim, Ergonomics 60 (4) (2017) 563–576.
[20] A. Mansur, Y. Makihara, Y. Yagi, Inverse dynamics for action recognition, TCYB 43 (4) (2013) 1226–1236.
[21] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments, TPAMI 36 (7) (2014) 1325–1339.
[22] Cmu graphics lab: Carnegie-mellon motion capture (mocap) database, 2003.
[23] D. Pavllo, D. Grangier, M. Auli, QuaterNet: a quaternion-based recurrent model for human motion, BMVC, 2018.
[24] M. Dong, C. Xu, Skeleton-based human motion prediction with privileged supervision, TNNLS (2022) 1–14.
[25] A. Jain, A.R. Zamir, S. Savarese, A. Saxena, Structural-RNN: deep learning on spatio-temporal graphs, in: CVPR, 2016, pp. 5308–5317.
[26] W. Mao, M. Liu, M. Salzmann, H. Li, Learning trajectory dependencies for human motion prediction, in: ICCV, 2019, pp. 9488–9496.
[27] C. Zhong, L. Hu, Z. Zhang, Y. Ye, S. Xia, Spatio-temporal gating-adjacency GCN for human motion prediction, in: CVPR, 2022, pp. 6437–6446.
[28] Á. Martínez-González, M. Villamizar, J. Odobez, Pose transformers (POTR): human motion prediction with non-autoregressive transformers, in: ICCVW, 2021, pp. 2276–2284.
[29] A. Escande, N. Mansard, P. Wieber, Hierarchical quadratic programming: fast online humanoid-robot motion generation, IJRR 33 (7) (2014) 1006–1028.
[30] J. Reher, A.D. Ames, Inverse dynamics control of compliant hybrid zero dynamic walking, in: ICRA, 2021, pp. 2040–2047.
[31] Musculoskeletal model-based inverse dynamic analysis under ambulatory conditions using inertial motion capture, Med. Eng. Phys. 65 (2019) 68–77.
[32] P. Zell, B. Rosenhahn, Learning inverse dynamics for human locomotion analysis, NCA 32 (15) (2020) 11729–11743.
[33] Y. Niu, C. Fritzen, H. Jung, I. Buethe, Y. Ni, Y. Wang, Online simultaneous reconstruction of wind load and structural responses- theory and application to canton tower, Comput.-Aided Civ. Infrastruct. Eng. 30 (8) (2015) 666–681.
[34] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, Z. Ding, 3D human pose estimation with spatial and temporal transformers, in: ICCV, 2021, pp. 11636–11645.
[35] J. Libovický, J. Helcl, D. Marecek, Input combination strategies for multi-source transformer decoder, in: Conference on Machine Translation, 2018, pp. 253–260.

Ju Dai is currently a Research Assistant in Peng Cheng Laboratory (PCL), Shenzhen, China. She received both BS and MS degree in Electronic Engineering, China University of Geosciences (CUG),Wuhan, China, in 2011 and 2014, respectively, and the PhD degree in Signal Processing in Dalian University of Technology (DUT), Dalian, China, in 2020. She worked in PCL as Postdoctoral Research Fellow from 2020 to 2022. Her research interests include human pose estimation, motion prediction, computer animation, person re-identification and saliency detection.

Hao Li received the BS degree in School of New Materials and Energy from Xuchang university, Henan, China, in 2020. He is currently a MS candidate in School of Software, Beihang University, Beijing, China. His research interest is in deep learning, motion prediction, computer animation.

Rui Zeng is currently a PhD candidate in School of Computer Science, Beihang University, Beijing, China. He received BS degree in Computer Science and Engineering, Beihang University, Beijing, China, in 2018. His research interests include computer animation, character movement, and data-driven motion analysis.

Junxuan Bai is currently a lecturer at the Institute of Artificial Intelligence in Sports (IAIS), Capital University of Physical Education and Sports (CUPES), Beijing, China. He received a BS degree in Mathematics from Dalian Maritime University, Dalian, China, in 2012, and an MS and a PhD degree in Computer Science from Beihang University, Beijing, China, in 2015 and 2021, respectively. He worked at China Mobile Research Institute as a researcher from 2021 to 2022. His interests include computer animation, motion synthesis, motion analysis, human pose estimation, and virtual surgery.

Feng Zhou received the BS degree in computer science from Beijing Union University in 2009, and the MS degree in computer science and application from Yun Nan University in 2014. He received the PhD degree in computer science from the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, in 2020. He is currently a lecturer with the School of Information Science and Technology, North China University of Technology, Beijing, China. His research interests include pattern recognition, image processing, virtual reality, computer graphics and computer vision.

Junjun Pan is currently a professor in School of Computer Science, Beihang University. He received both BS and MS degree in School of Computer Science, Northwestern Polytechnical University, China. In 2006, he studied in National Centre for Computer Animation (NCCA), Bournemouth University, UK as PhD candidate with full scholarship. In 2010, he received the PhD degree and worked in NCCA as Postdoctoral Research Fellow. From 2012 to 2013, he worked as a Research Associate in Center for Modeling, Simulation and Imaging in Medicine, Rensselaer Polytechnic Institute, USA. In November 2013, he was appointed as Associate Professor in School of Computer Science, Beihang University, China. His research interests include virtual surgery and computer animation.