A Quadruple Diffusion Convolutional Recurrent Network for Human Motion Prediction

Qianhui Men, Edmond S. L. Ho, Hubert P. H. Shum, *Senior Member, IEEE*, and Howard Leung

***Abstract*— Recurrent neural network (RNN) has become popular for human motion prediction thanks to its ability to capture temporal dependencies. However, it has limited capacity in modeling the complex spatial relationship in the human skeletal structure. In this work, we present a novel diffusion convolutional recurrent predictor for spatial and temporal movement forecasting, with multi-step random walks traversing bidirectionally along an adaptive graph to model interdependency among body joints. In the temporal domain, existing methods rely on a single forward predictor with the produced motion deflecting to the drift route, which leads to error accumulations over time. We propose to supplement the forward predictor with a forward discriminator to alleviate such motion drift in the long term under adversarial training. The solution is further enhanced by a backward predictor and a backward discriminator to effectively reduce the error, such that the system can also look into the past to improve the prediction at early frames. The two-way spatial diffusion convolutions and two-way temporal predictors together form a quadruple network. Furthermore, we train our framework by modeling the velocity from observed motion dynamics instead of static poses to predict future movements that effectively reduces the discontinuity problem at early prediction. Our method outperforms the state of the arts on both 3D and 2D datasets, including the Human3.6M, CMU Motion Capture and Penn Action datasets. The results also show that our method correctly predicts both high-dynamic and low-dynamic moving trends with less motion drift.**

***Index Terms*— Human motion prediction, body joint dynamics, diffusion convolutions, recurrent neural network, bi-directional predictor.**

I. Introduction

HUMAN motion prediction has attracted much attention in real-world applications where a precise estimation of movements in future frames are needed for a fast system reaction. Examples include predicting pedestrian behaviours in autonomous driving [1] and controlling virtual characters in computer graphics [2]. In contrast to action recognition [3]–[5] with fully observed human movements, anticipating motion aims at predicting the future moving trend from partially observed motion seed, and the challenges mainly come from the highly temporal uncertainties on complex topological structures formed by body joints. The goal of correctly predicting motion trend becomes not only spatially estimating plausible poses frame by frame, but also maintaining dynamics between frames.

To deal with the above challenges, classical data-driven solutions adopt probabilistic models to interpret human motion using Hidden Markov Model [6] or Gaussian process priors [7]. Such models depend on strong assumptions in statistical distributions, which limits the scope of prediction. The emergence of recurrent neural network (RNN) allows the prediction of motions with complex dynamics [8]–[11], as these networks use both motion history and the current pose to learn the temporal dependencies. Despite the improved accuracy, it is still challenging for the RNN-based model to precisely preserve the motion dynamics during prediction.

In this paper, we investigate three problems in existing motion prediction approaches with an RNN-based structure: 1) Mining the spatial interdependency among body joints; 2) Reducing temporal discontinuity at early prediction; 3) Preserving motion trend in long-term prediction.

In terms of mining spatial interdependency, we form a bi-directional diffusion graph on joints with adaptive connectivity to capture the dependencies within multiple spatial steps. Vanilla RNN generates unrealistic movements without spatial modeling [8], it is usually accompanied by a limb-level aggregation [11]–[13] while ignoring the abundant communications among joints, which ends up with an inaccurate pose estimation. Here, we focus on a more generalized solution to explore the topology of the graph formed by joints without body part constraints. By regarding each human joint as a graph node, we make our graph connectivity to be adaptive with network training to model flexible joint combinations without skeletal restrictions. We then perform graph convolutions [14] along multi-step random walks on the adaptive graph topology with a forward and backward diffusion process. Unlike the majority of existing methods that only model graph convolutions with one-way propagation, we integrate both forward and backward node information along the random walks, as the movement of different joints may also affect each other.

Regarding the temporal discontinuity at early prediction, we solve it by modeling motion velocity to encode continuous dynamics from the motion seed instead of raw poses. When synthesizing future movements, the discontinuity problem

Manuscript received June 26, 2020; revised September 23, 2020; accepted November 1, 2020. Date of publication November 13, 2020; date of current version September 3, 2021. This work was supported in part by the grants from the City University of Hong Kong (Project No. 9220077 and 9678139) and in part by the Royal Society (Ref: IES\R2\181024 and IES\R1\191147). This article was recommended by Associate Editor Z.-J. Zha. *(Corresponding author: Edmond S. L. Ho.)*

Qianhui Men and Howard Leung are with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR, China (e-mail: qianhumen2-c@my.cityu.edu.hk; howard@cityu.edu.hk).

Edmond S. L. Ho is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K. (e-mail: e.ho@northumbria.ac.uk).

Hubert P. H. Shum is with the Department of Computer Science, Durham University, Durham DH1 3LE, U.K. (e-mail: hubert.shum@durham.ac.uk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2020.3038145.

Digital Object Identifier 10.1109/TCSVT.2020.3038145

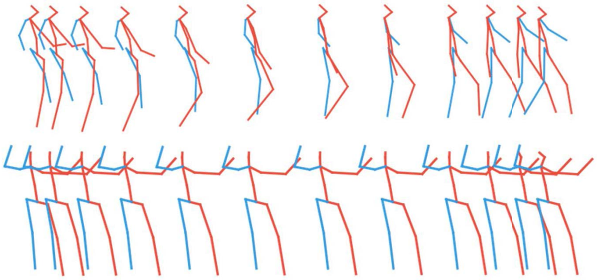


Fig. 1. Illustration of a high-dynamic motion (top) and a low-dynamic motion (bottom) in Human3.6M.

describes the irregular jump between the given motion and prediction. Residual connections [8], [15], [16] attempted to eliminate this by modeling the static poses to predict the dynamic velocities, where the discontinuity still exists as the motion dynamics is indeed not observed by the model. This motivates us to train the velocity in a consistent way, i.e., predicting the next velocity from the previous velocity rather than the previous pose, to maintain the moving regularities inherited from its seed motion dynamics. As a result, it shows a better continuity than residual connections. We further propose a velocity-pose reconstruction loss that optimizes the poses reproduced from the predicted velocity to ensure not to create unexpected movements.

To preserve the motion trend in long-term prediction, we propose a bi-directional predictor enhanced by a bi-discriminator to adversarially revise the generated forward and backward motion dynamics. From a single forward predictor [8], [10], [17], prediction errors are rapidly accumulated along the temporal domain since RNN models fail to keep the long-term knowledge in recurrent steps, causing the generated motion drifting to a wrong direction. To this end, we train a backward predictor to encode the velocity in reversed timesteps, such that the model recovers the context from the beginning dynamics that are lost during long sequence transition. Furthermore, the forward and backward predictors together with a bi-directional discriminator will guide the generated velocity sequence to detect and revise its error from both past and future dynamics through adversarial training. To reduce the model complexity, we also leverage the similarity between the predictor and discriminator in the same direction using a weight sharing structure.

In particular, our predictor is formed by embedding the multi-step diffusion convolutions in the gated recurrent unit (GRU) [18] to synchronously learn the spatial-temporal relationship of motion dynamics under a recurrent sequence-to-sequence (seq2seq) [19] pipeline. With dual directions in both space and time illustrated above, we achieve a quadruple diffusion convolutional recurrent network (Q-DCRN) for a precise motion dynamic prediction.

Comparing with the state of the art, we test on Mean Angle Error (MAE) as previous motion prediction works [8], [15], [16]. We also verify the predicted sequence with position-based metrics, i.e., Mean Per Joint Position Error (MPJPE) and Percentage of Correct Keyjoints (PCK) [20], to better tell whether a prediction follows the ground truth in pose level. Experimental results show that in terms of different metrics, our Q-DCRN outperforms the state of the arts on both 2D and 3D human pose datasets. The qualitative study also shows that the proposed method correctly preserves both high- and low-dynamic motions in long-term prediction, where previous work could not handle both cases. Here, a high-dynamic motion refers to an active motion state with more movements and a low-dynamic motion is the opposite (see Fig. 1). We also verify our improvements with ablation studies.

To summarize, the main contributions of this paper are:

- We propose a bi-directional diffusion graph under adaptive joint connectivity to mine the spatial interdependency for human motion prediction;
- We propose to model velocity from the seed motion dynamics to reduce temporal discontinuity at early prediction meanwhile optimizing the restored poses to avoid unexpected generations;
- We propose a bi-directional temporal predictor to reduce error accumulation from both past and future motion dynamics in an adversarial manner.

The rest of the paper is arranged as follows. Section II reviews the background research related to our work. Section III explains the proposed Q-DCRN prediction framework. Section IV analyses the experiment results and discusses our system. Lastly, Section V concludes this paper.

II. Related Work

We first review how existing research learns the spatial structure in sequential-based networks (i.e., Spatial Perception). We then summarize the background efforts in reducing the initial discontinuity (i.e., Temporal Discontinuity at Early Prediction), and the long-term errors (i.e., Long-term Motion Drift). After that, we present different types of parameterizing during training and their evaluation metrics (i.e., Parameterizations).

A. Spatial Perception

Sequential learning is the common approach to modeling temporal dynamics of human motion, since body joints are highly correlated with each other, it is equally important to consider the inherent spatial structure for generating a natural pose in the meanwhile. Butepage *et al.* [12] originally proposed a hierarchical encoder based on the kinematic tree using fully-connected layers, which outperforms its experimental counterpart without the structural prior. Similarly, Wang *et al.* [11] learned the high-level spatial representations by encoding hierarchical features extracted from different body components, and predict batch of frames at once to prevent the mean pose problem. In contrast to [11] and [12], Aksan *et al.* [21] considered skeleton hierarchy at the output stage for reconstructing controllable poses, and their idea can also be attached to existing works as extra structure-aware layers to further promote motion prediction performance. While in these works, the subdivision of joints into groups is a strong assumption under the articulated chain, and it overlooks the characteristic joint-level correlations.

Graph convolutional network (GCN) [22] is an alternative solution to integrally consider all joints as graph nodes. By merging the features of a joint with its nearby neighbours, GCN shows potential in modeling human pose under graph structures. When combining with the recurrent framework, GCN shows great advantages in analysing graph-based sequential data. For example, Seo *et al.* [23] modeled natural language represented by the nearest neighbour graph and learned temporal regularities using the RNN pipeline. In parsing motion patterns, Si *et al.* [24] exploited spatial-temporal graph convolution on dynamic skeleton sequence to boost the performance of action recognition. In this paper, we adapt the method originally for traffic network modeling [25] to our motion prediction task with multiple spatial and temporal steps to anticipate future movements. Since motion dynamics have more complex topology structures and more stochastic temporal variations, as discussed in [26] and [27], using fixed graph connection limits the spatial proximity to the predefined configuration (i.e., kinematic chain in skeletal structure). Therefore, we design our graph connectivity to be adaptive, so it is capable of learning the underlying dependencies among joints, and temporally we use a bi-discriminator to rectify the motion following a realistic moving pattern.

Recent researches also adopt GCN for motion predictions over innovative graph structures. Li *et al.* [28] constructed a multiscale graph structure based on different body components for motion prediction. While this method provides a comprehensive coarse-to-fine modeling, extra knowledge is required to group the body into skeleton subsets, which makes it deterministic and hard to be transferred to other skeletal structures. Moreover, the cross fusion of the multi-level structures in [28] also increases the computational complexity, resulting in a slower prediction process. Similar to our graph structure, Cui *et al.* [29] defined adaptive joint connectivities and achieved impressive prediction results under a deep GCN framework. However, their joint information can only be updated from its neighbour joints one step away. In this work, we conduct diffusion convolutions on joints by integrating information several steps away to capture global dependencies, which also provides more insights on the understanding of graph structure.

B. Temporal Discontinuity at Early Prediction

The temporal discontinuity in the beginning is harmful as it delivers wrong initial information to its following prediction, which may derive an unexpected motion sequence with a large error rate. In heuristic research for motion prediction, a representative residual network [8] was first proposed to estimate velocity, which has achieved great success in reducing initial discontinuity of the generated sequence compared with previous attempts [30], [31] predicting only static poses. This triggers many sequential-based motion prediction frameworks [13], [16], [32] introducing residual connection into their baselines. One step of residual connection means that the system outputs velocity from the pose, and adds the velocity back to the previous pose to predict the next step. However, the initial error remains notable during prediction as these methods only encode pose features while unseen to the dynamics from the motion seed. This causes inconsistency in preserving the moving trend for prediction, which violates the overall coherence of motion dynamics. In our case, we model the velocity from the given motion and observe a better continuity property.

C. Long-Term Motion Drift

The phenomenon of error accumulation during testing is originally observed in [30] who proposed an Encoder-Recurrent-Decoder (ERD) network and a multi-layer Long Short-Term Memory network (LSTM-3LR) to decode motion frame by frame. To detect the error, they suggested curriculum learning [33] to increasingly perturb input to mimic the distribution of the noisy prediction. The idea of noise scheduling is later absorbed in [31] who introduced Structure-RNN (SRNN) of mixture units interactions concerning an artificial spatial-temporal graph. Unlike ERD and SRNN, Martinez *et al.* [8] proposed a sampling-based loss to synthesize the next frame completely from its previous predicted pose. The method performs less satisfactorily in the long run for its invisibility of real motions. Later, a convolutional seq2seq network [15] is defined to identify spatial-temporal motion correlations. However, their learned temporal dependency is restricted by a deterministic filter size, causing an intensive long-term dynamic loss in prediction. Recently, Dong and Xu [32] attempted to reduce long-term error by looking back at previous frames with spatial attention. Chen *et al.* [34] avoided motion drift by generating early prediction controlled by the action label, while our model is label-agnostic and is also feasible for long-term prediction.

With the assistance of generative adversarial network (GAN), the generative model is able to produce realistic motions with less motion drift. Gui *et al.* [35] first incorporated a fidelity and a continuity discriminator with a residual generator to fix the prediction process. Later in [16], RNN was equipped with an extrinsic factor to find the intended probabilistic space of poses with the assist of a bi-directional discriminator. Note that their adversarial training aims to predict probabilistic priors, while we explore the native ability of a bi-discriminator to correct the predicted motion from two temporal directions in an effective weight-sharing strategy.

In [28], [29], temporal convolution network (TCN) is adopted to process motion history. By aggregating high-level temporal information, TCN shows an advantage over RNN in short-term prediction by generating smoothed poses. However, this advantage becomes weak in the long term especially for high dynamic motions, with the side effect of losing dynamic details. In this paper, we enhance RNN with a velocity-based discriminator to correct the generated moving trend, which eventually performs better in preserving long-term high dynamics compared with TCN-based methods [28], [29].

D. Parameterizations

The method of parameterizing human motion inevitably affects the outcome of final prediction, such as exploiting joint positions is more interpretable than joint angles but may generate invalid articulations. In most cases, input motion is parameterized as exponential maps, which obtains satisfactory results. Pavllo *et al.* [36] employed quaternion representation accompanying with the property of orientation interpolation across frames, and this brings a smooth path in the estimation. Holden *et al.* [2], [37] learned latent feature representations by operating 3D joint positions, which benefits multiple application fields like motion generation, recovery, and comparison. While training on 3D position suffers from skeleton constraints such as bone stretching, in [17] and [21], they modeled joint angles and tested on both angle and position spaces of their generations for a more comprehensive evaluation under different parameterizations. Following their work, for Human3.6M [38] and CMU MoCap datasets [39] we train on joint angles as they are invariant of bone length constraints and thus stabilizing the model fitting. In the test phase, we compare the joint angle as a standard metric used in previous models, and also joint position to convince of the prediction quality. The experiment on the Penn Action dataset [40] is carried out on key joint positions because of its data representation format in 2D space.

III. The Quadruple Diffusion Convolutional Recurrent Network

Our goal is to holistically learn the spatial-temporal joint correlations to preserve the motion trend. To achieve this, we propose an innovative approach to modeling joint dynamics in the velocity field under a graph-based sequential network architecture, with dual directions in both space and time.

We first explain the problem definition and introduce the notation that will be used throughout our framework. In general, a human motion sequence consists of consecutive poses, and each of them is represented by multiple joints. We assume that the interaction within two joints is directed and heterogeneous, i.e., the influence from joint p to joint q is different from q to p, which better models the effect of the body hierarchical structure [41]. Taking “arm swing” as an example, shoulder dynamics largely determine hand movements, while the influence will be smaller from hand to shoulder. However, this diversity cannot be modeled by an undirected graph where two opposite directions are weighted equally. Under this observation, a human pose can be constructed under a directed graph $G = (V, E)$, where V is the vertex set with K nodes, i.e., $|V| = K$, and E is the edge set. $A \in \mathbb{R}^{K\times K}$ is the graph adjacency matrix denoting the spatial proximity between nodes. Here, A is not symmetric in order to represent the inequality in the two-way connectivity. Given a prefix of human poses $X_{1:t} = [x_1, x_2, \ldots, x_t]$, where $x_i \in \mathbb{R}^K$ is defined on graph G at time i, the purpose of motion prediction is to estimate the motion postfix $X_{(t+1):T}$. Since we operate on the velocity domain, our task is characterized as estimating $[\nabla x_{t+1}, \ldots, \nabla x_T]$ from $[\nabla x_2, \ldots, \nabla x_t]$ under G, where the backward difference $\nabla x_i = x_i - x_{i-1}$ denotes the motion velocity at time i.

A. Bi-Directional Spatial Formation

We construct bi-directional diffusion convolutions on an adaptive graph structure to discover the spatial interdependency among joints. Diffusion convolutions [42], [43] aggregate messages passing within high-order neighbours by formulating the node communication as a diffusion process with multiple steps, comparing to standard GCN that only considers local node correlations. Since the joint dynamics can be influenced by the joints from several spatial steps away and vice versa, such as the movements of the joints in legs and arms always affecting each other to maintain the body balance, we regard the spatial dynamics flow as a divergent and convergent diffusion process separately to simulate upstream and downstream node communications. This is because under a conventional directed graph, the diffusion will only apply along a single direction from the root node to the child node within several steps [44], i.e., a divergence random path. Here, the extra convergent path is to complement the divergence in order to model the two-way information delivery, such that the child node can also influence its root node.

The diffusion processes are conducted on a novel graph structure with adaptive joint connectivity. In existing graph-centered networks [43], [45], [46], the topology structure of graph reflected by the node connectivity A is unweighted and artificially defined. In diffusion convolution, [43] provided a general case under an unweighted and undirected graph for node classification tasks, where the node connections are with equal importance. Their model is expected to learn the dominant graph structure that can discriminate against a certain type of cluster, regardless of the connectivity strengths between nodes. However, in motion prediction, the unweighted structure cannot quantify the joint dependency, which may lead to ambiguous joint movements. Furthermore, the predefined topology in human modeling indicates only the joints connected by bones are communicative, which ignores the abundant collaborative information among latent connections [3]. For example, the connection between two feet is important as it symbolizes the gait pattern during locomotion, but it will not be highlighted under the traditional setup. Therefore, instead of manually defining A that restricting the graph descriptiveness within the kinematic tree structure, we set A as learnable during network training to reveal the inherent connection strengths among joints acquired by the real motion data. Here, A is randomly initialized following a standard uniform distribution within the range $[0, 1]$.

With the adaptive graph structure, we then define a two-way diffusion convolution with polynomial recurrences to mine the interdependency of joints within multiple spatial steps. More specifically, in a diffusion process [47] with divergent random paths, a weighted combination $\sum_{n=0}^{N} \theta_n (D^{-1}A)^n$ is used to estimate the graph stationary distribution $\xi \in \mathbb{R}^{K\times K}$ with a truncation at step N, and θ_n is the nth factorization. This polynomial quantifies the effect of root nodes on their child nodes within N spatial steps spreading from the upstream. $D^{-1}A$ is the normalized adjacency matrix, where D is a degree matrix with its diagonal elements representing the row summation of the absolute A. The transpose A^T describes the spatial affinity for downstream diffusion process, which can be used to capture the impact of child nodes on their ascendant nodes. The diffusion convolution operation with dual random walks (denoted as $*_G$) is defined by:

$$H_t *_G \Theta = \sum_{n=0}^{N}((D_u^{-1}A)^n H_t \Theta_{u,n} + (D_d^{-1}A^T)^n H_t \Theta_{d,n}), \tag{1}$$

where $H_t \in \mathbb{R}^{K \times F}$ is the input features of the current step t with F denoting the latent feature dimension, Θ is the weights of the convolution filter to be trained and $\Theta_{:,n} \in \mathbb{R}^{F \times P}$ with P representing the dimension of output features. D_u and D_d are upstream and downstream diagonal matrices normalizing divergence and convergence on G, respectively. When $n = 0$, the two terms in Eq. (1) are merged and no diffusion is conducted. The dual directional diffusion procedure of our spatial structure is illustrated in Fig. 2.

To facilitate the refinement of the diffusion procedure, here we use an N-step diffusion along the two-way random walks on the spatial graph of human dynamics. Diffusion with multiple delivery steps gets access to the combination of different levels of impact. A lower-order n will only grasp the interactions between a few nodes, which is effective in describing the movements with a small body scope such as “waving hand”. A higher-order n could show spatial dependencies among a set of nodes, which is valuable in characterizing global physical coordination like “walking" and “jumping". The choice of total diffusion step N is empirical (see Fig. 11), as more steps will refine the diffusion process with the random walks traversing more often along the joints in a close relationship, on the other hand, it yields a more complicated model.

B. Bi-Directional Temporal Modeling

As observed from the bi-directional computation for time series [48], modeling temporal sequences in the forward and backward directions equips the system with rich contextual information from both past and future conditions. This is extremely useful for human motion prediction who will also borrow the information from the future dynamics to revise promptly in order to keep the long-term motion trend.

Under a seq2seq recurrent architecture, we propose a bi-directional predictor to encode the forward and backward motion dynamics. In the traditional single-predictor setup that only considers the forward direction [8], long-term movements are not guaranteed because a current pose only has access to the dynamics in the past and drift itself into a wrong moving direction. To alleviate this motion drift, we propose a novel two-way predictor to make the system aware of its own generated dynamics from the past and the future.

Furthermore, we also propose an adversarial bi-discriminator to reinforce the predictor such that it can adjust its own forward and backward generation synchronously according to the real motion dynamics. From previous work, when a single directional discriminator is used [35], the long-term errors are easily accumulated due to the difficulty in correcting small mistakes at early prediction. This is because the recurrent temporal modeling tends to focus more on the latest inputs. The function of the backward discriminator is to help the predictor correct the beginning predicted frames to reduce error accumulation.

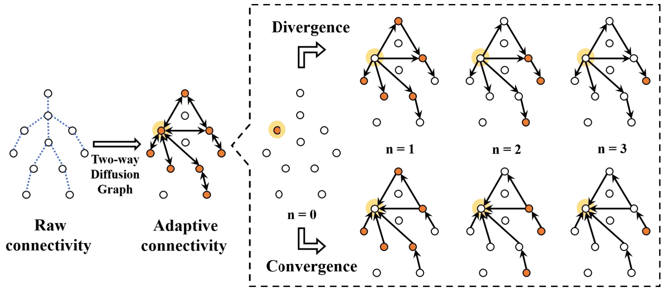


Fig. 2. An example of the dual directional 3-step diffusion graph concentrated on the “left shoulder” joint. The arrow between graph nodes represent the diffusion direction. The nodes in orange are the activated graph nodes that taking part in the feature fusion at the current step.

We also design a model compression method that could efficiently communicate between the bi-predictor and the bi-discriminator since they both need to encode the motion dynamics, i.e., we share the structures and weights between them within the same directions. This helps the common component to quickly converge to the optimal motion manifold and prevents the complicated GAN training from scratch.

The bi-directions of both spatial diffusion and temporal predictor together form a quadruple diffusion convolutional recurrent network (Q-DCRN) as shown in Fig. 3. In the framework, we consider the sampling-based inference (i.e., feeding in its generation per step) in the bi-predictor such that it is bi-directional knowledgeable of its own dynamics, and the teacher forcing learning (i.e., feeding in the ground truth per step) in the bi-discriminator such that it revises the predictor with real dynamics.

Here, we elaborate the details of our bi-predictor and bi-discriminator constructions. We formalize the forward predictor (i.e., BiS-DCRN) using a diffusion convolutional GRU (denoted as GRU_{*G}) as the basic recurrent unit. As an alternative to LSTM [49], GRU [18] has comparable performance with more portable gate mechanisms. Intuitively, we embed the dual directional diffusion convolution (Eq. (1)) into the GRU cell as a substitute for the matrix multiplication inside each gate. By absorbing current motion velocity ∇x_t and the previous hidden state h_{t-1} as input, a one-step diffusion convolution transition based on GRU can then be formatted as

$$h_t = \text{GRU}_{*G}([\nabla x_t, h_{t-1}]; \mathbf{w}), \tag{2}$$

where $\mathbf{w}$ is the convolution kernel set. The diffusion convolution $*_G$ is conducted on the update gate z_t, the reset gate r_t, and the candidate c_t of GRU, and we illustrate its detailed operations in z_t as an example in Fig. 4. The same operations are conducted for r_t. In c_t, the h_{t-1} in the structure is replaced by the dot product $r_t \odot h_{t-1}$, and f becomes *tanh*. The hidden state $h_t = z_t \odot h_{t-1} + (1 - z_t) \odot c_t$ follows the standard GRU architecture.

Next, we encode the prefix of motion dynamics frame by frame. The encoded hidden state along with the last frame observation is utilized to activate the decoder. The entire predictor is under a seq2seq backbone. After translating the input motion velocity into high-dimensional expression under GRU_{*G}, the output will go through a linear projection converted back to velocity space. The decoder will decode

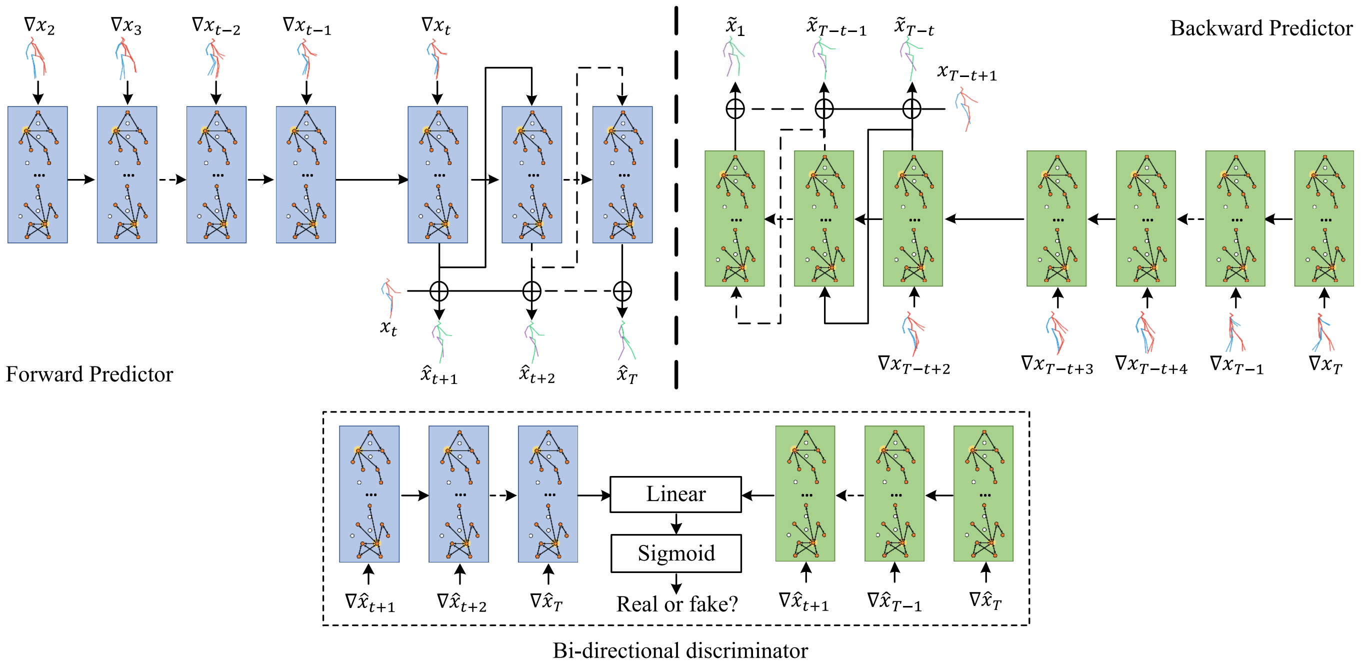


Fig. 3. The proposed Q-DCRN framework (unrolled version) with the outline of dual directional processes in both space and time. The blue and green boxes denote the GRU cells with diffusion graphs in forward and backward chronological directions, respectively. The skeleton in red and blue represents the ground truth posture, and the one in green and purple represents prediction. We attach two skeletons to represent the velocity of two adjacent frames. Inside the dotted line is the discriminator structure for adversarial training. $\hat{x}_i$ and $\tilde{x}_i$ are used to indicate forward and backward predicted poses. Parameters of boxes in the same color are shared during training.

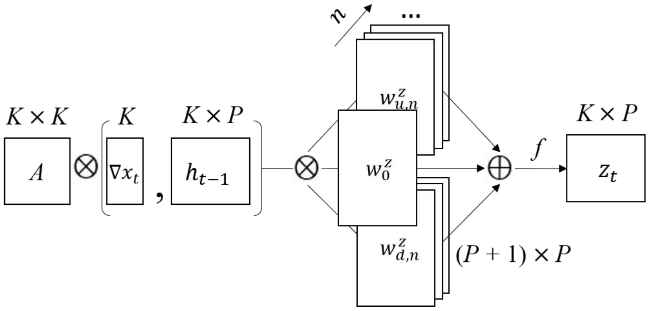


Fig. 4. Illustration of the diffusion convolutional structure in the update gate of GRU. The operators [,], ⊗, and ⊕ denote concatenation, matrix multiplication, and matrix addition, respectively. $w^z_{u,n}$ $w^z_{d,n}$ are the upstream and downstream convolution kernels for the diffusion step n. Note that we only have a single kernel w^z_0 when $n = 0$. The function f denotes activation (σ for z_t).

the predictive velocities under a sampling-based mechanism as in [8]. We follow the same steps for the backward direction by predicting the backward velocity. Our discriminator consists of a forward and a backward diffusion convolutional GRU layer which is shared from the forward and the backward predictor, respectively. The bi-discriminator encodes the generated velocity frame by frame in two directions. The final forward and backward states are concatenated by a linear layer ($K \times P \times 2 \rightarrow 1$) with *sigmoid* activation to output the probability as shown in the lower part of Fig. 3.

C. Velocity-Informed Training

Given the quadruple prediction system, we now explain how to train the system with velocity from the observed motion dynamics to keep continuity at early prediction, and how we optimize the model to ensure a plausible generated sequence in terms of the intra-frame poses and the inter-frame dynamics with the co-operation of two proposed losses.

To reduce the initial frame jump, we propose a training strategy to uniformly interpret the motion velocity from motion observation to prediction. The velocity acts as an explicit indicator to measure the body moving trend [50]. Compared with raw poses, predicting velocity mitigates the loss of temporal dynamics over time, which prevents changeless poses or so-called "dying out". With smaller magnitudes of input values, velocities also assist to regularize the network regression with good generalization ability. However, the general operation to include velocity is to use the residual architecture [8], [13], [16] that outputs the velocity from its observed pose sequence, which leads to the inconsistent dynamics between the prediction and the seed moving trend. To avoid this, we directly learn our system from the observed velocity to predict the future velocity, and this preserves the initial continuity in the generated temporal dynamics.

For the optimization of our framework, we wish the generated motion produces not only a plausible pose at each frame but also an overall right moving dynamics. This is because a generated motion can be intuitively measured under 1) the consecutive pose set and 2) the temporal velocity variation, where the second measurement is usually overlooked by existing research [9], [21], [51].

To this end, we propose a velocity-pose reconstruction loss to penalize the reproduced poses from velocity, together with a general adversarial loss to regularize the dynamics on velocity space. The whole network will optimize alternatively according to these two constraints and search the optimal solution for the predicted motion.

1) The Velocity-Pose Reconstruction Loss: We propose a novel velocity-pose reconstruction loss to measure the generated velocity in pose domain. The “velocity-pose” is defined as deriving the current pose based on the velocity over time and the initial pose. Specifically, for each temporal direction, we first compute the pose displacement by accumulating the predicted velocity sequences and then add it to the initial pose to generate the current pose. The “reconstruction loss” denotes the mean squared error between the ground-truth poses with the generated pose sequence. Since similar velocity chains can derive completely different pose sequences, it is risky to optimize the predictor on the velocity domain [17] when the network is blind to the generated poses. Therefore, we rebuild the poses from the predicted velocity frame by frame, and minimize the loss in the pose level, so that the generated motion is controllable.

Practically, the bi-predictor will output the joint velocities, and we then reduce the cost based on the iteratively derived pose sequence according to the composed objective function with two independent terms calculating forward and backward losses separately:

$$\begin{aligned}\mathcal{L}_{recons} &= \frac{1}{T-t}\Big(\sum_{i=t+1}^{T} \parallel x_i - \hat{x}_i \parallel_2^2 + \sum_{j=1}^{T-t} \parallel x_j - \tilde{x}_j \parallel_2^2\Big)\\ &= \frac{1}{T-t}\Bigg(\sum_{i=t+1}^{T} \Big\| x_i - (x_t + \sum_{i'=t+1}^{i} \nabla \hat{x}_{i'})\Big\|_2^2\\ &\quad + \sum_{j=1}^{T-t} \Big\| x_j - (x_{T-t+1} - \sum_{j'=j+1}^{T-t+1} \nabla \tilde{x}_{j'})\Big\|_2^2\Bigg), \qquad (3)\end{aligned}$$

where $\mathcal{L}_{recons}$ denotes the reconstruction loss conducted on the bi-directional pose sequences.

2) The Velocity-Based Adversarial Loss: We then show the details of how we form our adversarial loss in the velocity domain. With the bi-discriminator encoding the velocity trend, the adversarial loss will guide the generated velocities in two directions to follow the ground truth moving dynamics.

After minimizing the velocity-pose reconstruction loss (i.e., $\mathcal{L}_{recons}$), the optimized predictor weights will be reused in the discriminator (denoted as D) within the same direction to be further updated with respect to the adversarial loss $\mathcal{L}_{adv}$, which is computed by:

$$\begin{aligned}\mathcal{L}_{adv} &= \mathbb{E}_X \log D([\nabla X_{t+1:T}, \nabla X_{T:t+1}]|\mathbf{w}_f, \mathbf{w}_b, \mathbf{w}_0)\\ &\quad + \mathbb{E}_{\hat{X}} \log(1 - D([\nabla \hat{X}_{t+1:T}, \nabla \hat{X}_{T:t+1}]|\mathbf{w}_f, \mathbf{w}_b, \mathbf{w}_0)),\end{aligned} \qquad (4)$$

where $\mathbf{w}_0$ represents the kernel parameters for the linear layer, $\mathbf{w}_f$ and $\mathbf{w}_b$ are the shared forward and backward parameters from the bi-predictor respectively, and $\nabla X_{T:t+1}$ is the reverse of velocity sequence $\nabla X_{t+1:T}$ in time order. The adversarial training follows the minimax optimization:

$$\min_{\mathbf{w}_f} \max_{\mathbf{w}_f, \mathbf{w}_0} \mathcal{L}_{adv}. \qquad (5)$$

By reusing the learned weights $\mathbf{w}_f$ from D, the forward predictor can quickly converge to its target distribution. Note that we do not update $\mathbf{w}_b$ in this step since we want to regulate the forward generation as our final prediction rather than the backward generation. The sharing mechanism will not only keep the prediction consistent with the ground truth motion but also help save computational memory.

IV. Experiments

In this section, we validate the proposed Q-DCRN on both short and long-term predictions. The experiments are conducted on various benchmark datasets commonly used in motion prediction tasks. We then compare the results with the state of the arts and justify the effectiveness of different components of our model.

A. Datasets

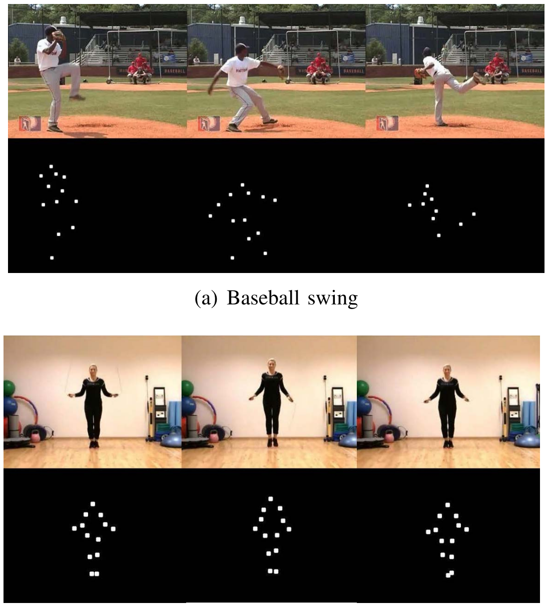
1) Human3.6M: We first experiment on Human3.6M [38], which is a large and canonical 3D human pose dataset for motion analysis. Human3.6M captures 7 actors performing 15 activities with diverse motion dynamics, such as periodic actions with moving regularities like “walking” and “eating”, and aperiodic action with intensive variations like “posing” and “walking dog”. In each frame, there are 32 joints represented by 3D angles in the format of the exponential map. As in [8], global translation and rotation are discarded together with the joint angles in constant standard deviations. The motion sequence is downsized to 25 frames per second. We test on subject #5 while training on the others, and set 50 frames as motion seeds and 25 frames for inference following previous experimental setup [8], [15].

2) CMU MoCap: Following [15], we conduct the second experiment on the CMU Motion Capture dataset (CMU MoCap) [39]. The CMU MoCap database captures 5 main activities produced by 144 actors, which serves over 2000 recordings. This dataset is very challenging with complex sports actions such as “soccer” and “basketball". The skeleton contains 38 joints in each 3D pose. We employ the same criteria of data cleaning as [15]. Human interactions and motions with multiple topics are removed as well as the motion categories with less than 6 trials. The final set contains 8 motion types. We conduct the same pre-processing steps as Human3.6M.

3) Penn Action Dataset: We also experiment on the Penn Action dataset [40] to test the robustness of our approach towards 2D pose forecasting. The Penn Action dataset consists of 2326 trials of human action annotated by 13 joints in the 2D pose. It contains 15 different categories with diverse complexity range as shown in Fig. 5. As in [17], [52], the dataset is split into 1258 samples for training and 1068 for testing. Following [17], we input the initial velocity and predict the next 16 frames of poses.

B. Baselines and Experimental Settings

1) Baselines: In this work, three action-specific models are used for comparison, which are RNN-based models: ERD [30], LSTM-3LR [30] and SRNN [31]. The action-specific model aims to train an individual prediction model for each action. The more general and more



(a) Baseball swing

(b) Jump rope

Fig. 5. Example frames of the Penn Action dataset. The upper rows are sampled from RGB action videos and the bottoms are the corresponding extracted 2D joint positions.

challenging multi-label algorithm aims to train a universe model for all action categories. Our approach follows the intention of multi-label algorithms. We then compare with the state-of-the-art multi-label algorithms related to our network architecture under two types of baselines, which are RNN or CNN-based models: RRNN [8], 3D-PFNet [10], RMA [32], TP-RNN [17], VGRU [10], QuaterNet [36], BiHMP-GAN [16], and ConvSeq2seq [15]; GCN-based models: LDR [29] and DMGNN [28]. To demonstrate the effectiveness of our velocity modeling method, we also present the prediction results for modeling velocity consistently (denoted as VRNN) to compare with RRNN which models posture sequence with residual connections.

2) Evaluation Metrics: We first evaluate our method on the standard metric, i.e., the Mean Angle Error (MAE) calculated on the Euler angle. Besides the common measurement, we also adopt the positional metrics that cover the Mean Per Joint Position Error (MPJPE) and the Percentage of Correct Keypoint (PCK) to validate the predictive ability of models. Previous literature of motion prediction heavily relies on measuring the Euler angle distance and sampling the predicted poses qualitatively. However, merely using the Euler angle as quantitative criteria is unconvincing due to the non-unique solutions for a feasible pose [53]. Hence, we also measure the generated poses using positional metrics as complementary.

a) Mean angle error (MAE): Following the standard evaluation protocol adopted in [8], [15], [16], [32], we first use the mean error of Euler angle as the evaluation metric for a fair comparison among the baselines and the proposed method. The prediction error is calculated from the average of Euler angle difference per joint between prediction and reference. Note that the joint angles are represented by local orientations based on the kinematic chain in the human skeleton.

b) Mean per joint position error (MPJPE): As a common problem in Euler angle representation [53], similar poses may deduce completely different joint angle sets. To avoid such biased verification in the MAE metric, we also evaluate the generated poses on the protocol of MPJPE as suggested by [38], [52]. The MPJPE is to calculate the deviation of estimated joint points by converting the relative angles to absolute joint coordinates using forward kinematics.

c) Percentage of correct keypoint (PCK): To be consistent with TP-RNN [17] and 3D-PFNet [52], we also test PCK on Human3.6M and Penn Action datasets. The intention of PCK is to count the proportion of predicted joints detected within a radius of predefined threshold ρ (in meters) around the objective joints, which is commonly employed in 2D or 3D pose estimation [20], [54]–[56].

3) Implementation Details: We express motion velocity ∇x_i as a graph signal of $\mathbb{R}^K$ and utilize 64 units ($P = 64$) in the GRU cell under graph convolution. The maximum step for spatial diffusion N is set to 3 (see detailed analysis in Section IV-G). To stabilize the optimization process, we employ a scheduled training strategy to balance the predictor and discriminator. We optimize two steps of Eq. (3) followed by one step of adversarial training. The proposed model is trained using gradient descent optimizer with a regressive learning rate of 0.05 on Human3.6M and CMU MoCap, and 0.005 on the Penn Action dataset. We set the batch size to 16, and perform gradient clipping under l_2-norm. The entire network is implemented using the Tensorflow backend.

C. Comparisons on the Human3.6M Dataset

We first compare with the state-of-the-art RNN or CNN-based methods and report their MAE over future timestamps 80ms, 160ms, 320ms, 400ms (for short-term prediction) and 1000ms (for long-term prediction) on Human3.6M. The prediction accuracy comparisons are presented in Table I. We significantly outperform ERD, LSTM-3LR, and SRNN on four actions “walking”, “eating”, “smoking”, and “discussion” that are usually compared in previous works. Generally, VRNN outperforms RRNN even at the primary prediction (80ms), which shows the advantage of our velocity modeling manner over residual connections to improve temporal continuity at early prediction. The visualization results on keeping the continuity can be found in our supplementary video. In Table I, Q-DCRN outperforms the baseline methods on both short and long-term prediction, and the error accumulates slower compared with the other methods along the sampled timestamps.

We also qualitatively verify RRNN, ConvSeq2seq, and our Q-DCRN prediction results towards commonly examined actions on Human3.6M with two high-dynamic actions “walking” and “eating”, and two low-dynamic actions “smoking” and “discussion” (see Fig. 6). We observe that Q-DCRN better simulates the ground truth motion trends compared to the other two methods. For the high-dynamic “walking” action in Fig. 6(a), all three methods show reliable movements as the periodic pattern is easy to capture. We further observe that Q-DCRN gives a precise prediction of double arms staying

TABLE I

EVALUATIONS ON THE STATE-OF-THE-ART RNN OR CNN-BASED APPROACHES AT SHORT-TERM AND LONG-TERM MAE OF HUMAN3.6M DATASET. UNDERLINED VALUES REPRESENT THE LOWER ERROR BETWEEN RRNN AND VRNN. BOLD VALUES REPRESENT THE LOWEST ERROR AMONG ALL METHODS

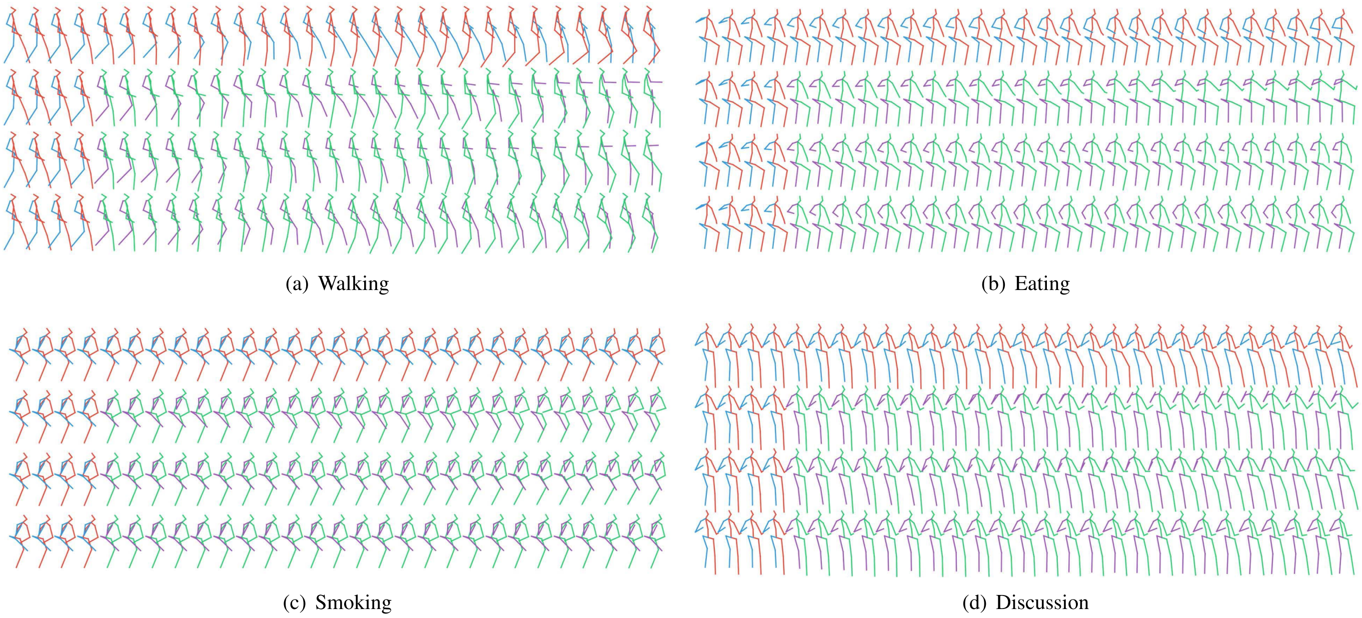
	Walking					Eating					Smoking					Discussion				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
ERD [30]	0.93	1.18	1.59	1.78	N/A	1.27	1.45	1.66	1.80	N/A	1.66	1.95	2.35	2.42	N/A	2.27	2.47	2.68	2.76	N/A
LSTM-3LR [30]	0.77	1.00	1.29	1.47	N/A	0.89	1.09	1.35	1.46	N/A	1.34	1.65	2.04	2.16	N/A	1.88	2.12	2.25	2.23	N/A
SRNN [31]	0.81	0.94	1.16	1.30	N/A	0.97	1.14	1.35	1.46	N/A	1.45	1.68	1.94	2.08	N/A	1.22	1.49	1.83	1.93	N/A
RRNN [8]	0.28	0.50	0.74	0.81	1.12	0.24	0.42	0.69	0.85	1.44	0.34	0.62	1.03	1.15	2.01	0.33	0.72	1.04	1.11	1.92
VRNN (Ours)	0.26	0.45	0.63	0.70	0.86	0.21	0.34	0.55	0.69	1.21	0.26	0.48	0.89	0.90	1.67	0.30	0.65	0.98	1.07	1.77
ConvSeq2seq [15]	0.28	0.48	0.68	0.77	1.08	0.21	0.35	0.57	0.72	1.27	0.27	0.49	0.93	0.91	1.68	0.31	0.65	0.91	1.02	2.01
RMA [32]	0.28	0.45	0.62	0.68	0.79	0.21	0.34	0.53	0.68	1.16	0.26	0.50	0.96	0.93	1.71	0.29	0.64	0.90	0.96	1.72
TP-RNN [17]	0.25	0.41	0.58	0.65	0.77	0.20	0.33	**0.53**	0.67	**1.14**	0.26	0.47	0.88	0.90	1.66	0.30	0.66	0.96	1.04	1.74
VGRU [10]	0.34	0.47	0.64	0.72	N/A	0.27	0.40	0.64	0.79	N/A	0.36	0.61	**0.85**	0.92	N/A	0.46	0.82	0.95	1.21	N/A
QuaterNet [36]	0.21	**0.34**	0.56	0.62	N/A	0.20	0.35	0.58	0.70	N/A	0.25	0.47	0.93	0.90	N/A	**0.26**	**0.60**	**0.85**	**0.93**	N/A
BiHMP-GAN [16]	0.33	0.52	0.63	0.67	0.85	0.20	0.33	0.54	0.70	1.20	0.26	0.50	0.91	0.86	**1.11**	0.33	0.65	0.91	0.95	1.77
Q-DCRN (Ours)	**0.20**	0.36	**0.56**	**0.60**	**0.69**	**0.18**	**0.32**	0.56	**0.67**	1.18	**0.22**	**0.43**	0.87	**0.84**	1.58	0.32	0.69	0.98	1.04	**1.56**
	Directions					Greeting					Phoning					Posing				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
RRNN [8]	0.43	0.69	0.84	0.94	1.49	0.53	0.88	1.34	1.53	2.11	0.60	1.14	1.56	1.72	1.98	0.40	0.76	1.41	1.68	2.55
VRNN (Ours)	0.37	0.58	0.77	0.86	1.37	0.50	0.84	1.27	1.45	1.77	0.57	1.11	1.48	1.63	1.71	0.44	0.83	1.41	1.65	2.51
ConvSeq2seq [15]	0.39	0.60	0.80	0.91	1.45	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
RMA [32]	0.40	0.61	0.77	0.86	1.42	0.52	0.86	1.26	1.43	1.79	0.59	1.11	1.47	1.59	1.73	**0.26**	**0.54**	1.14	1.41	2.43
TP-RNN [17]	0.38	0.59	0.75	0.83	1.38	0.51	0.86	1.27	1.44	1.81	0.57	1.08	1.44	1.59	1.68	0.42	0.76	1.29	1.54	2.47
Q-DCRN (Ours)	**0.28**	**0.45**	**0.62**	**0.70**	**1.31**	**0.38**	**0.67**	**1.11**	**1.32**	1.78	**0.53**	**1.00**	**1.39**	**1.56**	**1.60**	0.30	0.66	1.28	1.52	**2.26**
	Purchases					Sitting					Sitting Down					Taking Photo				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
RRNN [8]	0.59	0.83	1.16	1.24	2.35	0.47	0.77	1.25	1.49	2.15	0.54	1.03	1.58	1.81	2.81	0.33	0.64	0.98	1.10	1.54
VRNN (Ours)	0.60	0.83	1.13	1.21	2.32	0.40	0.64	1.04	1.18	1.68	0.43	0.80	1.17	1.32	1.98	0.27	0.54	0.85	0.98	1.36
ConvSeq2seq [15]	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
RMA [32]	0.59	0.84	1.14	1.19	2.33	0.40	0.64	1.04	1.22	1.71	0.41	0.77	1.14	1.29	2.07	0.27	0.52	0.80	0.92	1.21
TP-RNN [17]	0.59	0.82	1.12	1.18	2.28	0.41	0.66	1.07	1.22	1.74	0.41	0.79	1.13	1.27	**1.93**	0.26	0.51	0.80	0.95	1.35
Q-DCRN (Ours)	**0.46**	**0.68**	**1.08**	**1.13**	**2.16**	**0.29**	**0.51**	**0.88**	**1.05**	**1.63**	**0.37**	**0.73**	**1.03**	**1.15**	1.95	**0.18**	**0.38**	**0.64**	**0.78**	**1.17**
	Waiting					Walking Dog					Walking Together					Average				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
RRNN [8]	0.34	0.67	1.15	1.35	**2.27**	0.53	0.89	1.21	1.35	1.94	0.28	0.56	0.79	0.84	1.36	0.42	0.74	1.12	1.26	1.94
VRNN (Ours)	0.31	0.61	1.11	1.32	2.46	0.54	0.95	1.29	1.45	2.03	0.24	0.51	0.72	0.75	1.29	0.38	0.68	1.02	1.14	1.73
ConvSeq2seq [15]	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
RMA [32]	0.33	0.65	1.12	1.30	2.28	0.53	0.87	1.16	1.33	2.00	0.28	0.52	0.68	0.71	1.31	0.37	0.66	0.98	1.10	1.71
TP-RNN [17]	0.30	0.60	1.09	1.31	2.46	0.53	0.93	1.24	1.38	1.98	0.23	0.47	0.67	0.71	1.28	0.37	0.66	0.99	1.11	1.71
Q-DCRN (Ours)	**0.26**	**0.56**	**0.99**	**1.18**	2.33	**0.46**	**0.79**	**1.10**	**1.20**	**1.82**	**0.20**	**0.40**	**0.57**	**0.62**	**1.20**	**0.31**	**0.57**	**0.90**	**1.02**	**1.60**

behind the legs while its competitors fail to do so, which shows the effectiveness of globally modeling joint dependencies along the spatial graph. For the action "eating" in Fig. 6(b), there is an interesting investigation that both RRNN and ConvSeq2seq move the active arm to its opposite direction compared with ground truth. This is because the error propagation issues resulted in large posture deviation in long-term motion prediction. We can see that through narrowing the deviation at the early phase, Q-DCRN is able to maintain the right motion trend inherited from its seed sequence.

Other than preserving the high-dynamic trend, we also show a better prediction in low-dynamic motions. For "smoking" in Fig. 6(c), RRNN performs an unexpected action of putting down the leg. This is because the residual connections in RRNN force the decoded prediction to move, which makes it difficult to synthesize low-dynamic or motionless sequences. While Q-DCRN could keep the static trend with the input velocity closes to zero. For "discussion" in Fig. 6(d), both RRNN and ConvSeq2seq fail to catch the pace of the arm movements, which results in wrong predictions for the arm direction detected in the long term. Such observations suggest that Q-DCRN can handle both high-dynamic and low-dynamic motions precisely following the real poses. Please refer to the supplementary video for more qualitative comparisons on high-dynamic and low-dynamic predictions.

In "discussion", we also notice a better visualization result (Fig. 6(d)) but a worse MAE ("discussion" in Table I). To further investigate the inconsistency between the visualization and quantitative results, we test the MPJPE on the four actions and their average in Fig. 7(a). For all methods, "walking" and "eating" on average have lower MPJPE than "smoking" and "discussion" because these actions contain more repetitive patterns that are easily captured. We also report the long-term PCK accuracy in Fig. 7(b), and Q-DCRN already succeeds under a small threshold (0.025), which means more predicted joints are falling within the neighbour region of real joints in long-term prediction. Note that comparing with other methods, we achieve the best performance (the lowest MPJPE and the highest PCK accuracies) on these actions. This aligns with the visualization in Fig. 6 that we are the closest to the ground truth poses, which also indicates that positional evaluation is more reliable than Euler angle-based metric.

We also compare with the recent GCN-based prediction methods in Table II with 320ms, 400ms (short term) and more timesteps 520ms, 640ms, 760ms, 880ms, 1000ms (long term). In the short term, both LDR and DMGNN produce better numerical results as their employing of TCN generates smoothed movements from motion history with lower errors at the beginning of prediction. However, the smoothness may somehow degrade long-term high dynamic motions such as "walking" and "walking together". We found that Q-DCRN performs better in such motions by preserving the long-term dynamic trend. When comparing with DMGNN, this advantage is more obvious with over half of the lower errors laying



(a) Walking

(b) Eating

(c) Smoking

(d) Discussion

Fig. 6. Qualitative comparisons with the state-of-the-art RNN or CNN-based approaches on the Human3.6M dataset. For each action, the top sequence refers to the ground truth. The second, third and bottom sequences correspond to RRNN, ConvSeq2seq, and our Q-DCRN, respectively. The initial four poses are the seed frames, followed by one second of prediction.

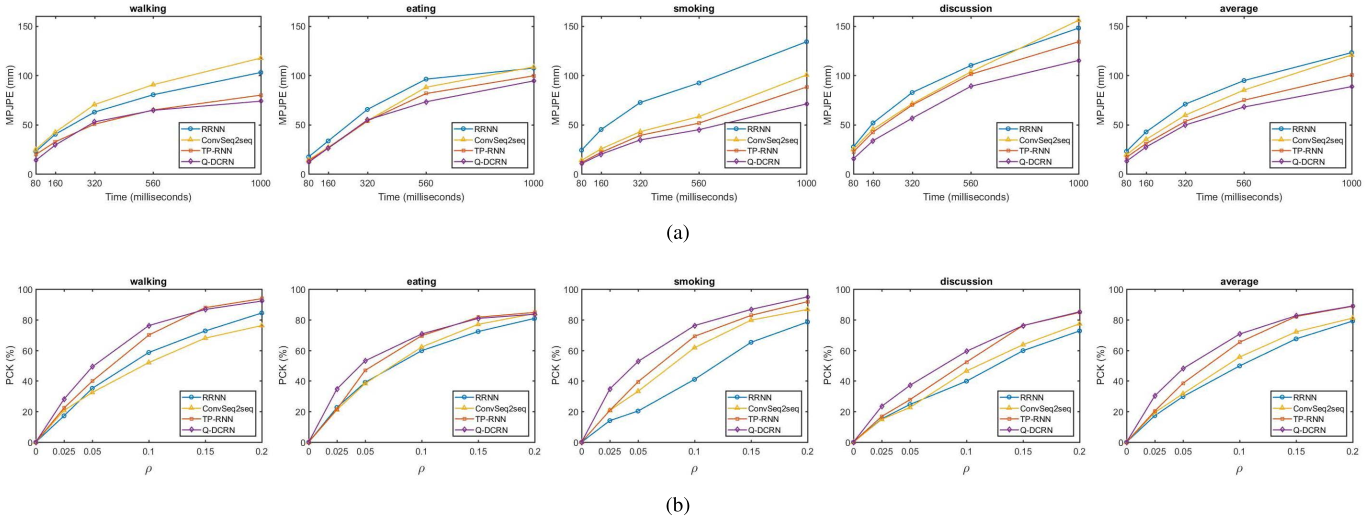


Fig. 7. Evaluations on the two positional metrics of Human3.6M with (a) MPJPE ↓ curves along the prediction timeline and (b) PCK ↑ curves at 1000ms under different thresholds ρ. ↓ the lower the better, ↑ the higher the better.

in our approach. In practice, those high-dynamic motions are very common in our daily life and may also indicate some dangerous situations such as walking across the road, thus are highly valued in motion prediction tasks [2], [11], [57].

We further give two examples in Fig. 8 to show the differences in the generated high-dynamic sequences. In Fig. 8(a), we observe that DMGNN tends to lose the active walking trend with an over-smoothed prediction. While our result still keeps the long-term walking cycle with relatively large steps similar to the ground truth. Figure 8(b) is a “walking dog” action with lots of movements in arms and legs. For DMGNN, their prediction is losing the moving dynamics by generating mean poses and eventually results in unnatural poses. In contrast, our prediction still preserves the active movements such as raising the right hand to keep balance.

D. Comparisons on the CMU MoCap Dataset

The MAE results on CMU MoCap are shown in Table III and the average comparisons across all 8 motion categories are given in Table IV. We achieve a comparable result on CMU MoCap with most of the best predictions falling in our approaches. We observe from the angular result that Q-DCRN works well especially on actions with legible intentions or consistent changes such as “sitting” or “basketball signals”, but shows higher errors on the actions with large accelerations like “jumping” or “running”. From the average performance in

TABLE II

EVALUATIONS ON THE STATE-OF-THE-ART GCN-BASED APPROACHES AT SHORT-TERM AND LONG-TERM MAE OF HUMAN3.6M DATASET

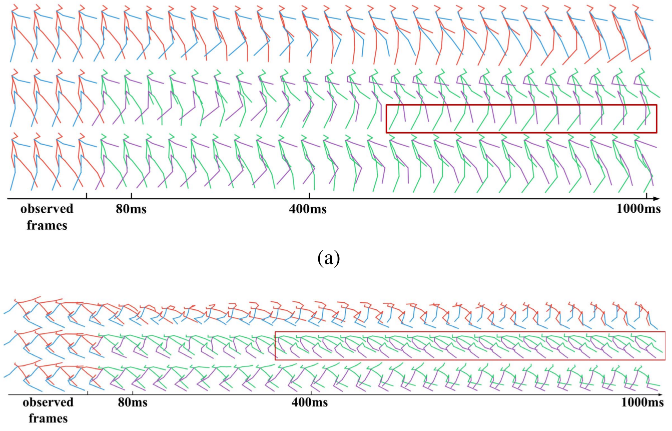
	Walking							Eating						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.46**	**0.57**	N/A	N/A	N/A	N/A	0.71	**0.49**	0.64	N/A	N/A	N/A	N/A	**0.97**
DMGNN [28]	0.49	0.58	0.67	0.71	0.74	0.70	0.78	**0.49**	**0.59**	**0.77**	0.91	0.99	1.06	1.14
Q-DCRN (Ours)	0.56	0.60	**0.66**	**0.70**	**0.69**	**0.67**	**0.69**	0.56	0.67	0.79	**0.85**	**0.89**	**1.05**	1.18
	Directions							Greeting						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.59**	**0.68**	N/A	N/A	N/A	N/A	**0.95**	**0.87**	**0.98**	N/A	N/A	N/A	N/A	**1.33**
DMGNN [28]	0.65	0.71	1.00	1.09	1.23	1.34	1.40	0.94	1.12	**1.57**	**1.51**	**1.64**	1.82	1.80
Q-DCRN (Ours)	0.62	0.70	**0.80**	**0.94**	**1.14**	**1.28**	1.31	1.11	1.32	1.69	1.63	1.69	**1.79**	1.78
	Purchases							Sitting						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.88**	**1.08**	N/A	N/A	N/A	N/A	**1.49**	**0.69**	1.01	N/A	N/A	N/A	N/A	**1.38**
DMGNN [28]	1.05	1.14	1.57	1.71	1.86	2.20	2.42	0.76	**0.97**	1.21	1.29	1.46	1.59	1.63
Q-DCRN (Ours)	1.08	1.13	**1.34**	**1.41**	**1.56**	**1.93**	2.16	0.88	1.05	**1.19**	**1.27**	**1.44**	**1.56**	1.63
	Waiting							Walking Dog						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.84**	1.15	N/A	N/A	N/A	N/A	**1.21**	**0.93**	**1.14**	N/A	N/A	N/A	N/A	**1.38**
DMGNN [28]	0.88	**1.10**	**1.33**	**1.58**	**1.88**	**2.11**	2.17	1.16	1.34	1.85	1.97	2.16	2.18	2.22
Q-DCRN (Ours)	0.99	1.18	1.46	1.73	2.02	2.25	2.33	1.10	1.20	**1.45**	**1.50**	**1.71**	**1.77**	1.82

	Smoking							Discussion						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.79**	0.82	N/A	N/A	N/A	N/A	**1.08**	**0.72**	**0.81**	N/A	N/A	N/A	N/A	**0.84**
DMGNN [28]	0.81	**0.77**	**0.78**	**0.82**	**0.96**	**1.24**	1.48	0.92	0.99	**1.20**	**1.34**	**1.39**	**1.35**	1.40
Q-DCRN (Ours)	0.87	0.84	0.89	0.95	1.10	1.35	1.58	0.98	1.04	1.25	1.41	1.51	1.58	1.56
	Phoning							Posing						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.63**	**0.78**	N/A	N/A	N/A	N/A	**1.33**	**0.91**	**1.07**	N/A	N/A	N/A	N/A	**1.34**
DMGNN [28]	1.29	1.43	**1.22**	1.39	1.52	1.61	1.62	1.06	1.34	**1.46**	**1.38**	**1.52**	**1.76**	1.96
Q-DCRN (Ours)	1.39	1.56	1.27	**1.34**	**1.47**	**1.55**	1.60	1.28	1.52	1.71	1.64	1.78	2.07	2.26
	Sitting Down							Taking Photo						
Time (milliseconds)	320	400	520	640	760	880	1000	320	400	520	640	760	880	1000
LDR [29]	**0.87**	**0.93**	N/A	N/A	N/A	N/A	**1.42**	**0.54**	**0.71**	N/A	N/A	N/A	N/A	1.20
DMGNN [28]	0.93	1.05	**1.18**	**1.37**	**1.51**	**1.59**	1.68	0.58	**0.71**	0.91	0.99	1.10	1.21	1.32
Q-DCRN (Ours)	1.03	1.15	1.29	1.52	1.69	1.83	1.95	0.64	0.78	**0.89**	**0.96**	**1.05**	**1.10**	**1.17**
	Walking Together													
Time (milliseconds)	320	400	520	640	760	880	1000							
LDR [29]	**0.49**	**0.54**	N/A	N/A	N/A	N/A	1.38							
DMGNN [28]	0.50	0.57	0.82	0.96	1.07	1.14	1.47							
Q-DCRN (Ours)	0.57	0.62	**0.67**	**0.75**	**0.80**	**0.83**	**1.20**							

TABLE III

EVALUATIONS ON MAE OF CMU MOCAP DATASET

	Basketball					Basketball Signal					Directing Traffic					Jumping				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
RRNN [8]	0.50	0.80	1.27	1.45	1.78	0.41	0.76	1.32	1.54	2.15	0.33	0.59	0.93	1.10	2.05	0.56	0.88	1.77	2.02	2.40
VRNN (Ours)	0.44	0.69	1.10	1.23	**1.77**	0.17	0.33	0.62	0.75	1.37	0.30	0.60	0.98	1.12	2.27	**0.38**	0.66	1.50	1.73	2.11
ConvSeq2seq [15]	0.39	0.66	1.14	1.31	2.18	0.34	0.64	1.15	1.35	1.91	0.25	0.60	0.92	1.01	2.05	0.41	0.67	1.45	1.64	2.08
BiHMP-GAN [16]	0.37	0.62	1.01	**1.11**	1.83	0.32	0.56	1.01	1.18	1.88	**0.25**	0.51	0.85	0.96	**1.95**	0.39	**0.57**	**1.31**	**1.50**	1.93
Q-DCRN (Ours)	**0.34**	**0.55**	**1.00**	1.19	2.34	**0.09**	**0.18**	**0.35**	**0.44**	**0.92**	0.26	**0.41**	**0.76**	**0.92**	2.07	0.39	0.68	1.45	1.59	**1.72**
	Running					Soccer					Walking					Wash Window				
Time (milliseconds)	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000	80	160	320	400	1000
RRNN [8]	0.33	0.50	0.66	0.75	1.00	0.29	0.51	0.88	0.99	1.72	0.35	0.47	0.60	0.65	0.88	0.30	0.46	**0.72**	**0.91**	1.36
VRNN (Ours)	0.35	0.66	1.08	1.20	0.89	0.21	0.35	**0.70**	0.83	**1.41**	0.34	0.46	0.61	0.70	1.06	0.27	0.41	0.75	0.98	1.35
ConvSeq2seq [15]	0.29	0.46	0.59	0.60	0.68	0.24	0.44	0.78	0.91	1.53	0.34	0.44	0.48	0.50	0.76	0.31	0.49	0.78	0.96	1.36
BiHMP-GAN [16]	**0.28**	**0.40**	**0.50**	**0.53**	**0.62**	0.26	0.44	0.72	**0.82**	1.51	0.35	0.45	**0.44**	**0.46**	**0.72**	0.31	0.46	0.77	0.92	**1.31**
Q-DCRN (Ours)	0.34	0.58	0.83	0.87	0.70	**0.20**	**0.33**	0.73	0.89	1.51	**0.32**	**0.43**	0.55	0.63	0.80	**0.22**	**0.38**	0.84	1.09	1.48



(a)

(b)

Fig. 8. Qualitative comparisons with DMGNN on high-dynamic motions. The three sequences refer to the ground truth, DMGNN, and Q-DCRN from top to bottom. (a) Walking. (b) Walking dog.

Table IV, our method has closer predictions overall to ground truth than the baselines towards the frequently compared angle distance.

TABLE IV

THE AVERAGE MAE OF CMU MOCAP DATASET

Time (milliseconds)	80	160	320	400	1000
RRNN [8]	0.38	0.62	1.02	1.18	1.67
VRNN (Ours)	0.31	0.52	0.92	1.07	1.53
ConvSeq2seq [15]	0.32	0.52	0.86	0.99	1.55
BiHMP-GAN [16]	0.32	0.50	0.83	**0.94**	1.47
Q-DCRN (Ours)	**0.27**	**0.44**	**0.81**	0.95	**1.44**

We further visualize "running" on CMU MoCap in Fig. 9 to qualitatively evaluate the performance of our method. In the running sequence generated by Q-DCRN, we find that from the 3rd to the 11th predicted frame (highlighted with red boxes), the torso slightly leans forward compared to the ground truth and ConvSeq2seq, which concurs with Table III that a relatively higher numerical error of running between 160ms and 400ms is observed. However, the frames generated by ConvSeq2seq tend to have more averaged poses and result in losing the *terminal swing* phase toward the end of the motion. Compared to ConvSeq2seq, the running poses of our Q-DCRN can clearly show the trend of raising or putting down legs in turn, which ensures a better prediction. This again, shows that a higher angle error rate does not necessarily indicate the generated motion is in bad quality.

To validate the generated poses, we also report the MPJPE results in Table V. In terms of the 3D position, our approach reduces the error rate substantially in most cases. Comparing with higher MAE of "running" and "walking" after 400ms, Q-DCRN performs lower MPJPE on these two actions which echos with Fig. 9 that for such active motions, we can better keep the real dynamics in long-term prediction. We further argue that MPJPE is more discriminative than MAE as observed in "directing traffic" and "soccer", where both methods yield similar quantitative results in angle space but Q-DCRN has much lower error than ConvSeq2seq in position space.

E. Comparisons on the Penn Action Dataset

To make a fair comparison with the methods [8], [17], [52] conducted on the Penn Action dataset, the experiments are

TABLE V

EVALUATIONS ON MPJPE OF CMU MOCAP DATASET

	Basketball					Basketball Signal					Directing Traffic					Jumping				
Time (milliseconds)	80	160	320	560	1000	80	160	320	560	1000	80	160	320	560	1000	80	160	320	560	1000
ConvSeq2seq [15]	22.1	41.0	78.4	130.9	172.8	15.6	30.6	60.0	99.7	129.4	63.1	112.6	222.8	263.4	262.0	**27.3**	**55.4**	111.7	171.9	228.4
Q-DCRN (Ours)	**21.2**	**37.1**	**70.3**	**117.3**	**147.9**	**3.5**	**8.1**	**18.0**	**31.5**	**61.8**	**16.8**	**24.9**	**49.9**	**97.8**	**170.5**	28.1	57.8	**110.6**	**144.8**	**166.6**
	Running					Soccer					Walking					Wash Window				
Time (milliseconds)	80	160	320	560	1000	80	160	320	560	1000	80	160	320	560	1000	80	160	320	560	1000
ConvSeq2seq [15]	**23.9**	**28.7**	**37.8**	55.7	70.8	37.0	76.5	183.1	178.4	203.7	**14.5**	28.7	54.9	71.9	97.2	21.6	44.5	84.5	113.0	144.8
Q-DCRN (Ours)	27.3	38.2	55.7	**48.4**	**62.4**	**18.4**	**40.0**	**77.7**	**113.8**	**152.9**	15.8	**26.2**	**51.2**	**68.9**	**77.1**	**10.2**	**21.6**	**49.8**	**78.7**	**109.1**

TABLE VI

EVALUATIONS ON PCK@0.05 (%) OF PENN ACTION DATASET

	Predicted frame															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
RRNN [8]	82.4	68.3	58.5	50.9	44.7	40.0	36.4	33.4	31.3	29.5	28.3	27.3	26.4	25.7	25.0	24.5
3D-PFNet [52]	79.2	60.0	49.0	43.9	41.5	40.3	39.8	39.7	40.1	40.5	41.1	41.6	42.3	42.9	43.2	43.3
TP-RNN [17]	84.5	72.0	64.8	60.3	57.2	55.0	53.4	52.1	50.9	50.0	49.3	48.7	48.3	47.9	47.6	47.3
Q-DCRN	**85.2**	**72.6**	**65.1**	**60.5**	**57.5**	**55.4**	**53.9**	**52.6**	**51.5**	**50.6**	**50.0**	**49.4**	**49.1**	**48.8**	**48.6**	**48.4**

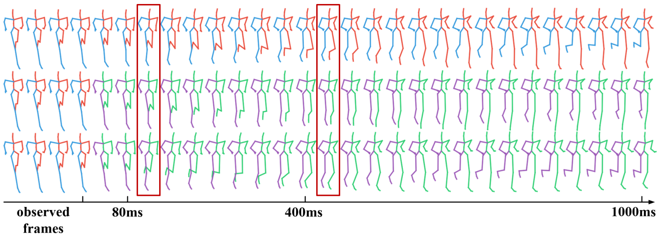


Fig. 9. Visualization of "running" on the CMU MoCap dataset. The three sequences refer to the ground truth, ConvSeq2seq, and Q-DCRN from top to bottom. ConvSeq2seq traps in mean poses with movement decay on legs while we still keep the active running mode as the ground truth.

evaluated on PCK at $\rho = 0.05$ (PCK@0.05), and the results are provided in Table VI. We achieve the state of the art at all predicted steps by a large margin with RRNN and 3D-PFNet, and superior to TP-RNN notably at longer prediction. The other three methods fail to preserve the prediction accuracy in the long term especially for RRNN, which suffers a drastic drop along with the predicted frames. This is because when the observed motion prefix is short (one frame for Penn Action dataset), the residual connection in RRNN may cause a large error accumulation with less information directing the decoder. 3D-PFNet is constructed under a plain RNN architecture, and TP-RNN improves it by designing multi-scale hierarchical RNNs to better learn the motion dynamics. However, these methods do not consider the latent relationship between the joints. For our case, we outperform them with the help of spatial modeling using graph convolutions. We also sustain the ground truth with a gentle accuracy decay by incorporating adversarial training to keep the long-term performance. The success on Penn Action dataset also highlights the generality of our proposed prediction method across different types of data modalities.

F. Ablation Studies

1) Network Structure: We then evaluate the effectiveness of our bi-directional spatial-temporal configurations. Starting from VRNN, we gradually add the key components back to Q-DCRN and test the performance at each step. We show the results of single direction in both space (divergence only) and time (forward only) which is denoted as DCRN, the performance of BiS-DCRN with bi-directional convolutions in space, BiS-DCRN with the forward discriminator only (denoted as BiS-DCRN *fwd dis.*) or with the backward discriminator only (denoted as BiS-DCRN *bwd dis.*), together with Q-DCRN by including the bi-directional discriminator. The error comparisons under MAE and MPJPE metrics are shown in Table VII of the average performance on Human3.6M. We found that compared with VRNN, there is a significant improvement of DCRN in both MAE and MPJPE, which yields that replacing fully connectivity with graph convolution gives potentials in identifying inner spatial dependencies during temporal propagation, thus outperforming individual RNN-based model.

TABLE VII

PREDICTION ERROR COMPARISONS UNDER DIFFERENT SPATIAL AND TEMPORAL CONFIGURATIONS

	Time (milliseconds)	80	160	320	400	1000
MAE	VRNN	0.38	0.68	1.02	1.14	1.73
	DCRN	0.31	0.59	0.95	1.07	1.66
	BiS-DCRN	0.31	0.58	0.94	1.06	1.66
	BiS-DCRN *fwd dis.*	0.32	0.58	0.92	1.04	1.64
	BiS-DCRN *bwd dis.*	**0.30**	0.57	0.92	1.06	1.65
	Q-DCRN	0.31	**0.57**	**0.90**	**1.02**	**1.60**
MPJPE	VRNN	22.6	43.0	77.8	91.5	145.9
	DCRN	19.1	39.0	72.0	84.4	131.8
	BiS-DCRN	18.9	37.2	68.2	81.0	129.9
	BiS-DCRN *fwd dis.*	19.2	37.5	67.9	80.1	128.9
	BiS-DCRN *bwd dis.*	**18.5**	37.0	70.7	83.2	131.2
	Q-DCRN	18.7	**36.9**	**67.9**	**79.7**	**127.3**

In addition, we observe that BiS-DCRN has comparable MAE but lower MPJPE compared with DCRN. This shows that BiS-DCRN is overall superior to DCRN by including the convergence process in diffusion convolutions. This is because the speed of a child node will more or less influence its root node(s) in terms of the correlations on the graph G. With both divergence and convergence convolution processes, the joints are aware of the dynamics of its neighbour joints from upstream and downstream random walks on the graph for a more perceptive and accurate spatial prediction.

Lastly, the improvement from BiS-DCRN to Q-DCRN shows the effectiveness of bi-directional temporal modeling. From both metrics, Q-DCRN demonstrates a better prediction especially in generating longer motions (320-1000ms), which

TABLE VIII

PREDICTION ERROR COMPARISONS (MPJPE) UNDER DIFFERENT GRAPH STRUCTURES

A	Type	80	160	320	400	1000
fixed	undirected	20.1	42.1	77.3	90.1	134.4
adaptive	undirected	19.8	36.9	68.6	80.6	129.1
adaptive	directed	**18.7**	**36.9**	**67.9**	**79.7**	**127.3**

illustrates the benefits of using adversarial training to reduce the error accumulation by amending the forward and backward dynamics. From the result of BiS-DCRN *fwd dis.*, we find that the single directional discriminator improves the prediction in the long term but may corrupt the short term compared with BiS-DCRN. This is because the forward discriminator may forget the information from the past dynamics, which makes it hard to revise the beginning velocities. On the contrary, we observe a better prediction at 80ms for BiS-DCRN *bwd dis.* but a large error in the long term, since the backward discriminator focuses more on the initial dynamics while losing the long-term information. By balancing the bi-directional discriminator, the final Q-DCRN is able to improve both short and long-term predictions compared with BiS-DCRN. We also observe that at the beginning of prediction (80ms), BiS-DCRN and Q-DCRN present similar angle results in MAE, while they show differently in position space from MPJPE. This further confirms our assumption that MPJPE is a more discriminative tool in measuring the generated movements compared with MAE.

2) Graph Structure: We also evaluate Q-DCRN under different adjacency matrices and graph types. The result in Table VIII shows that our system performs the best under the adaptive, directed graph structure.

a) Fixed vs. Adaptive: We compare the performance of the unweighted graph structure defined by the fixed adjacency matrix A, and our weighted graph structure defined by the adaptive A in the 1st and the 2nd rows of Table VIII. The fixed A is represented by a binary matrix, where the joints connected by bones are fixed at 1 with the others at 0. Note that since the joints under fixed A is an undirected graph, i.e., bone connections are undirected, we do not have the option for a directed graph under fixed A. Therefore, the effectiveness of the adaptive A is evaluated on the baseline of the undirected graph. From the results, we observe a significant improvement when A is adaptive. This is because the original fixed A restricts the information transitions only within the edges of bone connections, which neglects the useful implicit connections. Furthermore, all the connections take the same importance in fixed A, which contradicts the fact that different connections may contribute differently to the motion, such as the connection of knee and foot is more informative than the connection of spine and neck in a “running” action. By softening these two conditions on A, all nodes are flexibly connected with the trainable edge weights, which presents better performance than the fixed A.

b) Undirected vs. Directed : We verify the effectiveness of our directed graph by comparing it with its undirected counterpart, where the adjacency matrix A and its transpose are reduced to one symmetric matrix denoting the equivalent information transfer between a pair of nodes. The comparison results in the 2nd and the 3rd rows of Table VIII show that using a directed graph structure is more beneficial for a precise prediction.

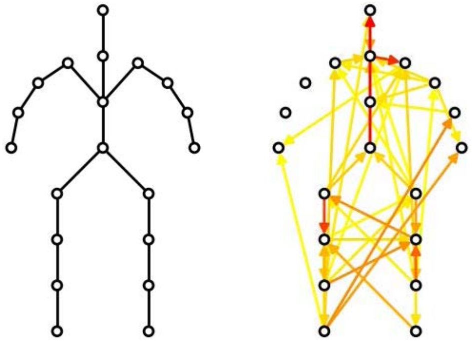


Fig. 10. Visualization of (a) the undirected joint connections of the fixed adjacency matrix, and (b) the top 40 joint connections of the learned adjacency matrix on Human3.6M. The arrow denotes the direction of the connection. The weight of the connection is visualized from a red to yellow scale, with the red color representing larger weights.

c) Visualization : We further visualize the adaptive A to show the learned spatial correlations comparing with the fixed undirected connections in Fig. 10. Since the adaptive A is not constrained to be positive, we select the top 40 absolute values of A representing the most significant connections in the information delivery between joints. The position of the value in A indicates the direction of the connection, e.g., A_{pq} is pointing from joint p to joint q. From Fig. 10(b), we observe that many selected edges between joints are not connected by bones, which highlights the importance of the implicit connections. We also find many connections between legs are selected. This makes sense as many of the learned motions are walking-related, where the movements of legs are dominant. Another interesting observation is that the connections are not necessarily symmetric for the left and right body—more edges are associated with the right arm than the left arm, which may be due to some natural habits that most actors are likely right-handed.

3) Loss Functions: First, we test the effectiveness of the proposed velocity-pose reconstruction loss. We compare it with directly calculating the mean squared error based on the velocity between ground truth and prediction. Second, the ablation for our velocity-based adversarial loss is conducted on training the discriminator with the generated pose rather than velocity, where the predictor and the discriminator do not share the weights and structures since they are modeling different motion features.

The comparison results are shown in Table IX. In the top row when only the velocity is considered in the optimization, the prediction error is fast accumulated as the system cannot guarantee the quality of the generated poses. In the middle row when only the static pose is considered, no penalization is added on the motion dynamics, which leads to a biased prediction with large error rates in both the short and long term. In the bottom row, the system improves the prediction with the optimal solution by synchronously considering the static pose from the reconstruction loss and the velocity dynamics from the adversarial loss.

TABLE IX

PREDICTION ERROR COMPARISONS (MPJPE) UNDER DIFFERENT LOSS FUNCTIONS

$\mathcal{L}_{recons}$	$\mathcal{L}_{adv}$	80	160	320	400	1000
velocity-based	velocity-based	18.7	37.2	69.7	81.8	130.1
velocity-pose	pose-based	19.7	39.3	68.2	80.7	129.5
velocity-pose	velocity-based	**18.7**	**36.9**	**67.9**	**79.7**	**127.3**

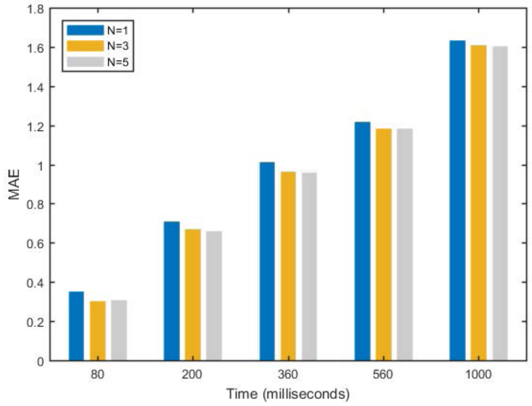


Fig. 11. Prediction error comparisons under different maximum diffusion steps, N, in the dual random walks.

G. Parameters and Model Efficiency

As mentioned in Section III-A, the value of maximum diffusion step N controls the approximation of the global graph distributions that the diffusion process would converge to. A biased N may lead to either inadequate or redundant graph description. From Fig. 11, we found that using only one diffusion step will lose information spread along multiple nodes, causing a large error rate. A maximum of five steps gives a plausible result with the price of more filters to deduce higher-order diffusions. Hence, we employ three diffusion steps which can sufficiently and efficiently describe the spatial dependency.

We also compare the number of parameters and the prediction time for each method in Table X. Q-DCRN uses the least parameters with relatively lower time cost, especially among the graph-based methods. We adopt graph embedding that is shared among all nodes under a recurrent network, which reduces the proportion of the learnable weights in contrast to [8], [15]. When comparing with [28] and [29], Q-DCRN avoids artificially crafting deep graph convolutions to extract features of different receptive fields, which yields a more efficient model.

H. Discussion

In our adaptive graph connectivity, A can be regarded as an attention map that represents the significance of joint pairs, which can be even extended to multi-head attentions [58] with different attention combinations to further improve the fitting ability of our model. It is also beneficial to adopt other attention mechanisms, such as considering temporal attention [59] to strengthen the network with the important memories from the past and future dynamics, if under bi-directional settings.

TABLE X

TRAINING PARAMETERS AND PREDICTION TIME (25 FRAMES) USED IN EACH METHOD

	Method	# parameters	Testing time (ms)
Non-graph	RRNN [8]	$\sim 3.4m$	1.7
	ConvSeq2seq [15]	$\sim 16.6m$	1.9
Graph-based	DMGNN [28]	$\sim 62.6m$	11.6
	LDR [29]	$\sim 2.1m$	2.4
	Q-DCRN	$\sim 0.2m$	2.2

One of the main challenges for RNN to process long sequences is losing long-term dependency. In this work, we adopt adversarial training to enhance its prediction results. Many other techniques that are orthogonal to our work, such as adaptively skipping the state update to reduce sequential operations [60], hierarchically integrating the temporal information several steps away [61], or efficiently reusing gate matrices with sparse representations [62] in RNN can also be well employed to further boost the performance in terms of prediction accuracy and computational cost.

We also find that the proposed Q-DCRN is effective in predicting the possible movements tracked from 2D videos, which sheds light on two potential future works: One is that we can realize 3D pose prediction from RGB videos [63] by integrating our framework with any 2D to 3D recovery algorithms; Another is directly predicting image outputs without the middle step of extracting joint features [64]. As we do not consider any specific bone constraints or body hierarchy, our proposed framework is not limited to the human skeleton but also compatible with any forms of data under graph structure, but we do rely on other designs like CNN to learn the representative local features from images in the first hand.

Since current motion predictions heavily depend on the precision of the detected pose, which is technically hard to achieve especially in crowded scenes [57], [65] or under a depth camera [34], [66], how to develop a robust system under noisy supervision may also benefit the prediction community. There are already some successful attempts, such as reconstructing the motion history for denoising [11] or estimating distributions with multiple future possibilities [67]. Pairing these methods with Q-DCRN to reduce the impact of noisy input will be another potential direction to explore.

V. CONCLUSION

We propose a quadruple diffusion convolutional recurrent network to preserve motion trend for human dynamic prediction. We encode spatial structure as an adaptive diffusion graph with bi-directional random walks in multiple spatial steps, and perform graph convolution on the recurrent seq2seq network to decode temporal dependencies. A bi-directional temporal predictor together with a bi-discriminator is designed in an efficient weight-sharing manner to fit and revise the short and long-term motion trends. The network is constructed directly on the velocity with a reconstruction loss on poses, which has proved to be more powerful at reducing discontinuity at early prediction than residual connections in RNN-based architecture. Experimental results on both angular and positional metrics suggest that the proposed Q-DCRN is able to preserve the motion trend with lower prediction errors to generate realistic moving dynamics.

Acknowledgment

The authors would like to thank Daniel Organisciak for polishing the paper.

References

[1] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[2] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016.

[3] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[4] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio–temporal visual attention on skeleton image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2405–2415, Aug. 2019.

[5] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu, and Y. Zhang, "Skeleton-based action recognition with gated convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3247–3257, Nov. 2019.

[6] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient nonlinear Markov models for human motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1314–1321.

[7] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.

[8] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2891–2900.

[9] Y. Zhou, Z. Li, S. Xiao, C. He, Z. Huang, and H. Li, "Auto-conditioned recurrent networks for extended complex human motion synthesis," in *Proc. Int. Conf. Learn. Represent.*, Jun. 2018, pp. 1–13.

[10] A. Gopalakrishnan, A. Mali, D. Kifer, L. Giles, and A. G. Ororbia, "A neural temporal model for human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12116–12125.

[11] H. Wang, E. S. L. Ho, H. P. H. Shum, and Z. Zhu, "Spatio-temporal manifold learning for human motions via long-horizon modeling," *IEEE Trans. Vis. Comput. Graphics*, early access, Aug. 22, 2019, doi: 10.1109/TVCG.2019.2936810.

[12] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6158–6166.

[13] X. Guo and J. Choi, "Human motion prediction via learning local structure representations and temporal dependencies," in *Proc. AAAI*, vol. 33, 2019, pp. 2580–2587.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[15] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5226–5234.

[16] J. N. Kundu, M. Gor, and R. V. Babu, "BIHMP-GAN: Bidirectional 3D human motion prediction GAN," in *Proc. AAAI*, vol. 33, 2019, pp. 8553–8560.

[17] H.-K. Chiu, E. Adeli, B. Wang, D.-A. Huang, and J. C. Niebles, "Action-agnostic human pose forecasting," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1423–1432.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[19] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[20] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, Dec. 2013.

[21] E. Aksan, M. Kaufmann, and O. Hilliges, "Structured prediction helps 3D human motion modelling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7144–7153.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[23] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Process.*, 2018, pp. 362–373.

[24] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1227–1236.

[25] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[26] R. Li, S. Wang, F. Zhu, and J. Huang, "Adaptive graph convolutional neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3546–3553.

[27] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12026–12035.

[28] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian, "Dynamic multiscale graph neural networks for 3D skeleton based human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 214–223.

[29] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3D human motion prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6519–6527.

[30] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4346–4354.

[31] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5308–5317.

[32] M. Dong and C. Xu, "On retrospecting human dynamics with attention," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 708–714.

[33] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.

[34] L. Chen, J. Lu, Z. Song, and J. Zhou, "Recurrent semantic preserving generation for action prediction," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Feb. 19, 2020, doi: 10.1109/TCSVT.2020.2975065.

[35] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. Moura, "Adversarial geometry-aware human motion prediction," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 786–803.

[36] D. Pavllo, C. Feichtenhofer, M. Auli, and D. Grangier, "Modeling human motion with quaternion-based neural networks," *Int. J. Comput. Vis.*, vol. 128, no. 4, pp. 855–872, Apr. 2020.

[37] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *Proc. SIGGRAPH ASIA Tech. Briefs SA*, 2015, pp. 1–4.

[38] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[39] CMU. (2003). *Graphics Lab Motion Capture Database*. [Online]. Available: http://mocap.cs.cmu.edu/

[40] W. Zhang, M. Zhu, and K. G. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2248–2255.

[41] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7912–7921.

[42] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1993–2001.

[43] J. Klicpera, S. Weißenberger, and S. Günnemann, "Diffusion improves graph learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 13354–13366.

[44] X. Zhang, Y. Li, D. Shen, and L. Carin, "Diffusion maps for textual network embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7587–7597.

[45] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1506–1515.

[46] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 1–9.

[47] S.-H. Teng, "Scalable algorithms for data and network analysis," *Found. Trends Theor. Comput. Sci.*, vol. 12, nos. 1–2, pp. 1–274, 2016.

[48] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[49] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[50] V. M. Zatsiorsky and V. M. Zaciorskij, *Kinetics of Human Motion*. Champaign, IL, USA: Human Kinetics, 2002.

[51] B. Wang, E. Adeli, H.-K. Chiu, D.-A. Huang, and J. C. Niebles, "Imitation learning for human pose prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7124–7133.

[52] Y.-W. Chao, J. Yang, B. Price, S. Cohen, and J. Deng, "Forecasting human dynamics from static images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 548–556.

[53] R. Parent, *Computer Animation: Algorithms and Techniques*. Boston, MA, USA: Newnes, 2012.

[54] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[55] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4903–4911.

[56] L. Ge *et al.*, "3D hand shape and pose estimation from a single RGB image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10833–10842.

[57] K. Mangalam, E. Adeli, K.-H. Lee, A. Gaidon, and J. C. Niebles, "Disentangling human dynamics for pedestrian locomotion forecasting with noisy supervision," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2784–2793.

[58] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[59] Z. Ji, K. Xiong, Y. Pang, and X. Li, "Video summarization with attention-based encoder–decoder networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1709–1717, Jun. 2020.

[60] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip RNN: Learning to skip state updates in recurrent neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[61] H. Fan, L. Zhu, and Y. Yang, "Cubic LSTMs for video prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8263–8270.

[62] A. Kusupati, M. Singh, K. Bhatia, A. Kumar, P. Jain, and M. Varma, "Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9017–9028.

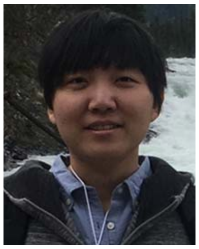
[63] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7035–7043.

[64] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, "Predrnn: Recurrent neural networks for predictive learning using spatiotemporal LSTMs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 879–888.

[65] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 961–971.

[66] J. Butepage, H. Kjellstrom, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4563–4570.

[67] A. Hernandez, J. Gall, and F. Moreno, "Human motion prediction via spatio-temporal inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7134–7143.



Qianhui Men received the B.S. degree from the Dalian University of Technology, and the M.S. degree from the National University of Singapore. She is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong. Her research interests include human motion analysis, computer vision, and machine learning.

Edmond S. L. Ho received the B.Sc. (Hons.) degree in computer science from the Hong Kong Baptist University in 2003, the M.Phil. degree in computer science from the City University of Hong Kong in 2006, and the Ph.D. degree from The University of Edinburgh in 2010.

He is currently a Senior Lecturer with the Department of Computer and Information Sciences, Northumbria University, Newcastle, U.K. Prior to this, he was a Research Assistant Professor with the Department of Computer Science, Hong Kong Baptist University. His research interests include computer graphics, computer vision, robotics, motion analysis, and machine learning.

Hubert P. H. Shum (Senior Member, IEEE) received the bachelor's and master's degrees from the City University of Hong Kong, and the Ph.D. degree from The University of Edinburgh. He was the Director of Research/Associate Professor/Senior Lecturer at Northumbria University, a Post-Doctoral Researcher with RIKEN Japan, and a Research Assistant at The City University of Hong Kong. He is currently an Associate Professor in Computer Science with Durham University. He led funded research projects as the Principal Investigator awarded by EPSRC, Ministry of Defence (DASA), and the Royal Society. He has published over 100 research articles in the fields of computer graphics, computer vision, motion analysis, and machine learning.

Howard Leung received the B.Eng. degree in electrical engineering from McGill University, Canada, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. He is supervising the 3D Motion Capture Laboratory at the City University of Hong Kong. His current research interests include 3D human motion analysis and retrieval, intelligent tools for Chinese handwriting education, web-based learning technologies, and brain informatics. He has received Best Paper Awards for his papers in a number of conferences, including the Sixth International Conference on Machine Learning and Cybernetics (ICMLC 2007), 31st Computer Graphics International (CGI 2014), and 36th Computer Graphics International (CGI 2019).