

# Overlooked Poses Actually Make Sense: Distilling Privileged Knowledge for Human Motion Prediction

Xiaoning Sun<sup>1</sup>, Qiongjie Cui<sup>1</sup>, Huaijiang Sun<sup>1</sup>, Bin Li<sup>2</sup>, Weiqing Li<sup>1</sup>, and  
Jianfeng Lu<sup>1</sup>

<sup>1</sup> Nanjing University of Science and Technology, China

{sunxiaoning,cuiqiongjie,sunhuaijiang,liweiqing,lujf}@njust.edu.cn

<sup>2</sup> Tianjin AiForward Science and Technology Co., Ltd., China libin@aiforward.com

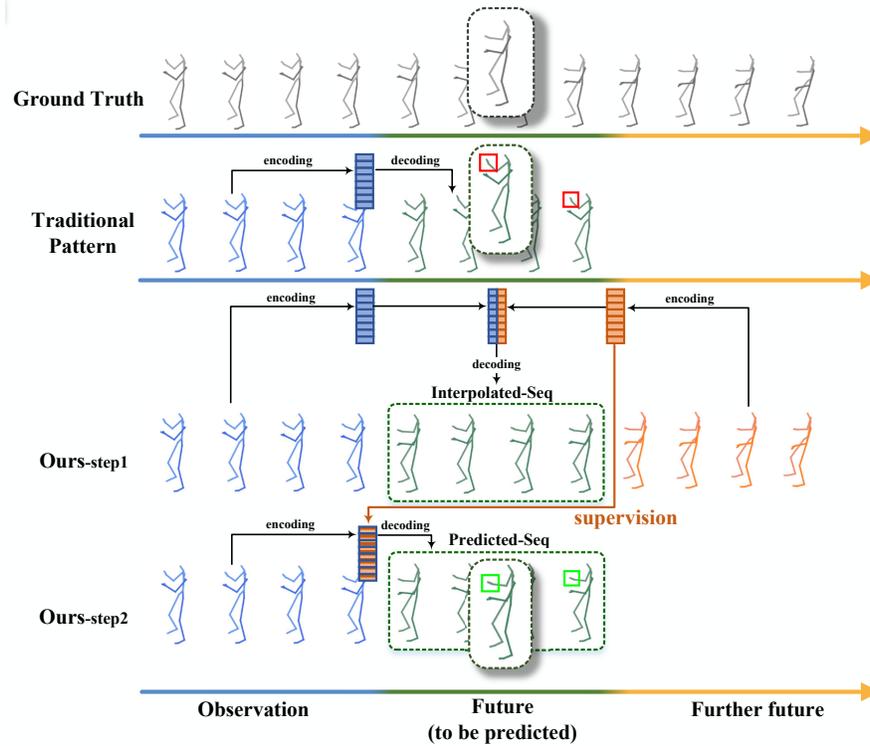
**Abstract.** Previous works on human motion prediction follow the pattern of building a mapping relation between the sequence observed and the one to be predicted. However, due to the inherent complexity of multivariate time series data, it still remains a challenge to find the extrapolation relation between motion sequences. In this paper, we present a new prediction pattern, which introduces previously overlooked human poses, to implement the prediction task from the view of interpolation. These poses exist *after* the predicted sequence, and form the privileged sequence. To be specific, we first propose an InTerPolation learning Network (ITP-Network) that encodes both the observed sequence and the privileged sequence to interpolate the in-between predicted sequence, wherein the embedded Privileged-sequence-Encoder (Priv-Encoder) learns the privileged knowledge (PK) simultaneously. Then, we propose a Final Prediction Network (FP-Network) for which the privileged sequence is not observable, but is equipped with a novel PK-Simulator that distills PK learned from the previous network. This simulator takes as input the observed sequence, but approximates the behavior of Priv-Encoder, enabling FP-Network to imitate the interpolation process. Extensive experimental results demonstrate that our prediction pattern achieves state-of-the-art performance on benchmarked H3.6M, CMU-Mocap and 3DPW datasets in both short-term and long-term predictions.

**Keywords:** Human motion prediction, Privileged knowledge

## 1 Introduction

Human motion prediction, aimed at forecasting future poses according to the historical motion sequence, has been widely applied in the field of autonomous driving [38], human-machine interaction [21] and action detection [13]. To make the best of human body structure information, recent methods employ graph convolutional networks (GCNs) [34,7,8,43] to model spatial dependencies or topological relations of body joints.

Classical sequential networks such as Recurrent Neural Networks (RNNs) [19,12,5] and Long Short-Term Memory (LSTM) [11,6,35] have been used to



**Fig. 1.** Comparison between the traditional human motion prediction pattern and ours. Previous works predict future motions on historical ones by direct extrapolation, which may bring about outcomes opposite to targets (see red boxes in the figure). However, our pattern a) introduces poses from further future to learn a PK representation by interpolating the in-between poses during the first step, and b) extrapolates (i.e. predicts) the sequence given historical poses only, but with PK to be distilled that offers supervision on the process of encoding and decoding during the second step.

tackle certain sequence prediction frame-by-frame, yet suffer from error accumulation and convergence to the static mean pose [22,14]. Temporal Convolutional Networks (TCNs) [2,50] and the Discrete Cosine Transform (DCT) [34,31,32] method have also proved their ability to encode temporal information, and to be coordinated with GCNs to better predict human motions in terms of both spatial and temporal domain [34,7,31,32]. All the methods mentioned above follow the same prediction pattern, which is building a mapping relation between the sequence observed and the one to be predicted. However, it still remains tough to directly extrapolate a multivariate time series based on the historical sequence, and unexpected results or unreasonable poses may arise.

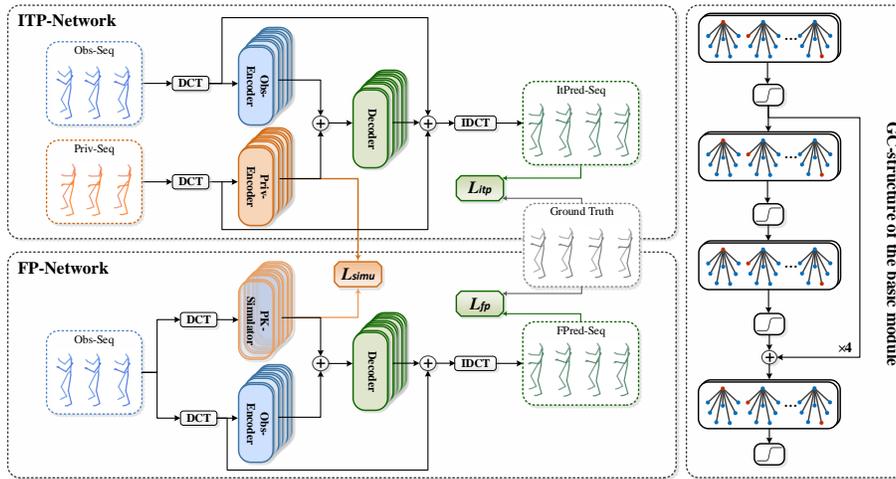
Inspired by the fact that, generally, **sequence interpolation is easier to operate and often yields an overall better result than extrapolation**, we introduce the overlooked poses which exist after the predicted sequence, in

the hope of constructing a prediction pattern that **shares similar spirit with interpolation**. As these poses are only observable for interpolation and should not appear in the final prediction, we regard the information provided by them as privileged information. The concept of learning with privileged information is first presented by Vapnik *et al.* [46] in which the additional information of training samples can be used during training but discarded while testing. As is shown in the top half part of our prediction pattern in Figure 1, when the privileged sequence is introduced, the poses to be predicted finally are first interpolated as we hope. With this preliminary step done, another problem that arises is how to let the prediction step acquire the privileged information, since the raw privileged sequence would not exist then.

Recently, it has been proved that knowledge distillation is able to transfer knowledge between different models, and has been applied to diverse fields such as image dehazing [17], object detection [48], and online action detection [53]. This advancement enlightens us to tackle the question above by designing a novel prediction network (see the bottom half of our pattern in Figure 1), which distills privileged knowledge (PK) from the aforementioned interpolation period. The distillation process exactly reflects the effect of PK supervision, enabling the model to be armed with a certain amount of PK gradually without taking the privileged sequence as input. In this way, this model is able to **keep the merits of interpolation**, while **predicting in the standard manner of extrapolation** finally.

In this paper, we name the two steps of networks proposed above as InTerpolation learning Network (ITP-Network) and Final Prediction Network (FP-Network), respectively. As our networks are created on the basis of GCN structure, the entire model is named as PK-GCN (see Figure 2). This model embodies a new pattern of human motion prediction, instead of the traditional one that directly finds an extrapolation relation between the observed sequence and the one to be predicted. Specifically, the Privileged-sequence-Encoder (Priv-Encoder) in ITP learns a PK representation of the privileged sequence during interpolation, then the novel PK-Simulator embedded in FP implements the PK distillation process by approximating the PK representation, with only the observed sequence as input. Meanwhile, a PK simulation loss function is presented to measure this approximation.

The idea of introducing future information to provide additional supervision has been practiced in prediction-related tasks, such as using predicted future frames to assist online action detection [49], or encapsulating future information in observation representations with Jaccard similarity measures [10] for action anticipation, among which the information obtained from *future* are still limited to the traditional observed-predicted research window. However, 3D human motion data is more elaborated with abundant yet subtle spatio-temporal information, and focusing on this two-section research window is not enough. Compared with existing works, our novelty lies in that we extend it to the observed-predicted-privileged window, in the seek of additional yet tailored information to provide supervision. To the best of our knowledge, this is the first work to in-



**Fig. 2.** The backbone of our PK-GCN. ITP is composed of an Observed-sequence-Encoder (Obs-Encoder), a Privileged-sequence-Encoder (Priv-Encoder) and a decoder to interpolate the in-between sequence. FP is composed of an Obs-Encoder, a PK-Simulator and a decoder. These basic modules share the same architecture of graph convolutional layers (GC-Layers) and residual structures shown in the right sub-figure. During the interpolation period, a PK representation is learned by Priv-Encoder, so that the PK-Simulator is able to distill PK by approximating this representation (we measure their similarity by PK simulation loss  $L_{simu}$ ). Therefore, FP could take as input only the observed sequence but predict motions with PK assisted.

volve poses existing after the predicted sequence to learn PK for human motion prediction.

In summary, our contributions are as follows: (i) ITP-Network: introducing the privileged sequence to learn PK during interpolation period; (ii) FP-Network: with the novel PK-Simulator distilling PK, to predict in the extrapolation manner while keeping the merits of interpolation; (iii) Entire model of PK-GCN: SOTA performance on benchmarked H3.6M, CMU-Mocap and 3DPW datasets.

## 2 Related Work

### 2.1 Human Motion Prediction

Since human motion data is composed of a time series of human poses, many works on human motion prediction rely on sequential networks such as RNNs [19,12,5], LSTM [11,35,44] or GRU [35,40,9] to extract temporal information, and has achieved success. Notably, there occurs a problem that the first predicted frame is discontinuous with the last observed frame, so Martinez *et al.* [35] propose a residual single-layer GRU model which moves from the usual LSTM architectures [25] and predicts velocities rather than poses. Meanwhile,

a zero-velocity baseline is proposed in this work to prove that the repetition of the last observed pose even outperform prediction performances in [11,19]. However, the crucial defect of RNNs, error accumulation, may lead to failure cases of convergence to unexpected mean poses. Other classical architectures such as CNN-based [22,26] make it possible to predict in a sequence-to-sequence manner rather than frame-by-frame, while GAN-based [14,15] are aimed at generating more realistic motions.

Recently, GCNs gain huge popularity to model spatial dependencies on predict human motions [34,31,7,24,8,43,28,30,54]. Mao *et al.* [34] first introduce a generic graph instead of the previous skeletal kinematic tree [3], with DCT encoding the short-term history in frequency domain. To depict the topological relations of body joints, Cui *et al.* [7] present a dynamic GCN, which learns both the natural connectivity of joint pairs and the implicit links of geometrically separated joints. Different multi-scale GNN [23]/GCN [8] models are developed to further explore the features of human motions. Sofianos *et al.* [43] realizes cross-talk over space and time within one graph and factorizes it into separable matrices for better understanding on both features. Meanwhile, stochastic motion prediction [1,52,32,33,29] gradually draw attention, which generate diverse yet plausible motion sequences given only one observed.

All the models mentioned above follow the traditional prediction pattern, limiting their research window within the observed-predicted sequence. Differently, our work goes further by adding a privileged window containing overlooked poses, so that a PK representation can be learned and distilled through our two-step model, to serve as a supplement of supervision information to assist prediction tasks.

## 2.2 Knowledge Distillation

Early knowledge distillation is utilized for deep model compression [16], wherein the knowledge from the larger teacher network helps the smaller student network to learn better [36]. The teacher provides the soft target for the student to imitate the output logits and intermediate features [45,42,4,51]. Subsequently, it has been proved that knowledge distillation is able to transfer knowledge between different models, and has its applications in image classification [41,45], action detection [53], semantic segmentation [27], trajectory prediction [37] and stochastic motion prediction [29]. However, the teachers in these models provides additional information about training samples either with no privileged information involved, or with it involved but irrelevant to distill knowledge from future further away. In addition, our design of distillation process ensures that a) PK would be distilled into FP as much as possible, and b) only the observed sequence would be taken as input. We embed the PK-Simulator in FP to be fed with the observed sequence but approximate the PK representation learned by Priv-Encoder, and present a PK simulation loss  $L_{\text{simu}}$  (also denoted as PKSL) to measure this approximation.

### 3 Proposed Method

We present a new formulation for human motion prediction. Different from existing works, our notation involves three research windows (i.e. the observed-predicted-privileged sequence), with  $N$  poses  $\mathbf{X}_{1:N} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N]$  given,  $T$  poses  $\mathbf{X}_{N+1:N+T}$  to predict, and particularly another  $P$  poses after the predicted sequence  $\mathbf{X}_{N+T+1:N+T+P}$  as the privileged sequence, where  $\mathbf{x}_i \in \mathbb{R}^K$ , with  $K$  parameters depicting each pose based on joint coordinates or angles. The entire architecture of PK-GCN is shown in Figure 2. To be specific, ITP-Network encodes both the observed sequence and the privileged sequence to interpolate the predicted sequence, during which PK is learned and to be distilled to the next step, enabling FP-Network to predict in the standard manner of extrapolation finally, but with PK assisted.

#### 3.1 Spatio-Temporal Encoding

To encode both the spatial and temporal information of human motions, we employ the remarkable encoding method from [34]. We treat each human skeleton as a fully-connected graph of  $K$  nodes, and use an adjacency matrix  $A \in \mathbb{R}^{K \times K}$  to depict its edges. Formally, we define  $H^{(l)} \in \mathbb{R}^{K \times F^{(l)}}$  to be fed into the  $l$ th layer of GCN (i.e. the  $l$ th GC-Layer),  $W^{(l)} \in \mathbb{R}^{F^{(l)} \times F^{(l+1)}}$  as weight matrix, and output (also the input of next GC-Layer) as:

$$H^{(l+1)} = \sigma(A^{(l)}H^{(l)}W^{(l)}), \quad (1)$$

where  $\sigma(\cdot)$  is an activation function. Particularly, a DCT-based representation is employed to extract temporal information, which transforms the observed sequence into its DCT coefficients  $H^{(1)} \in \mathbb{R}^{K \times C}$  before being input into our network, with  $C$  the DCT coefficient number. The output is  $H^{(last)} \in \mathbb{R}^{K \times C}$ , and should also be transformed into its sequence form by IDCT.

#### 3.2 Network Structure

**The InTerPolation learning Network (ITP-Network)**, based on residual GCN architecture, is composed of two encoders to encode the observed sequence (Obs-Encoder) and the privileged sequence (Priv-Encoder), and one decoder to interpolate the in-between sequence (see the top half of Figure 2). In each of the encoders/decoder, we employ 4 residual blocks and 2 separate GC-Layers with one at the beginning and one at the end (see the right part of Figure 2). Each residual block is composed of 2 GC-Layers, and the activation function is set to  $\tanh(\cdot)$ . The outputs of the Obs-Encoder and the Priv-Encoder are integrated by element-wise addition.

It should be emphasized that PK *only* serves as a supplement of supervision information, and it is unreasonable to hold this advantage as an excuse to diminish the original predictability of Obs-Encoder in FP. Additionally, overly complicated PK is not conducive to distill. Therefore, in this PK preparation

step, we intend to reduce the relative weight of Priv-Encoder. However, this variation will lead to degradation on the interpolation performance, so we alternatively set the weight ratio of the outermost residual connection to 0.7:0.3, sum of which is 1 in line with the residual weight in FP.

Since the input of both encoders and the output of the decoder are all DCT coefficients of the corresponding sequences, motivated by zero-velocity baseline [35] and inspired by the padding operation in [34], we propose a new data pre-processing method which replicates the last pose of the observed sequence,  $\mathbf{x}_N$ ,  $T + P$  times (represented as  $\mathbf{ObsX}_{1:N+T+P}$ ), and the first pose of the privileged sequence,  $\mathbf{x}_{N+T+1}$ ,  $N + T$  times (represented as  $\mathbf{PrivX}_{1:N+T+P}$ ), to form residual vectors in frequency domain as the input  $H_{\text{obs}}^{(1)}$ ,  $H_{\text{priv}}^{(1)}$ . After epochs of training to find the interpolation relation  $\mathcal{I}$ :

$$H_{\text{itp}}^{(last)} = \mathcal{I}(H_{\text{obs}}^{(1)}, H_{\text{priv}}^{(1)}), \quad (2)$$

where the interpolation result  $H_{\text{itp}}^{(last)}$  is close to the ground truth, the Priv-Encoder could obtain its most accurate representation of PK, which is aimed to be distilled to the final prediction period.

**The Final Prediction Network (FP-Network)** consists of an Obs-Encoder to encode the observed sequence, a novel PK-Simulator to approximate the representation obtained from Priv-Encoder, and a decoder to extrapolate the predicted sequence. We construct each module the same as in ITP with residual blocks and GC-Layers. We pad the observed sequence to the length of  $N + T + P$  the same as in ITP, and operate DCT transformation to form the input  $H_{\text{obs}}^{(1)}$  to be fed into the network, in the seek of a predictive function  $\mathcal{P}$ :

$$H_{\text{fp}}^{(last)} = \mathcal{P}(H_{\text{obs}}^{(1)}), \quad (3)$$

with distilled PK assisted. Since the privileged sequence does not exist at this moment, the embedded PK-Simulator should undertake the assistant role by distilling PK but without direct use of raw privileged data. A PK simulation loss is introduced in this step, to measure the discrepancy between the output of PK-Simulator and Priv-Encoder (i.e. how well does PK-Simulator imitate the PK representation learned by Priv-Encoder), and notably, the PK representation learned by Priv-Encoder has been well-trained previously and should remain fixed. Moreover, the distillation process is not damage-free. The PK-Simulator is doomed not to obtain entire PK from ITP, but is still capable of reflecting PK to some extent.

### 3.3 Training

Human poses are depicted mainly based on their 3D coordinates (i.e. 3D positions) or angles of body joints. In this section, we follow [34,31,8,43] to employ Mean Per Joint Position Error (MPJPE) for position-based representation, and Mean Angle Error (MAE) [35,34,43] for angle-based representation.

**Loss for ITP-Network.** The MPJPE interpolation loss is expressed as:

$$L_{\text{itp(mp)}} = \frac{1}{(N+T+P)J} \sum_{n=1}^{N+T+P} \sum_{j=1}^J \|\hat{\mathbf{m}}_{j,n} - \mathbf{m}_{j,n}\|_2, \quad (4)$$

where  $\hat{\mathbf{m}}_{j,n} \in \mathbb{R}^3$  is the interpolated  $j$ th joint position of the pose  $\mathbf{x}_n$ , and  $\mathbf{m}_{j,n}$  is the corresponding ground truth one. Note that the length of the interpolated sequence should be consistent with the padded observed/privileged sequences (i.e.  $N+T+P$ ), and is denoted as  $\mathbf{ITPredX}_{1:N+T+P}$ . This design is inspired by [34], which calculates the sum of  $l_2$  over the entire sequence sample, and we extend it to the longer observed-predicted-privileged window to fit our privileged design. Meanwhile, the minimization on  $L_{\text{itp(mp)}}$  ensures that the optimal PK representation is obtained by Priv-Encoder, denoted as  $\mathbf{E}_m$ .

Similarly, the MAE interpolation loss is expressed as:

$$L_{\text{itp(ma)}} = \frac{1}{(N+T+P)K} \sum_{n=1}^{N+T+P} \sum_{k=1}^K \|\hat{\mathbf{a}}_{k,n} - \mathbf{a}_{k,n}\|_1, \quad (5)$$

where  $\hat{\mathbf{a}}_{k,n}$  is the interpolated  $k$ th angle of the pose  $\mathbf{x}_n$ , and  $\mathbf{a}_{k,n}$  is the corresponding ground truth one. We minimize  $L_{\text{itp(ma)}}$  to gain the optimal output of the Priv-Encoder represented as  $\mathbf{E}_a$ .

**Loss for FP-Network.** As FP is aimed at imitating the interpolation process, then the input form should maintain the length of  $N+T+P$  in previous step, and the predicted sequence is also the same, denoted as  $\mathbf{FPredX}_{1:N+T+P}$ . But notably, we only take into account the loss of  $N+T$  poses because the following  $P$  ones are not our prediction target, and the corresponding ground truth (i.e. the privileged sequence) is not observable for this step. To measure the discrepancy between the predicted sequence and its ground truth, we present MPJPE prediction loss

$$L_{\text{fp(mp)}} = \frac{1}{(N+T)J} \sum_{n=1}^{N+T} \sum_{j=1}^J \|\hat{\mathbf{m}}_{j,n} - \mathbf{m}_{j,n}\|_2 \quad (6)$$

for position-based joint representation and MAE prediction loss

$$L_{\text{fp(ma)}} = \frac{1}{(N+T)K} \sum_{n=1}^{N+T} \sum_{k=1}^K \|\hat{\mathbf{a}}_{k,n} - \mathbf{a}_{k,n}\|_1 \quad (7)$$

for angle-based one. Meanwhile, we denote the output of the PK-Simulator as  $\mathbf{S}_m$  for MPJPE and  $\mathbf{S}_a$  for MAE, which approximate  $\mathbf{E}_m$  and  $\mathbf{E}_a$  respectively to implement distillation process. The PK simulation loss  $L_{\text{simu}}$  (i.e. PKSL) we introduce to measure this approximation is formulated as:

$$L_{\text{simu(mp)}} = \|\mathbf{S}_m - \mathbf{E}_m\|_F \quad \text{or} \quad L_{\text{simu(ma)}} = \|\mathbf{S}_a - \mathbf{E}_a\|_F. \quad (8)$$

Therefore, the MPJPE/MAE total loss are:

$$L_{\text{total(mp)}} = L_{\text{fp(mp)}} + \lambda L_{\text{simu(mp)}} \quad \text{or} \quad L_{\text{total(ma)}} = L_{\text{fp(ma)}} + \lambda L_{\text{simu(ma)}}, \quad (9)$$

where  $\lambda$  is the relative weight of the final prediction error and the simulation error, and is set to 0.6.

## 4 Experiments

To evaluate our proposed model, we conduct experiments on benchmarked motion capture datasets, including Human3.6M (H3.6M), CMU-Mocap, and 3DPW datasets. In this section, we first introduce these datasets, the evaluation metrics, and the baselines which we compare with, and then present our results.

### 4.1 Datasets and Evaluation Metrics

**Human3.6M** (H3.6M) [18], depicting seven actors which perform 15 actions (walking, eating, smoking, discussing, etc.), is the most widely used dataset for human motion processing tasks. Each human pose is represented by 32 joints to form a skeleton. In line with previous methods [35,22,14,34], we set the global rotation and translation to zero and down-sample the sequences to 25 frames per second. We train our model on Subject 1 (S1), S6, S7, S8, S9 and test on S5 the same as [35,22,14,34].

**CMU-Mocap**. Based on [22], eight actions are selected for experiments after removing sequences with multiple actors, insufficient poses, or actions with repetitions. The other pre-processing operations are the same as on H3.6M.

**3DPW**. The 3D Pose in the Wild dataset [47] consists of 60 sequences about both indoor and outdoor actions captured by a moving camera at 30 fps. We follow the setting in [34] to evaluate the effectiveness of our models on challenging activities.

**Evaluation Metrics**. We employ Mean Per Joint Position Error (MPJPE) [18] in millimeter [34,8,43] to measure the position discrepancy between the interpolated/predicted joint and the corresponding ground truth, and Mean Angle Error (MAE) [35,34,43] to measure the angle one. As is stated in [34] that MAE may fail due to ambiguous representation, we focus mainly on MPJPE results.

### 4.2 Baselines

Recent methods are involved to evaluate the effectiveness of our model, denoted as Res.sup. [35], Traj-GCN [34], MSR-GCN [8] and STS-GCN [43]. Specifically, Traj-GCN [34] is the first work that introduces an unconstrained graph convolutional network on human motion prediction. MSR-GCN [8] presents a GCN with multi-scale residual structure which abstracts human poses on different levels. STS-GCN [43] realizes cross-talk of space and time within only one graph, and is able to factorize the graph into separable space and time matrices for better

**Table 1.** Comparisons of short-term MPJPE error on H3.6M dataset. The best result is highlighted in bold.

millisecond (ms)	walking				eating				smoking				discussion				directions			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [35]	29.4	50.8	76.0	81.5	16.8	30.6	56.9	68.7	23.0	42.6	70.1	82.7	32.9	61.2	90.9	96.2	35.4	57.3	76.3	87.7
Traj-GCN [34]	12.3	23.0	39.8	46.1	8.4	<b>16.9</b>	33.2	40.7	8.0	16.2	31.9	38.9	12.5	27.4	58.5	71.7	9.0	19.9	43.4	<b>53.7</b>
MSR-GCN [8]	12.2	22.7	38.6	45.2	8.4	17.1	<b>33.0</b>	<b>40.4</b>	8.0	16.3	31.3	38.2	12.0	26.8	57.1	69.7	8.6	<b>19.7</b>	<b>43.3</b>	53.8
STS-GCN [43]	16.7	27.1	43.4	51.0	11.7	18.9	35.2	43.3	12.3	19.8	35.6	42.3	15.3	28.1	58.5	71.9	12.8	23.7	51.1	63.2
PK-GCN	<b>8.9</b>	<b>15.9</b>	<b>28.0</b>	<b>31.6</b>	<b>8.1</b>	17.7	33.6	41.8	<b>7.4</b>	<b>14.3</b>	<b>24.4</b>	<b>29.2</b>	<b>10.3</b>	<b>22.9</b>	<b>42.0</b>	<b>47.2</b>	<b>8.6</b>	23.7	46.5	56.2
millisecond (ms)	greeting				phoning				posing				purchases				sitting			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [35]	34.5	63.4	124.6	142.5	38.0	69.3	115.0	126.7	36.1	69.1	130.5	157.1	36.3	60.3	86.5	95.9	42.6	81.4	134.7	151.8
Traj-GCN [34]	18.7	38.7	77.7	93.4	10.2	21.0	42.5	52.3	13.7	29.9	66.6	84.1	15.6	32.8	65.7	79.3	10.6	<b>21.9</b>	<b>46.3</b>	57.9
MSR-GCN [8]	16.5	37.0	77.3	93.4	<b>10.1</b>	20.7	41.5	51.3	12.8	29.4	67.0	85.0	14.8	32.4	66.1	79.6	10.5	22.0	<b>46.3</b>	57.8
STS-GCN [43]	18.7	34.9	71.6	86.4	13.7	22.4	43.6	53.8	16.4	30.4	67.6	84.7	19.1	35.8	70.2	83.1	15.2	25.1	49.8	60.8
PK-GCN	<b>13.3</b>	<b>27.2</b>	<b>67.3</b>	<b>83.1</b>	11.4	<b>20.2</b>	<b>37.7</b>	<b>43.2</b>	<b>9.1</b>	<b>23.6</b>	<b>65.8</b>	<b>81.2</b>	<b>15.2</b>	<b>31.4</b>	<b>57.9</b>	<b>68.0</b>	<b>10.1</b>	24.6	47.8	<b>57.3</b>
millisecond (ms)	sittingdown				takingphoto				waiting				walkingdog				walkingtogether			
	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Res. sup [35]	47.3	86.0	145.8	168.9	26.1	47.6	81.4	94.7	30.6	57.8	106.2	121.5	64.2	102.1	141.1	164.4	26.8	50.1	80.2	92.2
Traj-GCN [34]	16.1	31.1	61.5	75.5	9.9	20.9	45.0	56.6	11.4	24.0	50.1	61.5	23.4	46.2	83.5	96.0	10.5	21.0	38.5	45.2
MSR-GCN [8]	16.1	31.6	62.5	76.8	9.9	21.0	44.6	56.3	10.7	23.1	<b>48.3</b>	<b>59.2</b>	<b>20.7</b>	42.9	<b>80.4</b>	<b>93.3</b>	10.6	20.9	37.4	<b>43.9</b>
STS-GCN [43]	22.1	37.2	66.5	79.4	14.5	23.4	47.8	59.4	14.5	24.8	50.4	62.0	26.5	46.1	83.6	97.3	14.3	23.9	41.3	49.1
PK-GCN	<b>11.5</b>	<b>27.5</b>	<b>56.8</b>	<b>67.3</b>	<b>7.6</b>	<b>16.1</b>	<b>39.7</b>	<b>51.3</b>	<b>9.5</b>	<b>23.0</b>	55.9	63.6	21.3	<b>42.4</b>	83.7	95.1	<b>9.4</b>	<b>19.3</b>	<b>36.3</b>	44.8

**Table 2.** Comparisons of long-term MPJPE error on some actions in H3.6M dataset. The best result is highlighted in bold.

millisecond (ms)	walking		eating		smoking		discussion		directions		greeting	
	560	1000	560	1000	560	1000	560	1000	560	1000	560	1000
Res. sup [35]	81.7	100.7	79.9	100.2	94.8	137.4	121.3	161.7	110.1	152.5	156.1	166.5
Traj-GCN [34]	54.1	59.8	53.4	77.8	50.7	72.6	91.6	121.5	<b>71.0</b>	101.8	115.4	148.8
MSR-GCN [8]	52.7	63.0	<b>52.5</b>	77.1	49.5	71.6	88.6	117.6	71.2	<b>100.6</b>	116.3	147.2
STS-GCN [43]	58.0	70.2	57.4	82.6	55.5	76.1	91.1	118.9	79.9	109.6	106.3	136.1
PK-GCN	<b>42.5</b>	<b>47.0</b>	57.9	<b>69.7</b>	<b>33.3</b>	<b>60.2</b>	<b>75.8</b>	<b>112.0</b>	74.7	101.9	<b>90.1</b>	<b>108.8</b>

understanding of both features. We follow them to predict 10 (i.e. short-term) or 25 (i.e. long-term) poses with the observed 10 poses. It should be added that errors in [35,34,8] are shown by per frame, while the one in [43] is shown by averaging on all previous frames. To keep in line with Traj-GCN and MSR-GCN, we re-demonstrate the error in [43] as the same protocol in [35,34,8].

### 4.3 Implement Details

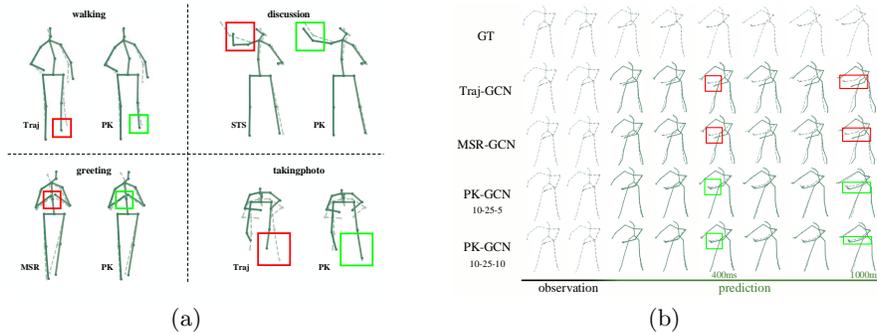
We implement our model using PyTorch [39] on an NVIDIA 2080TI GPU. The optimizer is ADAM [20], and the learning rate is set to 0.0005 with a 0.96 decay every two epochs. Each GC-Layer has a dropout rate of 0.5, with the learnable weight matrix  $\mathbf{W}$  of size  $256 \times 256$ . The batch size is set to 16. The ITP-Network and the FP-Network are trained separately. We first train the former one for 50 epochs, then fix it and train the latter for another 50 epochs. The size of our model is 2.8M for both stages.

### 4.4 Results

Following the baselines [34,8,43], we evaluate our prediction performance by showing the results of 400ms short-term and 1000ms long-term predictions on

**Table 3.** Comparisons of average MPJPE error (left) and MAE error (right) on H3.6M dataset. The best result is highlighted in bold.

millisecond (ms)	average-MPJPE						average-MAE					
	80	160	320	400	560	1000	80	160	320	400	560	1000
Res. sup [35]	34.7	62.0	101.1	115.5	135.8	167.3	0.36	0.67	1.02	1.15	-	-
Traj-GCN [34]	12.7	26.1	52.3	63.5	81.6	114.3	0.32	0.55	0.91	1.04	1.27	1.66
MSR-GCN [8]	12.1	25.6	51.6	62.9	81.1	114.2	-	-	-	-	-	-
STS-GCN [43]	16.3	28.1	54.4	65.8	85.1	117.0	0.31	0.57	0.91	1.03	1.22	1.61
PK-GCN	<b>10.8</b>	<b>23.3</b>	<b>48.2</b>	<b>57.4</b>	<b>76.1</b>	<b>106.4</b>	<b>0.29</b>	<b>0.54</b>	<b>0.85</b>	<b>0.96</b>	<b>1.15</b>	<b>1.57</b>



**Fig. 3.** (a) Visualized comparisons between three baselines [34,8,43] and PK-GCN on actions *walking*, *discussion*, *greeting*, *takingphoto* in H3.6M at the 1000ms testpoint (i.e. 25th frame). Performances marked in green boxes are better than in red. The dashed lines denote ground truth. (b) Visualized comparisons of short-term and long-term predictions on a motion sequence *Washwindow* in CMU-Mocap. We show the ground truth, Traj-GCN [34], MSR-GCN[8], ours with 5 privileged poses of PK, and ours with 10 privileged poses of PK. The prediction deviations on left wrist are marked with red/green boxes. Our results are much closer to the ground truth.

H3.6M, CMU-Mocap, and 3DPW with 10-pose sequence observed. The length of the privileged sequence offered to PK-GCN is also set to 10.

**H3.6M.** Table 1 and Table 2 show quantitative comparisons for MPJPE error between short/long-term prediction and the ground truth. We find that error obtained by PK-GCN is smaller than other baselines in most cases. Particularly, activities with larger ranges of motions, such as *discussion* and *takingphoto*, achieve more improvements than the others, in that the distilled PK restricts the overall moving trend/orientation of an activity from the future further away. Taking a sequence from *greeting* as an example, when the person makes a bow with hands folded in front, the prediction trajectory of hands is forced to move from the chest to down with PK foretelling that the hands will be at down in the end. On the other hand, the failure of *eating* and *sitting* confirms our assumption that activities with relatively motionless status are less likely to benefit much from our model, as the information in observations is sufficient to generate a nearly static sequence.

**Table 4.** Comparisons of average MPJPE error of short/long-term prediction on CMU-Mocap (left) and 3DPW (right) datasets. The best results are highlighted in bold.

millisecond (ms)	CMU-Average						3DPW-Average				
	80	160	320	400	560	1000	200	400	600	800	1000
Res. sup [35]	24.0	43.0	74.5	87.2	105.5	136.3	113.9	173.1	191.9	201.1	210.7
Traj-GCN [34]	11.5	20.4	37.8	46.8	55.8	86.2	35.6	67.8	90.6	106.9	117.8
MSR-GCN [8]	<b>8.1</b>	18.7	34.2	42.9	53.7	83.0	-	-	-	-	-
PK-GCN	9.4	<b>17.1</b>	<b>32.8</b>	<b>40.3</b>	<b>52.2</b>	<b>79.3</b>	<b>34.8</b>	<b>66.2</b>	<b>88.1</b>	<b>104.3</b>	<b>114.2</b>

We further present the average MPJPE and MAE errors in Table 3, wherein the omitted cells indicate no data source available. Meanwhile, visualization of prediction at 1000ms testpoint on four actions in H3.6M is given in Figure 3 (a).

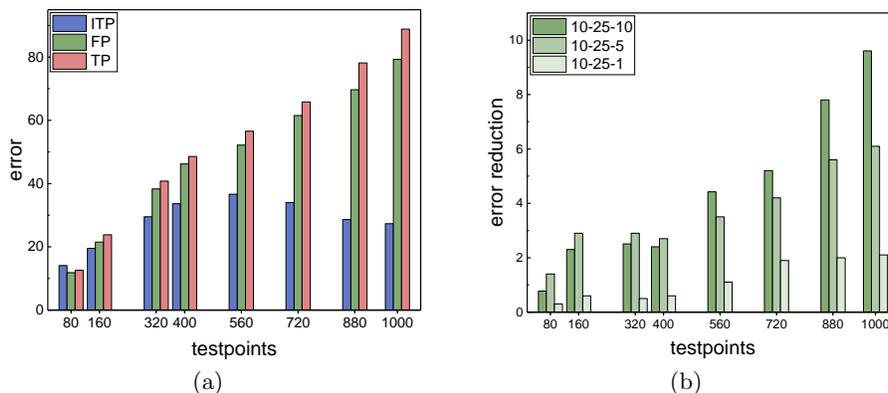
**CMU-Mocap and 3DPW.** Similar experiments are conducted on CMU-Mocap and 3DPW datasets, and the MPJPE errors of which are shown in Table 4. Apparently, our PK-GCN achieves state-of-the-art performance on CMU-Mocap among the baselines. In Figure 3 (b), qualitative visualization results of *Washwindow* are demonstrated with the ground truth. The prediction results of Traj-GCN [34], MSR-GCN[8], our PK-GCN with 5 privileged poses of PK, and ours with 10 privileged poses of PK, are from the top to the bottom. These comparisons show that our model performs better than that of the baselines. It should be emphasized that the first and second numbers behind our model indicate the length of the observed sequence and the predicted sequence, while the third number  $n$  means that we train our ITP-Network by introducing  $n$  privileged poses to learn PK, and distill it to assist the final prediction. Errors in Table 1, 2, 3, 4 are all obtained with PK of 10 poses.

In addition, the improvement achieved on 3DPW proves that PK is effective when faced with more complex data, but notably, it should be guaranteed that poses in one sample should be consistent. Two irrelevant actions in one sequence sample (such as *arguing* to *dancing*) is not suitable for PK distillation.

#### 4.5 Ablation Study

In this section, we conduct more experiments on three parts for deeper analysis on our PK-GCN. Following the previous notation, any numbers of the form  $a-b-c$  indicate predicting  $b$  poses on  $a$  historical poses, with  $c$  poses of PK introduced. **Interpolation results.** During the training of ITP, the loss between the interpolated sequence and the corresponding ground truth is minimized. In Figure 4 (a), among the eight testpoints within the range of 1000ms, the interpolation error show the performance of first increasing then decreasing, which is consistent with the variation trend of general interpolation problems, and the final prediction error appears a continuous increasing trend as expected.

A phenomenon easily neglected has emerged, that the interpolation process yields higher error than extrapolation at the beginning testpoint (i.e. 80ms), which seemingly contradicts our previous statement that interpolation often yields better results than extrapolation. In fact, human motions are of highly



**Fig. 4.** (a) The average MPJPE 10-25-10 interpolated error from ITP, final predicted error from FP, and error obtained under traditional prediction pattern (TP) on CMU-Mocap. (b) Comparison of the advancement achieved with different lengths of PK on CMU-Mocap.

smoothness with little difference between adjacent frames, and therefore the beginning predicted poses are quite close to observations. In other words, 10 poses of observation is sufficient to predict the following few frames directly. However, when the Priv-Encoder is involved in ITP to form an interpolation structure, the Obs-Encoder is compelled to sacrifice part of its original predictive property, to collaborate with Priv-Encoder for an *overall* (especially long-term) better prediction performance and thereby a relatively accurate representation of PK. Concerning FP-Network, where we reduce the weight of PKSL, it is acceptable even if the PK simulation is not *that* accurate, as PK just serves as an assistance in the extrapolation process, which leads to a continuous error-increase trend as standard extrapolation does, but lower than that without PK assisted. We regard the traditional prediction pattern that has no PK involved as TP, and the discrepancy between FP and TP is due to the distilled PK taking effects.

**Privileged sequences with different lengths.** We further set the length of the privileged sequence to 10, 5, 1 and 0, to evaluate how different lengths of privileged poses influence the final prediction performance. When the privileged length is down to 0, the two-step structure vanishes and is equivalent to predicting directly without PK (i.e. TP). In Figure 4 (b), we plot the advancement of PK-GCN trained with PK obtained from different lengths of privileged sequences on CMU-Mocap. From the figure, once PK is introduced, the final prediction performance is improved even when there only exists 1 privileged pose of PK. The higher error reduction of 10-25-5 than 10-25-10 at early testpoints indicates that too much PK may become a burden for short-term prediction.

**Baselines with PK.** In order to a) explore the impact of introducing PK on baselines under the traditional prediction pattern, and b) dispel the uncertainty whether our progress is *only* due to PK while irrelevant to our two-step struc-

**Table 5.** Comparisons of MPJPE error of 10-25-10 prediction on baselines with/without PSL and our PK-GCN on CMU-Mocap.

millisecond (ms)		Average					
		80	160	320	400	560	1000
Traj-GCN [34]	w/o PSL	13.9	24.8	43.3	52.7	55.8	86.2
	1*PSL/0.6*PSL	14.3/14.2	26.0/25.3	44.6/43.9	54.2/53.7	57.6/56.6	88.4/87.2
MSR-GCN [8]	w/o PSL	<b>10.5</b>	23.1	39.7	48.8	53.7	83.0
	1*PSL/0.6*PSL	11.2/10.6	24.0/23.6	41.6/39.3	49.5/48.8	55.5/54.3	85.6/85.0
PK-GCN		11.8	<b>21.5</b>	<b>38.3</b>	<b>46.2</b>	<b>52.2</b>	<b>79.3</b>

ture, we re-train Traj-GCN [34] and MSR-GCN [8] by also predicting poses in the privileged window and adding a loss (i.e. Privileged Sequence Loss, PSL) on them. Table 5 shows errors of baselines with/without PSL for 10-25-10 prediction. We further involve a reduction of PSL weight to 0.6, in line with the weight  $\lambda$  of PKSL in PK-GCN.

From the table, baselines with PSL yield even higher errors than without PSL, where the gap in-between exactly reflects the superiority of our two-step design. For baselines, when introducing PSL, the minimization of the total loss transforms the original 25-pose prediction into a 35-pose one. Since extrapolation becomes much harder when the prediction goes far, the last 10 poses, not belonging to the prediction object, waste big proportion of model resource and therefore lead to a worse performance of the previous 25 poses. Although an alleviation on PSL weight provides limited improvement, the direct use of PSL still harms the 25-pose prediction. However, our two-step spirit allows PK-GCN to remove these wastes during the training of FP-Network, focusing mainly on the prediction of 25 poses as we need.

## 5 Conclusion

In this paper, we introduce the overlooked poses existing after the predicted sequence to provide PK for human motion prediction from the view of interpolation. We extend the research window to the observed-predicted-privileged sequence, and propose a two-step model PK-GCN, wherein the ITP-Network learns a PK representation while interpolating the middle sequence, and the FP-Network predicts with the distilled PK assisted. Moreover, a novel PK-Simulator embedded in FP approximates the PK representation to implement the distillation process, enabling FP to imitate the behavior of ITP, and thereby keep the merits of interpolation during the final prediction. Our model outperforms the state-of-the-art methods on three benchmarked datasets, especially those activities with large ranges of motions.

**Acknowledgements.** This work was supported in part by the National Natural Science Foundation of China (NO. 62176125, 61772272).

## References

1. Aliakbarian, S., Saleh, F.S., Salzmann, M., Petersson, L., Gould, S.: A stochastic conditioning scheme for diverse human motion prediction. In: CVPR. pp. 5223–5232 (2020)
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
3. Butepage, J., Black, M.J., Kragic, D., Kjellstrom, H.: Deep representation learning for human motion prediction and classification. In: CVPR. pp. 6158–6166 (2017)
4. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: ICCV. pp. 4794–4802 (2019)
5. Corona, E., Pumarola, A., Alenyà, G., Moreno-Noguer, F.: Context-aware human motion prediction. In: CVPR. pp. 6992–7001 (2020)
6. Cui, Q., Sun, H., Li, Y., Kong, Y.: A deep bi-directional attention network for human motion recovery. In: IJCAI. pp. 701–707 (2019)
7. Cui, Q., Sun, H., Yang, F.: Learning dynamic relationships for 3d human motion prediction. In: CVPR. pp. 6519–6527 (2020)
8. Dang, L., Nie, Y., Long, C., Zhang, Q., Li, G.: Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In: ICCV. pp. 11467–11476 (2021)
9. Dong, M., Xu, C.: Skeleton-based human motion prediction with privileged supervision. *IEEE Transactions on Neural Networks and Learning Systems* (2022)
10. Fernando, B., Herath, S.: Anticipating human actions by correlating past with the future with jaccard similarity measures. In: CVPR. pp. 13224–13233 (2021)
11. Fragkiadaki, K., Levine, S., Felsen, P., Malik, J.: Recurrent network models for human dynamics. In: ICCV. pp. 4346–4354 (2015)
12. Gopalakrishnan, A., Mali, A., Kifer, D., Giles, L., Ororbia, A.G.: A neural temporal model for human motion prediction. In: CVPR. pp. 12116–12125 (2019)
13. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. In: CVPR. pp. 6047–6056 (2018)
14. Gui, L.Y., Wang, Y.X., Liang, X., Moura, J.M.: Adversarial geometry-aware human motion prediction. In: ECCV. pp. 786–803 (2018)
15. Hernandez, A., Gall, J., Moreno-Noguer, F.: Human motion prediction via spatio-temporal inpainting. In: ICCV. pp. 7134–7143 (2019)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
17. Hong, M., Xie, Y., Li, C., Qu, Y.: Distilling image dehazing with heterogeneous task imitation. In: CVPR. pp. 3462–3471 (2020)
18. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence* **36**(7), 1325–1339 (2013)
19. Jain, A., Zamir, A.R., Savarese, S., Saxena, A.: Structural-rnn: Deep learning on spatio-temporal graphs. In: CVPR. pp. 5308–5317 (2016)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
21. Koppula, H.S., Saxena, A.: Anticipating human activities for reactive robotic response. In: IROS. p. 2071 (2013)

22. Li, C., Zhang, Z., Lee, W.S., Lee, G.H.: Convolutional sequence to sequence model for human dynamics. In: CVPR. pp. 5226–5234 (2018)
23. Li, M., Chen, S., Zhao, Y., Zhang, Y., Wang, Y., Tian, Q.: Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In: CVPR. pp. 214–223 (2020)
24. Liang, M., Yang, B., Hu, R., Chen, Y., Liao, R., Feng, S., Urtasun, R.: Learning lane graph representations for motion forecasting. In: ECCV. pp. 541–556 (2020)
25. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: ECCV. pp. 816–833 (2016)
26. Liu, X., Yin, J., Liu, J., Ding, P., Liu, J., Liu, H.: Trajectorycnn: a new spatio-temporal feature learning network for human motion prediction. *IEEE Transactions on Circuits and Systems for Video Technology* **31**(6), 2133–2146 (2020)
27. Liu, Y., Chen, K., Liu, C., Qin, Z., Luo, Z., Wang, J.: Structured knowledge distillation for semantic segmentation. In: CVPR. pp. 2604–2613 (2019)
28. Liu, Z., Su, P., Wu, S., Shen, X., Chen, H., Hao, Y., Wang, M.: Motion prediction using trajectory cues. In: ICCV. pp. 13299–13308 (2021)
29. Ma, H., Li, J., Hosseini, R., Tomizuka, M., Choi, C.: Multi-objective diverse human motion prediction with knowledge distillation. In: CVPR. pp. 8161–8171 (2022)
30. Ma, T., Nie, Y., Long, C., Zhang, Q., Li, G.: Progressively generating better initial guesses towards next stages for high-quality human motion prediction. In: CVPR. pp. 6437–6446 (2022)
31. Mao, W., Liu, M., Salzmann, M.: History repeats itself: Human motion prediction via motion attention. In: ECCV. pp. 474–489 (2020)
32. Mao, W., Liu, M., Salzmann, M.: Generating smooth pose sequences for diverse human motion prediction. In: ICCV. pp. 13309–13318 (2021)
33. Mao, W., Liu, M., Salzmann, M.: Weakly-supervised action transition learning for stochastic human motion prediction. In: CVPR. pp. 8151–8160 (2022)
34. Mao, W., Liu, M., Salzmann, M., Li, H.: Learning trajectory dependencies for human motion prediction. In: ICCV. pp. 9489–9497 (2019)
35. Martinez, J., Black, M.J., Romero, J.: On human motion prediction using recurrent neural networks. In: CVPR. pp. 2891–2900 (2017)
36. Mishra, A., Marr, D.: Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *arXiv preprint arXiv:1711.05852* (2017)
37. Monti, A., Porrello, A., Calderara, S., Coscia, P., Ballan, L., Cucchiara, R.: How many observations are enough? knowledge distillation for trajectory forecasting. In: CVPR. pp. 6553–6562 (2022)
38. Paden, B., Čáp, M., Yong, S.Z., Yershov, D., Frazzoli, E.: A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles* **1**(1), 33–55 (2016)
39. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
40. Pavllo, D., Feichtenhofer, C., Auli, M., Grangier, D.: Modeling human motion with quaternion-based neural networks. *International Journal of Computer Vision* **128**(4), 855–872 (2020)
41. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014)
42. Shen, Z., He, Z., Xue, X.: Meal: Multi-model ensemble via adversarial learning. In: AAAI. pp. 4886–4893 (2019)
43. Sofianos, T., Sampieri, A., Franco, L., Galasso, F.: Space-time-separable graph convolutional network for pose forecasting. In: ICCV. pp. 11209–11218 (2021)

44. Sun, J., Lin, Z., Han, X., Hu, J.F., Xu, J., Zheng, W.S.: Action-guided 3d human motion prediction. *NeurIPS* **34**, 30169–30180 (2021)
45. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: *ICCV*. pp. 1365–1374 (2019)
46. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. *Neural Networks* **22**(5-6), 544–557 (2009)
47. Von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: *ECCV*. pp. 601–617 (2018)
48. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: *CVPR*. pp. 4933–4942 (2019)
49. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: *CVPR*. pp. 5532–5541 (2019)
50. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: *AAAI* (2018)
51. Yang, C., Xie, L., Su, C., Yuille, A.L.: Snapshot distillation: Teacher-student optimization in one generation. In: *CVPR*. pp. 2859–2868 (2019)
52. Yuan, Y., Kitani, K.: Dlow: Diversifying latent flows for diverse human motion prediction. In: *ECCV*. pp. 346–364 (2020)
53. Zhao, P., Xie, L., Zhang, Y., Wang, Y., Tian, Q.: Privileged knowledge distillation for online action detection. *arXiv preprint arXiv:2011.09158* (2020)
54. Zhong, C., Hu, L., Zhang, Z., Ye, Y., Xia, S.: Spatio-temporal gating-adjacency gcn for human motion prediction. In: *CVPR*. pp. 6447–6456 (2022)