Forecasting Distillation: Enhancing 3D Human Motion Prediction with Guidance Regularization

Anonymous Authors

Abstract—**Human motion prediction aims to forecast future body poses from historically observed sequences, which is challenging due to motion's complex dynamics. Existing methods mainly focus on dedicated network structures to model the spatial and temporal dependencies. The predicted results are required to be strictly similar to the training samples with ℓ_2 loss in the current training pipeline. It needs to be pointed out that most approaches predict the next frame conditioned on the previously predicted sequence, where a small error in the initial frame could be accumulated significantly. In addition, recent work indicated that different stages could play different roles. Hence, this paper considers a new direction by introducing a model learning framework with motion guidance regularization to reduce uncertainty. The guidance information is extracted from a designed Fusion Feature Extraction network (FE-Net) while knowledge distilling is conducted through intermediate supervision to improve the multi-stage prediction network during training. Incorporated with baseline models, our guidance design exhibits clear performance gains in terms of 3D mean per joint position error (MPJPE) on benchmark datasets Human3.6M, CMU Mocap, and 3DPW datasets, respectively. Related code will be available on https://github.com/tempAnonymous2024/MotionPredict-GuidanceReg.**

Index Terms—**motion prediction, knowledge distillation, graph neural networks**

I. Introduction

Humans have an innate ability to predict how future evolution of actions could be extrapolated. Equipping machines with the ability to anticipate human behavior remains a paramount challenge, which has gradually garnered widespread attention from researchers in recent years, particularly with the surge in applications such as robotics, autonomous driving, and human-computer interaction [1], [2]. The task of human motion prediction can be described as predicting future possible human action sequences based on the given human pose sequences that already occurred. Anticipating the future movement of the 3D human skeleton is a complex and challenging task due to the complex spatial-temporal modality and the great uncertainty of the future.

Early works have used statistical or probability models for motion prediction such as nonlinear Markov models [3], Gaussian process dynamic models [4], and restricted Boltzmann machines [5]. Many deep neural networks have also been applied to tackle the problem. Recurrent Neural Networks(RNNs) [6], [7] have been employed since they are naturally designed to handle temporally correlated sequences. However, RNNs could meet gradient disappearance and gradient explosion problems, which makes it ineffective in processing long sequence data.

Recently, Graph Convolutional Networks (GCNs) [8]–[15] have received widespread attention and research in this field, since it is capable to model the spatial dependencies naturally. These approaches regard each human pose as a graph composed of joints and bones and use graph convolution to model spatial information. Many complicated and sophisticated GCNs have been designed to improve the model's ability to capture more various spatial-temporal relationships in the motions. For instance, MSR-GCN [11] and MGCN [16] have employed a coarse-to-fine strategy to learn more multi-scale correlations of motion data. Transformers and attentions have been combined with GCNs [17], [18] to model how human skeletons evolve spatially and temporally over time.

However, the aforementioned work only focuses on exploiting the spatial and temporal relationships and designing more complicated models, largely overlooking the study of the uncertainty property of motion. The uncertainty of motion prediction mainly refers to its challenging variation, especially for non-periodic behaviors. We observe that most of the existing models use the average loss form which counts the error of each future frame equally. Actually, the uncertainty of human motion is not equal in each future frame. Long-term error accumulation has been recognized as one of the biggest issues that bring performance degradation in motion prediction problems. As most approaches predict the next frame conditioned on the previously predicted sequence, a small error in the initial frame could be accumulated significantly. Hence, it's essential to tackle the accumulation error issue for the motion prediction task.

Latest methods such as PGBIG [14] employs a multi-stage network structure, which conduct motion prediction as a consecutive process composed of many round. Their work [14] proves that a different initial pose could bring sharper performance gains than an individual method. An intermediate target is employed which is adopted through an Accumulated Average Smoothing algorithm. Such manually designed intermediate target could introduce future information for early stages but it is lack exploiting motion dynamics. The proposed method considers a new direction by introducing a model learning framework with motion guidance regularization to reduce uncertainty. Our key insight lies in that the step-by-step prediction could be improved through the guidance regularization in a multi-stage pipeline. An encoder-decoder structure is designed for the multi-stage prediction network. The encoder uses the past sequence as input and includes a set of self-attention blocks to extract high-dimensional features. The guidance regularization information is extracted from a designed Fusion Feature Extraction network (FE-Net) while knowledge distilling is conducted through intermediate supervision to improve the feature extracting of multi-stage prediction network during training.

In order to evaluate the proposed method, extensive experiments are carried out on H3.6M, 3DPW and CMU Mocap datasets to study the impact of exploiting guidance regularization information for the final performance. Incorporated with baseline models, the results demonstrate that our method achieves competitive performance in both short-term and long-term motion prediction tasks. Besides, our prediction results are more smooth and natural, achieve high quality without the intractable shaking effects. In summary, the contributions of this paper are the following:

1) A guidance regularization method is proposed to tackle the issue of accumulation error of motion prediction caused by motion uncertainty. The role of uncertainty property in human motion prediction tasks has been studied. We hope our work will encourage more studies to rethink the value of uncertainty factors in motion prediction problems.
2) A novel motion prediction work involving intermediate supervision of guidance information is proposed. A knowledge distilling framework is designed to conduct intermediate supervision for the multi-stage prediction network during training. Extensive experimental results on the benchmark datasets validate our method outperforms competitors in most short-term prediction jobs and achieve competitive performance in long-term prediction jobs. The proposed method could bring more favorable gains than existing methods.
3) We delve deeper into the intermediate supervision of guidance information and carry out extensive experiments to pursue the problem of how could intermediate supervision affect the training and learning of the motion prediction models. Fruitful insights are given out by our ablation studies.

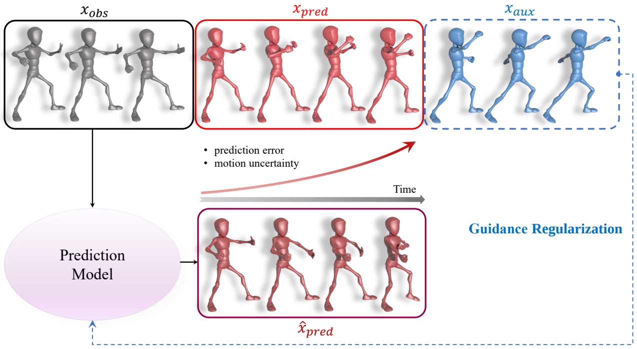

Fig. 1. Training pipeline of proposed method. Existing approaches rely on mapping the past sequences x_{obs} to the future sequences x_{pred}. Prediction error $\hat{x}_{pred}$ could increase over time. To address this issue, auxiliary sequences x_{aux} are introduced to provide guidance regularization.

II. Related Works

Traditional motion prediction works have employed Hidden Markov model [3], binary latent variables [5] or Gaussian process [4] to model sequences. Since motion prediction was often seen as a sequence-to-sequence task, many deep learning methods such as RNN-based [6] and feed-forward neural networks [19]–[23] have also been applied to the field

Recently, Graph Convolutional Networks (GCNs) have achieved great success in motion prediction task. LTD [8] proposed to model joint-wise trajectories by graph convolutional layers to strengthen the spatial correlation as well as model temporal information by DCT representations.

Such mechanism has been followed by lots of later work. JDM [9] and MPT [10] have took velocity information as input. Many works have tried to improve the graph representation for either scale [11], [24] or space-time separable [12], [13], [25] strategy. In order to improve the learning procedure, PGBIG [14] performed muti-stage prediction whose output corresponded to a smooth sequence in every stage and gradually approached the true value. Knowledge distillation is introduced into the prediction model in PKGCN [15]. Frequency decomposition and feature aggregation is used in DMAB [26] to encode temporal dynamic within the frequency domain. Prediction deviation is involved in DeFee [27] for multi-rounds prediction. Meanwhile, Transformer was also attempted for human motion prediction [17], [18], [28], [29].

Some stochastic approaches such as HumanMAC [30] and BeLFusion [31] used diffusion model to generate more continuous predictions. MotionMixer [32] and siMLPe [33] applied simple MLPs to fuse spacial and temporal information, and achieved reasonable performance.

III. Methodology

A. Problem Formulation

Human motion prediction aims to infer future motion sequences given past sequences. Inspired by existing methods, we denote $X_{1:N} = \{x_1, x_2, \ldots, x_N\}$ as the observed sequence with N consecutive human poses and $X_{N+1:N+P} = \{x_{N+1}, x_{N+2}, \ldots, x_{N+P}\}$ as the predicted sequence with P poses, where $x_i \in \mathbb{R}^{J \times D}$, with J joints and D dimensions (D is 3).

Traditional approaches typically focus on establishing a mapping function that maps $X_{1:N}$ to $X_{N+1:N+P}$ to capture their spatial-temporal dependencies. However, as time advances, prediction errors accumulate over time. In such cases, we believe that the subsequent sequence can provide accurate guidance regularization for prediction, thereby reducing the uncertainty of actions and minimizing long-term prediction errors, as illustrated in Fig. 1. Particularly, we introduce the auxiliary sequence denoted as $X_{N+P+1:N+P+K} = \{x_{N+P+1}, x_{N+P+2}, \ldots, x_{N+P+K}\}$ with K poses after the predicted sequence in pre-training, and transfer its information to the final prediction network to aid in the prediction process.

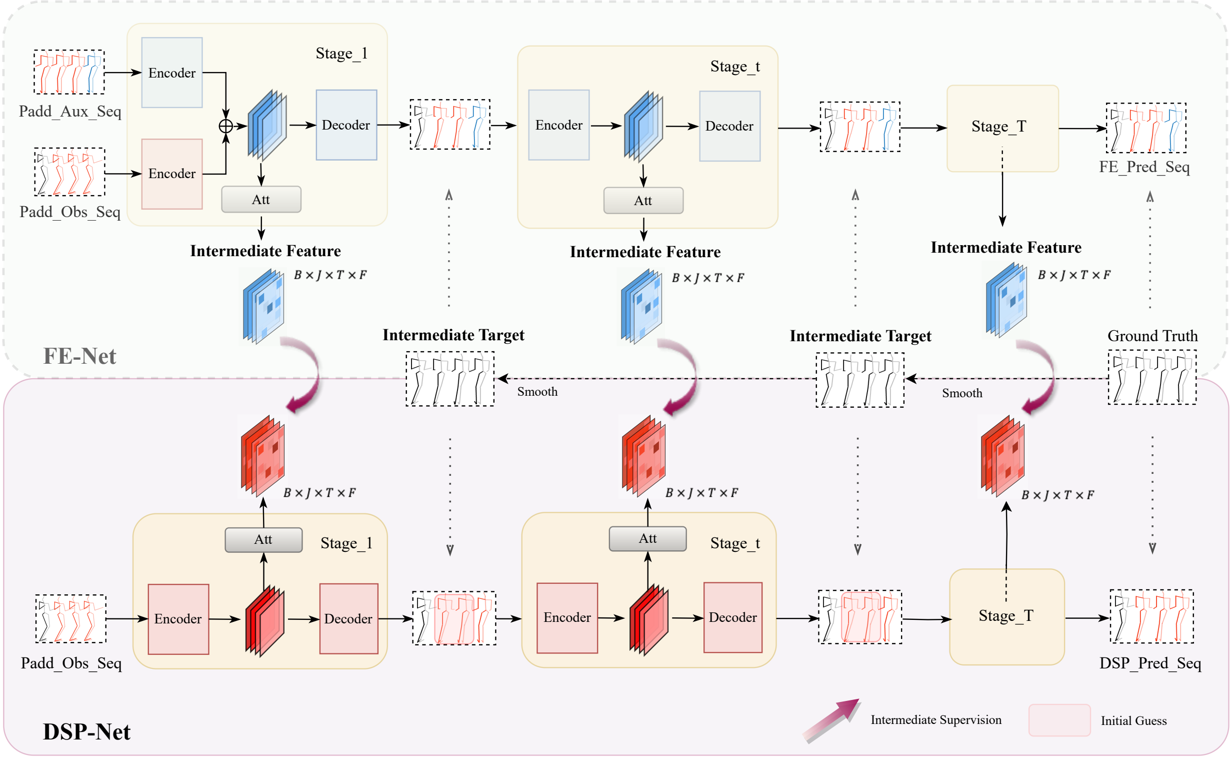

Fig. 2. Overview of proposed model for human motion prediction. FE-Net is used to encode the fusion features of the observed and auxiliary sequences to obtain the optimal feature representation. During the final prediction, DSP-Net distills the attention-enhanced feature to obtain guidance regularization from the auxiliary sequences. Both DSP-Net and FE-Net employ the same multi-stage prediction network, with each stage consisting of an Encoder-Decoder pair, whose output sequences are supervised by the ground truth and the smoothed results, enabling iterative optimization of the initial guess.

B. Network structure

The entire network structure is shown in Fig. 2. Both the FE-Net and the DSP-Net employ a similar multi-stage prediction network structure as described in III-C. Since the auxiliary sequence should be unavailable during training, we initially pre-train an FE-Net using the training data with its corresponding auxiliary sequence. Then the FE-Net is fixed as a teacher net while features of the observed sequence are extracted. A knowledge distilling operation is conducted, where the fusion feature from FE-Net is transferred through intermediate supervision to the prediction network DSP-Net.

a) Fusion Feature Extraction Network (FE-Net): The FE-Net fuses information from both the observed sequence and the auxiliary sequence, denoted as $\mathcal{M}^{\mathrm{FE}}(\cdot)$. Inspired by [8], [34], we duplicate the last pose of observed sequence $P+K$ times (represented as X_{padd_obs}) and duplicate the first pose of the auxiliary sequence forward $N+P$ times (represented as X_{padd_aux}). The FE-Net performs as below:

$$X_{\mathrm{FE}\_out} = \mathcal{M}^{\mathrm{FE}}\left(X_{padd\_obs}, X_{padd\_aux}\right) \tag{1}$$

where the output X_{FE_out} closely approximates the ground truth, the FE-Net can learn the most comprehensive fusion feature, which is then utilized as intermediate supervision for the DSP-Net.

b) Dual-Supervision Prediction Network (DSP-Net): The DSP-Net adopts similar multi-stage prediction network, denoted as $\mathcal{M}^{\mathrm{DSP}}(\cdot)$, which only receives X_{padd_obs} as input. The DSP-Net performs as below:

$$X_{\mathrm{DSP}\_out} = \mathcal{M}^{\mathrm{DSP}}\left(X_{padd\_obs}\right) \tag{2}$$

As depicted in the lower section of Fig. 2, the outputs of each stage of DSP-Net are supervised by the ground truth and the smoothed results. Through iterative backpropagation, the predictions are continuously refined to converge towards the ground truth. Additionally, we incorporate the fixed feature representations from the FE-Net as intermediate supervision through knowledge distillation, enabling the DSP-Net to receive guidance regularization simultaneously. Notably, we have designed an attention mechanism that aligns the features of both networks, facilitating effective information transfer. Indeed, the former supervision plays a primary role, while the latter serves as a supplementary aid.

C. Multi-stage prediction network

a) Overview: A multi-stage structure is employed to construct the proposed prediction network inspired by [14], which works as the backbone model for both FE-Net and DSP-Net. The whole multi-stage structure contains T Init-Guess stages described in III-D represented by $S^1, S^2, \ldots, S^T$, and each stage adopts the structure of encoder-decoder (note that the FE-Net's first stage has two encoders to handle different

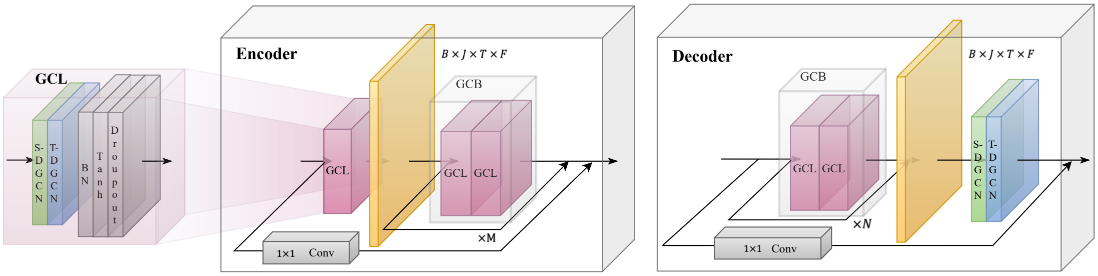

Fig. 3. The encoder-decoder architecture employed in the stage block mentioned in Fig. 2. The encoder and decoder are composed of the smallest computational units called GCL, as illustrated on the left side. Two GCLs make up a GCB. The encoder consists of a GCL, M residual GCBs, and a global residual structure. The decoder comprises N residual GCBs, S-DGCN, T-DGCN, and a global residual structure.

inputs). The output $\hat{X}^t$ of S^t can be defined as below ((3) for FE-Net, (4) for DSP-Net):

$$\hat{X}^t = \begin{cases} S^t\left(X_{padd\_obs}, X_{padd\_aux}\right), & t = 1 \\ S^t\left(\hat{X}^{t-1}\right), & t = 2, 3, \ldots, T \end{cases} \tag{3}$$

$$\hat{X}^t = \begin{cases} S^t\left(X_{padd\_obs}\right), & t = 1 \\ S^t\left(\hat{X}^{t-1}\right), & t = 2, 3, \ldots, T \end{cases} \tag{4}$$

b) Intermediate target supervision: For the FE-Net, our goal is to reconstruct both the prediction sequence and the auxiliary sequence, thus the initial guess should be $\hat{X}^t_{N+1:N+P+K}$. For the DSP-Net, we only focus on the prediction sequence, so the initial guess should be $\hat{X}^t_{N+1:N+P}$. Our objective is to iteratively optimize the initial guess until it closely approximates the ground truth. To achieve this, we apply a multi-stage smoothing algorithm (MSA) to the ground truth, generating the intermediate target $\tilde{X}^t_L = \left\{\tilde{x}^t_{N+1}, \tilde{x}^t_{N+2}, \ldots, \tilde{x}^t_{N+L}\right\}$ with length L to the initial guess of stage t. The MSA is defined as:

$$\tilde{x}^T_l = \text{MSA}\left(X_{N+1:N+l}\right) = \frac{1}{l-N}\sum_{i=N+1}^{l} x_i, l \in (N+1, L) \tag{5}$$

To clarify, the position of a frame can be calculated as the average of the positions of all previous frames. By recursively applying this calculation, we can obtain the intermediate targets for each stage.

D. Init-Guess stage

In each stage, the S-DGCN (Spatial Graph Convolutional Network) and T-DGCN (Temporal Graph Convolutional Network) are employed to capture the spatial-temporal dependencies of human motion sequences. Based on the S-DGCN and T-DGCN, we adopt an encoder-decoder architecture, as illustrated in Fig. 3.

a) S-DGCN: S-DGCN is defined to capture the correlations between joints. A learnable adjacency matrix $A_S \in \mathbb{R}^{J\times J}$ is defined to represent the strength of connections between joints, where J is the number of joints. Given the human motion sequence $X \in \mathbb{R}^{J\times T\times F}$ and a set of trainable S-DGCN weights $W_S \in \mathbb{R}^{F\times F'}$, where T is the sequence length and F is the numbers of the features, the output of S-DGCN could be computed as follows:

$$X' = A_S X W_S \tag{6}$$

Where $X' \in \mathbb{R}^{J\times T\times F'}$.

b) T-DGCN: T-DGCN is defined to learn the dependencies between the different trajectories. We defined a learnable adjacency matrix $A_T \in \mathbb{R}^{T\times T}$, given a set of trainable T-DGCN weights $W_T \in \mathbb{R}^{F'\times F'}$, by reversing the first two dimensions of X', the output of T-DGCN could be computed as below:

$$X'' = A_T(X')^\top W_T \tag{7}$$

Where $X'' \in \mathbb{R}^{T\times J\times F'}$, $(\cdot)^\top$ represents the transpose operation. Then, we transpose the first two dimensions of X'' back to obtain the final output $Y \in \mathbb{R}^{J\times T\times F'}$.

c) GCL: The graph convolutional layer (GCL) includes S-DGCN, T-DGCN, batch normalization, tanh activation and dropout, as shown in the right portion of Fig. 3. Two GCLs form a graph convolutional block (GCB).

d) Encoder: As shown in the middle portion of Fig. 3, the encoder consists of a GCL followed by M residual GCBs. The initial GCL maps the input from the action space of $\mathbb{R}^{J\times T\times D}$ to the feature space of $\mathbb{R}^{J\times T\times F}$, where F is 16, while the residual GCBs extract spatial-temporal features in the feature space. A global residual connection, implemented through a 1×1 convolution, is added between the input and output features, enabling the overall network to better learn the mapping.

e) Decoder: As shown in the left portion of Fig. 3, the decoder consists of N residual GCBs followed by S-DGCN and T-DGCN. The residual GCBs operate in the feature space, and the final S-DGCN and T-DGCN are used to map the pose sequence back from the feature space to the action space. Similarly, we employ a global residual connection using a 1×1 convolution that acts on the input features and the output sequence.

E. Intermediate supervision

When the pre-trained FE-Net performs optimally, the output feature $\mathbb{H}^t_{FE} \in \mathbb{R}^{J\times T\times F}$ of stage t's encoder contains the fusion features from both auxiliary and observed sequences. As mentioned in III-B, $\mathbb{H}^t_{FE}$ is employed as intermediate

TABLE I

COMPARISON OF SHORT-TERM PREDICTION FOR ALL ACTIONS AND AVERAGE ON HUMAN3.6M. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE SECOND BEST ARE MARKED BY UNDERLINE.

action	**walking**				**eating**				**smoking**				**discussion**			
millisecond	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***
Res-sup	29.4	50.8	76.0	81.5	16.8	10.6	56.9	68.7	23.0	42.6	70.1	82.7	32.9	61.2	90.9	96.2
LTD	12.3	23.0	39.8	46.1	8.4	16.9	33.2	40.7	7.9	16.2	31.9	38.9	12.5	27.4	58.5	71.7
MSR	12.1	22.7	38.6	45.2	8.4	17.1	33.0	40.4	8.0	16.3	31.3	38.2	12.0	26.8	57.1	69.7
PGBIG	10.2	19.8	34.5	40.3	7.0	15.1	30.6	38.1	6.6	14.1	28.2	34.7	**10.0**	**23.8**	53.6	66.7
DeFee*	10.2	19.9	35.5	42.5	**6.7**	**14.8**	30.7	38.3	**6.5**	**14.0**	28.8	35.6	10.0	23.8	54.5	68.1
Ours	**10.1**	**19.3**	**33.2**	**38.6**	7.4	15.2	**30.6**	**37.8**	6.9	14.1	**28.2**	**34.7**	10.3	24	**53.6**	**66.5**
action	**directions**				**greeting**				**phoning**				**posing**			
millisecond	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***
Res-sup	35.4	57.3	76.3	87.7	34.5	63.4	124.6	142.5	38.0	69.3	115.0	126.7	36.1	69.1	130.5	157.1
LTD	9.0	19.9	43.4	53.7	18.7	38.7	77.7	93.4	10.2	21.0	42.5	52.3	13.7	29.9	66.6	84.1
MSR	8.6	19.7	43.3	53.8	16.5	37.0	77.3	93.4	10.1	20.7	41.5	51.3	12.8	29.4	67.0	85.0
PGBIG	7.2	17.6	40.9	51.5	**15.2**	34.1	71.6	87.1	**8.3**	18.3	38.7	48.4	**10.7**	**25.7**	**60.0**	**76.6**
DeFee*	**6.9**	**16.7**	**39.6**	**50.2**	16.4	**32.8**	**68.8**	**82.5**	11.4	19.8	40.8	49.7	14.8	28.3	64.8	80.5
Ours	7.6	17.9	40.8	51.2	15.8	35.1	72.1	87.4	8.6	**18.3**	**38.5**	**48.1**	10.8	25.9	60.1	76.7
action	**purchases**				**sitting**				**sittingdown**				**takingphoto**			
millisecond	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***
Res-sup	36.3	60.3	86.5	95.9	42.6	81.4	134.7	151.8	47.3	86.0	145.8	168.9	26.1	47.6	81.4	94.7
LTD	15.6	32.8	65.7	79.3	10.6	21.9	46.3	57.9	16.1	31.1	61.5	75.5	9.9	20.9	45.0	56.6
MSR	14.8	32.4	66.1	79.6	10.5	22.0	46.3	57.8	16.1	31.6	62.5	76.8	9.9	21.0	44.6	56.3
PGBIG	**12.5**	**28.7**	**60.1**	**73.3**	8.8	19.2	42.4	53.8	13.9	27.9	57.4	71.5	8.4	18.9	42.0	53.3
DeFee*	16.6	32.5	67.6	80.6	14.0	23.3	47.5	58.7	**9.8**	29.1	61.8	**70.2**	**7.8**	**16.9**	**37.1**	**47.7**
Ours	12.8	29	60.4	73.5	**8.8**	**19**	**42.1**	**53.6**	13.8	**27.8**	**56.7**	70.7	8.6	18.8	41.8	53.1
action	**waiting**				**walkingdog**				**walkingtogether**				**average**			
millisecond	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***	***80***	***160***	***320***	***400***
Res-sup	30.6	57.8	106.2	121.5	64.2	102.1	141.1	164.4	26.8	50.1	80.2	92.2	34.7	62.0	101.1	115.5
LTD	11.1	24.0	50.1	61.5	23.4	46.2	83.5	96.0	10.5	21.0	38.5	45.2	12.7	26.1	52.3	63.5
MSR	10.7	23.1	48.3	59.2	20.7	42.9	80.4	93.3	10.6	20.9	37.4	43.9	12.1	25.6	51.6	62.9
PGBIG	**8.9**	20.1	43.6	54.3	18.8	**39.3**	73.7	86.4	8.7	18.6	**34.4**	**41.0**	**10.3**	**22.7**	47.4	58.5
DeFee*	9.3	**19.5**	**42.0**	**53.4**	**17.3**	40.9	**72.8**	**84.4**	**8.3**	19.1	35.9	41.5	-	-	-	-
Ours	9.1	20.3	43.7	54.3	19.1	39.6	74.2	87.1	8.8	**18.4**	34.6	41.2	10.6	22.8	**47.3**	**58.3**

*The code is not publicly available, so the best reported results from the paper is used [27]. Furthermore, it only focused on short-term predictions, no SOTA comparison of long-term prediction is included.

supervision for the output feature $\mathbb{H}^t_{DSP} \in \mathbb{R}^{J\times T\times F}$ of stage t's encoder in DSP-Net, which enables DSP-Net to learn the feature representation of the auxiliary sequences, while receiving guidance regularization during prediction. Additionally, to enhance the spatial dependency within $\mathbb{H}^t_{FE}$ and facilitate efficient information transfer to $\mathbb{H}^t_{DSP}$, we have devised an attention-based knowledge distillation approach. The attention map $\alpha_t \in \mathbb{R}^{J\times J}$ that represent relations between joints are calculated as below:

$$\alpha_t = softmax(\frac{\sigma(\mathbb{H})\varphi(\mathbb{H})^\top}{\sqrt{d_k}}) \tag{8}$$

Where $\sigma(\cdot)$ and $\varphi(\cdot)$ are the mapping function to generate query and key, d_k is the dimension of key. The enhanced feature is computed as follows:

$$\hat{\mathbb{H}} = \phi(\mathbb{H})\alpha_t \tag{9}$$

Where $\phi(\cdot)$ is the mapping function to generate value. $\hat{\mathbb{H}}^t_{FE}$ is used to supervise $\hat{\mathbb{H}}^t_{DSP}$, and the closer they are, the better we consider DSP-Net has obtained long-term guidance regularization.

F. Loss function

a) Loss for FE-Net: We utilize the $\mathcal{L}_2$ loss to measure the loss on the all outputs:

$$\mathcal{L}_{\text{FE}} = \frac{1}{T}\sum_{t=1}^{T}\left\|\hat{X}^t_{1:N+P+K} - \tilde{X}^t_{1:N+P+K}\right\|_2 \tag{10}$$

Note that FE-Net aims to simultaneously reconstruct both the prediction sequence and the auxiliary sequence in pre-training, the length of the loss should be $N + P + K$. When $\mathcal{L}_{FE}$ is minimized, the optimal representation of $\hat{\mathbb{H}}^t_{FE}$ is denoted as $\mathbb{V}^t$.

b) Loss for DSP-Net: In the final prediction, we only focus on the $\mathcal{L}_{\text{DSP}}$. Since the auxiliary sequence is unavailable during this phase and our main objective is to generate the prediction sequence with length $N+P$, the loss on all outputs is determined as follows:

$$\mathcal{L}_{target} = \frac{1}{T}\sum_{t=1}^{T}\left\|\hat{X}^t_{1:N+P} - \tilde{X}^t_{1:N+P}\right\|_2 \tag{11}$$

Moreover, it is desirable for the intermediate feature $\hat{\mathbb{H}}^t_{DSP}$ to learn from $\mathbb{V}^t$, with the objective of maximizing their

TABLE II

COMPARISON OF LONG-TERM PREDICTION FOR ALL ACTIONS AND AVERAGE ON HUMAN3.6M. OUR METHOD OUTPERFORMS SOTAS IN MOST CASES.

action	**walking**		**eating**		**smoking**		**discussion**		**directions**		**greeting**		**phoning**		**posing**	
millisecond	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***
Res-sup	81.7	100.7	79.9	100.2	94.8	137.4	121.3	161.7	110.1	152.5	156.1	166.5	141.2	131.5	194.7	240.2
LTD	54.1	59.8	53.4	77.8	50.7	72.6	91.6	121.5	71.0	101.8	115.4	148.8	69.2	103.1	114.5	173.0
MSR	52.7	63.0	52.5	77.1	49.5	71.6	88.6	117.6	71.2	100.6	116.3	147.2	68.3	104.4	116.3	174.3
PGBIG	48.1	56.4	51.1	76.0	46.5	69.5	87.1	118.2	69.3	**100.4**	110.2	143.5	65.9	102.7	**106.1**	164.8
Ours	**46.3**	**54.8**	**50.1**	**74.4**	**46.4**	**68.4**	**86.7**	**117.3**	**69.2**	100.5	**109.7**	**142.2**	**65.7**	**101.6**	106.5	**164.5**
action	**purchases**		**sitting**		**sittingdown**		**takingphoto**		**waiting**		**walkingdog**		**walkingtogether**		**average**	
millisecond	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***	***560***	***1000***
Res-sup	122.7	160.3	167.4	201.5	205.3	277.6	117.0	143.2	146.2	196.2	191.3	209.0	107.6	131.1	97.7	130.5
LTD	102.0	143.5	78.3	119.7	100.0	150.2	77.4	119.8	79.4	108.1	111.9	148.9	55.0	65.6	81.6	114.3
MSR	101.6	139.2	78.2	120.0	102.8	155.5	77.9	121.9	76.3	106.3	111.9	148.2	52.9	65.9	81.1	114.2
PGBIG	**95.3**	**133.3**	74.4	**116.1**	96.7	147.8	74.3	118.6	72.2	103.4	**104.7**	**139.8**	51.9	64.3	76.9	110.3
Ours	95.5	133.5	**74.4**	116.2	**96.1**	**147.2**	**73.9**	**116.8**	**71.9**	**103.3**	105.9	142.3	**51.3**	**61.2**	**76.6**	**109.6**

similarity. The formulation for calculating the distillation loss is as follows:

$$\mathcal{L}_{distill} = \frac{1}{T} \sum_{t=1}^{T} \left\| \mathbb{V}^t - \hat{\mathbb{H}}_{DSP}^t \right\|_2 \qquad (12)$$

Therefore, the total loss for DSP-Net is defined as below:

$$\mathcal{L}_{\text{DSP}} = \mathcal{L}_{target} + \lambda \mathcal{L}_{distill} \qquad (13)$$

where λ is the weight of the distillation loss. In this paper, we set it to 0.5.

IV. EXPERIMENTAL RESULTS

A. Datasets

a) Human3.6M (H3.6M): H3.6M is the most commonly used dataset for motion prediction, consisting of 15 actions (e.g., walking, sitting and posing) performed by seven actors (S1, S5$\sim$S9, S11). The actions are represented by 32 joints. The sequences are downsampled to 25 frames per second, like [13], [14], excluding global rotation and translation of the pose. We use the data of S5 and S11 as test and validation datasets, while the remaining datasets are used for training.

b) CMU-Mocap: CMU-Mocap dataset consists of 8 action categories with 25 body joints. The division of training and testing sets follows the method proposed in [14]. Additionally, Other details are similar to H3.6M.

c) 3DPW: 3DPW dataset contains indoor and outdoor actions captured by 30Hz. A single pose has 23 body joints. Official training, test and validation sets are employed in this work.

B. Evaluation metric and baselines

a) Evaluation metric: Mean Per Joint Position Error (MPJPE) is used as the evaluation metric that calculates the ℓ_2 distance between the predicted values and the ground truth for each frame:

$$\ell_{\text{MPJPE}} = \frac{1}{J(N+P)} \sum_{t=1}^{N+P} \sum_{j=1}^{J} \| \hat{x}_{j,t} - x_{j,t} \|_2 \qquad (14)$$

where $\hat{x}_{j,t}$ denotes the predicted jth joint position in frame t, and $x_{j,t}$ is the corresponding ground truth.

b) Baselines: We compare our work on these three datasets with Res-sup [34], LTD [8], MSR [11], PGBIG [14] and DeFee [27] which are the current state-of-the-art methods. All evaluations are under the same conditions to ensure a fair comparison. Besides, the code of DeFee is not published, the best reported results from the paper are used [27]. Additionally, DeFee [27] only focused on short-term predictions thus no comparison of long-term prediction is included.

c) Implementation details: All implementations are conducted on an NVIDIA GeForce RTX 3090 GPU using Adam optimizer. The learning rate is set to 0.005 with a 0.96 decay every two epochs. The length of the auxiliary sequence K is 10. In the multi-stage framework, we set the number of stages T is 4, and each encoder contains 1 GCB and the decoder contains 2 GCBs. We firstly pre-train the FE-Net for 100 epochs with batch size of 256, then train the DSP-Net for 50 epochs with batch size of 16.

C. Comparison with the State-of-the-art Methods

To validate the prediction performance of the proposed model, the quantitative comparison is conducted for both short-term and long-term prediction on all benchmark datasets.

a) H3.6M: The comparison of short-term and long-term prediction on H3.6M is reported in Table I and Table II respectively. Our approach demonstrates particularly effective results in long-term predictions, showing significant enhancements for actions such as ”takingphoto” and ”walking”. We believe that the introduction of guidance for these actions can enable better understanding and prediction of the whole trajectory. Additionally, our model outperforms other methods in most cases of short-term predictions, showcasing competitiveness. However, in extremely short-term predictions (e.g., 80ms and 160ms), our method does not show much significant improvement.

b) CMU-Mocap and 3DPW: Experimental results on CMU-Mocap and 3DPW are presented in Tables III and Table IV respectively. Only average MPJPE at test-points(e.g., 80ms, 160ms, 320ms, etc.) is listed due to space limitation. The results on CMU-Mocap demonstrate similarities to those achieved on H3.6M, highlighting the effectiveness of our method in long-term predictions. Moreover, our approach ex-

TABLE III
COMPARISON OF PREDICTION FOR THE AVERAGE PREDICTION ON CMU-MOCAP.

millisecond	80	160	320	400	560	1000
Res-sup	24.0	43.0	74.5	87.2	105.5	136.3
LTD	9.3	17.1	33.0	40.9	55.8	86.2
MSR	8.1	15.2	30.6	38.6	53.7	83.0
PGBIG	**7.6**	**14.3**	**29**	**36.6**	50.9	80.1
Ours	8.4	15.7	30.5	37.0	**50.1**	**74.5**

TABLE IV
COMPARISON OF PREDICTION FOR THE AVERAGE PREDICTION ON 3DPW.

millisecond	200	400	600	800	1000
Res-sup	113.9	173.1	191.9	201.1	210.7
LTD	35.6	67.8	90.6	106.9	117.8
MSR	37.8	71.3	93.9	110.8	121.5
PGBIG	29.3	58.3	79.8	94.4	104.1
Ours	**25.9**	**52.9**	**74.0**	**89.6**	**100.5**

cels on the challenging 3DPW dataset, outperforming various baselines in both long-term and short-term predictions.

D. Ablation study

In this section, the result of ablation study is presented to further analyze the proposed method. All experimental results are obtained on the H3.6M dataset.

TABLE V
ABLATION STUDY OF ARCHITECTURE ON H3.6M.

millsecond		80	160	320	400	560	1000	ave
w/o ItSup		10.2	22.8	48.1	59.4	78.1	111.1	55.0
with ItSup								
feature -based	w/o att	10.3	22.5	47.5	58.3	76.8	110.1	54.3
	λ=0.4	**10.1**	**22.3**	47.6	58.9	77.6	110.6	54.5
	λ=0.5	10.6	22.8	**47.4**	**58.3**	**76.6**	109.6	**54.2**
	λ=0.6	10.3	22.5	47.4	58.5	77.2	110.4	54.4
	λ=1.0	11.0	23.2	47.8	58.8	76.6	**109.4**	54.5
response-based		10.3	22.8	48.1	59.3	78.1	111.7	55.1

a) Architecture: A multi-stage prediction network has been adopted as an baseline model while several additional modules are introduced.

1) **Intermediate supervision (ItSup).** The average MPJPE for baseline decreases when intermediate supervision is introduced. For instance, the average MPJPE is 55.0 at 1000ms for baseline and decreases to 54.3 with intermediate supervision, with a significant improvement of 1.0 at 1000ms. These results further validate our hypothesis that intermediate supervision can enhance long-term prediction performance.
2) **Attention module.** After the inclusion of the attention module, we observe a further reduction in the average MPJPE. This finding highlights the necessity of the attention module in learning the spatial joint structure effectively. By incorporating attention mechanisms, our model can focus on important joints and capture relevant information.
3) **Loss function weight.**We vary the weight of $\mathcal{L}_{distill}$ (i.e. λ) from 0.4 to 1 and observe that as λ increases, there is a slight improvement in long-term prediction performance. However, the short-term prediction performance is noticeably affected. This indicates that intermediate supervision can only serve as auxiliary predictions.
4) **Knowledge type for distillation.** To investigate the effectiveness of distillation, additional experiment using response-based knowledge distillation is conducted. In this experiment, each stage's output sequence of DSP-Net is learned from the sequence predicted by FE-Net. We use ℓ_2 loss to measure the difference between them. The results indicate that distilling knowledge in the feature space outperforms distillation in the 3D space. This further validates the effectiveness of our encoder and attention module in capturing and transferring essential knowledge across the network.

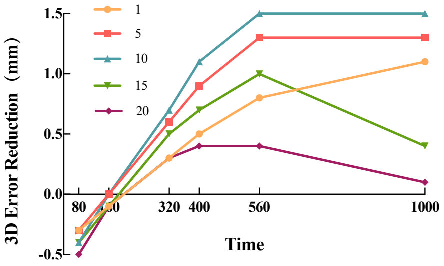

Fig. 4. Comparison of performance gains with different auxiliary sequence length on H3.6M.

b) The length of the auxiliary sequence: To investigate the impact of the length K of the auxiliary sequence on the the model, we conduct experiments with K values of 1, 5, 10, 15 and 20, comparing them with the results when K is 0 (i.e., without intermediate supervision). Fig. 4 illustrates the results of this ablation experiment. We find that introducing longer auxiliary sequences leads to better long-term prediction performance. However, as mentioned earlier, it also comes at the expense of weakened short-term prediction accuracy. Moreover, the introduction of excessively long auxiliary sequences can lead to redundant guidance regularization, thereby diminishing the overall accuracy.

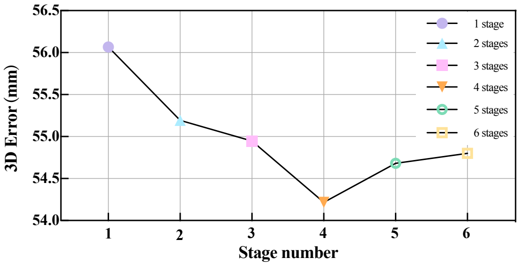

Fig. 5. Comparison of performance gains with different stage number on H3.6M.

c) The number of stages: The ablation study on the number of stages of baseline is conducted showing as Fig. 5.

Number of stages T is varied from 1 to 6, and the best performance gain is obtained when T is 4.

V. Conclusion

To sum up, a novel motion prediction framework with guidance regularization regulation is presented in this work. A knowledge distilling design is proposed where an FE-Net is designed as a teacher. During training, the guidance regularization information is extracted from FE-Net and then translated through intermediate supervision to improve the multi-stage prediction network DSP-Net. The proposed method has been evaluated on benchmark datasets H3.6M, 3DPW and CMU Mocap to evaluate the prediction performance. The results have shown that the proposed method outperforms state-of-art methods in both short-term and long-term prediction jobs. Extensive ablation studies have been carried out to present fruitful insights into the field. We believe this new perspective of guidance regularization could inspire other researchers and facilitate the prediction related work both in academics and industries in the future.

References

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

[2] H. S. Koppula and A. Saxena, "Anticipating human activities for reactive robotic response.," in *IROS*, p. 2071, Tokyo, 2013.

[3] A. M. Lehrmann, P. V. Gehler, and S. Nowozin, "Efficient nonlinear markov models for human motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1314–1321, 2014.

[4] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2007.

[5] G. W. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," *Advances in neural information processing systems*, vol. 19, 2006.

[6] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the IEEE international conference on computer vision*, pp. 4346–4354, 2015.

[7] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

[8] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9489–9497, 2019.

[9] P. Su, Z. Liu, S. Wu, L. Zhu, Y. Yin, and X. Shen, "Motion prediction via joint dependency modeling in phase space," in *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 713–721, 2021.

[10] Z. Liu, P. Su, S. Wu, X. Shen, H. Chen, Y. Hao, and M. Wang, "Motion prediction using trajectory cues," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13299–13308, 2021.

[11] L. Dang, Y. Nie, C. Long, Q. Zhang, and G. Li, "Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11467–11476, 2021.

[12] C. Zhong, L. Hu, Z. Zhang, Y. Ye, and S. Xia, "Spatio-temporal gating-adjacency gcn for human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6447–6456, 2022.

[13] M. Li, S. Chen, Z. Zhang, L. Xie, Q. Tian, and Y. Zhang, "Skeleton-parted graph scattering networks for 3d human motion prediction," *arXiv preprint arXiv:2208.00368*, 2022.

[14] T. Ma, Y. Nie, C. Long, Q. Zhang, and G. Li, "Progressively generating better initial guesses towards next stages for high-quality human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6437–6446, 2022.

[15] X. Sun, Q. Cui, H. Sun, B. Li, W. Li, and J. Lu, "Overlooked poses actually make sense: Distilling privileged knowledge for human motion prediction," *arXiv preprint arXiv:2208.01302*, 2022.

[16] H. Zhou, C. Guo, H. Zhang, and Y. Wang, "Learning multiscale correlations for human motion prediction," in *2021 IEEE International Conference on Development and Learning (ICDL)*, pp. 1–7, IEEE, 2021.

[17] Y. Cai, L. Huang, Y. Wang, T.-J. Cham, J. Cai, J. Yuan, J. Liu, X. Yang, Y. Zhu, X. Shen, *et al.*, "Learning progressive joint propagation for human motion prediction," in *European Conference on Computer Vision*, pp. 226–242, Springer, 2020.

[18] W. Mao, M. Liu, and M. Salzmann, "History repeats itself: Human motion prediction via motion attention," in *European Conference on Computer Vision*, pp. 474–489, Springer, 2020.

[19] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6158–6166, 2017.

[20] D. Pavllo, D. Grangier, and M. Auli, "Quaternet: A quaternion-based recurrent model for human motion," *arXiv preprint arXiv:1805.06485*, 2018.

[21] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5226–5234, 2018.

[22] Y. Li, Z. Wang, X. Yang, M. Wang, S. I. Poiana, E. Chaudhry, and J. Zhang, "Efficient convolutional hierarchical autoencoder for human motion prediction," *The Visual Computer*, vol. 35, no. 6, pp. 1143–1156, 2019.

[23] X. Liu, J. Yin, J. Liu, P. Ding, J. Liu, and H. Liu, "Trajectorycnn: a new spatio-temporal feature learning network for human motion prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 6, pp. 2133–2146, 2020.

[24] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 214–223, 2020.

[25] T. Sofianos, A. Sampieri, L. Franco, and F. Galasso, "Space-time-separable graph convolutional network for pose forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11209–11218, 2021.

[26] X. Gao, S. Du, Y. Wu, and Y. Yang, "Decompose more and aggregate better: Two closer looks at frequency representation learning for human motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6451–6460, 2023.

[27] X. Sun, H. Sun, B. Li, D. Wei, W. Li, and J. Lu, "Defeenet: Consecutive 3d human motion prediction with deviation feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5527–5536, 2023.

[28] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges, "A spatio-temporal transformer for 3d human motion prediction," in *2021 International Conference on 3D Vision (3DV)*, pp. 565–574, IEEE, 2021.

[29] C. Xu, R. T. Tan, Y. Tan, S. Chen, X. Wang, and Y. Wang, "Auxiliary tasks benefit 3d skeleton-based human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9509–9520, 2023.

[30] L.-H. Chen, J. Zhang, Y. Li, Y. Pang, X. Xia, and T. Liu, "Humanmac: Masked motion completion for human motion prediction," *arXiv preprint arXiv:2302.03665*, 2023.

[31] G. Barquero, S. Escalera, and C. Palmero, "Belfusion: Latent diffusion for behavior-driven human motion prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2317–2327, 2023.

[32] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer, and V. Belagiannis, "Motionmixer: Mlp-based 3d human body pose forecasting," *arXiv preprint arXiv:2207.00499*, 2022.

[33] W. Guo, Y. Du, X. Shen, V. Lepetit, X. Alameda-Pineda, and F. Moreno-Noguer, "Back to mlp: A simple baseline for human motion prediction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4809–4819, 2023.

[34] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2891–2900, 2017.