2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Directed Acyclic Graph Neural Network for Human Motion Prediction

Qin Li[1], Georgia Chalvatzaki[2], Jan Peters[2] and Yong Wang[1,*]

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561540

***Abstract*— Human motion prediction is essential in human-robot interaction. Current research mostly considers the joint dependencies but ignores the bone dependencies and their relationship in the human skeleton, thus limiting the prediction accuracy. To address this issue, we represent the human skeleton as a directed acyclic graph with joints as vertexes and bones as directed edges. Then, we propose a novel directed acyclic graph neural network (DA-GNN) that follows the encoder-decoder structure. The encoder is stacked by multiple encoder blocks, each of which includes a directed acyclic graph computational operator (DA-GCO) to update joint and bone attributes based on the relationship between joint and bone dependencies in the observed human states, and a temporal update operator (TUO) to update the temporal dynamics of joints and bones in the same observation. After progressively implementing the above update process, the encoder outputs the final update result, fed into the decoder. The decoder includes a directed acyclic graph-based gated recurrent unit (DAG-GRU) and a multi-layered perceptron (MLP) to predict future human states sequentially. To the best of our knowledge, this is the first time to introduce the relationship between bone and joint dependencies in human motion prediction. Our experimental evaluations on two datasets, CMU Mocap and Human 3.6m, prove that DA-GNN outperforms current models. Finally, we showcase the efficacy of DA-GNN in a realistic HRI scenario.**

I. Introduction

Human motion prediction emerges as important research for artificial intelligence applications, like human-robot interaction (HRI) [1]–[3]. For a robot to act proactively, it should plan strategies based on effective human motion prediction [1], [2]. This research considers the challenges of forecasting human states (usually represented as skeletons) over a long time horizon, given some motion history observed by ambient sensors (e.g., cameras, lasers, optical capture devices). Due to the high-dimensionality of human states, research in HRI mainly focuses on predicting partial states, either predicting the human as a point-mass [1], [3] or predicting states of parts of the human body [4]–[6]. Whole-body motion prediction (e.g., preserving the whole human skeleton during prediction) is still challenging [7], and this is the focus of this paper. Consider a scenario where a human approaches a robot assistant while extending his arm to hand-over an object. In such a case, the whole-body motion prediction is important for efficient robot motion planning [8].

[1]School of Automation, Central South University, Changsha 410083, China (email: liqin6@csu.edu.cn, ywang@csu.edu.cn).
[2]Intelligent Autonomous Systems, Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany georgia@robot-learning.de
The work of Chalvatzaki and Peters has been supported by the projects RoboTrust and Skills4Robots.
*Corresponding author: Yong Wang.

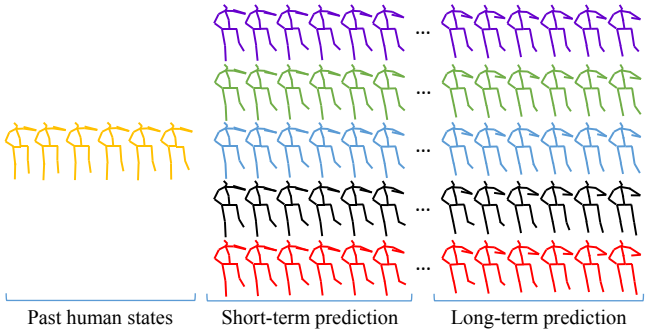

Fig. 1. **Example results.** [left] observed human states (in orange). The middle and right frames are the prediction results of [9] (in purple), [10] (in green), [11] (in blue), our DA-GNN (in black) and the ground truth (in red). As we can visually inspect, DA-GNN performs the best.

Early works designed models for whole-body motion prediction based on machine learning models [12]–[14]. These models have proven to be effective only in predicting simple and periodic motions, like *walking* [15]. Deep learning-based models have recently achieved impressive results for predicting complex motions [9]–[11], [15]–[26]. The main developments were based on recurrent neural networks (RNNs) [9], [16], [17] for modeling temporal dynamics of the observed human motion, and convolutional neural networks (CNNs) [22]–[24] for considering the joint attributes (i.e., 3D Cartesian positions or angular values) in the observed human states. However, those methods rarely investigate the skeletal representation and evolution during human motion, hence missing crucial information [10].

Researchers designed models based on graph neural networks (GNNs) to accommodate this issue [10], [11], [15], [25], [26]. They consider the human skeleton as an undirected skeleton graph, whose vertexes are joints and undirected edges are bones. However, these GNNs only treat a bone as a simple connection between two joints, neglecting its significance for the skeletal representation. Arguably, each bone has each own length and direction (bone attributes). The bone attributes are equally crucial as joint attributes since humans naturally recognize or predict motions by observing the joint positions and the bone movements in the skeleton. Moreover, bone dependencies can be captured by considering the skeletal kinematic chain. Joint and bone dependencies are complementary to each other [27]. Only by combining those can the skeletal representations of the observed human states be fully characterized, potentially leading to further improvement of human motion prediction. Exploring the relationship between bone and joint dependencies remains still an open research problem.

In this paper, we address this problem by considering the human skeleton as a directed acyclic graph (DAG) (Fig. 2),

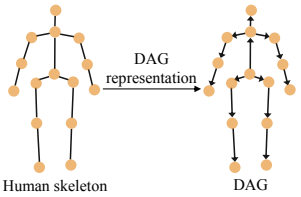

Fig. 2. Illustration of the DAG of the human skeleton.

with joints as vertexes and bones as directed edges, through which we obtain the distance and direction of bones. To learn the skeletal representation and evolution of the DAG, a GNN is required. Note that some GNNs can learn the dynamics of physical systems represented as DAGs [28]–[30]. However, these physical systems always have rich information, as they are driven by measurable external forces, i.e., a very different case from the human skeleton motion; hence these GNNs do not apply to human motion prediction.

We propose a novel GNN named DA-GNN that follows the encoder-decoder structure. The encoder contains stacked encoder blocks (EN-blocks), including a DA-GCO and a TUO. Inside an EN-block, the DA-GCO captures the relationship between joint and bone dependencies to update their respective attributes in a window of observations; the TUO updates the temporal dynamics of joints and bones in the same time-frame. All EN-blocks progressively implement the update process to finally output a DAG with updated joints and bones' attributes and their respective updated temporal dynamics. Then, this DAG serves as an initial hidden state for the decoder to predict future human states by sequentially employing a novel DAG-GRU followed by an MLP. The DAG-GRU updates a given hidden state, and the MLP predicts the next human state based on the updated hidden state. To the best of our knowledge, our work is the first to consider as additional features the relationship between bone and joint dependencies for human motion prediction. We conducted experiments on two human motion datasets: CMU Mocap and Human 3.6m. We show that DA-GNN outperforms the state-of-the-art (SoTA) models in predicting complex human motions over different time-scales. Moreover, we conducted an HRI experiment to showcase DA-GNN's benefit in human intention prediction in a realistic HRI scenario. Our code is available at `https://github.com/Qinli-zz/DA-GNN`.

II. Related Work

Human Motion Modeling and Prediction. Much research on the topic focuses on modeling and predicting a human's motion as a point-mass [1], [3], [31]. A SocialLSTM was designed in [3] that jointly predicts people's motions and paths in a crowded scene based on the social conventions of humans when navigating in shared scenes. An RNN-based generator was proposed in [31] for generating multiple socially-acceptable human motion trajectories given an observed past followed by an RNN-based discriminator for evaluating them. However, an important line of works aims at modeling and predicting part-body or whole-body human motion. Early works in this direction employed traditional machine learning models, like anHMM [12], [32] to design a style machine for synthesizing future human states given an observed sequence. A Gaussian process dynamic model was proposed by [13] for extracting nonlinear time-varying features. Recently, deep learning has fuelled research on human motion prediction. RNNs are used for modeling temporal dynamics of the observed human states [9], [16], [17]. A scalable model for casting an arbitrary spatio-temporal graph as a fully differentiable and jointly trainable RNN mixture was proposed by [16], while an RNN encoder-recursive decoder [9] combined representation learning with temporal dynamic modeling for feature extraction and prediction [33].

Other researchers consider the spatio-temporal dependencies in human states and designed models based on richer deep learning architectures [10], [11], [15], [22]–[26]. A novel approach to human state modeling based on CNNs was introduced in [22], where a convolutional long-term encoder encodes the observed states into a long-term hidden variable, and then a convolutional decoder with a small encoder-decoder structure predicts future states. A novel GNN model called symbiotic graph neural network (Sym-GNN) was proposed by [10]. The authors designed joint-scale graph convolution and part-scale graph convolution operators to extract multi-scale features of the observed human states. A dynamic multi-scale graph neural network (DMGNN) was presented in [25]. The core idea of DMGNN is to use a multi-scale graph to comprehensively model a human body's internal relations for motion feature learning.

Human Motion Modeling in HRI. Human motion prediction as an instance of intention inference is essential for robot planning in HRI and collaboration (HRC) settings [34]. Probabilistic methods [35]–[37] and filtering [8], [38] have been used extensively, however not being able to predict for long time-horizons. Recent approaches rely on deep learning architectures to predict human motion in interactive navigation scenarios like [1], [3], where humans are considered point masses. However, for HRI, more detailed information on the human motion is needed. LSTMs were used in [5] for hand motion prediction, and in [6] for upper-body motion generation and prediction. [4] proposed an integrated HRC framework, including both plan recognition and trajectory prediction based on part-body human motion prediction for the generation of safe and efficient robotic motions. In this work, we go one step further than related work and propose a novel graph-based model DA-GNN. In DA-GNN, the encoder can effectively characterize the human body, learning attributes and temporal dynamics of both joints and bones, along with the relationship between joint and bone dependencies. DA-GNN achieves SoTA performance on two challenging human motion datasets on various time-horizons, while DAGNN's benefit is also shown in better intention prediction when applied in HRI.

III. Preliminaries

A. Graph Neural Networks

GNNs are deep learning architectures that operate on graph analysis [39]. A graph is a unique non-Euclidean data structure that models a set of objects (vertexes) and their

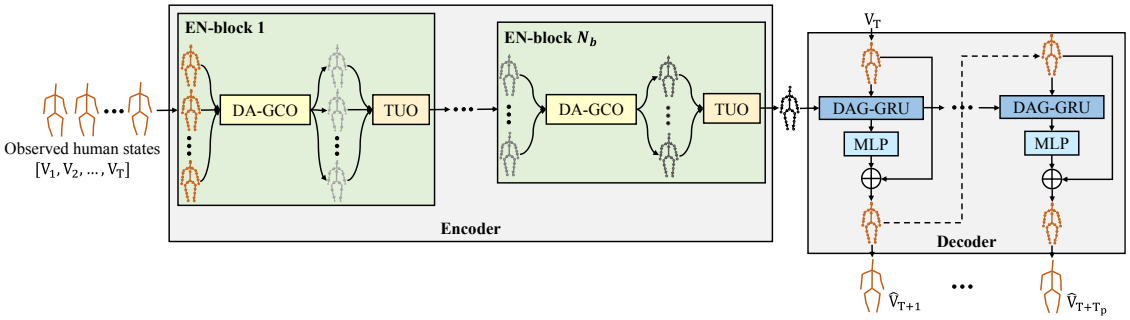

Fig. 3. The architecture of the proposed DA-GNN. Given the observed human states $[\mathrm{V}_1, \mathrm{V}_2, \mathrm{V}_3, \cdots, \mathrm{V}_T]$, we create DAGs of these human states and feed them into stacked EN-blocks, each of which includes a DA-GCO and a TUO, aiming to update the attributes of bones and joints along with their respective temporal dynamics in the observed human states. In this way, the encoder outputs the final update result represented as a DAG. The obtained DAG is then fed into a decoder, which is composed of a DA-GRU and a MLP, sequentially predicting future human states.

relationships (edges). Due to the convincing performance and high interpretability, GNNs received increased attention in various research areas [28], [40], [41]. For example, an interaction network based on GNNs was proposed in [28], which can reason on how objects interact in a complex system. In [41], a GNN-based supervised neural relational model that can infer the interactions and learn the dynamics from observation data on physical simulations was proposed.

B. Directed Acyclic Graphs

A DAG is a directed graph with no directed cycles. As shown in Fig. 2, it consists of vertices and edges, with each edge directed from one vertex to another. DAGs have the potential to solve the over-smoothing problem in GNNs. Over-smoothing indicates that as the number of layers increases, the performance of GNNs does not improve anymore, and the extracted node features become similar [42]. This problem mainly occurs when GNNs deal with undirected graphs. Unlike undirected graphs, DAGs have directional edges that represent unidirectional propagations of information between connected joints, making GNNs to treat connected nodes differently, thus reducing the similarity of their features.

IV. Directed Acyclic Graph Neural Network

Fig. 3 shows an illustration of DA-GNN. In what follows, we provide a comprehensive problem definition and analyze the components of DA-GNN.

A. Problem Definition

Let $\mathbf{V}_{1:T}$ be the observed human states, i.e., $\mathbf{V}_{1:T} = [\mathrm{V}_1, \mathrm{V}_2, \mathrm{V}_3, \cdots, \mathrm{V}_T] \in \mathbb{R}^{T\times M\times D}$. T is the temporal dimension of $\mathbf{V}_{1:T}$, i.e., the number of the observed human states. $\mathrm{V}_t \in \mathbb{R}^{M\times D}$ is the t^{th} observed human state, where M is the number of joints and D is the dimension of joint attributes. Our goal is to predict the future T_p (the forecast horizon) human states based on the observed $\mathbf{V}_{1:T}$.

We consider a human state as a DAG, where the joints and bones are the vertexes and directed edges, respectively. Taking V_t as an example, we denote it as $\mathcal{G}_h^t = (\mathcal{V}_h^t, \mathcal{E}_h^t)$, where $\mathcal{V}_h^t$ is a set of joints and $\mathcal{E}_h^t$ is a set of bones. To implement subsequent temporal computations, we use two matrices to indicate $\mathcal{V}_h^t$ and $\mathcal{E}_h^t$, respectively. Obviously, $\mathcal{V}_h^t$ equals to $\mathrm{V}_t \in \mathbb{R}^{M\times D}$; while the $\mathcal{E}_h^t$ need to be computed based on V_t. To do so, we consider each bone to be a vector, directing from a source joint to a target joint in V_t, and calculate the subtraction of these two joints' attributes to acquire the bone attributes. The subtraction order is from the centrifugal joint to the centripetal joint, following the same formulation as in [10]. Based on the above process, the attributes of all bones in V_t are obtained and generate $\mathcal{E}_h^t \in \mathbb{R}^{N\times D}$, where N is the number of bones. Given the above formalization, the resulting DAGs that represent all human states in $\mathbf{V}_{1:T}$ are $\{\mathcal{G}_h^1, \mathcal{G}_h^2, \cdots, \mathcal{G}_h^T\}$ respectively.

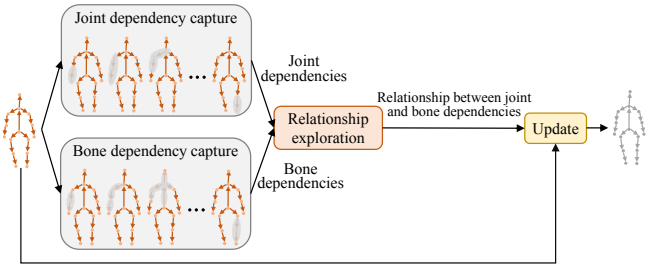

Fig. 4. Illustration of the update process of DA-GCO.

B. Encoder

The encoder comprises stacked EN-blocks, each of which updates the attributes and temporal dynamics of joints and bones in the observed human states. Each EN-block is composed of a DA-GCO and a TUO. Let us consider the first EN-block; its input is all observed human states represented as DAGs, i.e., $\{\mathcal{G}_h^t\}_{t=1:T}$. First, for each human state, the DA-GCO captures the joint and bone dependencies, and explores their relationship to update joint and bone attributes (see Fig. 4). Taking $\mathcal{G}_h^t = (\mathcal{V}_h^t, \mathcal{E}_h^t)$ as an example, the above process is as follows:

(1) Capture all joint dependencies in $\mathcal{V}_h^t$ using a graph convolutional layer: $C_\mathcal{V} = \sigma((A_\mathcal{V} \odot B_\mathcal{V})\mathcal{V}_h^t W_\mathcal{V})$, where $A_\mathcal{V} \in \{0,1\}^{M\times M}$ is the joints' adjacency matrix. $A_\mathcal{V}(i,j) = 1$ when the i^{th} and the j^{th} joints are connected. $\odot$ is the element-wise multiplication, σ is a sigmoid activation function, $B_\mathcal{V} \in \mathbb{R}^{M\times M}$ is the trainable importance matrix of $A_\mathcal{V}$, $W_\mathcal{V} \in \mathbb{R}^{D\times D}$ is a trainable parameter matrix, and $C_\mathcal{V} \in \mathbb{R}^{M\times D}$ contains the dependencies of all joints.

(2) Capture all bone dependencies in $\mathcal{E}_h^t$ using a second graph convolutional layer: $C_\mathcal{E} = \sigma((A_\mathcal{E} \odot B_\mathcal{E})\mathcal{E}_h^t W_\mathcal{E})$, where $A_\mathcal{E} \in \{0,1\}^{N\times N}$ is the bones' adjacency matrix. $A_\mathcal{E}(i,j) = 1$ when the i^{th} and the j^{th} bones are connected. $B_\mathcal{E} \in \mathbb{R}^{N\times N}$ is the trainable importance matrix of $A_\mathcal{E}$ and $W_\mathcal{E} \in \mathbb{R}^{D\times D}$

is a trainable parameter matrix. $C_{\mathcal{E}} \in \mathbb{R}^{N\times D}$ contains the dependencies of all bones.

(3) Explore the relationship between bone and joint dependencies, i.e., relationship exploration (RE). First, for each joint, we aggregate the dependencies of the bones which regard the joint as the target joint: $R^{\mathcal{T}}_{\mathcal{V}} = (A_{\mathcal{T}} \odot B_{\mathcal{T}})C_{\mathcal{E}}$, where $A_{\mathcal{T}} \in \{0,1\}^{M\times N}$ is a target adjacency matrix. $A_{\mathcal{T}}(i,j) = 1$ when the i^{th} joint is the target joint of the j^{th} bone. $B_{\mathcal{T}} \in \mathbb{R}^{M\times N}$ is the trainable importance matrix of $A_{\mathcal{T}}$. Then, for each joint, we aggregate the dependencies of the bones which regard the joint as the source joint: $R^{\mathcal{S}}_{\mathcal{V}} = (A_{\mathcal{S}} \odot B_{\mathcal{S}})C_{\mathcal{E}}$, where $A_{\mathcal{S}} \in \{0,1\}^{M\times N}$ is a source adjacency matrix. $A_{\mathcal{S}}(i,j) = 1$ when the i^{th} joint is the source joint of the j^{th} bone. $B_{\mathcal{S}} \in \mathbb{R}^{M\times N}$ is the trainable importance matrix of $A_{\mathcal{S}}$. Finally, for each bone, we aggregate the dependencies of its target joints and source joints respectively: $R^{\mathcal{T}}_{\mathcal{E}} = (A_{\mathcal{T}} \odot B_{\mathcal{T}})^{\top} C_{\mathcal{V}}$, $R^{\mathcal{S}}_{\mathcal{E}} = (A_{\mathcal{S}} \odot B_{\mathcal{S}})^{\top} C_{\mathcal{V}}$.

(4) Update $\mathcal{V}^t_h$ and $\mathcal{E}^t_h$ based on the obtained aggregation results: $\mathcal{V}^{t,up}_h = \mathrm{FC}^1_{D_{o,1}}(\mathcal{V}^t_h, R^{\mathcal{T}}_{\mathcal{V}}, R^{\mathcal{S}}_{\mathcal{V}})$, $\mathcal{E}^{t,up}_h = \mathrm{FC}^2_{D_{o,1}}(\mathcal{E}^t_h, R^{\mathcal{T}}_{\mathcal{E}}, R^{\mathcal{S}}_{\mathcal{E}})$. $\mathrm{FC}^1_{D_{o,1}}$ and $\mathrm{FC}^2_{D_{o,1}}$ are fully-connected layers with output dimension $D_{o,1}$. $\mathcal{V}^{t,up}_h \in \mathbb{R}^{M\times D_{o,1}}$ and $\mathcal{E}^{t,up}_h \in \mathbb{R}^{N\times D_{o,1}}$ constitute a DAG, i.e., $\mathcal{G}^{t,up}_h = (\mathcal{V}^{t,up}_h, \mathcal{E}^{t,up}_h)$. Following the above process across all observed human states, we obtain $\{\mathcal{G}^{t,up}_h\}_{t=1:T}$.

Continuing, the TUO updates the temporal dynamics of $\{\mathcal{G}^{t,up}_h\}_{t=1:T}$ as follows:

(1) Group joints and bones in all human states respectively, obtaining a joint sequence $\mathbf{S}_{\mathcal{V}} = \{\mathcal{V}^{t,up}_h\}_{t=1:T} \in \mathbb{R}^{T\times M\times D_{o,1}}$ and a bone sequence $\mathbf{S}_{\mathcal{E}} = \{\mathcal{E}^{t,up}_h\}_{t=1:T} \in \mathbb{R}^{T\times N\times D_{o,1}}$.

(2) Update the temporal dynamics of $\mathbf{S}_{\mathcal{V}}$ and $\mathbf{S}_{\mathcal{E}}$ by using two 2D convolutional layers: $\mathbf{S}^{up}_{\mathcal{V}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv2d}_1(\mathbf{S}_{\mathcal{V}})))$, $\mathbf{S}^{up}_{\mathcal{E}} = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv2d}_2(\mathbf{S}_{\mathcal{E}})))$, In these two layers, the values of the kernel size and the stride are k and s respectively along the temporal dimension, and are 1 along the spatial dimension. BN is a batch normalization layer and ReLU is a rectified linear unit, normalizing and nonlinearizing the convolutions respectively. $\mathbf{S}^{up}_{\mathcal{V}} = \{\hat{\mathcal{V}}^{t,up}_h\}_{t=1:\lceil T/s\rceil} \in \mathbb{R}^{\lceil T/s\rceil\times M\times D_{o,1}}$ and $\mathbf{S}^{up}_{\mathcal{E}} = \{\hat{\mathcal{E}}^{t,up}_h\}_{t=1:\lceil T/s\rceil} \in \mathbb{R}^{\lceil T/s\rceil\times N\times D_{o,1}}$ are sequences with updated temporal dynamics, where $\lceil * \rceil$ is the round up symbol.

(3) Generate the final updated DAGs based on $\mathbf{S}^{up}_{\mathcal{V}}$ and $\mathbf{S}^{up}_{\mathcal{E}}$, obtaining $\{\hat{\mathcal{G}}^{t,up}_h\}_{t=1:\lceil T/s\rceil}$, where $\hat{\mathcal{G}}^{t,up}_h = (\hat{\mathcal{V}}^{t,up}_h, \hat{\mathcal{E}}^{t,up}_h)$. These updated DAGs are fed into the next EN-block for further update.

Assuming the number of EN-blocks is N_b, all EN-blocks progressively implement the update process and finally output a final DAG $\hat{\mathcal{G}}^{up,N_b}_h = (\hat{\mathcal{V}}^{up,N_b}_h, \hat{\mathcal{E}}^{up,N_b}_h)$, where $\hat{\mathcal{V}}^{up,N_b}_h \in \mathbb{R}^{M\times D_{o,N_b}}$ and $\hat{\mathcal{E}}^{up,N_b}_h \in \mathbb{R}^{N\times D_{o,N_b}}$. D_{o,N_b} is the output dimension of fully-connected layers in the N^{th}_b EN-block[1]. $\hat{\mathcal{G}}^{up,N_b}_h$ is regarded as composed of joints and edges with final updated attributes and temporal dynamics.

[1]In the rest of this paper, we will simplify the output dimension of fully-connected layers in an EN block to the output dimension of the EN block.

C. Decoder

The decoder aims to predict future T_{p} human states based on $\hat{\mathcal{G}}^{up,N_b}_h$. To do so, we first design a variant of the gated recurrent unit [43], i.e., DAG-GRU, which receives a DAG as a hidden state, and updates the hidden state based on a given human state also represented as a DAG. Let $\mathcal{G}^t_s = (\mathcal{V}^t_s, \mathcal{E}^t_s)$ be the hidden state and $\mathcal{G}^t_h$ be the given human state, the process of DAG-GRU is as follows:

$$
\begin{aligned}
r &= \sigma(r_{\mathrm{in}}(\mathcal{G}^t_h) + r_{hid}(\text{DA-GCO}(\mathcal{G}^t_s))),\\
u &= \sigma(u_{\mathrm{in}}(\mathcal{G}^t_h) + u_{hid}(\text{DA-GCO}(\mathcal{G}^t_s))),\\
\mathcal{G}_r &= \tanh(c_{\mathrm{in}}(\mathcal{G}^t_h) + \mathrm{r} \odot c_{hid}(\text{DA-GCO}(\mathcal{G}^t_s))),\\
\mathcal{G}^{t+1}_s &= u \odot \mathcal{G}^t_s + (1-u) \odot \mathcal{G}_r
\end{aligned}
$$

where $r_{\mathrm{in}}(\cdot)$, $r_{\mathrm{hid}}(\cdot)$, $u_{\mathrm{in}}(\cdot)$, $u_{\mathrm{hid}}(\cdot)$, $c_{\mathrm{in}}(\cdot)$, and $c_{\mathrm{hid}}(\cdot)$ are trainable linear mappings. DA-GCO is used to refine $\mathcal{G}^t_s$ for state propagation enhancement (SPE). r and u are the reset and update signals, which determine the degree of reset and update of $\mathcal{G}^t_s$, respectively. $\mathcal{G}_r$ is the hidden state after being reset. $\mathcal{G}^{t+1}_s$ is the hidden state updated by u and $\mathcal{G}_r$.

Having designed DAG-GRU, we combine it with an MLP to design a recurrent connection to sequentially predict future T_p human states. In detail, regarding $\hat{\mathcal{G}}^{up,N_b}_h$ as an initial hidden state (i.e., $\mathcal{G}^0_s$), DAG-GRU updates $\mathcal{G}^0_s$ based on $\mathcal{G}^T_h$ and obtains $\mathcal{G}^1_s = (\mathcal{V}^1_s, \mathcal{E}^1_s)$. Then, $\mathcal{G}^1_s$ is fed into the MLP to predict $\mathcal{G}^{T+1}_h$, while a residual structure (RS) is used to stabilize the prediction: $\mathcal{G}^{T+1}_h = \mathcal{G}^T_h + \mathrm{MLP}(\mathcal{G}^1_s)$. $\mathcal{G}^{T+1}_h = (\mathcal{V}^{T+1}_h, \mathcal{E}^{T+1}_h)$, where $\mathcal{V}^{T+1}_h$ indicates the $T+1^{th}$ human state, i.e., $\hat{\mathrm{V}}_{T+1}$. After, DAG-GRU and MLP predict the $T+2^{th}$ human state $\hat{\mathrm{V}}_{T+2}$ based on $\mathcal{G}^1_s$ and $\mathcal{G}^{T+1}_h$. By iterating the above process, we can predict T_p human states, i.e., $\hat{\mathrm{V}}_{T+1}, \hat{\mathrm{V}}_{T+2}, \hat{\mathrm{V}}_{T+3}, \cdots, \hat{\mathrm{V}}_{T+T_p}$.

D. Training

We use Mini-batch Gradient Descent [44] to train DA-GNN, which iteratively updates model parameters based on the loss of a batch of training samples. A training sample consists of T past human states, and T_{p} future human states used as the ground truth, which are randomly intercepted from a human motion sequence in a training set. The loss of a batch of training samples is calculated by: $L = \frac{1}{N_{bt}} \sum_{n=1}^{N_{bt}} \left\| \mathbf{V}^n_{1:T_p} - \hat{\mathbf{V}}^n_{1:T_p} \right\|_1$ where N_{bt} is the batch size. $\hat{\mathbf{V}}^n_{1:T_p} \in \mathbb{R}^{T_p\times M\times D}$ and $\mathbf{V}^n_{1:T_p} \in \mathbb{R}^{T_p\times M\times D}$ are the prediction results and the ground truth corresponding to the n^{th} training sample, respectively.

V. Experiments

A. Datasets

CMU Mocap[2]: is a public human motion dataset. Here, each human motion sequence is obtained by capturing an actor's motion over a long time through an optical tracking system. A human state is composed of 3D-positions of 38 joints in the angular coordinate system. We transformed the 3D-positions into quaternion exponential maps according to [45]. There are 23 motion categories involved in this dataset. Following [22], we chose eight representative categories. Each category contains six motion sequences, where five

[2]Available at http://mocap.cs.cmu.edu/

TABLE I
ABLATION MAEs OF DA-GNN ON CMU MOCAP.

Components			Basketball	Basketball Signal	Directing Traffic	Jumping
RE	SPE	RS				
	✓	✓	1.59	1.09	1.92	1.99
✓		✓	1.46	0.93	1.85	1.83
✓	✓		1.55	1.15	1.99	2.00
✓	✓	✓	**1.44**	**0.90**	**1.80**	**1.80**
Components			Running	Soccer	Walking	Washing Window
RE	SPE	RS				
	✓	✓	0.75	1.30	0.57	1.10
✓		✓	0.63	1.32	**0.48**	1.04
✓	✓		0.71	1.34	0.59	1.15
✓	✓	✓	**0.62**	**1.29**	**0.48**	**0.99**

sequences were for training and the rest was for testing. The process of generating test samples is the same as generating training samples, as described in Sec. IV-D.

Human 3.6M (H3.6M) [46]: is a public large-scale human motion dataset. The capture process is the same as that of CMU Mocap. The number of joints within a human state is 32. The 3D-position of a joint is also in the angular coordinate system. We also transformed the 3D-positions into exponential maps. The number of categories is 15. There are 14 sequences for each category, from which we selected 12 sequences for training and the remaining one for testing.

B. Settings

We conducted experiments on one Nvidia RTX-2080 GPU. Following current works, we set T to 50 and T_p to 25. For the encoder, N_b was set to $= 6$. Each EN-block's output dimension was 32, 64, 128, 128, 256, and 256, respectively. k and s in each EN-block were 5 and 2. For the decoder, the output dimension of all trainable linear mappings and the fully-connected layers in DA-GCO was 256. MLP had three linear layers, and the output dimension of each layer was 256, 256, and 3. For training, we used Adam optimizer [47] with a learning rate 10^{-4}. Batch size and the number of training iterations were set to 32 and 10^4, respectively.

C. Influence of Key Components

To provide a deep understanding of DA-GNN, we evaluated the influence of its several components, i.e., RE in DA-GCO, SPE in DA-GRU, and RS in the decoder. We used the mean angle error (MAE) as the evaluation metrics, which indicates the mean of the Euclidean distances between the predicted human states and the ground truth in all test samples, corresponding to a motion category. Note that the average MAE is the average value of MAEs corresponding to all motion categories. Table I shows the MAEs for different design choices (with or without the above three components) in the architecture of DA-GNN on CMU Mocap. As we can see, when removing any component, the corresponding MAEs for all motion categories increase to some extent. This result effectively validates the positive influence of the ablated components on the prediction accuracy.

D. Comparative Evaluation

We compared DA-GNN against the SoTA models on both CMU Mocap and H3.6M. Since the exact train/test splits were randomly intercepted in human motion sequences, we re-evaluated the prediction results of SoTA models based on our splits for a fair comparison. The training settings are in V-B. Table II shows the MAEs of DA-GNN and SoTA models at 80ms, 160ms, 320ms, 400ms, and 1000ms on CMU Mocap. We can see that DA-GNN beats the SoTA models in most cases. Moreover, DA-GNN obtains the lowest MAEs at 1000ms for all motion categories, demonstrating the advantage of DA-GNN in long-term human motion prediction. Table III presents the MAEs of DA-GNN and SoTA models at 80ms, 160ms, 320ms, 400ms, and 1000ms on H3.6M. Once more, DA-GNN performs better than SoTA models overall. Note that Sym-GNN also considers both joint and bone attributes for human motion prediction. However, Sym-GNN does not explore the relationship between bone and joint dependencies. Meanwhile, Sym-GNN uses a normal GRU for prediction, while DA-GNN uses the proposed DAG-GRU with its unique SPE process. Thus, DA-GNN performs better than Sym-GNN.

E. HRI Experiment

To evaluate the efficacy of DA-GNN in HRI, we designed a social HRI task in a realistic scenario, during which a robot predicts a user's intention based on the user's partial motion and responds before the motion is completed. For such a task, human intention prediction is fundamental. Therefore, we designed an intention prediction method. In this method, a prediction model (i.e., DA-GNN) predicts the future motion based on the user's partial motion. A recognition model recognizes the user's intention using the combined partial observed motion and the predicted future motion. The recognition model is simply the combination of our encoder with a softmax layer [48]. To train these two models, we introduced an HRI-dedicated dataset AIR-Act2Act [49], which involves 10 interaction categories. Each category has 1500 human motion sequences obtained by three Kinect v2 cameras at 30 fps. We selected the sequences from two cameras for training and those from the other camera for testing. Each sequence includes 150 frames. For DA-GNN, T and T_p were set to 90 and 60, respectively. Eight EN-blocks were stacked in the encoder, and the output dimension of these EN-blocks was $\{32, 32, 64, 64, 128, 128, 256, 256\}$. The other settings were the same as Sec. V-B. For the recognition model, the encoder' settings were the same as above. To validate DA-GNN's advantage, we used Sym-GNN and LDR as the prediction models for comparison. Table IV shows the test accuracy comparisons of the intention prediction method designed based on these three models, respectively. It shows that the method based on DA-GNN obtains the best test accuracy, showing its advantage.

Furthermore, we embedded the intention prediction method based on DA-GNN into a realistic HRI scenario, including a user and a humanoid NAO robot. We used a Kinect v2 camera to record the user's behavior at 30 fps for three seconds. Meanwhile, we designed the joint angles to control Nao's response w.r.t. the user's behaviors. We conducted three interaction categories, i.e., *greeting*, *high-five* and *hit* in AIR-Act2Act. Taking *greeting* as an example (Fig. 5 [left]), we first assessed the motion prediction performance

TABLE II

MAEs OF DA-GNN AND SOTA MODELS AT 80MS, 160MS, 320MS, 400MS, AND 1000MS ON CMU MOCAP.

Motion Category	Basketball					Basketball Signal					Directing Traffic					Jumping				
time[ms]	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k
Res-sup [17]	0.49	0.77	1.26	1.45	1.77	0.42	0.76	1.33	1.54	2.17	0.31	0.58	0.94	1.10	2.06	0.57	0.86	1.76	2.03	2.42
CSM [22]	0.37	0.62	1.07	1.18	1.95	0.32	0.59	1.04	1.24	1.96	0.25	0.56	0.89	1.00	2.04	0.39	0.60	1.36	1.56	2.01
Sym-GNN [10]	0.33	0.48	0.91	1.06	1.47	0.12	0.21	0.38	0.49	0.94	0.20	0.41	0.75	0.87	1.84	0.32	0.55	1.40	1.60	1.82
Taj-GCN [15]	0.33	0.52	0.89	1.06	1.71	0.11	0.20	0.41	0.53	1.00	0.17	0.35	0.53	0.60	2.00	0.31	0.51	1.23	1.39	1.82
NAT [11]	0.34	0.49	0.89	1.02	1.49	0.15	0.24	0.48	0.61	0.92	0.20	0.41	0.65	0.77	1.83	0.38	0.56	1.29	1.45	1.81
LDR [26]	0.31	0.45	**0.76**	**0.89**	1.52	0.12	0.18	0.39	0.51	0.91	0.16	**0.30**	**0.45**	**0.58**	1.82	0.30	**0.41**	**0.91**	**1.17**	1.80
DA-GNN	**0.30**	**0.43**	0.88	1.02	**1.44**	**0.09**	**0.16**	**0.34**	**0.43**	**0.89**	**0.14**	0.37	0.71	0.83	**1.80**	**0.29**	0.51	1.38	1.57	**1.78**
Motion Category	Running					Soccer					Walking					Washing Window				
time[ms]	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k
Res-sup [17]	0.32	0.48	0.65	0.74	1.00	0.29	0.50	0.87	0.98	1.73	0.35	0.45	0.59	0.64	0.88	0.32	0.47	0.74	0.93	1.37
CSM [22]	0.28	0.41	0.52	0.57	0.67	0.26	0.44	0.75	0.87	1.56	0.35	0.44	0.45	0.50	0.78	0.30	0.47	0.80	1.01	1.39
Sym-GNN [10]	0.21	0.33	0.53	0.56	0.65	0.19	0.32	0.66	0.78	1.32	0.26	0.32	0.35	0.39	0.52	0.22	0.33	0.55	0.73	1.05
Traj-GCN [15]	0.33	0.55	0.73	0.74	0.95	0.18	0.30	0.61	0.71	1.40	0.33	0.45	0.49	0.53	0.61	0.22	0.33	0.57	0.75	1.20
NAT [11]	0.24	0.43	0.53	0.56	0.74	0.20	0.33	0.59	0.76	1.31	0.31	0.37	0.40	0.46	0.52	0.23	0.36	0.66	0.86	1.10
LDR [26]	0.31	0.39	0.53	0.68	0.87	0.17	0.29	**0.57**	**0.66**	1.29	0.29	0.41	0.54	0.61	0.64	0.20	0.31	0.54	0.70	1.04
DA-GNN	**0.18**	**0.30**	**0.51**	**0.54**	**0.62**	**0.16**	**0.28**	**0.57**	0.76	**1.27**	**0.23**	**0.28**	**0.33**	**0.36**	**0.48**	**0.18**	**0.30**	**0.52**	**0.68**	**0.99**

TABLE III

MAEs OF DA-GNN AND SOTA MODELS AT 80MS, 160MS, 320MS, 400MS, AND 1000MS ON H3.6M.

Motion Category	Walking					Eating					Smoking					Discussion					Waiting				
time[ms]	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k
Res-sup [17]	0.27	0.46	0.67	0.75	0.93	0.23	0.37	0.59	0.73	1.08	0.32	0.59	1.01	1.10	1.50	0.30	0.67	0.98	1.06	1.69	0.32	0.63	1.07	1.26	1.49
CSM [22]	0.33	0.54	0.68	0.73	0.86	0.22	0.36	0.58	0.71	1.24	0.26	0.49	0.96	0.92	1.62	0.32	0.67	0.94	1.01	1.86	0.30	0.62	1.09	1.30	1.40
Sym-GNN [10]	0.17	0.31	0.50	0.60	0.78	0.16	0.29	0.48	0.60	0.88	0.21	0.40	0.76	0.82	1.18	0.21	0.55	0.79	0.85	1.28	0.22	0.48	0.87	1.06	1.33
Traj-GCN [15]	0.18	0.31	0.49	0.58	0.69	0.16	0.29	0.50	0.62	1.12	0.22	0.41	0.86	0.81	1.57	0.20	0.55	0.79	0.85	1.70	0.23	0.50	0.91	1.29	1.33
NAT [11]	0.17	0.29	0.46	0.56	**0.58**	0.17	0.27	0.47	0.58	0.96	0.23	0.42	0.82	0.84	1.39	0.22	0.54	0.79	0.89	1.35	0.23	0.48	0.87	1.07	1.37
LDR [26]	0.16	0.29	0.47	0.58	0.73	0.16	0.27	0.49	0.64	0.97	0.22	0.40	0.79	0.82	**1.08**	0.19	**0.45**	0.78	0.83	**0.85**	0.21	0.48	0.87	1.15	1.34
DA-GNN	**0.15**	**0.27**	**0.44**	**0.55**	0.76	**0.13**	**0.26**	**0.45**	**0.57**	**0.85**	**0.20**	**0.38**	**0.75**	**0.80**	1.20	**0.18**	**0.53**	**0.77**	**0.84**	1.29	**0.19**	**0.45**	**0.85**	**1.02**	**1.30**
Motion Category	Directions					Greeting					Phoning					Posing					Walking Dog				
time[ms]	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k
Res-sup [17]	0.41	0.64	0.80	0.92	1.18	0.57	0.83	1.45	1.60	1.79	0.59	1.06	1.45	1.60	1.75	0.45	0.85	1.34	1.56	64	0.52	0.89	1.25	1.40	1.61
CSM [22]	0.39	0.60	0.80	0.91	0.99	0.51	0.82	1.21	1.38	1.45	0.59	1.31	1.51	1.65	1.77	0.29	0.60	1.21	1.37	1.45	0.59	1.00	1.32	1.44	1.58
Sym-GNN [10]	0.23	0.42	0.57	0.67	0.74	0.35	0.60	0.95	1.15	1.29	0.48	0.80	1.28	1.41	1.52	0.18	0.45	0.97	1.20	1.34	0.42	0.73	1.08	1.22	1.36
Traj-GCN [15]	0.26	0.45	0.71	0.79	0.88	0.36	0.60	0.95	1.13	1.28	0.53	1.02	1.35	1.48	1.62	0.19	0.44	1.01	1.24	1.37	0.46	0.79	1.12	1.29	1.35
NAT [11]	0.26	0.42	0.62	0.72	0.83	0.36	0.59	0.93	1.11	1.26	0.55	0.96	1.28	1.42	1.59	0.18	0.43	0.95	1.18	1.33	0.40	0.71	1.04	1.18	1.33
LDR [26]	0.29	0.47	0.59	0.68	0.76	0.35	0.58	**0.87**	**0.98**	1.27	0.45	**0.54**	**0.63**	**0.78**	1.50	0.17	0.44	0.94	**1.09**	1.33	0.45	0.71	**0.93**	**1.15**	1.32
DA-GNN	**0.20**	**0.39**	**0.56**	**0.65**	**0.72**	**0.33**	**0.57**	0.92	1.11	**1.25**	**0.43**	0.78	1.25	1.39	**1.48**	**0.16**	**0.41**	**0.93**	1.16	**1.29**	**0.39**	**0.69**	1.02	**1.15**	**1.30**
Motion Category	Purchases					Sitting					Sitting Down					Taking Photo					Walking Together				
time[ms]	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k	80	160	320	400	1k
Res-sup [17]	0.58	0.79	1.08	1.15	1.27	0.41	0.68	1.12	1.33	1.45	0.47	0.88	1.37	1.54	1.69	0.28	0.57	0.90	1.02	1.19	0.27	0.53	0.74	0.79	0.95
CSM [22]	0.63	0.91	1.19	1.29	1.33	0.39	0.61	1.02	1.18	1.25	0.41	0.78	1.16	1.31	1.42	0.23	0.49	0.88	1.06	1.20	0.27	0.52	0.71	0.74	0.90
Sym-GNN [10]	0.40	0.60	0.97	1.04	1.21	0.24	0.41	0.77	0.95	1.11	0.28	0.60	0.89	0.99	1.12	0.14	0.32	0.53	0.62	0.89	0.16	0.33	0.50	0.56	0.73
Traj-GCN [15]	0.43	0.65	1.05	1.13	1.25	0.29	0.45	0.80	0.97	1.15	0.30	0.61	0.90	1.00	1.16	0.14	0.34	0.58	0.70	1.00	0.15	0.34	0.52	0.57	0.70
NAT [11]	0.46	0.67	0.97	1.03	1.19	0.29	0.46	0.80	0.98	1.14	0.31	0.63	0.92	1.05	1.16	0.17	0.37	0.59	0.71	0.97	0.16	0.31	0.48	0.54	0.69
LDR [26]	0.43	0.58	**0.92**	1.04	1.20	0.27	0.43	0.75	1.01	1.14	0.29	0.62	0.87	0.98	1.13	0.15	0.33	0.54	0.71	0.95	0.15	0.33	0.49	0.54	0.70
DA-GNN	**0.37**	**0.57**	**0.92**	**1.01**	**1.17**	**0.22**	**0.40**	**0.73**	**0.92**	**1.09**	**0.25**	**0.58**	**0.86**	**0.96**	**1.10**	**0.13**	**0.30**	**0.51**	**0.64**	**0.87**	**0.12**	**0.30**	**0.46**	**0.53**	**0.68**

TABLE IV

TEST ACCURACY OF THE INTENTION PREDICTION METHOD BASED ON SYM-GNN, LDR, AND DA-GNN.

Method	Based on Sym-GNN	Based on LDR	Based on DA-GNN
Test accuracy[%]	86.54	90.23	**92.43**

(Fig. 5 [right]). As expected, the predicted results are similar to the real performed motions overall. Then, we did *greeting* 10 times where the user's behavior was slightly different each time and obtained eight correct intention predictions, which shows good accuracy of the intention prediction method. Afterward, we compared Nao's reaction time (i.e., the time from the start of the interaction till the start of Nao's response) when predicting the intention based on the partial observation against when recognizing the intention with the full observation. Note that we only used 150 motion frames recorded by the camera for the intention recognition, which were directly fed into the recognition model. As a result, the reaction time of the intention prediction and intention recognition is 4.79s and 5.87s, respectively, meaning a benefit of 1.08s in robot reaction when performing motion prediction.

VI. CONCLUSION

This paper proposed representing the human skeleton as a DAG, and we presented a novel DA-GNN model for human motion prediction. DA-GNN follows the encoder-decoder structure. The encoder progressively updates the attributes and the temporal dynamics of joints and bones in the observed human states. The decoder sequentially uses DAG-GRU and MLP to predict future states based on the update result obtained by the encoder. The experimental results obtained on CMU Mocap and Human 3.6m have validated the effectiveness of DA-GNN for human motion prediction over different prediction horizons. Furthermore, we have conducted an HRI-related experiment to showcase DA-GNN's benefit in realistic HRI scenarios. Our future work will focus on improving DA-RNN for more effective long-term prediction, and on systematically analyzing our model's embedding in human intention prediction.

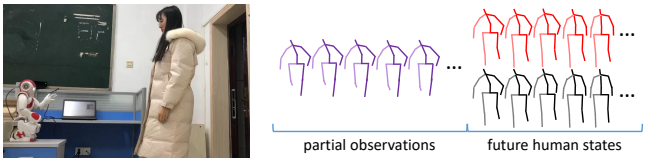

Fig. 5. [left] Illustration of *greeting*: a user enters into a service area and Nao greets the user. [right] Motion prediction results. The results in red and in black are the predicted human states by DA-GNN and the real human states the user preformed, respectively.

REFERENCES

[1] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *International Conference on Robotics and Automation*, 2019, pp. 6015–6022.

[2] G. Chalvatzaki, X. S. Papageorgiou, P. Maragos, and C. S. Tzafestas, "Learn to adapt to human walking: A model-based reinforcement learning approach for a robotic assistant rollator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3774–3781, 2019.

[3] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[4] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, 2020.

[5] Y. Wang, X. Ye, Y. Yang, and W. Zhang, "Collision-free trajectory planning in human-robot interaction through hand movement prediction from vision," in *International Confonference on Humanoid Robotics*, 2017, pp. 305–310.

[6] J. Bütepage, H. Kjellström, and D. Kragic, "Anticipating many futures: Online human motion prediction and generation for human-robot interaction," in *International Conference on Robotics and Automation*, 2018, pp. 4563–4570.

[7] G. Chalvatzaki, P. Koutras, J. Hadfield, X. S. Papageorgiou, C. S. Tzafestas, and P. Maragos, "Lstm-based network for human gait stability prediction in an intelligent robotic rollator," in *International Conference on Robotics and Automation*, 2019, pp. 4225–4232.

[8] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R.-D. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018.

[9] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *International Conference on Computer Vision*, 2015, pp. 4346–4354.

[10] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction," *arXiv preprint arXiv:1910.02212*, 2019.

[11] B. Li, J. Tian, Z. Zhang, H. Feng, and X. Li, "Multitask non-autoregressive model for human motion prediction," *arXiv preprint arXiv:2007.06426*, 2020.

[12] M. Brand and A. Hertzmann, "Style machines," in *Annual Conference on Computer Graphics and Interactive Techniques*, 2000.

[13] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2007.

[14] A. Lòpez-Mèndez, J. Gall, J. R. Casas, and L. J. Van Gool, "Metric learning from poses for temporal clustering of human motion," in *British Machine Vision Conference*, 2012, pp. 1–12.

[15] W. Mao, M. Liu, M. Salzmann, and H. Li, "Learning trajectory dependencies for human motion prediction," in *International Conference on Computer Vision*, 2019, pp. 9488–9496.

[16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5308–5317.

[17] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4674–4683.

[18] J. N. Kundu, M. Gor, and R. V. Babu, "Bihmp-gan: Bidirectional 3d human motion prediction gan," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 8553–8560.

[19] X. Guo and J. Choi, "Human motion prediction via learning local structure representations and temporal dependencies," in *AAAI Conference on Artificial Intelligence*, 2019, pp. 2580–2587.

[20] H. K. Chiu, E. Adeli, B. Wang, D. A. Huang, and J. C. Niebles, "Action-agnostic human pose forecasting," in *Winter Conference on Applications of Computer Vision*, 2019, pp. 1423–1432.

[21] P. Ghosh, J. Song, E. Aksan, and O. Hilliges, "Learning human motion models for long-term predictions," in *International Conference on 3D Vision*, 2017, pp. 458–466.

[22] C. Li, Z. Zhang, W. Sun Lee, and G. Hee Lee, "Convolutional sequence to sequence model for human dynamics," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5226–5234.

[23] 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. 6-8, pp. 1143–1156, 2019.

[24] D. Zecha, C. Eggert, M. Einfalt, S. Brehm, and R. Lienhart, "A convolutional sequence to sequence model for multimodal dynamics prediction in ski jumps," in *International Workshop on Multimedia Content Analysis in Sports*, 2018, pp. 11–19.

[25] 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," *arXiv preprint arXiv:2003.08802*, 2020.

[26] Q. Cui, H. Sun, and F. Yang, "Learning dynamic relationships for 3d human motion prediction," in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6519–6527.

[27] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.

[28] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende *et al.*, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems*, 2016, pp. 4502–4510.

[29] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *International Conference on Robotics and Automation*, 2019, pp. 1205–1211.

[30] A. E. Tekden, A. Erdem, E. Erdem, M. Imre, M. Y. Seker, and E. Ugur, "Belief regulated dual propagation nets for learning action effects on articulated multi-part objects," *arXiv preprint arXiv:1909.03785*, 2019.

[31] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2255–2264.

[32] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden markov model: Analysis and applications," *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.

[33] P. Rodriguez, J. Wiles, and J. L. Elman, "A recurrent neural network that learns to count," *Connection Science*, vol. 11, no. 1, pp. 5–40, 1999.

[34] Z. Ji, Q. Liu, W. Xu, Z. Liu, B. Yao, B. Xiong, and Z. Zhou, "Towards shared autonomy framework for human-aware motion planning in industrial human-robot collaboration," in *International Conference on Automation Science and Engineering*, 2020, pp. 411–417.

[35] Z. Wang, K. Mülling, M. P. Deisenroth, H. Ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic movement modeling for intention inference in human–robot interaction," *Int'l Journal of Robotics Research*, vol. 32, no. 7, pp. 841–858, 2013.

[36] A. K. Tanwani and S. Calinon, "A generative model for intention recognition and manipulation assistance in teleoperation," in *International Conference on Intelligent Robots and Systems*, 2017, pp. 43–50.

[37] H. C. Ravichandar and A. P. Dani, "Human intention inference using expectation-maximization algorithm with online model learning," *IEEE Trans. on Automation Science and Engineering*, vol. 14, no. 2, pp. 855–868, 2016.

[38] G. Chalvatzaki, X. S. Papageorgiou, C. S. Tzafestas, and P. Maragos, "Augmented human state estimation using interacting multiple model particle filters with probabilistic data association," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1872–1879, 2018.

[39] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[40] A. Sanchez-Gonzalez, N. Heess, J. T. Springenberg, J. Merel, M. Riedmiller, R. Hadsell, and P. Battaglia, "Graph networks as learnable physics engines for inference and control," *arXiv preprint arXiv:1806.01242*, 2018.

[41] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural relational inference for interacting systems," *arXiv preprint arXiv:1802.04687*, 2018.

[42] C. Cai and Y. Wang, "A note on over-smoothing for graph neural networks," *arXiv preprint arXiv:2006.13318*, 2020.

[43] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[44] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *International Conference on Computational Statistics*, 2010, pp. 177–186.

[45] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.
[46] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.
[47] P. D. Kingma and L. J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015, pp. 1–5.
[48] I. Goodfellow, Y. Bengio, and A. Courville, "Machine learning basics," *Deep Learning*, vol. 1, pp. 98–164, 2016.
[49] W.-R. Ko, M. Jang, J. Lee, and J. Kim, "Air-act2act: Human-human interaction dataset for teaching non-verbal social behaviors to robots," *arXiv preprint arXiv:2009.02041*, 2020.