HDG-ODE: A Hierarchical Continuous-Time Model for Human Pose Forecasting

Yucheng Xing Xin Wang
Department of Electrical and Computer Engineering, Stony Brook University
{yucheng.xing, x.wang}@stonybrook.edu

Abstract

Recently, human pose estimation has attracted more and more attention due to its importance in many real applications. Although many efforts have been put on extracting 2D poses from static images, there are still some severe problems to be solved. A critical one is occlusion, which is more obvious in multi-person scenarios and makes it even more difficult to recover the corresponding 3D poses. When we consider a sequence of images, the temporal correlation among the contexts can be utilized to help us ease the problem, but most of the current works only rely on discrete-time models and estimate the joint locations of all people within a whole sparse graph. In this paper, we propose a new framework, Hierarchical Dynamic Graph Ordinary Differential Equation (HDG-ODE), to tackle the 3D pose forecasting task from 2D skeleton representations in videos. Our framework adopts ODE, a continuous-time model, as the base to predict the 3D joint positions at any time. Considering the structural-property of the skeleton data in representing human poses and the possible irregularity caused by occlusion, we propose the use of dynamic graph convolution as the basic operator. To reduce the computational complexity introduced by the sparsity of the pose graph, our model takes a hierarchical structure where the encoding process at the observation timestamp is done in a cascade manner while the propagation between observations is conducted in parallel. The performance studies on several datasets demonstrate that our model is effective and can out-perform other methods with fewer parameters.

1. Introduction

With the progress in the computer vision field, human-related contents have attracted more and more attention, as they are very important in many real-world applications. Some examples are autonomous driving, human-computer interaction and anomaly detection. Among these contents, human poses are the most basic and important ones since their accurate and timely estimation or even forecasting usually serve as the first step for further analyzing complex activities.

Human pose estimation is not a novel topic but has been studied for many years. Its performance, however, is still not satisfactory enough to meet the demands of many practical applications. Since A. Toshev and C. Szegedy proposed DeepPose [69] to represent human poses by the locations of a set of joints from human skeletons and first time utilized the regression method of deep learning to estimate the corresponding 2D coordinates, a lot of efforts [78, 51, 87, 20, 86, 19, 79, 77, 54, 67, 95, 94, 3, 66, 74, 73] have been made to get the accurate estimation in the presence of occlusion resulted from the perspective limitation of images. These methods [14, 33, 55, 65, 34, 68, 38, 53, 16, 71, 58, 48, 99, 100, 84] are extended to multi-person scenarios, where the occlusion problem is more severe. Although some progress has been made, 2D locations cannot perfectly restore the human poses, and are never our final goals for this task. Recently, more and more works have been focusing on 3D pose estimation with raw images [15, 24, 6, 91] or extracted 2D poses [96, 21, 5, 18, 93, 39, 97, 83, 17, 43, 91] as input. Besides the inherent occlusion problems, the lack of depth in monocular images also limits the performance of algorithms. Although some schemes relying on multi-view images [50, 60, 94, 80, 72, 10, 70, 28, 40, 25, 11] or other sensors [54, 31, 27, 94] can ease this problem to some degree, we have to admit that paired images of different views or complementary sensors are not always available in most scenarios. The core problem is how to just utilize limited information in monocular images to extract the corresponding 3D poses of people. If a sequence of images (i.e. video) is available, the temporal context carried can be also helpful for the task [1, 81, 7, 47, 76]. They not only provide information to make up the missing joints caused by the occlusion and provide hints for joints' locations in the future, but also help group the detected joints belonging to a specific person in multi-person scenarios. So in this paper, we mainly focus on restoration from a sequence of monocular multi-person 2D skeletons to predict the corresponding 3D poses at a future time point.

Some related methods [64, 9, 75, 82, 41, 11, 98, 26, 13, 63, 36] usually applied graph neural networks to encode the

2D human joints and propagated the information through recurrent neural networks or directly used spatial-temporal graph convolutional networks. However, almost all of them only focused on single-person scenarios, while in most of cases, the relations among people in the scene are also helpful for the forecasting. Also, there exist several limitations in these works. Firstly, traditional recurrent neural networks, such as GRU or LSTM, all use discrete-time sequential models, which have a fixed propagation step and are not flexible enough to deal with different moving speeds of different human joints. The performance of these models will be compromised when the video frequency is not high or the cameras need to scan around in a certain period instead of shooting a fixed scene. For example, if the human pose is recorded every second in the video, discrete-time models cannot provide the forecasting result within 1 second. Instead, we propose to use neural ordinary differential equation (NeuralODE), a continuous-time model, to propagate information along the temporal axis in our framework. Secondly, we notice that any joint of a person is only connected to its nearest neighbors, which makes the adjacency matrix used in the graph convolution very sparse, and leads to inefficient calculations in turn. To reduce the meaningless calculations and increase the efficiency of our model, we propose to utilize a hierarchical structure to decompose a sparsely connected graph into several dense graphs and encode them in a cascade manner. By doing so, we can also deal with edges of different types separately, such as the edges between people and the edges connecting joints within each person. Lastly, the whole and fixed binary adjacency matrix is used for the graph convolution in the literature, although the actual connections of joints only occupy a very small part of the graph due to the occlusion. Instead, we propose to provide dynamic topology information at each time stamp. Different from works [21, 96, 84, 58, 42, 43] which learned another weight matrix from fully observed input, we propose to learn the weight matrix from partial input to more timely capture the effect of every specific connection in the graph and multiply it with the time-varying dynamic binary adjacency matrix to form the final one used in graph convolutions.

To summarize, the main contributions of this paper are three folds:

- We propose a continuous-time framework which takes the sequence of 2D skeletons extracted from a multi-person video as the input to forecast the multi-person 3D poses in any future time.
- We design our model as a hierarchical structure to reduce the computational complexity for multi-person scenarios, and also deal with different connection types in the graph.
- We utilize dynamic graph convolutions to deal with the irregular input caused by occlusions.

In the remaining of this paper, we will briefly introduce the related works in Sec. 2 and give the details of our model in Sec. 3. The performance of our model is demonstrated through experiments in Sec. 4. Finally, we will conclude our work in Sec. 5. The corresponding source code can be found at `https://github.com/SBU-YCX/HDG-ODE`.

2. Related Works

2.1. 3D Pose Estimation, Tracking and Forecasting

Single-Person Pose Estimation: Most of the 3D Single-Person Pose Estimation (SPPE) methods are proposed to recover the 3D poses from 2D ones extracted by a pre-trained 2D pose extractor such as Stacked Hourglass Network [51] or Convolutional Pose Machines [69]. Given 2D coordinates as input, Zeng *et al*. [93] proposed to split the human joints into local groups and estimate them separately. The same idea was used by Chen *et al*. [15], although they directly estimated 3D poses from raw images and applied different architectures for different parts. In Li *et al*.'s work [39], different subsets were exchanged to generate augment data for better training. Choi *et al*. [18] and Azizi *et al*. [5] used spectral graph convolutions to tackle the problem due to the structure property of input skeletons. Since images captured from monocular cameras have perspective limitations, Yu *et al*. [91] proposed perspective crop layers to eliminate the perspective effect, while Nie *et al*. [52] tried to use siamese architectures to decouple the view-dependent representation and pose-dependent representation. There are also some other works solved the perspective problems with the help of multi-view images, such as [50, 60, 80, 72, 94], by fusing the 2D poses from different views to get the original 3D results. Besides, other wearable sensors were used to complement the missing depth information in [94, 27], and Isogawa *et al*. [31] proposed to utilize an optical non-line-of-sight imaging system to obtain the 3D poses from photon images.

Multi-Person Pose Estimation: Inspired by 2D Multi-Person Pose Estimation (MPPE) methods, the works for the 3D estimation task can also be divided into two groups. One is top-down methods where a human detector is applied first and estimation is made for each person separately. Yang *et al*. [88] followed this line of work with graph convolutional networks (GCNs) [32], T. Xu and W. Takano [83] modified the stacked hourglass structure also with the graph convolutions. Li *et al*.'s [37] proposed a hierarchical architecture, which is the one closest to our work, but their goal was to augment the graph with additional labeled mesh data and integrate more information, which is totally different from our motivation. The other one is bottom-up approaches where joint locations are directly estimated, then associated and grouped into different people. In [24], Fabbri *et al*. proposed to use distance-based heuristic for the associations from the detected head joint with the highest confidence.

Cheng *et al.* [17] proposed to integrate both top-down and bottom-up branches to give better results. Besides, inspired by YOLO [59], some works [97, 6, 35, 49] explored to obtain the 3D poses in a single pass where a pivot joint was detected as well as the offsets for all other joints. Similar to single-person case, multi-view sources were also used in [25, 10, 70, 40, 28].

Pose Tracking and Forecasting: Since Andriluka *et al.* proposed PoseTrack [1] to first time introduce the Human Pose Tracking (HPT) task, contextual information has begun to be used for better estimating human poses in a sequence of inputs. Similar to other 3D pose tasks, most of works used extracted 2D pose sequence as the input and adopted top-down pipeline to mainly focus on single-person's pose tracking. Cai *et al.* [9] integrated spatio-temporal GCN into a stacked hourglass architecture to get the information of different scales. The same idea was applied in Wang *et al.*'s work [75], where the U-shaped GCN was used to extract multi-scale contexts. Liu *et al.* [41] proposed a graph attention spatio-temporal network with dilated temporal convolutions to get both local and global information, and Sofianos *et al.* [64] built a space-time-separable GCN for the 3D pose forecasting. Encoder-decoder architecture was utilized in [36, 63, 26] and Zheng *et al.* [98] solved the problem with spatio-temporal transformer. Furthermore, Chen *et al.* [11] proposed a fast method with multi-view inputs and Yuan *et al.* [92] utilized physics simulators and the reinforcement learning strategy in their work. Besides the skeleton joint representations, bone lengths and directions were used in [13, 82] and the extracted information was propagated through an LSTM.

2.2. Graph Neural Networks

Different from traditional convolutional neural networks (CNNs), graph neural networks (GNNs) are mainly proposed to deal with data having more general structures. GNNs can be divided into two groups, spatial and spectral graph convolutions, as described in [8]. The former one considers the graph as locally connected and the information is only shared between each node and its nearest neighbors. J. Atwood and D. Towsley [4] extended it to allow the communications among nodes within k-hop ($k >= 1$) connections by using the probabilistic transitions. The latter one converts the data into spectral domain first, multiplies it with the Laplacian of the topology matrix in the spectral domain to mimic the convolutional operation in the spatial domain. Compared with spatial graph convolutoins, spectral operations utilizes the global graph structure each time. However, since the kernel of the whole graph is large, to reduce the computational complexity, Defferrard *et al.* [23] proposed to approximate it with the k^{th} order polynomials of the Laplacian with the help of Chebyshev expansion to model the k-hop connections. T. Kipf and M. Welling [32] proposed graph convolutional networks (GCNs) to further simplify it with the 1^{st} order approximation of the covolution in ChebNet [23].

In the pose estimation tasks, GCNs were widely used in [88, 83, 37, 9, 75, 41]. However, unlike CNNs, traditional GCNs have shared weights for different neighbors. To tackle this problems, another learnable weight matrix was trained in [96, 21, 42]. But, in these works, the weight matrix were learned with full joints and the final matrix used for convolution was the multiplication between the weight matrix and the static skeleton matrix assuming all joints observable. In our work, to deal with the missing case caused by occlusion, we make the skeleton matrix dynamic according to the current occlusion status so that the learnt weight matrix can also be effective even though only partial joints given. Some other works [43, 58, 84] proposed to add more semantic connections to the graph, which increased the calculation complexity of the models since the joints of all people could not be fully-connected. In addition, the semantic graphs were still globally sparse although locally dense. In this work, we propose to decompose the whole graph in a hierarchical manner.

2.3. Neural Differential Equation

As mentioned earlier, almost all the existing pose tracking methods were built based on spatio-temporal graph convolutional networks (ST-GCNs) and recurrent neural networks (RNNs), which could only model the discrete-time data where all human joints were assumed to be observable although there existed missing and inaccurate inputs due to occlusions. In order to better model the continuous-time process of the irregular data, neural ordinary differential equation (NeuralODE) was proposed by Chen *et al.* [12], where deep neural networks were utilized to parameterize a nonlinear ordinary differential equation (ODE). Rubanova *et al.* [61] and Brouwer *et al.* [22] further extended it by introducing a recurrent structure to efficiently encode the input information into the underlying trajectories. Some other works [44, 45, 46] also inserted Brownain terms into ODE to make it a stochastic model that can capture the uncertainties in the dynamics. In these works, pose estimation was also mentioned as a part of experiment to prove their efficiency. However, when they dealt with the human joint data, they merely considered each joint sequence as an isolated trajectory and ignored the relations among these joints. Graph neural networks were introduced to parameterize the ODEs in [56, 29, 57], but all of them utilized the whole graph, which didn't take the sparsity problem into consideration in multi-person scenarios.

3. Hierarchical Dynamic-Graph Ordinary Differential Equation Model

In this section, we will introduce the details of our hierarchical ordinary differential equation model with dynamic graph convolutions, which we call as *HDG-ODE*. As shown in Figure 1, the framework of our model follows a hierarchical structure, where each level is a sub-graph with the state changes modeled through graph ordinary differential equation and parameterized with dynamic graph convolutions. In the remaining of this section, we will describe the details of each component as well as the advantages of such designs.

3.1. Problem Statement and Notations

Given a multi-person monocular video containing K people, and the 2D joints sequence up to time t is $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_t\}$, where $\mathbf{X}_t \in \mathcal{R}^{[K,N_j,2]}$ and N_j is the number of joints within each person. These 2D representations can be obtained from any existing 2D pose estimator. Our goal is to predict the corresponding 3D poses $\mathbf{Y}_\tau \in \mathcal{R}^{[K,N_j,3]}$ at a future time τ from $\mathbf{X}_t$, where τ can be any value within $(t, t+1]$. Like in many 2D estimators, the missing or occluded joints are represented as some singular values or estimated with a very low confidence, based on this, we can directly obtain the corresponding occlusion mask $\mathbf{M}_t$ along with $\mathbf{X}_t$ at each time t.

3.2. Hierarchical Graph Structure

As explored in many other works, compared to the traditional use of CNN, graph convolution is more suitable for human skeleton data due to the irregular connection of joints. The feature matrices $Z^{(l)}$ and $Z^{(l+1)}$ of two adjacent convolution layers are expressed as

$$Z^{(l+1)} = \sigma(WZ^{(l)}\tilde{A}), \tag{1}$$

where W is a learnable parameter matrix, $\tilde{A}$ is the normalized form of the adjacency matrix $A \in [0,1]^{N\times N}$ and N is the number of nodes in the graph. However, when modeling human skeletons, we notice that each joint node is only connected to a few other joints, which will make the static skeleton-based $\tilde{A}$ very sparse and lead to plenty of meaningless calculations in (1). Although some recent works tried to learn more semantic connections among different joints besides static ones, the sparsity problem cannot be eliminated. The semantic relationship among joints usually exists within a limited number of hops instead of the whole graph. Even after we add those semantic edges, we can separate all joints into several groups such that the connections within each group are dense but the ones among different groups are still sparse. This inspires us to adopt a hierarchical structure instead of a flatten one to represent the graph of human skeletons.

Besides, according to the physical property of human motion, different joints change their positions at different speeds over time. The location of the central joint usually has a slow and smooth change while the extremity joints, such as "wrist" and "ankle", have drastic and unexpected changes. Therefore, rather than allocating the same portion of computational resource to all joints, we can focus more on the latter ones. The cascade processing along the hierarchical structure also allows us to utilize the relative-central ones' locations to guide the estimation of positions of the relative-extreme joints.

Another reason for which we adopt the hierarchical structure is its advantage of dealing with different types of edges in the whole graph. For example, in the multi-person scenario, previous works commonly used top-down methods where each person in the scene would be processed individually, but actually, pose estimation and tracking can benefit from taking the relation among people into consideration. However, the edges describing the inter-person relations are different from those representing the intra-person limbs, and it will reduce the effect if we model them in one large graph. Intuitively, separating them into different groups and forming a graph at each level will be a better choice.

Specifically, as shown in Figure 2, for each frame of the video at time t, we get a set of 2D skeleton coordinates of all the people in the scene as input. To apply the hierarchical pipeline, we choose the "pelvis" joint as the root node of each person, whose coordinate can be also considered as the location of the person. All these root nodes form the 1st-level graph and encoded by a graph encoder

$$\mathbf{Y}_\tau^{(1)} = G_1(\mathbf{X}_t^{(1)}, \mathbf{H}_{<t}^{(1)}, \tilde{A}_t^{(1)}), \tag{2}$$

where $\mathbf{X}_t^{(1)} \in \mathcal{R}^{[K,1,2]}$ represents the 2D coordinates of all root nodes of K people, $\mathbf{Y}_\tau^{(1)} \in \mathcal{R}^{[K,1,3]}$ is the corresponding 3D one forecasted for the future time τ, $\mathbf{H}_{<t}^{(1)}$ is the historical feature, and $\tilde{A}_t^{(1)}$ is the normalized adjacency matrix with the edges calculated based on the distances among connected roots, i.e. the relations among people. There is an edge between two people if their root nodes' distance is smaller than a threshold (the average of the heights of joints in 2D skeleton, i.e. distance from "head" to "ankle" joint in our setting), which indicates they are related. Then, all the limb-root joints, including "neck", "left shoulder", "right shoulder", "left hip" and "right hip", as well as the root node are grouped to form the 2nd-level graph within each person. To utilize the result of the 1st-level graph, the output of this level is designed as the spatial shifts of the limb-root joints' locations from the root node. We can express the process as below

$$\begin{aligned} &\mathbf{D}_\tau^{(2,i)} = G_2(\mathbf{X}_t^{(2,i)}, \mathbf{Y}_{\tau,i}^{(1)}, \mathbf{H}_{<t}^{(2,i)}, \tilde{A}_t^{(2,i)}), i \in [1, K] \\ &\mathbf{Y}_\tau^{(2,i)} = \mathbf{D}_\tau^{(2,i)} + \mathbf{Y}_{\tau,i}^{(1)}, \end{aligned} \tag{3}$$

where $\mathbf{X}_t^{(2,i)} \in \mathcal{R}^{[1,6,2]}$ is the 2D coordinates vector of all limb-root nodes as well as root node of person i, $\mathbf{Y}_{\tau,i}^{(1)}$ is the 3D root node's position of this person obtained by (2), $\mathbf{D}_\tau^{(2,i)}$ and $\mathbf{Y}_\tau^{(2,i)}$ represent respectively the future 3D position shift and final position predicted. Finally, we group each limb-root joint to the corresponding limb-extreme joint to form

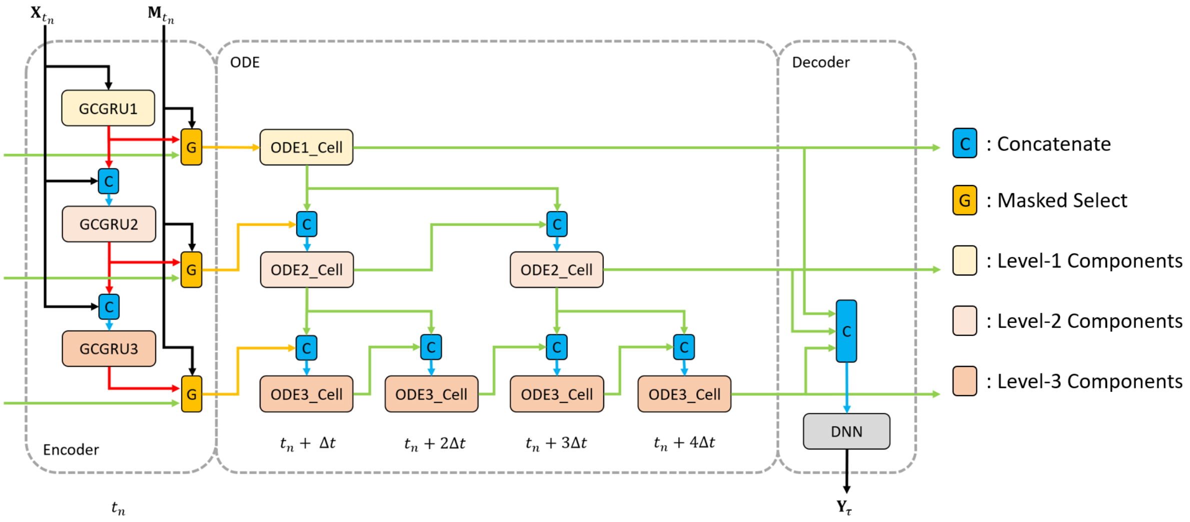

Figure 1. The framework of HDG-ODE. $\mathbf{X}_{t_n}$ are the 2D joints coordinates at time t_n, $\mathbf{M}_{t_n}$ is the occlusion mask obtained from $\mathbf{X}_{t_n}$. $\mathbf{Y}_\tau$ are the predicted 3D coordinates at the future time τ.

the 3rd-level graph. Similarly, the encoding process within each lamb group j of person i can be expressed as

$$\mathbf{D}_\tau^{(3,i,j)} = G_3(\mathbf{X}_t^{(3,i,j)}, \mathbf{Y}_{\tau,j}^{(2,i)}, \mathbf{H}_{<t}^{(3,i,j)}, \tilde{A}_t^{(3,i,j)}), j \in [1,5]$$
$$\mathbf{Y}_\tau^{(3,i,j)} = \mathbf{D}_\tau^{(3,i,j)} + \mathbf{Y}_{\tau,j}^{(2,i)}, \qquad (4)$$

where $\mathbf{X}_t^{(3,i,j)} \in \mathcal{R}^{[1,3,2]}$ is the vector of the 2D coordinates representing all nodes of person i within the limb group j, $\mathbf{Y}_{\tau,j}^{(2,i)}$ is the 3D location of the lamb-root j calculated by (3), $\mathbf{H}_{<t}^{(3,i,j)}$ and $\tilde{A}_t^{(3,i,j)}$ represent the corresponding part of the historical feature and the adjacency matrix in this level. 3D shift $\mathbf{D}_\tau^{(3,i,j)}$ is used to get the final 3D coordinates $\mathbf{Y}_\tau^{(3,i,j)}$ at future time τ.

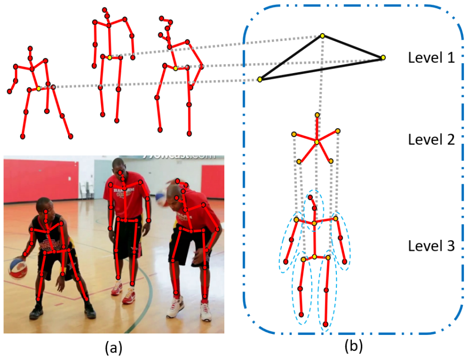

Figure 2. Illustration of the hierarchical graph structure: (a) the skeleton representations of a multi-person scerario; (b) the corresponding 3-level hierarchical structure.

3.3. Time-Varying Dynamic Graph Convolution

A most important impact factor to the performance of human pose estimation is the occlusion, no matter the occlusion is caused by other objects or the self-occlusion happens within a person. To get the 2D skeleton inputs with some 2D pose estimation methods, the visible joints can be directly extracted from the raw images while the occluded ones are missing. Even though the missing joints can be estimated by some intuition, they are less accurate due to the uncertainties. When we use such 2D poses as inputs to recover the corresponding 3D poses, called the lifting process, the inaccuracy will accumulates and compromises data processing during the information communication within the graph convolution operation. Therefore, we hope our graph encoder can be reasonably adaptive to deal with the missing and inaccurate cases.

In the literature work that apply the traditional graph convolution for this task, the adjacency matrix A and its normalized form $\tilde{A}$ in (1) usually keep constant to represent the static connections within the graph, i.e. all the joints are assumed to be fully available, no matter they are observable or occluded. However, as we described, only those visible ones are accurate and worth being integrated to guide the forecasting. So we propose to use the time-varying dynamic graph convolution to only encode visible joints input at each timestamp. Inspired by some literature work [96], the learning of a hybrid adjacency-weight matrix may be written as

$$Z_t^{(l+1)} = \sigma(WZ_t^{(l)}\rho(W_{\tilde{A}} \odot \tilde{A})), \qquad (5)$$

where $W_{\tilde{A}}$ is a learnable weight matrix applied to assign the information along different edges with different weights to solve the weight-sharing problem [21, 42] and $\rho(\cdot)$ is a nonlinear function. However, we notice that in (5), $W_{\tilde{A}}$ is learnt based on the fully-observable data and the $\tilde{A}$ is the static one without taking the occlusion cases into consideration. In other words, although the learnt weights work well when all joints are available, they may not be effective if partial of joints are occluded and missing. Based on this, we make a further step and propose to also take the occlusion status into consideration when obtaining the dynamic weight matrix. Specifically, at each timestamp t, we will drop the occluded joints in the original static adjacency matrix A to form a time-varying adjacency matrix A_t, and this dynamic adjacency matrix will be used in both training and testing, i.e. (5) can be rewritten as

$$Z_t^{(l+1)} = \sigma(W Z_t^{(l)} \rho(W_{\tilde{A}} \odot \tilde{A}_t)), \tag{6}$$

where $\tilde{A}_t$ is the normalized form of the real adjacency matrix at time t. By doing so, the learning of the weight matrix $W_{\tilde{A}}$ will only focus on the observable joints to make sure it can also be effective when only partial joints are available. In the testing phase, the hybrid-matrix $W_{\tilde{A}} \odot \tilde{A}_t$ can ignore the missing joints and adaptively adjust the weights along existing edges for information integration among joints.

3.4. Parallel Ordinary Differential Equation

Besides the partial occlusion, which leads to incomplete graph input at each frame and can be regarded as spatial missing, there are also many cases where a person is totally occluded, and causes the temporal missing within a sequence. Previous works almost rely on the discrete-time sequential models , which are not flexible enough since the missing in real world can vary from several frames to dozens of frames.

To make the model more robust, we adopt ordinary differential equation (ODE), a continuous-time model in this paper, which can be expressed as

$$\begin{aligned} \frac{d\mathbf{H}_t}{dt} &= F(\mathbf{H}_t), \\ \mathbf{H}_t &= \mathbf{H}_{t_{n-1}} + \int_{t_{n-1}}^{t} F(\mathbf{H}_\tau) d\tau. \end{aligned} \tag{7}$$

Compared with traditional recurrent networks, it can be used to predict the values over the interval of any length. Specially, in our model, we use time-varying dynamic graph convolution mentioned in Sec. 3.3 to parameterize the derivatives $F(\cdot)$. For efficiency, we compute the integration in (7) by Euler Method and we can rewrite it as

$$\mathbf{H}_{t,F} = \mathbf{H}_{t-\Delta t,F} + F(\mathbf{H}_{t-\Delta t,F}, \tilde{A})\Delta t, \tag{8}$$

where $\mathbf{H}_{t,F}$ is the hidden feature, $\tilde{A}$ is the normalized adjacency matrix assuming all joints observable, and Δt is the step-size. At an observation time t_n, the features of observable joints are updated through the same graph convolution encoder as described in Sec. 3.2, i.e.

$$\mathbf{H}_{t_n,G} = G(\mathbf{X}_{t_n}, \mathbf{Y}_{t_n}, \mathbf{H}_{<t_n}, \tilde{A}_{t_n}), \tag{9}$$

where $G = \{G_1, G_2, G_3\}$ represents the cascade process in (2)-(4) parameterized by our time-varying dynamic graph convolution networks, $\mathbf{X}_{t_n}$ and $\mathbf{Y}_{t_n}$ are the 2D and 3D joint locations, $\mathbf{H}_{<t_n} = \mathbf{H}_{t_n - \Delta t, F}$ is the feature from the ODE model and $\tilde{A}_{t_n}$ is the normalized adjacency matrix of the graph formed by observable joints. After the update, the new feature at the time instant t_n becomes

$$\mathbf{H}_{t_n} = \mathbf{M}_{t_n} \odot \mathbf{H}_{t_n,F} + (\mathbf{1} - \mathbf{M}_{t_n}) \odot \mathbf{H}_{t_n,G}, \tag{10}$$

where $\mathbf{M}_{t_n}$ is the occlusion mask of the input described in Sec. 3.1. Here, we want to emphasize that $\tilde{A}_{t_n}$ in (9) and $\tilde{A}$ in (8) may not be the same according to the observation states of input. As we described in last section, the input may contain missing values, but all the missing values have been completed by the encoder in the encoding process and we can assume the graph is fully observable in the propagation process.

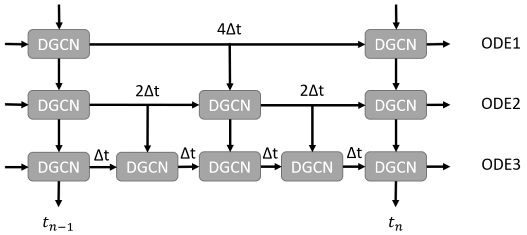

Figure 3. Illustration of the parallel ODE pipeline.

Here, we can further demonstrate the advantages of hierarchical structure. For efficiency, we adopt Euler method to approximately calculate the integration, but as we know, it will predict the curve as a straight line if we make the propagation step Δt too large and equal to the interval between two observations as $t_n - t_{n-1}$. To make the model more expressive, we usually make Δt smaller, i.e.

$$\Delta t = \frac{1}{k}(t_n - t_{n-1}). \tag{11}$$

However, as we discussed in Sec. 3.2, different joints of human body have different rates of location changes, we don't need to set k to the same value at different levels in our model. Instead, we can adopt a parallel pipeline and make it small in shallow levels while enlarge it as the level goes deeper as shown in Figure 3.

4. Experiments

4.1. Experimental Setups

4.1.1 Datasets

MuPoTS-3D[49]: This multi-person dataset contains sequences captured in 20 real scenes. Each sequence has up to three subjects, as well as provides raw images, 2D pose annotations and the corresponding 3D ones. The lengths of data vary from 125 frames to 800 frames, and we split them into segments with the length of 50 frames. We divide all the data into training, validation and testing sets with the ratio of 0.8: 0.1: 0.1. The real occlusion mask are given as well, and the final data are obtained by the element-wise multiplication between the full data and the occlusion masks. Before being fed into the models, data from each sequence are first normalized.

Human3.6M[30]: This dataset consists of 3.6 million of single-person frames. 11 subjects conducting 15 activities are captured by cameras. Both 2D and 3D annotations are provided where 2D ones come from cameras while 3D ones are obtained using a motion capture system. Different from commonly used training methods, we only use data from one camera to train the model. We performed the same data partition and normalization as above.

4.1.2 Evaluation Metrics

Mean Per Joint Position Error (MPJPE): This metric measures the average position error over all joints and frames in the sequence, which can be expressed as

$$MPJPE = \frac{1}{N_f}\sum_f \frac{1}{N_j^{(f)}}\sum_j ||P_{pred}^{(f)}(j) - P_{gt}^{(f)}(j)||_2, \quad (12)$$

where N_f is the total number of frames, $N_j^{(f)}$ is the total number of joints in the frame f, $P_{gt}^{(f)}(j)$ and $P_{pred}^{(f)}(j)$ are the ground-truth and the predicted 3D position of the joint j in the frame f.

Percentage of Correct Keypoints (PCK): PCK calculates the percentage of correctly predicted key-points, where a key-point j in the frame f is considered to be correct if the distance $d_p^{(f)}(j)$ between the 3D ground-truth $P_{p,gt}^{(f)}(j)$ and the predicted 3D position $P_{p,pred}^{(f)}(j)$ are smaller than a person-specific threshold S_p^f, i.e.

$$PCK = \frac{\sum_f \sum_p \sum_j \delta(d_p^{(f)}(j) < S_p^{(f)})}{\sum_f \sum_p \sum_j 1}, \quad (13)$$

Here we adopt the setting in MPII dataset[2] where the threshold is the ratio of head link's length $h_p^{(f)}$ of the person p, i.e. the distance between "head" joint and "upper neck", and the PCK can be represented as

$$PCKh@\alpha = \frac{\sum_f \sum_p \sum_j \delta(\frac{d_p^{(f)}(j)}{h_p^{(f)}} < \alpha)}{\sum_f \sum_p \sum_j 1}, \quad (14)$$

where α is the ratio parameter.

4.1.3 Implementation Details

As introduced in Sec. 3.2, the hierarchical graph has 3 levels, containing K, 6 and 3 nodes each, and our HDG-ODE model has a hierarchical structure of 3 levels, with an encoder and an ODE at each level. For each encoder, we use a single-layer graph-GRU, using our time-varying dynamic graph convolutions with the feature size of 6. ODE models are parameterized with graph neural networks of 1, 2 and 3 convolutional layers at the three levels respectively, with the hidden feature size of 16. The propagation steps for the three levels are respectively $[1/30s, 1/60s, 1/120s]$, and the data sampling interval is $1/30s$. The experiments are conducted on single NVIDIA 1080 Ti GPU. During the training, we use the $Adam()$ optimizer with the learning rates of $[0.01, 0.001, 0.0001]$ to minimize the mean squared error (MSE) between the predicted 3D positions and the ground-truth of the visible joints, and 100 epochs are run at each learning rate. To speed up the training, it will be early-stopped if there is no further loss decrease in 20 epochs.

4.2. Experiment Results

4.2.1 Overall Performance

We conduct experiments over MuPoTS-3D dataset to evaluate the performance of our proposed model on the multi-person 3D pose tracking task. For comparison, we choose some representative methods, including discrete-time recurrent models and continuous-time ones, as the baselines. Specifically we take the following methods into consideration:

- **STGCN:** This discrete-time model consists of several concatenated spatio-temporal convolutional blocks (ST-Conv blocks), which is a common structure that is widely used for action recognition [85], pose forecasting [75] and other prediction tasks [89]. In our experiments, STGCN has 2 ST-Conv blocks, and recurrently processes each clip within the whole 2D pose sequence and give the 3D predictions. The output dimension of each layer in the blocks is 32.

- **Graph-GRU:** It is a variant of conventional Gated Recurrent Unit (GRU), where the linear operation in the conventional GRU is replaced by graph convolution. It works as the backbone in [90]. We construct the baseline with 2 Graph-GRU layer and a linear decoder. The output dimension of each layer in the blocks is 32.

- **SemGCGRU:** It is an improved version of traditional Graph-GRU, where the fixed binary adjacency matrix is replaced by a learnable weight matrix to capture the relations among nodes [96]. In our implementation, the structure and the feature size are the same as Graph-GRU.

- **ODE-RNN:** In this model [62], the deep learning modules, specifically dense neural networks, are incorporated to parameterize a non-linear ordinary differential equation (ODE) to model the continuous-time process of data sequence. In our experiments, the dense network contains 2 linear layers and the dimension of hidden features is 32.

- **Graph-ODE:** It is a GNN variant of ODE-RNN, with a GCN to learn the derivative of ODE and a GRU to integrate the information of observations. In our model comparison, the GCN has 2 hidden layers with ReLU activation functions and the output dimension of each hidden layer is 32. The output dimension of GRU is 32 as well. In other words, to compare with discrete models, we just replace the GRU module in Graph-GRU with an ODE module.

Although continuous-time models can provide predictions at any time, even smaller than the data sampling rate, we set the step of them to be equal to the original data interval used in discrete-time models for fair comparison. From the results in Table 1, we can see that our HDG-ODE can achieve the best result with the smallest mean prediction error and the highest percentage of corrected prediction.

	MPJPE(↓)	PCKh@α (↑)		
		α=0.1	α=0.5	α=1.0
Discrete				
STGCN [89, 75]	0.6004	13.03%	60.35%	77.76%
Graph-GRU [90]	0.5573	14.86%	62.39%	80.04%
SemGCGRU [96]	0.4375	17.09%	69.61%	85.96%
Continuous				
ODE-RNN [61]	0.4396	20.21%	71.78%	85.45%
Graph-ODE [56]	0.4066	19.80%	71.57%	87.00%
Ours				
HDG-ODE	**0.3038**	**29.72**%	**78.67**%	**92.18**%

Table 1. Testing Performance of different models on MuPoTS-3D.

4.2.2 Ablation Study

In this part, we will separately demonstrate the effectiveness of each designing component in our proposed model by some experiment results, including *the continuous ODE model*, *the pipeline taking all people into consideration*, *the new time-varying dynamic graph convolution* and *the proposed hierarchical structure*. The evaluation is sequential by adding the components one by one, and each time we only choose the current best model to conduct the experiment for the next component.

Discrete-Time Model vs. Continuous-Time Model: From Table 1, compared to traditional discrete-time models, no matter our HDG-ODE or other continuous-time models have a better performance. Different from discrete models which only execute one propagation between two observations, the continuous-time ones conduct multiple-step propagation so that the information on each joint will be affected and corrected by its neighbors several times before getting the final prediction. Also, due to the property of ODE, the continuous-time model can provide predictions at any time in the interval between two observations, instead of just obtaining predictions at the observation instants as done by discrete-time ones.

Isolated Top-Down Pipeline vs. Relational Top-Down Pipeline: As we described previously, most of existing multi-person pose tracking works adopt the top-down processing pipeline, where each person in the scenario is tracked separately as an isolated trajectory. nstead, we take into consideration the relation among people to infer their relative movements, which in turn affects the future positions of specific joints. In Table 2, by comparing the models in isolated top-down pipeline with the ones in our relational pipeline, we can observe the improvement.

	Isolated Pipeline		Relational Pipeline	
	MPJPE	PCKh@1.0	MPJPE	PCKh@1.0
STGCN	0.6104	77.72%	0.6004	77.76%
Graph-GRU	0.5738	78.46%	0.5573	80.04%
Graph-ODE	0.4203	86.89%	0.4066	87.00%

Table 2. Comparison between the isolated top-down pipeline and our relational top-down pipeline on MuPoTS-3D.

Static Graph Convolution vs. Time-Varying Dynamic Graph Convolution: In the real cases, occlusion is inevitable. To better tackle the problem caused by the input data with missing or inaccurate values due to occlusions, we propose to replace the static graph convolution used in most of works with our time-varying dynamic graph convolution in the HDG-ODE model. As shown in Table 3, the dynamic graph convolution helps eliminate the negative influence of inaccurate inputs as well as adjust the integration weights of edges to the most suitable ones according to occlusion states, so the models can have a better performance than before.

	Original Graph Conv		Our Graph Conv	
	MPJPE	PCKh@1.0	MPJPE	PCKh@1.0
Graph-GRU	0.5573	80.04%	0.3844	86.45%
Graph-ODE	0.4066	87.00%	0.3706	88.74%

Table 3. Comparison between original static graph convolutions (GC) and our time-varying dynamic convolutions (DG) on MuPoTS-3D.

Flatten Structure vs. Hierarchical Structure: To solve the sparsity problem explained in Sec. 3, our HDG-ODE adopts a hierarchical structure. In Table 4, compared to models in the flatten structure with a large sparse graph, the corresponding hierarchical ones perform better for models with graph convolutions. The reasons leading to the improvement lie in two aspects: On one hand, by decomposing the whole sparse graph into a set of relatively dense ones, the parameters to learn are reduced, as shown in Table 5. This helps save a significant amount of computational resource, and the released resource can be reallocated to the joints with more complicated movements, e.g. making the corresponding network deeper or reducing the propagation step of ODE. On the other hand, the cascade processing along the hierarchical structure can provide the guidance among the levels. As for the ODE-RNN, the hierarchical one has a worse result because the structure is mainly proposed to deal with the graph sparsity problem. However, ODE-RNN doesn't utilize the graph information and only cares about isolated joint trajectories, so the hierarchical structure will make the model more complex and reduce the performance.

	Flatten Structure		Hierarchical Structure	
	MPJPE	PCKh@1.0	MPJPE	PCKh@1.0
ODE-RNN	0.4396	85.45%	0.4938	82.40%
GC-ODE	0.4066	87.00%	0.3451	90.51%
DG-ODE	0.3706	88.74%	0.3038	92.18%

Table 4. Performance Comparison between the flatten structure and the hierarchical structure on MuPoTS-3D.

	#Params (↓)	Memory (↓)
DG-ODE	10649	0.76M
HDG-ODE	2643	0.66M

Table 5. Complexity Comparison between the flatten Dynamic-Graph ODE (DG-ODE) and our hierarchical one (HDG-ODE).

4.2.3 Robustness Analysis

Besides the multi-person scenarios, we also conduct experiments of single-person scenarios on Human3.6M dataset. Different from MuPoTS-3D, no occlusion information is provided within the data, so we make occlusion masks by ourselves and test the robustness of our model under different observable ratios. Specifically, for a sequence, we assume only p_t of the frames are observable while in the remaining ones, the input skeletons are totally missing due to the occlusion by buildings or other large objects. For each observable frame, we assume there exist self-occlusions such that only p_s of the joints are available. To make the corresponding mask, both p_t of the frames and p_s of the joints on each frame are randomly chosen and the total observable ratio can be expressed as $p_t \times p_s$. The results are shown in Table 6. When the observable ratio decreases as a result of occlusions, the performance gets worse for all models. However, our HDG-ODE always performs the best among all models studied under all circumstances.

5. Conclusion

In this paper, we propose a new continuous-time model called HDG-ODE to forecast the future 3D human pose representations in multi-person videos given the 2D pose sequence as input. By using ordinary differential equations and dynamic graph convolutions, we alleviate the occlusion problem brought by perspective effects of monocular images. Besides, with the help of hierarchical structure, we reduce the computational complexity of our model and are able to allocate more resources to the human joints which have more complicated movement. Our experiments demonstrate that our model can well recover the corresponding 3D poses from the single-view 2D poses and outperform literature works on the human pose forecasting task. Currently, the maximum number of people in the scenes of the video is limited. In the future, we will make our model more adaptive and test it in more crowded scenarios.

	$p_t = 0.8, p_s = 0.5$		$p_t = 0.6, p_s = 0.4$	
	MPJPE	PCKh@1.0	MPJPE	PCKh@1.0
STGCN	0.3966	80.19%	0.4376	75.82%
Graph-GRU	0.3723	83.34%	0.4188	77.39%
ODE-RNN	0.3527	84.33%	0.3826	80.66%
Graph-ODE	0.3473	83.98%	0.4019	78.54%
HDG-ODE	**0.3052**	**88.29**%	**0.3543**	**82.36**%

Table 6. Performance of different models on Human3.6M with different observable ratios.

References

[1] Mykhaylo Andriluka, Umar Iqbal, Eldar Insafutdinov, Leonid Pishchulin, Anton Milan, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking, 2017.

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele.

[3] Bruno Artacho and Andreas Savakis. Unipose: Unified human pose estimation in single images and videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[4] James Atwood and Don Towsley. Diffusion-convolutional neural networks, 2015.

[5] Niloofar Azizi, Horst Possegger, Emanuele Rodolà, and Horst Bischof. 3d human pose estimation using möbius graph convolutional networks, 2022.

[6] Abdallah Benzine, Florian Chabot, Bertrand Luvison, Quoc Cuong Pham, and Catherine Achard. Pandanet: Anchor-based single-shot multi-person 3d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning temporal pose estimation from sparsely labeled videos. In *Advances in Neural Information Processing Systems 33*, 2019.

[8] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs, 2013.

[9] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[10] He Chen, Pengfei Guo, Pengfei Li, Gim Hee Lee, and Gregory Chirikjian. Multi-person 3d pose estimation in crowded scenes based on multi-view geometry. *arXiv preprint arXiv:2007.10986*, 2020.

[11] Long Chen, Haizhou Ai, Rui Chen, Zijie Zhuang, and Shuang Liu. Cross-view tracking for multi-human 3d pose estimation at over 100 fps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations, 2018.

[13] Tianlang Chen, Chen Fang, Xiaohui Shen, Yiheng Zhu, Zhili Chen, and Jiebo Luo. Anatomy-aware 3d human pose estimation in videos. *arXiv preprint arXiv:2002.10322*, 2020.

[14] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] Zerui Chen, Yan Huang, Hongyuan Yu, Bin Xue, Ke Han, Yiru Guo, and Liang Wang. Towards part-aware monocular 3d human pose estimation: An architecture search approach. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 715–732, Cham, 2020. Springer International Publishing.

[16] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] Yu Cheng, Bo Wang, Bo Yang, and Robby T. Tan. Monocular 3d multi-person pose estimation by integrating top-down and bottom-up networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7649–7659, June 2021.

[18] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision (ECCV)*, 2020.

[19] Chia-Jung Chou, Jui-Ting Chien, and Hwann-Tzong Chen. Self adversarial training for human pose estimation. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 17–30, 2018.

[20] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[22] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series, 2019.

[23] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 2016.

[24] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Stefano Alletto, and Rita Cucchiara. Compressed volumetric heatmaps for multi-person 3d pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[25] Qi Fang, Qing Shuai, Junting Dong, Hujun Bao, and Xiaowei Zhou. Reconstructing 3d human pose by watching humans in the mirror. In *CVPR*, 2021.

[26] Kehong Gong, Bingbing Li, Jianfeng Zhang, Tao Wang, Jing Huang, Michael Bi Mi, Jiashi Feng, and Xinchao Wang. Posetriplet: Co-evolving 3d human pose estimation, imitation, and hallucination under self-supervision. In *CVPR*, 2022.

[27] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4318–4329, June 2021.

[28] Congzhentao Huang, Shuai Jiang, Yang Li, Ziyue Zhang, Jason Traish, Chen Deng, Sam Ferguson, and Richard Yi Da Xu. End-to-end dynamic matching network for multi-view multi-person 3d pose estimation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*, page 477–493, Berlin, Heidelberg, 2020. Springer-Verlag.

[29] Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21, page 705–715, New York, NY, USA, 2021. Association for Computing Machinery.

[30] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[31] Mariko Isogawa, Ye Yuan, Matthew O'Toole, and Kris M. Kitani. Optical non-line-of-sight physics-based 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[33] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[34] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[35] Jogendra Nath Kundu, Ambareesh Revanur, Govind Vitthal Waghmare, Rahul Mysore Venkatesh, and R. Venkatesh Babu. Unsupervised cross-modal alignment for multi-person 3d pose estimation, 2020.

[36] Jogendra Nath Kundu, Siddharth Seth, Varun Jampani, Mugalodi Rakesh, R. Venkatesh Babu, and Anirban Chakraborty. Self-supervised 3d human pose estimation via part guided novel image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] Han Li, Bowen Shi, Wenrui Dai, Yabo Chen, Botao Wang, Yu Sun, Min Guo, Chenlin Li, Junni Zou, and Hongkai Xiong. Hierarchical graph networks for 3d human pose estimation, 2021.

[38] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[39] Shichao Li, Lei Ke, Kevin Pratama, Yu-Wing Tai, Chi-Keung Tang, and Kwang-Ting Cheng. Cascaded deep monocular 3d human pose estimation with evolutionary training data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Jiahao Lin and Gim Hee Lee. Multi-view multi-person 3d pose estimation with plane sweep stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11886–11895, June 2021.

[41] Junfa Liu, Juan Rojas, Zhijun Liang, Yihui Li, and Yisheng Guan. A graph attention spatio-temporal convolutional networks for 3d human pose estimation in video. *arXiv preprint arXiv:2003.14179*, 2020.

[42] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X*, page 318–334, Berlin, Heidelberg, 2020. Springer-Verlag.

[43] Shengyuan Liu, Pei Lv, Yuzhen Zhang, Jie Fu, Junjin Cheng, Wanqing Li, Bing Zhou, and Mingliang Xu. Semi-dynamic hypergraph neural network for 3d pose estimation. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 782–788. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[44] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural sde: Stabilizing neural ode networks with stochastic noise, 2019.

[45] Yingru Liu, Yucheng Xing, Xuewen Yang, Xin Wang, Jing Shi, Di Jin, and Zhaoyue Chen. Learning continuous-time dynamics by stochastic differential networks, 2020.

[46] Yingru Liu, Yucheng Xing, Xuewen Yang, Xin Wang, Jing Shi, Di Jin, Zhaoyue Chen, and Jacqueline Wu. Continuous-time stochastic differential networks for irregular time series modeling. In Teddy Mantoro, Minho Lee, Media Anugerah Ayu, Kok Wai Wong, and Achmad Nizar Hidayanto, editors, *Neural Information Processing*, pages 343–351, Cham, 2021. Springer International Publishing.

[47] Zhenguang Liu, Haoming Chen, Runyang Feng, Shuang Wu, Shouling Ji, Bailin Yang, and Xun Wang. Deep dual consecutive network for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 525–534, June 2021.

[48] Zhengxiong Luo, Zhicheng Wang, Yan Huang, Liang Wang, Tieniu Tan, and Erjin Zhou. Rethinking the heatmap regression for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13264–13273, June 2021.

[49] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb, 2017.

[50] Rahul Mitra, Nitesh B. Gundavarapu, Abhishek Sharma, and Arjun Jain. Multiview-consistent semi-supervised learning for 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[51] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation, 2016.

[52] Qiang Nie, Ziwei Liu, and Yunhui Liu. Unsupervised 3d human pose representation with viewpoint and pose disentanglement, 2020.

[53] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[54] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. Multi-source deep learning for human pose estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2344, 2014.

[55] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018.

[56] Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations, 2019.

[57] Michael Poli, Stefano Massaroli, Clayton M. Rabideau, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Continuous-depth neural models for dynamic graph prediction, 2021.

[58] Zhongwei Qiu, Kai Qiu, Jianlong Fu, and Dongmei Fu. Dgcn: Dynamic graph convolutional network for efficient multi-person pose estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):11924–11931, Apr. 2020.

[59] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[60] Edoardo Remelli, Shangchen Han, Sina Honari, Pascal Fua, and Robert Wang. Lightweight multi-view 3d pose estimation through camera-disentangled representation. 2020.

[61] Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent odes for irregularly-sampled time series, 2019.

[62] Yulia Rubanova, Tian Qi Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems 32*, pages 5321–5331. 2019.

[63] Luca Schmidtke, Athanasios Vlontzos, Simon Ellershaw, Anna Lukens, Tomoki Arichi, and Bernhard Kainz. Unsupervised human pose estimation through transforming shape templates. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2494, June 2021.

[64] Theodoros Sofianos, Alessio Sampieri, Luca Franco, and Fabio Galasso. Space-time-separable graph convolutional network for pose forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11209–11218, October 2021.

[65] Kai Su, Dongdong Yu, Zhenqi Xu, Xin Geng, and Changhu Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[66] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5686–5696, 2019.

[67] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[68] Zhi Tian, Hao Chen, and Chunhua Shen. Directpose: Direct end-to-end multi-person pose estimation. *ArXiv*, abs/1911.07451, 2019.

[69] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.

[70] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. Voxelpose: Towards multi-camera 3d human pose estimation in wild environment. In *European Conference on Computer Vision (ECCV)*, 2020.

[71] Ali Varamesh and Tinne Tuytelaars. Mixture dense regression for object detection and human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[72] Bastian Wandt, Marco Rudolph, Petrissa Zell, Helge Rhodin, and Bodo Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13294–13304, June 2021.

[73] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11855–11864, June 2021.

[74] Jian Wang, Xiang Long, Yuan Gao, Errui Ding, and Shilei Wen. Graph-pcnn: Two stage human pose estimation with graph pose refinement. In *The European Conference on Computer Vision (ECCV)*, 2020.

[75] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos, 2020.

[76] Manchen Wang, Joseph Tighe, and Davide Modolo. Combining detection and tracking for human pose estimation in videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[77] Rui Wang, Zhongzheng Cao, Xiangyang Wang, Zhi Liu, and Xiaoqiang Zhu. Human pose estimation with deeply learned multi-scale compositional models. *IEEE Access*, 7:71158–71166, 2019.

[78] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[79] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

[80] Rongchang Xie, Chunyu Wang, and Yizhou Wang. Metafuse: A pre-trained fusion model for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[81] Yuliang Xiu, Jiefeng Li, Haoyu Wang, Yinghong Fang, and Cewu Lu. Pose Flow: Efficient online pose tracking. In *BMVC*, 2018.

[82] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[83] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16105–16114, June 2021.

[84] Xixia Xu, Qi Zou, and Xue Lin. Adaptive hypergraph neural network for multi-person pose estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):2955–2963, Jun. 2022.

[85] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition, 2018.

[86] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[87] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[88] Yiding Yang, Zhou Ren, Haoxiang Li, Chunluan Zhou, Xinchao Wang, and Gang Hua. Learning dynamics via graph neural networks for human pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8074–8084, June 2021.

[89] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of he 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[90] Bing Yu, Haoteng Yin, and Zhanxing Zhu. ST-UNet: A spatio-temporal u-network for graph-structured time series modeling. *arXiv*, abs/1903.05631, 2019.

[91] Frank Yu, Mathieu Salzmann, Pascal Fua, and Helge Rhodin. Pcls: Geometry-aware neural reconstruction of 3d pose with perspective crop layers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9064–9073, June 2021.

[92] Ye Yuan, Shih-En Wei, Tomas Simon, Kris Kitani, and Jason Saragih. Simpoe: Simulated character control for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7159–7169, June 2021.

[93] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 507–523, Cham, 2020. Springer International Publishing.

[94] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[95] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[96] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[97] Jianan Zhen, Qi Fang, Jiaming Sun, Wentao Liu, Wei Jiang, Hujun Bao, and Xiaowei Zhou. Smap: Single-shot multi-person absolute 3d pose estimation, 2020.

[98] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11656–11665, October 2021.

[99] Tianfei Zhou, Wenguan Wang, Si Liu, Yi Yang, and Luc Van Gool. Differentiable multi-granularity human representation learning for instance-aware human semantic parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1622–1631, June 2021.

[100] Bin Xiao Zhaoxiang Zhang Jingdong Wang Zigang Geng, Ke Sun. Bottom-up human pose estimation via disentangled keypoint regression. In *CVPR*, 2021.