Full-Body Motion Reconstruction with Sparse Sensing from Graph Perspective

Feiyu Yao[1], Zongkai Wu[2*], Li Yi[3,4,5*]
[1]2012 Lab, Huawei Technologies Co., Ltd
[2]Fancy Technology
[3]Tsinghua University
[4]Shanghai Artificial Intelligence Laboratory
[5]Shanghai Qi Zhi Institute
yaofeiyu1@huawei.com, wuzongkai@fancy.tech, ericyi@mail.tsinghua.edu.cn

Abstract

Estimating 3D full-body pose from sparse sensor data is a pivotal technique employed for the reconstruction of realistic human motions in Augmented Reality and Virtual Reality. However, translating sparse sensor signals into comprehensive human motion remains a challenge since the sparsely distributed sensors in common VR systems fail to capture the motion of full human body. In this paper, we use well-designed Body Pose Graph (BPG) to represent the human body and translate the challenge into a prediction problem of graph missing nodes. Then, we propose a novel full-body motion reconstruction framework based on BPG. To establish BPG, nodes are initially endowed with features extracted from sparse sensor signals. Features from identifiable joint nodes across diverse sensors are amalgamated and processed from both temporal and spatial perspectives. Temporal dynamics are captured using the Temporal Pyramid Structure, while spatial relations in joint movements inform the spatial attributes. The resultant features serve as the foundational elements of the BPG nodes. To further refine the BPG, node features are updated through a graph neural network that incorporates edge reflecting varying joint relations. Our method's effectiveness is evidenced by the attained state-of-the-art performance, particularly in lower body motion, outperforming other baseline methods. Additionally, an ablation study validates the efficacy of each module in our proposed framework.

Introduction

Continuously full-body motion reconstruction from sparse motion sensing is crucial for applications in Augmented Reality and Virtual Reality (AR/VR), which demands highly accurate human motion poses to render vivid avatars in the digital world and do interactions. Common VR systems are composed by head-mounted displays and handheld controllers. These devices can provides resourceful abundant upper body motion information, yet they are unable to provide corresponding lower body motion data. The significant sparsity inherent in known data distribution makes the generation of realistic full-body motion a particularly challenging endeavor for conventional methods based on human kinematics (Company 2018) and matching motions (Ahuja et al. 2021).

Various learning-based methods have been made to generate full-body avatars from sparse inputs in AR/VR (Dittadi et al. 2021) (Du et al. 2023) (Jiang et al. 2022) (Jiang et al. 2022). These methods in these diverse studies essentially entail the extraction of features from sparse sensor data, devoid of considerations for human body joint relationships. Subsequently, these extracted features are integrated into various network architectures that similarly also lack a profound consideration of the interdependence among human body joints. The homogenization of these methodologies confines the development of reconstructing human motion from sparse inputs to the realm of network structure updates. Also, the absence of sufficient human body information contributes to a notable disparity between the reconstructed outcomes of the lower human body and actual motion dynamics.

To solve the problems mentioned above, we consider the human body from graph perspective and propose BPG to represent full body. The task is then transformed to be predicting missing nodes in established BPG. Considering the limited information available on missing nodes, the BPG is initialized and updated referring to node properties. The first stage is processing node features. Position feature and angle feature are fused since they share different transformation law and distribution. Temporal Pyramid Structure is proposed on fusing frame-level and clip-level features to build temporal properties for feature representation. To model spatial properties, features of limb joints and trunk joints are generated separately referring to the human skeleton dynamic. The generated motion features are assigned to be initial features in BPG. In Node Feature Updating stage, the nodes in BPG is updated referring to joint relations. We split the node relations into static skeleton relations, dynamic skeleton relations and latent relations. Then the node features in BPG are updated in Graph Convolution Network with expressive edges generated from node relations.

Our main contributions are summarized as follows:

- We are the first to conduct research on full body pose reconstruction with sparse sensing from graph perspective. The task is viewed as predicting missing nodes in an established graph.
- We propose a framework to reconstruct full body motions via Body Pose Graph (BPG). Motion features with temporal and spatial properties are generated and assigned

*Corresponding authors.

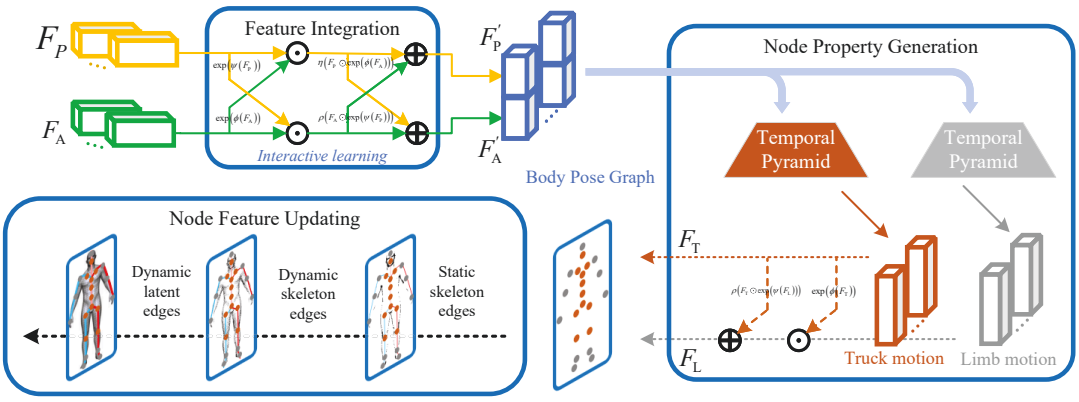

Figure 1: Illustration of our proposed structure. Inputs are sparse sensor position and rotational signals from VR system. Feature Integration module integrates position feature and rotation feature with different physical properties with interactive learning. In Node Property Generation module, motion temporal property is achieved through Temporal Pyramid module. To gain motion spatial property, the limb motion feature is composed by trunk motion features and limb local motion features. The trunk and limb feature then serve as initial node features in Body Pose Graph. In Node Feature Updating, graph convolution network with different edges modeling different joint relations is applied to update nodes.

to be initial node features. Then a Graph Neural Network with expressive edges is applied to updated nodes. Full body motion sequence is generated from the node-associated joint movements.

- Experiment results demonstrate that our framework achieves state of the art performance on on full-body avatar estimation from sparse inputs. Further analysis shows the contribution of each component to the performance improvement, especially in the lower body joints.

Related Work

Full-Body Motion Reconstruction from Sparse Inputs

Primary researches on this area make attempts on fully-body motion reconstruction from 6 IMU sensors on human body (head, arms, pelvis and legs). (von Marcard et al. 2017) proposes a joint optimization framework based on statistic body models. (Huang et al. 2018) applies learning method BiRNN with body models to do the estimation. (Yi, Zhou, and Xu 2021) proposes a multi-stage learning based method where multiple subtasks and losses are designed to restrain pose generation. (Yi et al. 2022) relies on physical models to refine the poses generated from learning methods.

However, requiring six IMUs is still excessive, as they are costly and logistically inconvenient to deploy. Reconstructing full body motion from common VR system will be more advantageous and flexible. (Ahuja et al. 2021) first utilizes sparser inputs from current consumer-grade VR systems (with headset and hand controllers) to estimate full-body motion. It makes full body motion reconstruction much easier. However, it estimates poses based on matching from a dataset with only 5 types of activities. It can hardly handle diverse activities out of dataset. (Yang, Kim, and Lee 2021) proposes a Gated Recurrent Unit - based method to estimate lower-body pose while achieving upper-body with IK solver. It queries the confidence of lower-body motion reconstruction especially when upper-body and lower-body have weak correlations. Thus apart from sensor data from VR devices, it also requires a sensor on human waist. (Dittadi et al. 2021) proposes a VAE(Variance AutoEncoders)-based method to generate poses from VR devices. However, it assumes the directions of pelvis in each frame should be the same. (Jiang et al. 2022) proposes a Transformer-based structure to generate global orientation and local joint orientations. Orientations will then be input to body models to generate joint poses. In the domain of human body reconstruction utilizing only head-mounted devices, several methods have also been developed and explored. (Li, Liu, and Wu 2023) estimates full-body human motions from only egocentric video for diverse scenarios. (Winkler, Won, and Ye 2022) proposes a reinforcement learning based framework and, together with physical simulator, can generate vivid leg motions even when the input is only the 6D transformations of the HMD. Despite these methods' promising performance, leveraging joint motion relations in the human body can likely yield better results. Hence, our work introduces this prior knowledge through a graph-based approach.

Graph Neural Networks

Graph Neural Networks (Kipf and Welling 2017)(Zhang and Chen 2018) process data that can be represented as graph. Nodes representations will be iteratively updated by messages passed from their neighbors. Typical message passing methods include convolution-based methods (Kipf and Welling 2017) and attention-based (Veličković et al. 2018). One research field related to sparsity is handling graphs with missing nodes. (Chen et al. 2020) develops a distribution match based GNN Transformer-like method for attribute-missing graph. (Taguchi, Liu, and Murata 2018) introduces Gaussian mixture model to represent missing data in Graph

Convolutional Network. (Jiang and Zhang 2020) utilizes a partial message-passing method to transmit observed features observed features in GCN-based model. (Rossi et al. 2022) handles missing features in graph by minimizing Dirichlet energy and leading to a diffusion-type differential equation on graph.

Although they handle the graphs missing data, however, the graphs that these methods focus on have three characteristics. First, the node features have great similitudes. Also, node relations tend to be qualitative and simple. And the downstream tasks (for example node prediction) do not rely much on the accurate quantity of node features. A typical example will be graph in bibliographic data. While full-body motion reconstruction from sparse inputs needs accurate quantity node features. The joints also have their own characteristics and the relations between them have motion meanings. The methods mentioned above are unsuitable for handling joint missing human joint graph.

Graph Neural Networks in Human Pose Estimation

While there are currently no methods solving full-body motion estimation from sparse inputs utilizing Graph Neural Networks (GNN), GNN, renowned for their heightened interpretability, have found widespread application in tasks associated with human pose. For example, action recognition (Sofianos et al. 2021; Liu et al. 2020c; Cheng et al. 2020b,a; Chen et al. 2021b; Si et al. 2019; Zhang, Xu, and Tao 2020; Li et al. 2019; Shi et al. 2019b; Chen et al. 2021a; Shi et al. 2019a; Duan et al. 2022; Shi et al. 2020; Ye et al. 2019) and 3D human pose estimation from 2D (Azizi et al. 2022; Liu et al. 2020b,a; Kundu et al. 2021; Zhao et al. 2019; Zou and Tang 2021; Li et al. 2020; Liu, Zou, and Tang 2020).

(Sofianos et al. 2021) is the first work that applies GNN in action recognition. After that, various researches have been proposed on GNN-based method in action recognition. Some focus on improving the graph structure itself. For example, (Cheng et al. 2020b) proposes shift operations and lightweight point-wise convolutions to provide flexible receptive field for graph. (Si et al. 2019) integrates GNN with attention mechanism and lstm to increase representation ability. Some focus on empowering node relations to be more expressive. (Shi et al. 2019b) generates graph edges with directions based on kinematics while (Shi et al. 2020; Ye et al. 2019) generates edges containing temporal information or spatial information by learning methods.

Different from action recognition tasks which focus on pose classification, 3D human pose estimation aims to reduce the estimation errors on all joints in pose. In order to solve this harder task, various methods focus on more powerful graph structure and more contextual edges. The more powerful graph includes graph in Non-euclidean space (Azizi et al. 2022), hypergraph neural networks (Liu et al. 2020b), and so on. More advanced graph updating methods are also proposed, for example (Liu et al. 2020a; Kundu et al. 2021; Zhao et al. 2019; Zou and Tang 2021; Yan, Xiong, and Lin 2018). Also, more human priors are utilized. For example, (Li et al. 2020) establishes dynamic GNN based on human motion prediction. (Liu, Zou, and Tang 2020) reveals the importance of decoupling global information from joints. (Zeng et al. 2021), (Lee and KIM 2022) analyzes the multi-hop relations between human graph nodes and models then in updating methods.

The various methods mentioned above motivate us to introduce graph in fully-body pose estimation from sparse inputs. However, these methods mainly focus on tasks where sparsity hardly exists. Features for almost all joints are supplied as input. While our task only provides sensor data on 3 joints as inputs and expects accurate estimation in 22 joints. Above mentioned methods in related human pose methods can hardly be applied to our task.

Methods

In this section, we first formalize the process of full body motion reconstruction with sparse sensing and understand the task from graph perspective. Then building process of BPG is introduced. After node initialization, BPG is updated referring to several joint relations and all joint motions are generated.

Problem Formulation

This work focuses on full-body motion reconstruction with measurements from one headset and two hand controllers, a common configuration in commercial VR device. The inputs are cartesian coordinates $\mathbf{p}^{1\times 3}$ and orientations in axis-angle representation $\mathbf{\Phi}^{1\times 3}$ of headset and hand controllers. The outputs are local rotation angles between joints and their parent joints θ. considering real-time requirements in application scenarios, the issue is formalized an online problem:

$$\theta_{\mathrm{N}}^{1:\mathrm{F}} = f\left(\{\mathbf{p}^{\mathbf{w}}, \mathbf{\Phi}^{\mathbf{w}}\}_{(\mathrm{N\text{-}K}):\mathrm{N}}^{1:\mathrm{S}}\right), \tag{1}$$

in which $S = 3$ corresponds to the number of joints tracked by the VR system, $F = 22$ is the number of joints used to represent the full-body motion. The movements of joints in current frame (N) are generated from sensor data in previous K frames ((N-K):N). The final full human body can be rendered from the outputs θ with human body model.

From graph perspective, we view the full-body as a graph with 22 nodes. For N-th frame full-body motion reconstruction, motions of 3 nodes in graph are known and used. They are the positional and angular motions of head and two hands in previous K frames. Thus the task is transformed to be completion of the missing 19 nodes in graph. Feature Integration module and Node Property Generation module extract various features from movement sequences of three known nodes, assign features to different nodes and endow these node features with characteristics corresponding to the joints properties.

Feature Integration module integrates different sensor signals as normalization. Node Property Generation module generates features with temporal and spatial properties for nodes as initial value. In Node Feature Updating module, the node features are updated by GCN with expressive edges.

Node Feature Initialization

Given the constraints of a limited number of known nodes and the valuable information associated with them, the

sparse sensor data from sensors undergoes an abstraction process. This process leads to the extraction of features related to the whole body motions. Subsequently, these extracted features are assigned to all nodes, thus serving as the initialization for the graph structure.

Feature Integration The angular measurements and positional measurements have totally different distribution and follows different math laws for transformation, we propose Feature Integration module to fuse them.

The measurements from each VR system device are joint position $\mathbf{p} \in R^{1\times3}$ and joint angular representation $\theta \in R^{1\times6}$ (elements in rotation matrix $R^{3\times3}$ first row and second row). We augment the features by differentiate the measurements and get joint velocity $\mathbf{v} \in v^{1\times3}$ and joint angular velocity representation $\omega_t \in R^{1\times6}$(elements in rotation velocity matrix $R_{t-1}^{-1}R_t$ first row and second row). Joint position and joint velocity compose translation feature $F_{P_{sensor}}$. Joint angular representation and joint angular velocity representation compose joint rotation feature $F_{A_{sensor}}$.

$$F_{P_{sensor}}^{S\times6} = \begin{bmatrix} (\mathbf{p}_t^1, \mathbf{v}_t^1)^{1\times6} \\ \cdots \\ (\mathbf{p}_t^s, \mathbf{v}_t^s)^{1\times6} \end{bmatrix} \tag{2}$$

$$F_{A_{sensor}}^{S\times12} = \begin{bmatrix} (\theta_t^1, \omega_t^1)^{1\times12} \\ \cdots \\ (\theta_t^s, \omega_t^s)^{1\times12} \end{bmatrix} \tag{3}$$

$F_{P_{sensor}}$ and $F_{A_{sensor}}$ describes the joint motion from different perspective and shares totally different geometric properties. Thus, to enhance the feature representation ability and eliminate feature geometric differences, we utilize dual interactive learning to generate new feature referring to (Zhu et al. 2020).

$$\begin{aligned} F_{\mathrm{P}}' &= F_{\mathrm{P}} \odot \exp\left(\phi\left(F_{\mathrm{A}}\right)\right) - \rho\left(F_{\mathrm{A}} \odot \exp\left(\psi\left(F_{\mathrm{P}}\right)\right)\right) \\ F_{\mathrm{A}}' &= F_{\mathrm{A}} \odot \exp\left(\psi\left(F_{\mathrm{P}}\right)\right) + \eta\left(F_{\mathrm{P}} \odot \exp\left(\phi\left(F_{\mathrm{A}}\right)\right)\right) \end{aligned} \tag{4}$$

ψ, ρ are 1D convolutional layers. exp is used to map different features onto similar distribution spaces. The Feature Integration module is applied to integrate the motion information of nodes and output fused node features that incorporate both positional and angular information.

Node Temporal Motion Property Generation As stated in the Problem Formulation, the framework's input comprises motion information from k preceding frames, while the output entails the motion of the current frame. Thus this module is designed to mitigate temporal disparities in features originating from different frames. Also, As highlighted in (Martinez, Black, and Romero 2017), motion continuity stands out as a distinct characteristic of human motion. Motion details within each frame can be inferred from the surrounding contextual frames (clip). In this section, we apply the modeling of motion continuity as a guidance to mitigate temporal disparities.

For node temporal properties, to better capture the joint motion temporal properties, we design Temporal Pyramid Structure in Figure 2. The inputs are k previous frame features and the outputs are high dimensional temporal features of current frame. The feature extractor is based on SCI-Block (Liu et al. 2022), which is a CNN-based time series model with output dimension adjustable. In Temporal Pyramid Structure, three Feature extractor are applied. First one and second one extract frame level and clip level features. The two level features are concatenated in an interleaved manner. The third extractor is applied to generate motion features for current frame.

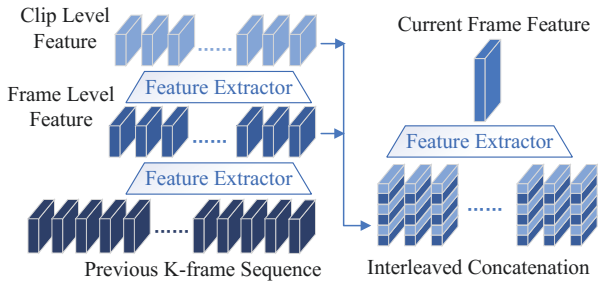

Figure 2: Temporal Pyramid Structure

Node Spatial Motion Property Generation Human motions in different joints have different spatial properties. (Leteneur et al. 2013) claims the important impact of trunk on human body motion and reveals the different property of trunk and limb joints. As the human skeleton kinematic (Shi et al. 2019b) (Hu et al. 2021) reveals, joints farther from the center of the human body are always physically controlled by an adjacent joint which is closer to the center. In this context, limb joints act as child joints relative to trunk joints, resulting in limb joint motions being composed of both local limb joint movements and trunk joint motions. To address the challenge of predicting joints located distantly from the body's center and to capture this directed control relationship, we propose a unidirectional interactive learning approach. This method guides the extraction of limb motion features by leveraging the guidance from trunk motion features. This module's mechanism is described as followed.

$$F_{\mathrm{L}} = F_{\mathrm{L}} \odot \exp\left(\phi\left(F_{\mathrm{T}}\right)\right) + \rho\left(F_{\mathrm{T}} \odot \exp\left(\psi\left(F_{\mathrm{L}}\right)\right)\right) \tag{5}$$

F_T and F_L are trunk motion features and limb motion features generated by temporal pyramid separately. ϕ and ψ are convolutional networks for generating sub-structure level features. The interactive learning mechanism used here is similar to Feature Integration module. The trunk and limb motion features generated are then assigned to corresponding trunk nodes (joint 0,1,2,3,4,5,6,9,12,13,14 in Figure 3) and limb nodes (joint 7,8,10,11,15,16,17,18,19,20,21 in Figure 3) as initial features.

Node Feature Updating

Node Feature Updating aims to capture diverse joint relationships through a Graph Convolutional Network with expressive edges. In Section 1, we initially revisit the vanilla

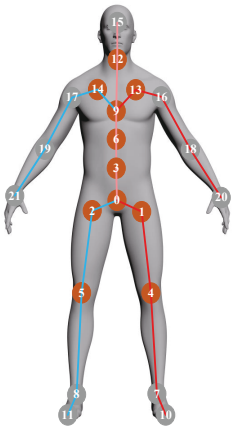

Figure 3: Index of human body joints

Graph Convolutional Network, updating joint features considering the static human skeleton as edges. Subsequently, Section 2 introduces Node Updating using a graph convolutional network with expressive edges.

Vanilla GCN We will first review the vanilla graph convolution network. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, it consists of the nodes $\mathcal{V}$ and the edges $\mathcal{E}$. We revisit a generic GCN layer defined as follows:

$$\mathbf{X}^{'} = \sigma(\mathbf{W}\mathbf{X}\mathbf{A}) \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is an adjacency matrix with N nodes, indicating the connections between nodes. In 2D-3D human pose estimation, a task predicts 3d coordinates from pictures, the adjacency matrix is often established referring to human skeleton. If there is a bone connection between joint j and the joint i, then $a_{ij} = 1$. Otherwise, the value will be set to zero $a_{ij} = 0$. We denote the input node features as $\mathbf{X} \in \mathbb{R}^{N\times C_{in}}$. Each node corresponding to a C_{in}-dimensional feature vector. The learnable weight matrix $\mathbf{W} \in \mathbb{R}^{C_{in}\times C_{\text{out}}}$ is set to adjust feature's dimension to be expected. The $\sigma(\cdot)$ is common activation function.

Consider i-th node, the node feature of node i is X_i. Corresponding adjacency matrix slice will be A_i, with j-th element being a_{ij}. S_i represents the joints have bone connections with joint i. $a_{ij} = 1$ if $j \in S_i$ and $a_{ij} = 0$ if $j \notin S_i$. The update of i-th node in vanilla GCN is expressed as:

$$\mathbf{X}_i' = \sigma\left(\sum_{j\in S_i} \mathbf{W} X_j a_{ij}\right) \tag{7}$$

Although lots of improvements has been made in past, the Graph updating method updates each node synchronously, which assumes that useful information is evenly distributed and the confidence of joint features is at the same level.

Node Updating with Expressive Edges Vanilla graph convolution network has limited representation ability. It assumes that features in each node is reliable and node can be represented well by updates referring to constant graph edges built by human skeletons. Also, other strong hidden relationships among joint nodes exist and change with actions, these relations can hardly be modeled by vanilla GCN. For example, when the human is running, there is a strong relation between hand joint and foot joint. But when the human is sitting, there is no such strong relation. Considering above two limitations, we proposed GCN with multiple kinds of edge learned. To be specific, the edges in graph are dynamic corresponding to the current joint state instead of being constant. When human action changes, the edges can change simultaneously to better represent the node relations.

In our task, edges are represented as an adjacency matrix $\mathbf{A}^h \in \mathbb{R}^{22\times 22}$.

$$\mathbf{A}^h = \mathbf{A}^s + \mathbf{A}^l \tag{8}$$

$$\mathbf{A}^s = \mathbf{A}^{ss} + \mathbf{A}^{ds} \tag{9}$$

$\mathbf{A}^s \in \mathbb{R}^{22\times 22}$ is skeleton relation adjacency matrix, it describes the relations exist in human skeleton (to be specific, all the edges drawn in Figure 3). $\mathbf{A}^l \in \mathbb{R}^{22\times 22}$ is latent relation adjacency matrix, it describes potential links between nodes (node links not exist in Figure 3). Static skeleton relation adjacency matrix $\mathbf{A}^{ss} \in \mathbb{R}^{22\times 22}$ is built referring to the human skeleton of SMPL model. Joints connected in human skeleton will have edges with non-zero constant value in the corresponding place. Dynamic skeleton relation adjacency matrix $\mathbf{A}^{ds} \in \mathbb{R}^{22\times 22}$ models the relations among joints in skeleton. It is also built referring to human skeleton of SMPL. However, the value of the edges will be determined by the features of nodes in graph. The values in $\mathbf{A}^{ds}$ and $\mathbf{A}^l$ are learned by MLP structure seperately.

$$A = W_1\phi(W_0X + B_0) + B_1 \tag{10}$$

in which, $X \in \mathbb{R}^{b\times nf}$, $W_0 \in \mathbb{R}^{h\times nf}$, $B_0 \in \mathbb{R}^h$, $W_1 \in \mathbb{R}^{o\times h}$, $B_1 \in \mathbb{R}^o$. X is joint feature. b is batch size, n is number of nodes and f is the dimension of feature. h is the dimension of the hidden layer. o is the dimension of output. ϕ is the ReLU activation function.

The nodes are updated with above adjacency matrix. The final output of BPG are the axis-angle of each joint, which, togther with SMPL human model (Pavlakos et al. 2019), will be referred to generate the position of each joint .

Training and Loss

The loss function is composed of rotational loss, positional loss and bone symmetric loss.

$$L_{\text{final}} = L_{\text{rot}} + L_{\text{pos}} + L_{\text{bone}} \tag{11}$$

L_{rot} is absolute error loss on all joint axis-angles. L_{pos} is absolute error loss on all joint positions. The accuracy of axis-angle and position of each joint is both crucial for full body reconstruction. (Jiang et al. 2022). L_{bone} is human skeleton symmetric loss. It emphasises the relative position relations among joints and introduce human body priors for optimization.

$$L_{bone} = \sum_{i_l,j_l,i_r,j_r} (||\hat{Y}^{pos}_{i_l} - \hat{Y}^{pos}_{j_l}|| - ||\hat{Y}^{pos}_{i_r} - \hat{Y}^{pos}_{j_r}||) \tag{12}$$

Methods	MPJRE	MPJPE	MPJVE
Final IK	16.77	18.09	59.24
CoolMoves	5.20	7.83	100.54
LoBSTr	10.69	9.02	44.97
AvatarPoser	3.21	4.18	29.40
AvatarPoser *	3.01	4.11	27.79
Our method *	**2.49**	**3.34**	**22.84**

Table 1: Performance Comparison among our method and baselines in AMASS dataset. Notice that * means the results are trained in our machine.

Here, $\hat{Y}_i^{pos}$ represents the predicted position of joint i. $(i_l, j_l) \in set_r$ are right human skeleton bones shown as blue lines in Figure 3, $(i_r, j_r) \in set_l$ are left human skeleton bones shown as red lines in Figure 3.

Experiemnt

Data Preparation and Evaluation Metrics

CMU (Lab 2000), BMLrub (Troje 2002) and HDM05 (Müller et al. 2007) in AMASS (Mahmood et al. 2019) dataset are employed . The datasets are randomly partitioned into training and testing subsets, comprising 90% and 10% of the data respectively, following the same setting as (Jiang et al. 2022).

The metrics utilized for overall performance comparison are MPJRE (Mean Per Joint Rotation Error $[^\circ]$), MPJPE (Mean Per Joint Position Error $[\mathrm{cm}]$), and MPJVE (Mean Per Joint Velocity Error $[\mathrm{cm/s}]$). In ablation study, to reveal the effect of each component on motion reconstruction, we list the estimated position error on each lower body joint.

Performance Comparison with Baseline Method

We compare our method with baseline methods in Table 1. To be specific, there are Final IK, CoolMoves(Ahuja et al. 2021), LoBSTr (Yang, Kim, and Lee 2021), VAE-HMD (Dittadi et al. 2021), AvatarPoser (Jiang et al. 2022). The results are referring to (Jiang et al. 2022). To be fair, we retrained AvatarPoser on our platform. Our method attains superior results across all three metrics, outperforming all other methods. By representing human body as Graph and modeling spatial-temporal relations among joints, our method surpasses than baseline method, notably in predicting unseen lower body joints, as illustrated in Table 3.

Performance Comparison with Offline Method

Offline methods refer to methods outputting n length human pose sequence instead single frame in each inference. AGRoL (Du et al. 2023) is the state-of-art Offline method. In our method, we use 41 frame sensor sequence as input and ouput 1 frame in each inference. In Table 2, AGRoL_{41} represents that the lengths of input sequence and output sequence are both 41. Thanks to the feature generation method and graph based architecture, which are special designed for human body, our method performs better than AGRoL in all criteria in same condition. When extending the output sequence length to 192, AGRoL demonstrates commendable performance in MPJVE metric. However, this enhancement of the MPJVE metrics did not translate into superior results for the MPJRE and MPJPE metrics, which are more important for the restructure task. In contrast, our proposed methodology exhibits superior performance in both the MPJRE and MPJPE metrics, further substantiating its efficacy.

Methods	MPJRE	MPJPE	MPJVE
VAE-HMD	4.11	6.83	37.99
AGRoL $_{41}$	2.59	3.64	23.24
AGRoL $_{196}$	2.66	3.71	**18.59**
Our method	**2.49**	**3.34**	22.84

Table 2: Performance Comparison with offline methods

Index	AvatarPoser	Our method	Improvement
Joint 1	3.8	3.1	18.84%
Joint 2	3.8	3.2	15.79%
Joint 4	6.9	5.5	20.29%
Joint 5	6.9	5.5	20.29%
Joint 7	10.1	7.9	21.78%
Joint 8	10.1	8.0	20.80%
Joint 10	10.8	8.4	22.22%
Joint 11	11.0	8.7	20.91%
All Joints	4.11	3.34	18.73%

Table 3: Joint Position Error performance comparison between our method and AvatarPoser in lower body joints [cm]

Ablation Study

To dissect individual component functions, we conduct ablation studies across various cases. Findings are outlined in Table 4. Given our method's targeted focus on mitigating substantial estimation errors in lower body joints, we employ MPJPE-lower-body (MPJPE on joints 1, 2, 4, 5, 7, 8, 10, 11) to directly exemplify performance.

- No Bone Symmetric Loss: The bone symmetric loss is not utilized in the framework.
- No Spatial Property: The node features are not generated separately for trunk joints and limb joints and the relations between trunk and limb is not considered.
- No Temporal Property: The temporal pyramid structure is replaced by temporal feature extractor.
- No Feature Initialization: The Feature Integration process is replaced by simple MLP structure.
- Vanilla GCN: The nodes in BPG are updated through Vanilla GCN instead of the GCN with expressive edges.

As evident from Table 4, the absence of modules induces notable performance declines, particularly in the lower body region. This proves the efficacy of each component.

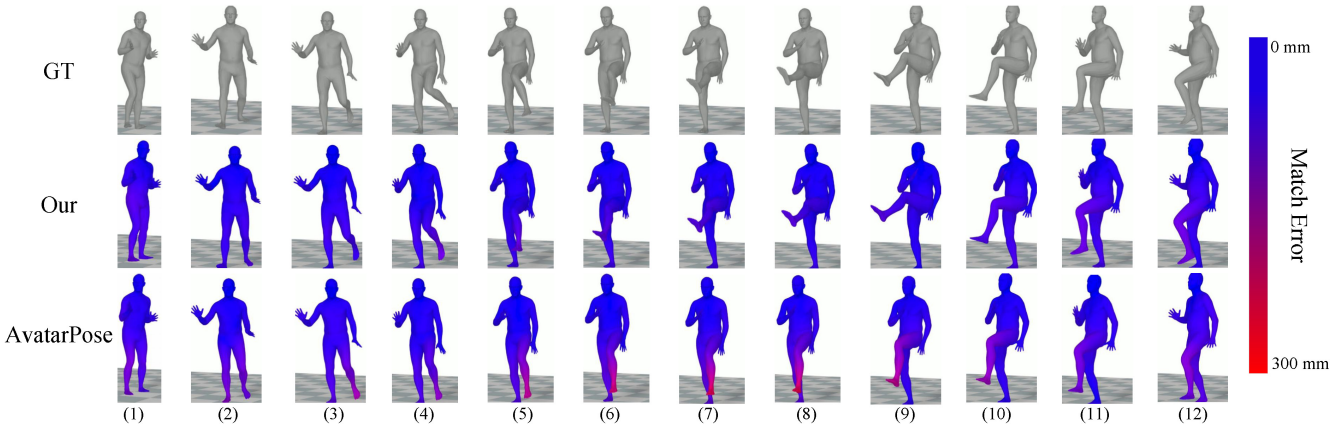

Figure 4: Visualization of estimated poses on an avatar involves a series of frames portraying a human front kick action. It encompasses three rows: the top row showcases avatars with ground truth (GT) poses, while the subsequent two rows display avatars generated by our approach and AvatarPoser. These avatars are color-coded to denote errors in each mesh.

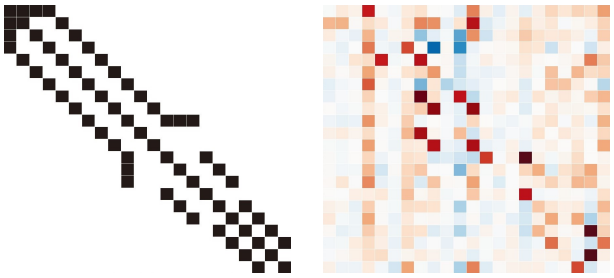

Figure 5: Left diagram depicts a 0-1 adjacency matrix representation of the skeletal connectivity within the human body. Conversely, right diagram showcases an adjacency matrix generated by the GCN with expressive edges. The deeper the color, the stronger the relationship between the nodes. Red indicates positive correlation, while blue indicates negative correlation.

Configuration	MPJPE	MPJPE-lower-body
No Bone Symmetric Loss	3.53	6.75
No Spatial Property	3.60	6.88
No Temporal Property	3.71	7.10
No Feature Initialization	3.53	6.80
Vanilla GCN	3.58	6.88
Default	3.34	6.29

Table 4: Ablation study

Visualization of Estimated Pose on Avatar

In order to better analyze the estimation performance, we visualize the estimated poses on the whole avatar in figure 4. Each mesh triangle in avatar is rendered referring to the error of each estimated mesh vertex. Red represents large mesh vertex estimation error. The avatars in the first row show the ground truth poses. The avatars in second row and the third row are generated by our proposed method and baseline. As can be seen, the avatar generated by our method accomplishes the whole process of lifting and lowering the leg with little mesh error while the one generated by AvatarPoser accomplishes the action with errors and stiffness. Especially in frames (5) (6) (7) (8) (9), avatars generated by AvatarPoser can hardly even raise left leg as high as the ground truth.

Analysis of Expressive Edges

As shown in Figure 5, the adjacency matrix generated by GCN with expressive edges (right-hand figure) shows more joint relations than the static 0-1 adjacency matrix generated from the quantification of the human skeletal structure (left-hand figure). This indicated that, benefiting from the potent expressive capabilities of of GCN with expressive edges, our approach has yielded more comprehensive joint relationships in than human skeleton.

Conclusion

In this study, we approach the task of full-body motion reconstruction from sparse sensor input through a graph-based perspective, introducing the Body Pose Graph to represent the human body. In Node Feature Initialization step, different kind of VR system device features are first integrated. The new generated features are then processed to achieve spatial properties and temporal properties of joint motions before serving as initial node features in Body Pose Graph. Temporal property is generated by Temporal Pyramid Structure and Spatial property is generated referring to joint motion spatial relations. In the Node Feature Updating stage, we employ GNN with expressive edges to update node features within the Body Pose Graph. Our approach demonstrates exceptional estimation performance as evidenced by comprehensive evaluations. Ablation studies validate the effectiveness of individual components. Visualizations of learned edges and estimated poses on avatars provide insights into learned motion relationships and our method's prowess in mesh-scale representations.

References

Ahuja, K.; Ofek, E.; Gonzalez-Franco, M.; Holz, C.; and Wilson, A. D. 2021. CoolMoves: User Motion Accentuation in Virtual Reality. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, volume 5, 1–23.

Azizi, N.; Possegger, H.; Rodolà, E.; and Bischof, H. 2022. 3D Human Pose Estimation Using Möbius Graph Convolutional Networks. In *Proceedings of the European Conference on Computer Vision*, 160–178.

Chen, X.; Chen, S.; Yao, J.; Zheng, H.; Zhang, Y.; and Tsang, I. W. 2020. Learning on Attribute-Missing Graph. *IEEE transactions on pattern analysis and machine intelligence*, 44(2): 740–757.

Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; and Hu, W. 2021a. Channel-Wise Topology Refinement Graph Convolution for Skeleton-Based Action Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13359–13368.

Chen, Z.; Li, S.; Yang, B.; Li, Q.; and Liu, H. 2021b. Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 1113–1122.

Cheng, K.; Zhang, Y.; Cao, C.; Shi, L.; Cheng, J.; and Lu, H. 2020a. Decoupling GCN with DropGraph Module for Skeleton-Based Action Recognition. In *Proceedings of the European Conference on Computer Vision*, 536–553.

Cheng, K.; Zhang, Y.; He, X.; Chen, W.; Cheng, J.; and Lu, H. 2020b. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 183–192.

Company, R. 2018. Final IK. https://assetstore.unity.com/packages/tools/animation/final-ik-14290.

Dittadi, A.; Dziadzio, S.; Cosker, D.; Lundell, B.; Cashman, T.; and Shotton, J. 2021. Full-Body Motion from a Single Head-Mounted Device: Generating SMPL Poses from Partial Observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11687–11697.

Du, Y.; Kips, R.; Pumarola, A.; Starke, S.; Thabet, A.; and Sanakoyeu, A. 2023. Avatars Grow Legs: Generating Smooth Human Motion From Sparse Tracking Inputs With Diffusion Model. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 481–490.

Duan, H.; Zhao, Y.; Chen, K.; Lin, D.; and Dai, B. 2022. Revisiting Skeleton-Based Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2969–2978.

Hu, W.; Zhang, C.; Zhan, F.; Zhang, L.; and Wong, T.-T. 2021. onditional Directed Graph Convolution for 3D Human Pose Estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, 602–611.

Huang, Y.; Kaufmann, M.; Aksan, E.; Black, M. J.; Hilliges, O.; and Pons-Moll, G. 2018. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6): 1–15.

Jiang, B.; and Zhang, Z. 2020. Incomplete graph representation and learning via partial graph neural networks. arXiv:2003.10130.

Jiang, J.; Streli, P.; Qiu, H.; Fender, A.; Laich, L.; Snape, P.; and Holz, C. 2022. AvatarPoser: Articulated Full-Body Pose Tracking from Sparse Motion Sensing. In *Proceedings of the European Conference on Computer Vision*, 443–460.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kundu, J. N.; Seth, S.; Jamkhandi, A.; YM, P.; Jampani, V.; Chakraborty, A.; and R, V. B. 2021. Non-local Latent Relation Distillation for Self-Adaptive 3D Human Pose Estimation. In *Advances in Neural Information Processing Systems*, volume 34, 158–171.

Lab, C. G. 2000. CMU Graphics Lab Motion Capture Database. http://mocap.cs.cmu.edu/.

Lee, J. Y.; and KIM, I. 2022. Multi-hop Modulated Graph Convolutional Networks for 3D Human Pose Estimation. In *British Machine Vision Conference*.

Leteneur, S.; Simoneau, E.; Gillet, C.; Dessery, Y.; and Barbier, F. 2013. Trunk's natural inclination influences stance limb kinetics, but not body kinematics, during gait initiation in able men. *PloS one*, (1): e55256.

Li, J.; Liu, K.; and Wu, J. 2023. Ego-Body Pose Estimation via Ego-Head Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17142–17151.

Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; and Tian, Q. 2019. Actional-structural graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3595–3603.

Li, M.; Chen, S.; Zhao, Y.; Zhang, Y.; Wang, Y.; and Tian, Q. 2020. Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 214–223.

Liu, K.; Ding, R.; Zou, Z.; Wang, L.; and Tang, W. 2020a. A Comprehensive Study of Weight Sharing in Graph Networks for 3D Human Pose Estimation. In *Proceedings of the European Conference on Computer Vision*, 318–334.

Liu, K.; Zou, Z.; and Tang, W. 2020. Learning Global Pose Features in Graph Convolutional Networks for 3D Human Pose Estimation. In *Proceedings of the Asian Conference on Computer Vision*, 1429–1442.

Liu, M.; Zeng, A.; Chen, M.; Xu, Z.; Lai, Q.; Ma, L.; and Xu, Q. 2022. SCINet: Time Series Modeling and Forecasting with Sample Convolution and Interaction. In *Advances in Neural Information Processing Systems*, 5816–5828.

Liu, S.; Lv, P.; Zhang, Y.; Fu, J.; Cheng, J.; Li, W.; Zhou, B.; and Xu, M. 2020b. Semi-Dynamic Hypergraph Neural Network for 3D Pose Estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 782–788.

Liu, Z.; Zhang, H.; Chen, Z.; Wang, Z.; and Ouyang, W. 2020c. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 143–152.

Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; and Black, M. J. 2019. AMASS: Archive of motion capture as surface shapes. *Proceedings of the IEEE/CVF international conference on computer vision*, 5442–5451.

Martinez, J.; Black, M. J.; and Romero, J. 2017. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2891–2900.

Müller, M.; Röder, T.; Clausen, M.; Eberhardt, B.; Krüger, B.; and Weber, A. 2007. Documentation Mocap Database HDM05. Technical Report CG-2007-2, Institut für Informatik II, Universität Bonn.

Pavlakos, G.; Choutas, V.; Ghorbani, N.; Bolkart, T.; Osman, A. A. A.; Tzionas, D.; and Black, M. J. 2019. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 10975–10985.

Rossi, E.; Kenlay, H.; Gorinova, M. I.; Chamberlain, B. P.; Dong, X.; and Bronstein, M. M. 2022. On the Unreasonable Effectiveness of Feature Propagation in Learning on Graphs With Missing Node Features. In *Learning on Graphs Conference*, volume 198, 11:1–11:16.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019a. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12026–12035.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2019b. Skeleton-based action recognition with directed graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7912–7921.

Shi, L.; Zhang, Y.; Cheng, J.; and Lu, H. 2020. Skeleton-Based Action Recognition With Multi-Stream Adaptive Graph Convolutional Networks. *IEEE Transactions on Image Processing*, 29: 9532–9545.

Si, C.; Chen, W.; Wang, W.; Wang, L.; and Tan, T. 2019. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1227–1236.

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

Taguchi, H.; Liu, X.; and Murata, T. 2018. Graph convolutional networks for graphs containing missing features. *Future Generation Computer Systems*, 117: 155–168.

Troje, N. F. 2002. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5): 2–2.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

von Marcard, T.; Rosenhahn, B.; Black, M. J.; and Pons-Moll, G. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Computer graphics forum*, 36(2): 349–360.

Winkler, A.; Won, J.; and Ye, Y. 2022. QuestSim: Human Motion Tracking from Sparse Sensors with Simulated Avatars. In *SIGGRAPH Asia 2022 Conference Papers*, 1–8.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Yang, D.; Kim, D.; and Lee, S.-H. 2021. Lobstr: Real-time lower-body pose prediction from sparse upper-body tracking signals. *Computer Graphics Forum*, 40(2): 265–275.

Ye, F.; Pu, S.; Zhong, Q.; Li, C.; Xie, D.; and Tang, H. 2019. Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In *Proceedings of the 28th ACM international conference on multimedia*, 55–63.

Yi, X.; Zhou, Y.; Habermann, M.; Shimada, S.; Golyanik, V.; Theobalt, C.; and Xu, F. 2022. Physical Inertial Poser (PIP): Physics-aware Real-time Human Motion Tracking from Sparse Inertial Sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13167–13178.

Yi, X.; Zhou, Y.; and Xu, F. 2021. TransPose: Real-time 3D Human Translation and Pose Estimation with Six Inertial Sensors. *ACM Transactions on Graphics (TOG)*, 40(4): 1–13.

Zeng, A.; Sun, X.; Yang, L.; Zhao, N.; Liu, M.; and Xu, Q. 2021. Learning skeletal graph neural networks for hard 3d pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11436–11445.

Zhang, M.; and Chen, Y. 2018. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems*, 5165–5175.

Zhang, X.; Xu, C.; and Tao, D. 2020. Context aware graph convolution for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14333–14342.

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D. N. 2019. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3425–3435.

Zhu, S.; Pan, S.; Zhou, C.; Wu, J.; Cao, Y.; and Wang, B. 2020. Graph geometry interaction learning. In *Advances in Neural Information Processing Systems*, volume 33, 7548–7558.

Zou, Z.; and Tang, W. 2021. Modulated Graph Convolutional Network for 3D Human Pose Estimation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 11477–11487.