ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240479

Quater-GCN: Enhancing 3D Human Pose Estimation with Orientation and Semi-Supervised Training

Xingyu Song, Zhan Li, Shi Chen and Kazuyuki Demachi

The University of Tokyo
songxingyu0429@gmail.com, {lizhan, shichen, yypr9411}@g.ecc.u-tokyo.ac.jp

Abstract. 3D human pose estimation is a vital task in computer vision, involving the prediction of human joint positions from images or videos to reconstruct a skeleton of a human in three-dimensional space. This technology is pivotal in various fields, including animation, security, human-computer interaction, and automotive safety, where it promotes both technological progress and enhanced human well-being. The advent of deep learning significantly advances the performance of 3D pose estimation by incorporating temporal information for predicting the spatial positions of human joints. However, traditional methods often fall short as they primarily focus on the spatial coordinates of joints and overlook the orientation and rotation of the connecting bones, which are crucial for a comprehensive understanding of human pose in 3D space. To address these limitations, we introduce Quater-GCN (Q-GCN), a directed graph convolutional network tailored to enhance pose estimation by orientation. Q-GCN excels by not only capturing the spatial dependencies among node joints through their coordinates but also integrating the dynamic context of bone rotations in 2D space. This approach enables a more sophisticated representation of human poses by also regressing the orientation of each bone in 3D space, moving beyond mere coordinate prediction. Furthermore, we complement our model with a semi-supervised training strategy that leverages unlabeled data, addressing the challenge of limited orientation ground truth data. Through comprehensive evaluations, Q-GCN has demonstrated outstanding performance against current state-of-the-art methods. The full version of this paper, along with the code and data, can be found at [39].

1 Introduction

3D human pose estimation is a critical task aimed at predicting the spatial positions of human joints within three-dimensional space. This process is foundational for various applications, including action recognition [48, 16], synthetic data augmentation [41], and 3D reconstruction [31, 22]. The task of 3D human pose estimation can be categorized based on viewpoint settings into multi-view and monocular views. The monocular view, in comparison to the multi-view approach, offers advantages in terms of lower equipment costs and greater flexibility for real-world applications. However, the reduced accuracy of monocular estimation presents significant challenges, thereby garnering increased interest.

Recently, the development of deep learning model has significantly improved the outcomes of 3D human pose estimation [50, 24, 32, 27]. Current deep learning-based techniques fall into two main categories: one-stage methods [26, 28, 45] and 2D-to-3D methods [21, 27, 32]. One-stage methods directly derive the 3D coordinates to construct the human pose from images in an end-to-end manner. Conversely, 2D-to-3D methods initiate by estimating the 2D pose from the input image, which is then extrapolated into 3D space. Comparing to one-stage methods, the superior performance of 2D-to-3D methods is attributed to advancements in 2D human pose detection and the leveraging of temporal information across multiple frames. Moreover, the intermediate 2D pose estimation stage significantly lowers the data volume and simplifies the complexity of the 3D estimation task. Thus, we are motivated to explore the potential of enhancing 3D human pose prediction through high-quality estimated 2D pose data.

However, traditional deep learning models for 2D-to-3D pose lifting focus on the spatial coordinates of joints solely which do not explicitly model the orientation or the rotation of the bones connecting these joints. This orientation information is crucial for understanding the pose in three-dimensional space, as it provides insights into the direction in which a limb is facing or moving. In addition, for complex scenarios where multiple body parts are closely interacting or occluded, the lack of orientation information can lead to ambiguities that are difficult to resolve based on position information alone. These ambiguities can result in less robust pose estimations in challenging situations (see Section 4 for details).

Therefore, determining the optimal model architecture to encode the structural information of human body, including both position and orientation information, is our primary focus. Within the domain of 2D-to-3D pose estimation approaches, recent advancements in deep learning can be categorized into three distinct architectures: Temporal Convolutional Networks (TCN)-based architectures [32, 21], Transformer architectures [20, 52, 19], and Graph Convolutional Networks (GCN)-based architectures [50, 13, 53]. Comparing to another two kinds of architectures, GCN-based architecture stand out for their ability to explicitly maintain the structural integrity of both 2D and 3D human poses throughout the convolutional process, and a more parameter-efficient process, especially when dealing with graph-structured data both on coordinate and orientation. These distinctive capabilities of GCN-based models to conserve the pose structure during estimation highlights their potential for achieving more refined outcomes in 3D pose estimation tasks (see Section 2 for details).

In this paper, we introduce an innovative Directed Graph Convolutional Network, named Quater-GCN (Q-GCN). The Q-GCN not only captures spatial dependencies by analyzing the positions of each node joint but also integrates dynamic context through examining the rotation of each bone in the 2D space. Similarly, as for pose construc-

tion, our method extends beyond solely predicting the coordinates of each joint, it also infers the orientation of each bone in 3D space, crafting a more sophisticated pose representation.

However, regressing the prediction using insufficient orientation ground truth proves challenging. Additionally, calculating the 4D Orientation of each bone joint using only the 3D coordinates of each joint node can be challenging and typically yields incomplete results due to the absence of directional information in space [8]. Moreover, gathering precise orientation data typically requires an expensive motion capture setup [32]. Therefore, we have developed a semi-supervised training strategy that effectively uses unlabeled data by mapping the predicted 4D Orientations back to the rotations in 2D space of each bone joint.

The primary contributions of our work can be summarized in three key aspects: (1) The introduction of a distinctive 2D-to-3D pose lifting method that incorporates bone joint orientations, significantly enhancing model performance. (2) The development of a semi-supervised training approach, ingeniously leveraging unlabeled data to overcome the scarcity of orientation training data. (3) Demonstrated improvements in 3D pose estimation accuracy over existing state-of-the-art methods.

2 Related Work

2.1 2D-to-3D Pose Lifting

Recent advancements in 2D-to-3D pose lifting models can be categorized into three primary types: TCN-, Transformer-, and GCN-based architectures. TCN and Transformer methods are known for their broad receptive fields, effectively processing long 2D pose sequences through strided convolutions. TCN-based methods like [32, 21] have enhanced pose lifting with their architecture designs. For example, [32] introduces a semi-supervised technique called back-projection that uses unlabeled video data to boost accuracy. [21] incorporates an attention mechanism and multi-scale dilated convolutions to address temporal inconsistency and improve accuracy by focusing on key frames. Transformer-based approaches [20, 52, 19] also show promise by utilizing strided structures to handle depth ambiguities and improve spatial and temporal feature encoding. This technique enables these models to cover the full video sequence, enhancing the potential for accurate pose estimation.

In contrast, GCN-based models are superior in maintaining the structural integrity of both 2D and 3D poses, preserving joint relationships and offering parameter efficiency, especially with graph-structured data. The foundational work on GCNs [15] and further developments like ST-GCN [48], which introduced spatial temporal graph convolution for action recognition, and DGCN [37], which models skeleton data as a directed acyclic graph, highlight their capabilities. Recent implementations in pose lifting [50, 13, 53, 43] demonstrate GCNs' versatility. Notably, [13] and [50] explore advanced graph convolutional techniques that address dynamic joint dependencies and enhance local feature detection, while [53] focuses on learning semantic relationships between nodes.

2.2 Orientation-based Motion Representation

The concept of orientation first emerged in human motion representation through mesh generation tasks. The Skinned Multi-Person Linear model (SMPL) [22] serves as a foundational approach in human motion capture, introducing orientation-based representation of the human body. Inspired by SMPL, subsequent models such as CAPE [25], MANO [34], SMPL-X [31], and STAR [29] have extended the framework to include detailed body shape modeling, facial expressions, hand movements, and the depiction of clothed human figures.

In the realm of non-parametric models, OriNet [23] employs limb orientations to depict 3D poses, coupling the orientation with the bounding box of each limb region to enhance the correlation between images and predictions. Yet, the orientation error does not regress through iteration. [8] presents an innovative method for estimating the complete position and rotation of skeletal joints. It utilizes virtual markers to provide ample data, allowing for the accurate deduction of rotations with straightforward post-processing steps.

In 3D human pose estimation task, [9] first proposes a framework based on Mask Region-based Convolutional Neural Networks (R-CNN) and extended to integrate the joint feature, body boundary, body orientation and occlusion condition together. PONet [44] then estimates the 4D Orientation of these limbs by taking advantage of the local image evidence to recover the 3D pose. Similarly, PedRecNet [2] supports body and head orientation estimation based on full body bounding box input.

2.3 Semi-supervised Training

Semi-supervised learning combines a small amount of labeled data with a large amount of unlabeled data during training. In the context of pose representation, Generative Adversarial Networks (GANs) are valuable for data augmentation, helping to distinguish realistic poses from unrealistic ones in datasets where only 2D annotations are available. For instance, [40] employs GANs to work with unpaired 2D/3D datasets and includes a 2D projection consistency term to ensure accuracy. [49] introduces a novel multi-source discriminator that differentiates between predicted 3D poses and ground-truth data from real-world images. Additionally, [6] describes a weakly supervised method for 3D pose estimation using an adversarial setup with a new Random Projection layer. [30] recommends using the ordinal depths of human joints as a minimal supervision signal to make use of the variety found in 2D human pose datasets. Simultaneously, [32] details back-projection, an effective semi-supervised training method that utilizes unlabeled video data to train GCNs for 3D pose estimation using TCN.

3 Methodology

This section introduces the whole structure of Quater-GCN (Q-GCN) and the semi-supervised training strategy we employ. Q-GCN extracts both temporal information from position of node joint and rotation from bone joint with a sequence, and reconstructs a more sophisticated human pose representation by 3D coordinates and orientations. Figure 1 shows the whole architecture of the model. The input of Q-GCN are the 2D keypoint coordinates estimated from a video, and the 2D bone joint rotations derived from these coordinates. The output of Q-GCN are the 3D positions of each node joint and the orientations of each bone joint.

3.1 Graph Configuration

In 2D space, node joint positions can be represented as coordinate sequence within a video as $P = \{p_{t,j} \in \mathbb{R}^2 | t = 1, 2, \ldots, T; n = 1, 2, \ldots, N\}$, where T and N represent the number of frames in the sequence and the number of node joints in the human skeleton, respectively. We also incorporate a sequence of rotations for each bone

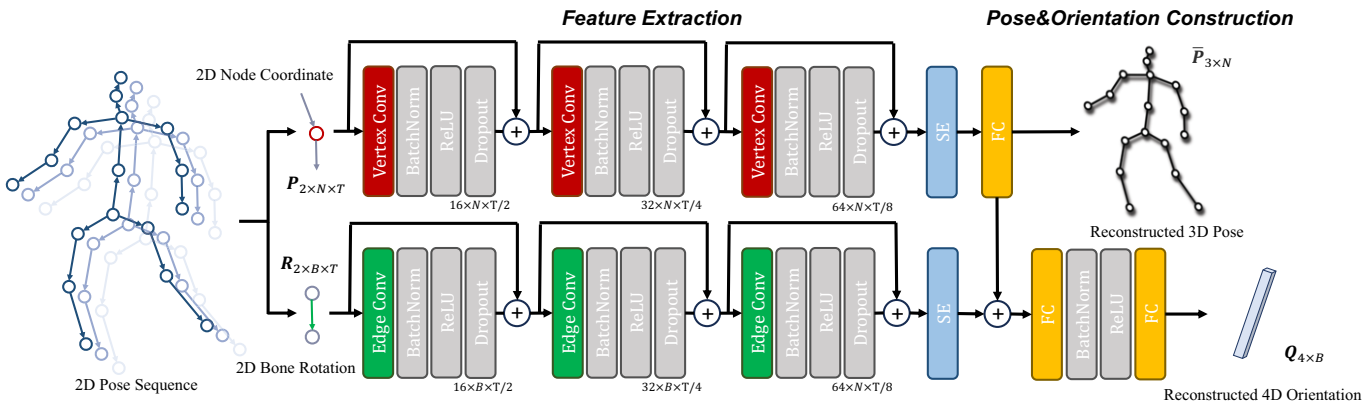

Figure 1. Whole architecture of Q-GCN. Q-GCN begins by dividing the input 2D pose sequence into node coordinates and bone rotations, which are represented as vertices and edges in a directed graph, respectively. It then extracts spatial-temporal features from these vertices and edges, incorporating a residual connection with each convolution operation. Following this feature extraction, Q-GCN reconstructs the human 3D pose and 4D orientation using a fully-connected (FC) layer that includes a Squeeze and Excitation (SE) block.

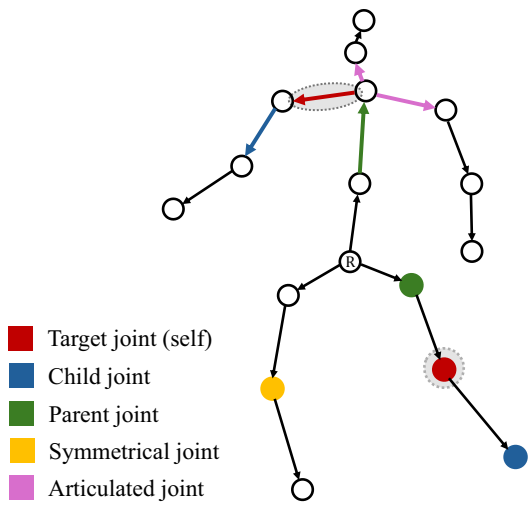

Figure 2. Whole configuration of the directed graph and the sampling strategy in Q-GCN. In this graph, vertices and edges are organized in a hierarchical structure, with the root node (typically the pelvis node) serving as the initialization point. The sampling strategy is illustrated by marking the target vertex and edge joint with a dot circle, while different colors on the joints indicate the subsets they belong to, as defined by the sampling strategy.

joint within the 2D coordinate space, expressed as $R = \{r_{t,b} \in \mathbb{R}^2 | t = 1, 2, 3, \ldots, T; b = 1, 2, 3, \ldots, B\}$, where B denotes the number of bone joints, and $r_{t,b}$ contains a 2-tuple of the cosine and sine values of the rotation angle $\theta \in [-\pi, \pi]$ for each bone joint b, starting from the initial position. All rotations are defined within the Local Coordinate System (LCS). In LCS, the origin point is established at the parent node of each bone, with the initial position (0 degrees) aligned with the direction in which the parent bone extends from this origin. Utilizing LCS is crucial as it preserves the spatial relationships between a bone joint and its parent.

Despite with the ignorance of the kinematic dependencies between joint and bones from the previous GCNs, we represent the skeleton data of human pose as a directed acyclic graph (DAG) inspired by [37], with each key node joint as vertex while each key bone joint as edge, as shown in Figure 2. We define this directed graph as $G = \{(v_t, e_t) | t = 1, 2, \ldots, T\}$, where $v_t = \{v_{t,n} | t = 1, 2, \ldots, T; n = 1, 2, \ldots, N\}$, and $e_t = \{e_{t,b} | t = 1, 2, \ldots, T; b = 1, 2, 3, \ldots, B\}$ represent the sets of key node joints and bone joints respectively. The features of vertex $v_{t,n}$ and edge $e_{t,b}$ are initialized with their respective 2D coordinates $p_{t,n}$ and rotations $r_{t,b}$.

Q-GCN predicts both 3D positions and orientations to reconstruct human pose representation. The 3D coordinates of the node joints in a human pose are denoted as $\bar{P} = \{\bar{p}_n \in \mathbb{R}^3 | n = 1, 2, \ldots, N\}$. In addition, we represent the output orientations in 3D space using 4D Quaternions, expressed as $Q = \{q_b \in \mathbb{R}^4 | b = 1, 2, 3, \ldots, B\}$ along with the root coordinate in 3D space, $\bar{p}_{root} \in \mathbb{R}^3$. Quaternions are preferred because they not only avoid Gimbal lock but require fewer dimensions compared to a 6D matrix [8]. Similar to the rotations in the 2D system, orientations in 3D space are defined within the LCS. The relative orientation of a child bone joint to its parent is calculated using the Quaternions in the World Coordinate System (WCS) as follows:

$$q_{child|LCS} = Inv(q_{parent|WCS}) \times q_{child|WCS} \quad (1)$$

where $Inv(q)$ denotes the inverse of the Quaternion q, and $\times$ denotes Quaternion multiplication. This design enhances the understanding of the internal dynamics and dependencies exerted from parent node to child node, as established by prior research [13].

3.2 Feature Extraction

Similar to [48], we first implement a basic spatial-temporal graph convolution block to extract the features both for positions of node and rotations of bones within the graph.

3.2.1 Sampling Strategy

We define a neighbor set $\mathcal{B}_n^v$ as a spatial graph convolutional filter for vertex $v_{t,n}$, and set $\mathcal{B}_b^e$ for edge $e_{t,b}$. Consequently, for the convolutional filter of vertex, we define four distinct neighbor subsets: (1) the vertex itself; (2) the subset of parent vertices, which includes vertices that directly point to the target vertex (closer to root vertex); (3) the subset of child vertices, which comprises vertices directly pointed by the target vertex; and (4) the subset of symmetrical vertices. The inclusion of the symmetrical vertex subset addresses the issue of pendant vertices (also known as leaf vertices), such as the left or right

hand, which do not have child vertex. Relying solely on the feature extraction of its parent vertex can result in a poor global representation [50]. In addition, for the convolutional filter for edges, similar to vertices, we also define four distinct neighbor subsets: (1) the edge itself; (2) the subset of parent edge, which includes edges directly point to the target vertex; (3) the subset of child edges, which are the edges directly pointed by the target edge; and (4) the subset of articulating edges. The inclusion of the articulating edge subset addresses the specific needs of edges like the left (or right) shoulder or neck (pink bone joints in Figure 2). These bone joints start from the root node and are articulated with one another, sharing close spatial dependencies. Therefore, the kernel size K is set to 4 both for vertex and edge filters, corresponding to the 4 subsets.

To implement the subsets, mappings $h^v_{t,n} \rightarrow \{0, \ldots, K-1\}$ and $h^e_{t,b} \rightarrow \{0, \ldots, K-1\}$ are used to index each subset with a numeric label. Therefore, this convolutional operations of vertex and edge can be written as

$$f^v_{out}(v_{t,n}) = \sum_{v_{t,n'} \in \mathcal{B}^v_n} \frac{1}{Z_{t,n'}} f^v_{in}(v_{t,n'}) W^v(h^v_{t,n}(v_{t,n'})) \quad (2)$$

$$f^e_{out}(e_{t,b}) = \sum_{e_{t,b'} \in \mathcal{B}^e_b} \frac{1}{Z_{t,b'}} f^e_{in}(e_{t,b'}) W^e(h^e_{t,b}(v_{t,b'})) \quad (3)$$

where the functions $f^v_{in}(v_{t,n'}) : v_{t,n'} \rightarrow \mathbb{R}^2$ and $f^e_{in}(e_{t,b'}) : e_{t,b'} \rightarrow \mathbb{R}^2$ denote the mappings that retrieve the attribute features of neighbor node joint $v_{t,n'}$ and neighbor bone joint $e_{t,b'}$ of $v_{t,n}$ and $e_{t,b}$ respectively. Note that the attribute features encapsulate both position of nodes and rotation of bone joints. $Z_{t,n'}$ and $Z_{t,b'}$ serve as normalization factors, equal to the cardinality of their respective subsets. The weight functions $W^v(h_{t,n}(v_{t,n'}))$ and $W^e(h_{t,b}(v_{t,b'}))$ correspond to the mappings for $\mathcal{B}^v_n$ and $\mathcal{B}^e_b$ respectively, which are implemented by indexing a $(2, K)$ tensor.

3.2.2 Dependency Representation

Within a pose frame, the graph convolution, as determined by the sampling strategy, is consistently implemented using adjacency matrices [50, 16, 13]. Accordingly, for a directed graph containing N vertices and B edges, we define an $N \times N$ adjacent matrix $\mathbf{A}^v$ for the vertices and a $B \times B$ adjacent matrices $\mathbf{A}^e$ for edges. The elements of these matrices represent the relationships between the corresponding vertices or edges, facilitating the propagation of information through the graph based on these defined connections.

However, an adjacency matrix that lacks hierarchical spatial information is not adequate for representing the directed edges within a directed graph. Inspired by [37], we employ incidence matrices for both vertices and edges to address this limitation. Furthermore, we define two $N \times B$ incidence matrices $\mathbf{P}^v$ and $\mathbf{C}^v$, where the elements indicate whether a given edge is the parent or child edge of a vertex. Similarly, we define two $B \times N$ incidence matrices $\mathbf{P}^e$ and $\mathbf{C}^e$, to specify whether a vertex is the parent or child of an edge. For instance, for a parent edge (or vertex) of a vertex v_n (or an edge e_b), the corresponding element in the parent incidence matrix $\mathbf{P}^v$ (or $\mathbf{P}^e$) is set to 1. Conversely, for its child edge (or vertex), the corresponding element in the child incidence matrix $\mathbf{C}^v$ (or $\mathbf{C}^e$) is set to 1, with all other elements set to 0. This structure enhances the graph representation by clearly defining and utilizing the hierarchical relationships between vertices and edges within the data.

3.2.3 Adaptive Representation

Inspired by [18], we also incorporate an adaptive design to enhance the flexibility of the ST-GCN block. Specifically, utilizing K spatial sampling strategies, we employ the sum of the incidence matrices for vertices, $\sum_{k=0}^{K-1} A^v_k$, and for edges, $\sum_{k=0}^{K-1} A^e_k$. This allows for the implementation of Equations 2 and 3 using these matrices as follows:

$$\mathbf{H}^v_t = \sum_{k=0}^{K-1} [\bar{\mathbf{A}}^v_k, \bar{\mathbf{P}}^e_k, \bar{\mathbf{C}}^e_k] \mathbf{F}^v_k \mathbf{W}^v_k \quad (4)$$

$$\mathbf{H}^e_t = \sum_{k=0}^{K-1} [\bar{\mathbf{A}}^e_k, \bar{\mathbf{P}}^v_k, \bar{\mathbf{C}}^v_k] \mathbf{F}^e_k \mathbf{W}^e_k \quad (5)$$

where $\bar{\mathbf{A}}_k = \mathbf{\Lambda}_k^{\frac{1}{2}} \mathbf{A}_k \mathbf{\Lambda}_k^{\frac{1}{2}}$ represents the normalized adjacency matrix of $\mathbf{A}_k$ for both vertices and edges. Following the approach used in [15], $\mathbf{\Lambda}^{ii}_k = \sum_n (\bar{\mathbf{A}}^{in}_k) + \alpha$ forms a diagonal matrix, with α set to 0.001 to prevent empty rows. $[\cdot]$ denotes the concatenation operation. $\mathbf{W}_k$ represents the weighting function for Equations 2 and 3, corresponding to a weight tensor of the 1×1 convolution operation. $\mathbf{F}_k$ specifies the attribute features of all the neighbor joints sampled into the subset k. This structured approach facilitates comprehensive spatial-temporal feature extraction, essential for dynamic pose estimation tasks.

Therefore, each convolution layer in Q-GCN is implemented using a $1 \times T$ classical 2D convolution layer, where T represents the temporal kernel size. The output from this layer, $\mathbf{H}_t$, is sequentially processed through a batch normalization layer, which is followed by a ReLU activation layer, and then a dropout layer, collectively forming a single convolutional block. Additionally, a residual connection [10] is integrated in each convolution layer to enhance the learning process.

3.3 Pose and Orientation Construction

For pose and orientation construction, drawing inspiration from [12, 1], we initially employ a Squeeze and Excitation (SE) block to recalibrate the channel-wise features for both coordinates and rotations. This enhances the model's sensitivity to informative features for both pose and orientation. Subsequently, we utilize a fully-connected layer to integrate multi-scale feature maps, which helps in predicting the final 3D poses with enhanced accuracy. In terms of orientation, the process begins by concatenating the rotation features with the predicted 3D coordinates. This concatenated output is then processed through two fully-connected layers. Between these layers, we insert a batch normalization layer followed by a ReLU activation layer. The complete architecture of Q-GCN is illustrated in Figure 1.

3.4 Semi-supervision Strategy

Inspired by [32], we introduce a semi-supervised training strategy to address the shortage of labeled ground truth of Quaternion regression of orientation. Figure 3 illustrates this strategy, which integrates both supervised and unsupervised components. Initially, in the supervised phase (first half batch), we feed Q-GCN with labeled 2D node coordinates and labeled 2D bone rotations. For the loss functions, we utilize the mean per-joint position error (MPJPE) loss [32] to regress the predicted 3D node coordinates based on the ground truth. To accurately regress the predicted 4D bone Quaternions, we develop an

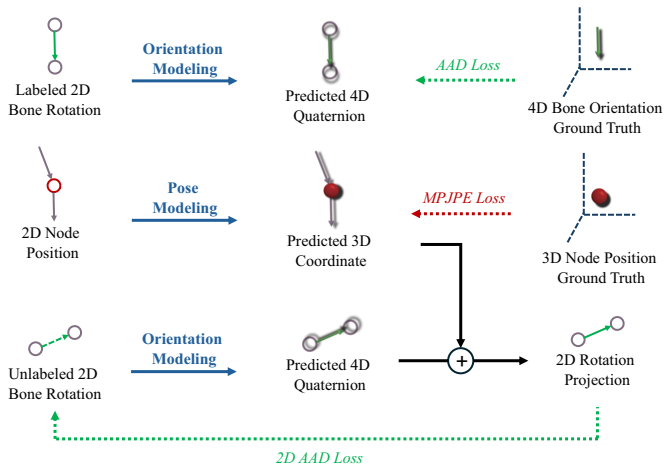

Figure 3. Semi-supervised training strategy for orientation regression. The unlabeled 2D rotations in the latter half of the batch are regressed using projected 2D rotations. These projections are derived from combining the predicted 4D orientations with the predicted 3D positions, both of which are initially trained using labeled data during the first half of the batch.

Average Angular Distance (AAD) loss function aimed at minimizing the angular distance between the ground truth and the predicted values:

$$\mathcal{L}_{angular} = \frac{1}{T}\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B} 2\arccos\left(Re(\bar{q}_{t,b} \times conj(q_{t,b})\right) \quad (6)$$

where $\bar{q}_{t,b}$ and $q_{t,b}$ represent the ground truth and predicted Quaternion of bone joint b at frame t, respectively. The functions $Re(\cdot)$ and $conj(\cdot)$ return the real part and the conjugate of a Quaternion, respectively.

In the unsupervised phrase (last half batch), which deals with unlabeled 2D bone rotations and corresponding labeled coordinates, we first use the model, trained with the initial batch, to predict the 4D bone orientations and corresponding 3D node coordinates. Following this, we integrate the predicted 3D coordinates with the 4D Quaternions to facilitate the computation of the 2D rotation projection. This setup allows us to regress the unlabeled data with these projections using a 2D AAD loss:

$$\mathcal{L}^{2D}_{angular} = \frac{1}{T}\frac{1}{B}\sum_{t=1}^{T}\sum_{b=1}^{B} |\bar{\theta}_{t,b} - \theta_{t,b}| \quad (7)$$

Where $\bar{\theta}_{t,b}$ and $\theta_{t,b}$ denote the ground truth and the predicted rotation angle of bone joint b at frame t.

4 Experimental Results

4.1 Datasets and Evaluation

Our method is assessed using three public datasets, with *Human3.6M* [14] and *HumanEva-I* [38] focusing on human major-part keypoints, while *H3WB* [56] on human whole-body keypoints. Consistent with established practices in prior research [50, 32, 13], our training on *Human3.6M* involves data from subjects S1, S5, S6, S7, and S8, with testing conducted on subjects S9 and S11. For *HumanEva-I*, we use data depicting the actions "walk" and "jog" performed by subjects S1, S2, and S3, applying it to both training and testing. In the case of *H3WB*, we adhere to the settings outlined in [56], utilizing a training set of 80k 2D-3D pairs and testing on half of the total available test samples.

Our evaluation protocols include the Mean Per-Joint Position Error (MPJPE) and the Pose-aligned MPJPE (P-MPJPE), also referred to as $Protocol\#1$ and $Protocol\#2$, respectively. For *Human3.6M*, both protocols are implemented, whereas for *HumanEva*, we solely apply $Protocol\#2$. In addition, considering the sparse research on orientation evaluation [33], we empoly our mean Average Angular Distance (mAAD) loss for assessment, as detailed in Equation 6.

4.2 Implementation Details

For 2D pose detection, we apply the methodologies utilized in *Human3.6M* and *HumanEva* as detailed in [13], using CPN [4] and MRCNN [11] respectively for detection. Additionally, we conduct experiments with ground truth (GT) 2D pose detection for all three datasets, noting that H36W is evaluated solely on the GT 2D whole-body pose due to its extensive keypoint coverage.

Regarding model settings, we adapt the sizes of the graph convolutional filters to match the structure of the 2D pose, with filters set to accommodate 17 node and 16 bone joints in *Human3.6M* and 16 node and 15 bone joints in *HumanEva*, respectively. For *H3WB*, due to the extensive number of keypoints, we categorize the whole-body keypoints into distinct groups: body, face, left hand, and right hand. Each group's filter size in our model is specifically tailored to match the number of node and bone joints associated with that particular body part. For example, in the configuration for the right hand within H3WB, the filters for vertex and edge are set to 21 and 20, respectively, corresponding to the 21 node joints and 20 bone joints that comprise the right hand. To assess the efficacy of the proposed model, particularly in orientation construction and semi-supervised learning, additional ablation experiments are performed on the *Human3.6M* dataset.

In terms of hyperparameters, batch sizes are set at 512 for *Human3.6M*, 256 for *HumanEva-I*, and 128 for *H3WB*, in line with [50, 56]. Consistent with [50], the ranger optimizer is used, and the model is trained using the MPJPE loss for 80 epochs for *Human3.6M* and 1000 epochs for *HumanEva-I*, starting with an initial learning rate of 0.01. The dropout rate is maintained at 0.25, and data augmentation via horizontal flipping is applied during both training and testing phases.

4.3 Comparable Results

Tables 1 and 2 showcase comparisons of Q-GCN against state-of-the-art (SOTA) methods on the *Human3.6M* dataset under $Protocol\#1$ and $Protocol\#2$, respectively. For the *HumanEva* dataset, Table 3 displays results under $Protocol\#2$ alongside other SOTA methods. Overall, Q-GCN surpasses these methods in terms of average results across both evaluation protocols, achieving the lowest average MPJPE loss on *Human3.6M* with 2D GT pose as input, and the lowest P-MPJPE loss on *Human3.6M* and on *HumanEva* when using both detected 2D poses (by MRCNN) and 2D GT poses. Particularly with GT pose inputs, Q-GCN demonstrates substantial performance improvements across nearly all action classes. Additionally, GLA-GCN [50], which also incorporates temporal information within a GCN-based model, achieves superior outcomes comparing to other methods. This highlights the crucial contribution of temporal information and GCN-based architecture in capturing the dynamics of human movement over time and the relational dependencies between body parts, essential for accurate 3D pose estimation. However, as

Method	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez et al. [27] (ICCV'17)	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Fang et al. [7] (AAAI'18)	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Pavlakos et al. [30] (CVPR'18)	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Lee et al. [17] (ECCV'18) †	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Zhao et al. [53] (CVPR'19) *	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	**55.0**	67.8	61.0	42.1	60.6	45.3	57.6
Ci et al. [5] (ICCV'19)	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Pavllo et al. [32] (CVPR'19) †	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Cai et al. [3] (ICCV'19) † *	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Xu et al. [46] (CVPR'20) †	**37.4**	43.5	42.7	42.7	46.6	59.7	41.3	45.1	52.7	60.2	45.8	43.1	47.7	33.7	37.1	45.6
Liu et al. [21] (CVPR'20) †	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Zeng et al. [51] (ECCV'20) †	46.6	47.1	43.9	41.6	45.8	49.6	46.5	40.0	53.4	61.1	46.1	42.6	43.1	31.5	32.6	44.8
Xu and Takano [47] (CVPR'21) *	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
Zhou et al. [55] (PAMI'21) †	38.5	45.8	40.3	54.9	**39.5**	**45.9**	**39.2**	43.1	**49.2**	71.1	**41.0**	53.6	44.5	33.2	34.1	45.1
Li et al. [20] (CVPR'22) †	39.2	43.1	**40.1**	**40.9**	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
Shan et al. [36] (ECCV'22) †	38.9	**42.7**	40.4	41.1	45.6	49.7	40.9	**39.9**	55.5	59.4	44.9	42.2	**42.7**	29.4	29.4	**42.8**
Yu et al. [50] (ICCV'23) † *	41.3	44.3	40.8	41.8	45.9	54.1	42.1	41.5	57.8	62.9	45.0	42.8	45.9	29.4	29.9	44.4
Our Q-GCN (T=243, CPN) † *	41.1	43.3	40.4	41.3	44.9	53.2	41.7	41.1	54.9	65.2	43.5	**41.3**	**42.7**	**29.1**	**29.2**	43.5
Martinez et al. [27] (ICCV'17)	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Lee et al. [17] (ECCV'18) †	32.1	36.6	34.3	37.8	44.5	49.9	40.9	36.2	44.1	45.6	35.3	35.9	30.3	37.6	35.5	38.4
Zhao et al. [53] (CVPR'19)	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
Ci et al. [5] (ICCV'19)	36.3	38.8	29.7	37.8	34.6	42.5	39.8	32.5	36.2	39.5	34.4	38.4	38.2	31.3	34.2	36.3
Liu et al. [21] (CVPR'20) †	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Xu and Takano [47] (CVPR'21) *	35.8	38.1	31.0	35.3	35.8	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	30.7	35.8
Zheng et al. [54] (ICCV'21) †	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
Li et al. [20] (CVPR'22) †	27.7	32.1	29.1	28.9	30.0	33.9	33.0	31.2	37.0	39.3	30.0	31.0	29.4	22.2	23.0	30.5
Shan et al. [36] (ECCV'22) †	28.5	30.1	**28.6**	27.9	29.8	33.2	31.3	27.8	**36.0**	**37.4**	29.7	29.5	28.1	21.0	21.0	29.3
Yu et al. (ICCV'23) † *	26.5	27.2	29.2	**25.4**	**28.2**	31.7	**29.5**	26.9	37.8	39.9	29.9	**27.0**	**27.3**	**20.5**	**20.8**	**28.5**
Our Q-GCN (T=243, GT) † *	**26.1**	**26.8**	28.8	26.1	28.5	**31.1**	29.9	**26.4**	37.1	39.4	**29.6**	28.1	28.3	20.7	**20.2**	**28.5**

Table 1. Reconstruction error on *Human3.6M* under $Protocol\#1$. Top table: 2D pose sequences detected by CPN as input. Bottom table: 2D pose sequences with GT as input. (†) uses temporal information. (*) uses GCN-based model. Lower is better, best in bold, second best underlined.

Method	Dir.	Disc.	Eat.	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez et al. [27] (ICCV'17)	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang et al. [7] (AAAI'18)	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Pavlakos et al. [30] (CVPR'18)	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Lee et al. [17] (ECCV'18) †	34.9	35.2	43.2	42.6	46.2	55.0	37.6	38.8	50.9	67.3	48.9	35.2	**31.0**	50.7	34.6	43.4
Cai et al. [3] (ICCV'19) † *	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	33.2	39.0
Pavllo et al. [32] (CVPR'19) †	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Xu et al. [46] (CVPR'20) †	**31.0**	**34.8**	34.7	34.4	36.2	43.9	31.6	33.5	**42.3**	49.0	37.1	33.0	39.1	26.9	31.9	36.2
Chen et al. [4] (ICCV'20) †	32.9	35.2	35.6	34.4	36.4	42.7	**31.2**	32.5	45.6	50.2	37.3	32.8	36.3	26.0	23.9	35.5
Liu et al. [21] (CVPR'20) †	32.3	35.2	33.3	35.8	35.9	**41.5**	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	35.6
Shan et al. [35] (MM'21) †	32.5	36.2	33.2	35.3	35.6	42.1	32.6	31.9	42.6	**47.9**	36.6	**32.1**	34.8	24.2	25.8	35.0
Shan et al. [36] (ECCV'22) †	31.3	35.2	32.9	33.9	35.4	39.3	32.5	31.5	44.6	48.2	36.3	32.9	34.4	23.8	**23.9**	**34.4**
Yu et al. [50] (ICCV'23) † *	32.4	35.3	32.6	34.2	**35.0**	42.1	32.1	31.9	45.5	49.5	**36.1**	32.4	35.6	**23.5**	24.7	34.8
Our Q-GCN (T=243, CPN) † *	31.1	34.9	**32.4**	**33.7**	36.3	42.8	31.6	**31.2**	44.7	48.6	36.9	32.4	35.4	24.1	24.4	34.7
Martinez et al. [27] (ICCV'17)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.1
Ci et al. [5] (ICCV'19)	24.6	28.6	24.0	27.9	27.1	31.0	28.0	25.0	31.2	35.1	27.6	28.0	29.1	24.3	26.9	27.9
Yu et al. [50] (ICCV'23) † *	20.2	21.9	21.7	**19.9**	21.6	24.7	22.5	20.8	28.6	**33.1**	22.7	**20.6**	20.3	15.9	16.2	22.0
Our Q-GCN (T=243, GT) † *	**20.1**	**21.4**	**21.5**	20.3	**21.1**	**24.2**	**21.2**	**20.3**	**27.2**	34.2	**21.5**	21.4	**20.0**	14.8	**15.1**	**21.6**

Table 2. Reconstruction error after rigid alignment on *Human3.6M* under $Protocol\#2$. Top table: 2D pose sequences detected by CPN as input. Bottom table: 2D pose sequences with GT as input. (†) uses temporal information. (*) uses GCN-based model. Lower is better, best in bold, second best underlined.

noted in the top sections of Tables 1 and 2, actions such as Discussion (Disc), Taking Photos (Photo), Posing (Pose), and Sitting (Sit) tend to exhibit higher errors with methods leveraging temporal information and GCN-based architecture, especially when using detected 2D poses. This may be due to the lower movement amplitude of these actions compared to others like Walking Dog (WalkD.), Walking (Walk), and Walking Together (WalkT.), which are easier for GCN-based models to capture due to their more pronounced dynamic features.

To gain deeper insights into how our method performs on different body parts, we conducted 2D-to-3D lifting experiments on human whole-body keypoints. Table 4 presents results comparing our Q-GCN method to state-of-the-art (SOTA) methods on the *H3WB* dataset, evaluated using $Protocol\#1$. Overall, Q-GCN consistently achieves top performance, ranking within the top two for the lowest error across all tests, indicating its strong capability in modeling complex pose structures with numerous vertices and edges. Additionally, Table 5 details the performance of various methods in constructing 4D orientation, evaluated using mean Average Angular Distance (mAAD). Q-GCN outperforms all competing methods, demonstrating superior effectiveness in accurately modeling every body part.

4.4 Ablation Studies

The ablation studies were conducted from two perspectives: the effect of different receptive fields and a component-wise comparison.

Table 6 compares various SOTA methods on the *Human3.6M* dataset, applying different receptive fields to GT 2D poses under $Protocol\#1$. Generally, the results indicate that a larger receptive field tends to yield better performance across all methods. Notably, Q-GCN outperforms other methods, particularly when utiliz-

Method	Walk			Jog			Avg
	S1	S2	S3	S1	S2	S3	
Martinez et al. [27] (ICCV'17)	19.7	17.4	46.8	26.9	18.2	18.6	24.6
Fang et al. [7] (AAAI'18)	19.4	16.8	37.4	30.4	17.6	16.3	23.0
Pavlakos et al. [30] (CVPR'18)	18.8	12.7	29.2	23.5	15.4	14.5	19.0
Lee et al. [17] (ECCV'18) †	18.6	19.9	30.5	25.7	16.8	17.7	21.5
Pavllo et al. [32] (CVPR'19) †	13.9	10.2	46.6	20.9	13.1	13.8	19.8
Liu et al. [21] (CVPR'20) †	13.1	9.8	26.8	**16.9**	12.8	13.3	15.5
Zheng et al. [54] (ICCV'21) †	14.4	10.2	46.6	22.7	13.4	13.4	20.1
Li et al. [19] (TMM'22) †	14.0	10.0	32.8	19.5	13.6	14.2	17.4
Zhang et al. [52] (CVPR'22) †	12.7	10.9	**17.6**	22.6	15.8	17.0	16.1
Yu et al. [50] (ICCV'23) † *	12.5	**9.1**	26.9	18.5	**12.7**	12.8	15.4
Our Q-GCN (T=27, MRCNN) † *	**12.1**	9.3	25.4	17.9	12.9	**12.7**	**15.1**
Li et al. [19] (TMM'22) †	9.7	7.6	15.8	12.3	9.4	11.2	11.1
Yu et al. [50] (ICCV'23) † *	8.7	6.8	11.5	**10.1**	**8.2**	9.9	9.2
Our Q-GCN (T=27, GT) † *	**8.5**	**6.4**	**10.8**	10.7	8.9	**9.6**	**9.1**

Table 3. Reconstruction error after rigid alignment on *HumanEva* under $Protocol\#2$. Top table: 2D pose sequences detected by MRCNN as input. Bottom table: 2D pose sequences with GT as input. (†) uses temporal information. (*) uses GCN-based model. Lower is better, best in bold, second best underlined.

Method	All	Body	Face / aligned†	Hand / aligned‡
SMPL-X [31]	188.9	166.0	208.3 / 23.7	170.2 / 44.4
CanonPose[42]*	186.7	193.7	188.4 / 24.6	180.2 / 48.9
SimpleBaseline [27]*	125.4	125.7	115.9 / 24.6	140.7 / 42.5
CanonPose[42] *w* 3D sv.*	117.7	117.5	112.0 / 17.9	126.9 / 38.3
Large SimpleBaseline[27]*	112.3	112.6	110.6 / **14.6**	**114.8** / **31.7**
Jointformer [24]	88.3	84.9	66.5 / 17.8	125.3 / 43.7
Our Q-GCN	**82.4**	**79.6**	**63.2** / 15.4	119.6 / 39.2

Table 4. Reconstruction error w/o rigid alignment on *H3WB* under $Protocol\#1$. (*) output normalized predictions. (Sv.) for supervision. Lower is better, best in bold, second best underlined. MPJPE metric in mm. All results are pelvis aligned, except † and ‡ show nose and wrist aligned results for face and hands, respectively.

ing larger receptive fields. However, with a smaller receptive field (e.g., $T = 27$), GCN-based models show lower performance compared to transformer-based models like [20], likely due to the expansive receptive field afforded by the attention mechanism of transformer.

Table 7 details an ablation study on key component designs of Q-GCN, focusing on orientation construction, semi-supervised training strategies, and the use of directed graphs (implemented via incidence matrices). We established a baseline model that includes only vertex convolution and pose construction using an undirected graph, regressed by 3D coordinates. The addition of orientation entails incorporating edge convolution and orientation construction regressed by 4D Quaternions. "With semi-supervision" indicates the application of the semi-supervised training strategy, which can only be implemented alongside orientation construction. "With directed graph" refers to the implementation of directed graph convolution network. The results demonstrate that the fully-equipped Q-GCN significantly outperforms the other configurations, confirming the effectiveness of the designed components. Comparing setups $\#2$ and $\#3$, it is evident that the semi-supervised training strategy effectively addresses the lack of Quaternion annotations for orientation regression. Additionally, comparison between setups $\#3$ and $\#5$ shows that with both including orientation construction, the impact of semi-supervision is more significant than that of using a directed graph. Comparisons among $\#1$, $\#2$, and $\#4$ suggest that implementing orientation construction yields more substantial benefits than using a directed graph. These findings further underscore the importance of orientation information in accurate 3D human pose estimation.

Method	All	Major-part	Upper-body	Lower-body	Hands
SMPL-X [31]	123	72	89	64	167
Jointformer [24]	77	66	72	49	103
GLA-GCN [50]	79	54	63	41	91
Our Q-GCN	**67**	**32**	**41**	**27**	**83**

Table 5. Reconstruction error on *H3WB* under mAAD loss, scaled by 10^3. Lower is better, best in bold, second best underlined.

Medhod	Frames	MPJPE (mm)
Pavllo et al. [32] (CVPR'19) †	$T = 27$	40.6
Liu et al. [21] (CVPR'20) †	$T = 27$	38.9
Li et al. [20] (CVPR'22) †	$T = 27$	**34.3**
Yu et al. [50] (ICCV'23) † *	$T = 27$	34.4
Our Q-GCN † *	$T = 27$	34.8
Pavllo et al. [32] (CVPR'19) †	$T = 81$	38.7
Liu et al. [21] (CVPR'20) †	$T = 81$	36.2
Li et al. [20] (CVPR'22) †	$T = 81$	32.7
Yu et al. [50] (ICCV'23) † *	$T = 81$	**31.5**
Our Q-GCN † *	$T = 81$	31.9
Pavllo et al. [32] (CVPR'19) †	$T = 243$	37.8
Liu et al. [21] (CVPR'20) †	$T = 243$	34.7
Zhang et al. [52] (CVPR'22) †	$T = 243$	21.6
Yu et al. [50] (ICCV'23) † *	$T = 243$	28.5
Our Q-GCN † *	$T = 243$	**21.3**

Table 6. Comparison with state-of-the-art methods on *Human3.6M* under $Protocol\#1$, implemented with different receptive fields of ground truth 2D pose. (*) uses GCN model.

#	Method	*Human3.6M*		*HumanEva-I*	
		CPN	GT	MRCNN	GT
1	Baseline	47.1	36.5	23.4	11.6
2	With orientation ($\mathcal{L}_{angular}$)	40.2	30.1	20.9	10.7
3	With orientation & semi-supervision	36.0	27.2	17.9	10.1
4	With directed graph	45.2	31.7	21.5	11.2
5	With orientation & directed graph	39.6	29.7	19.6	9.8
6	Q-GCN (With all)	**34.7**	**21.6**	**15.1**	**9.1**

Table 7. Ablation study on key designs of our Q-GCN. The results are based on the average value of $Protocol\#2$ implemented with 27 receptive fields for various 2D pose detections of the *Human3.6M* and *HumanEva-I*.

5 Conclusion

In this paper, we tackle the shortcomings of current 3D pose estimation methods that focus on spatial position but overlook the orientation or rotation of bones, which are crucial for understanding poses in 3D space. We introduce Quater-GCN, a novel graph convolutional network for 2D-to-3D pose lifting that incorporates both orientation and position data. This approach is further refined by a semi-supervised training strategy for 4D Quaternion regression, providing a more sophisticated pose representation. Rigorous evaluations across public datasets show that Q-GCN consistently outperforms state-of-the-art methods, especially with ground truth 2D poses, demonstrating its robustness and accuracy. Ablation studies highlight key components like orientation regression, directed graphs, and semi-supervised learning as significant contributors to our system's performance. This paper emphasizes the importance of integrating orientation data and semi-supervised learning to enhance 3D human pose estimation.

References

[1] S. Banik, E. Avagyan, A. M. Gracia, and A. Knoll. Posegraphnet++: Enriching 3d human pose with orientation estimation, 2023.

[2] D. Burgermeister and C. Curio. Pedrecnet: Multi-task deep neural network for full 3d human pose and orientation estimation. In *IV*, 2022.

[3] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, 2019.

[4] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[5] H. Ci, C. Wang, X. Ma, and Y. Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, 2019.

[6] D. Drover, M. V. Rohith, C. Chen, A. Agrawal, A. Tyagi, and C. P. Huynh. Can 3d pose be learned from 2d projections alone? *CoRR*, abs/1808.07182, 2018.

[7] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *AAAI*, 2018.

[8] M. Fisch and R. Clark. Orientation keypoints for 6d human pose estimation. *CoRR*, abs/2009.04930, 2020.

[9] Y. Gu, H. Zhang, and S. Kamijo. Multi-person pose estimation using an orientation and occlusion aware deep learning network. *Sensors*, 2020.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017.

[12] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[13] W. Hu, C. Zhang, F. Zhan, L. Zhang, and T. Wong. Conditional directed graph convolution for 3d human pose estimation. *CoRR*, abs/2107.07797, 2021.

[14] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.

[15] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[16] M. Korban and X. Li. Ddgcn: A dynamic directed graph convolutional network for action recognition. In *ECCV*, 2020.

[17] K. Lee, I. Lee, and S. Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *ECCV*, 2018.

[18] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *AAAI*, volume 32, 2018.

[19] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 25:1282–1293, 2022.

[20] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In *CVPR*, 2022.

[21] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *CVPR*, 2020.

[22] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 851–866. 2023.

[23] C. Luo, X. Chu, and A. L. Yuille. Orinet: A fully convolutional network for 3d human pose estimation. *CoRR*, abs/1811.04989, 2018. URL http://arxiv.org/abs/1811.04989.

[24] S. Lutz, R. Blythman, K. Ghostal, M. Matthew, C. Simms, and A. Smolic. Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation. *ICPR*, 2022.

[25] Q. Ma, J. Yang, A. Ranjan, S. Pujades, G. Pons-Moll, S. Tang, and M. J. Black. Learning to dress 3d people in generative clothing. In *CVPR*, June 2020.

[26] X. Ma, J. Su, C. Wang, H. Ci, and Y. Wang. Context modeling in 3d human pose estimation: A unified perspective. In *CVPR*, pages 6238–6247, 2021.

[27] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *CVPR*, 2017.

[28] G. Moon and K. M. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020.

[29] A. A. A. Osman, T. Bolkart, and M. J. Black. STAR: A sparse trained articulated human body regressor. In *ECCV*, 2020.

[30] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3d human pose estimation. *CoRR*, abs/1805.04095, 2018. URL http://arxiv.org/abs/1805.04095.

[31] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10975–10985, 2019.

[32] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. *CoRR*, abs/1811.11742, 2018. URL http://arxiv.org/abs/1811.11742.

[33] D. Pavllo, D. Grangier, and M. Auli. Quaternet: A quaternion-based recurrent model for human motion. *CoRR*, abs/1805.06485, 2018. URL http://arxiv.org/abs/1805.06485.

[34] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), Nov. 2017.

[35] W. Shan, H. Lu, S. Wang, X. Zhang, and W. Gao. Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3446–3454, 2021.

[36] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In *ECCV*, 2022.

[37] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, 2019.

[38] L. Sigal, A. O. Balan, and M. J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 2010.

[39] X. Song, Z. Li, S. Chen, and K. Demachi. Quater-gcn: Enhancing 3d human pose estimation with orientation and semi-supervised training, 2024. URL https://arxiv.org/abs/2404.19279.

[40] H. F. Tung, A. W. Harley, W. Seto, and K. Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. *CoRR*, abs/1705.11166, 2017. URL http://arxiv.org/abs/1705.11166.

[41] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. *CoRR*, abs/1701.01370, 2017. URL http://arxiv.org/abs/1701.01370.

[42] B. Wandt, M. Rudolph, P. Zell, H. Rhodin, and B. Rosenhahn. Canonpose: Self-supervised monocular 3d human pose estimation in the wild. *CoRR*, abs/2011.14679, 2020. URL https://arxiv.org/abs/2011.14679.

[43] J. Wang, S. Yan, Y. Xiong, and D. Lin. Motion guided 3d pose estimation from videos. In *ECCV*, pages 764–780. Springer, 2020.

[44] J. Wang, S. Huang, X. Wang, and D. Tao. Ponet: Robust 3d human pose estimation via learning orientations only. *CoRR*, abs/2112.11153, 2021. URL https://arxiv.org/abs/2112.11153.

[45] T. Wehrbein, M. Rudolph, B. Rosenhahn, and B. Wandt. Probabilistic monocular 3d human pose estimation with normalizing flows. In *ICCV*, pages 11199–11208, 2021.

[46] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *CVPR*, pages 899–908, 2020.

[47] T. Xu and W. Takano. Graph stacked hourglass networks for 3d human pose estimation. In *CVPR*, pages 16105–16114, 2021.

[48] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[49] W. Yang, W. Ouyang, X. Wang, J. S. J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. *CoRR*, abs/1803.09722, 2018. URL http://arxiv.org/abs/1803.09722.

[50] B. X. Yu, Z. Zhang, Y. Liu, S.-h. Zhong, Y. Liu, and C. W. Chen. Glagcn: Global-local adaptive graph convolutional network for 3d human. *arXiv preprint arXiv:2307.05853*, 2023.

[51] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, and S. Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In *ECCV*, 2020.

[52] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In *CVPR*, pages 13232–13242, 2022.

[53] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pages 3425–3435, 2019.

[54] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding. 3d human pose estimation with spatial and temporal transformers. In *ICCV*, pages 11656–11665, 2021.

[55] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu. Hemlets posh: Learning part-centric heatmap triplets for 3d human pose and shape estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. ISSN 0162-8828.

[56] Y. Zhu, N. Samet, and D. Picard. H3wb: Human3.6m 3d wholebody dataset and benchmark. In *ICCV*, 2023.