

Flexible Graph Convolutional Network for 3D Human Pose Estimation

Abu Taib Mohammed Shahjahan
a_shahj@live.concordia.ca

Concordia University
Montreal, QC, Canada

A. Ben Hamza
hamza@ciise.concordia.ca

Abstract

Although graph convolutional networks exhibit promising performance in 3D human pose estimation, their reliance on one-hop neighbors limits their ability to capture high-order dependencies among body joints, crucial for mitigating uncertainty arising from occlusion or depth ambiguity. To tackle this limitation, we introduce Flex-GCN, a flexible graph convolutional network designed to learn graph representations that capture broader global information and dependencies. At its core is the flexible graph convolution, which aggregates features from both immediate and second-order neighbors of each node, while maintaining the same time and memory complexity as the standard convolution. Our network architecture comprises residual blocks of flexible graph convolutional layers, as well as a global response normalization layer for global feature aggregation, normalization and calibration. Quantitative and qualitative results demonstrate the effectiveness of our model, achieving competitive performance on benchmark datasets. Code is available at: <https://github.com/shahjahan0275/Flex-GCN>

1 Introduction

The objective of 3D human pose estimation is to predict the 3D positions of body joints from images or videos. This task is essential for interpreting human movements and actions in various computer vision applications, including sports performance analytics and pedestrian behavior analysis [1]. For instance, accurately identifying skeletal joints is crucial for assessing sports activities, as it enables a meaningful evaluation of athletes' performance.

Existing approaches to 3D human pose estimation can generally be categorized into one- and two-stage methods. One-stage approaches, also known as direct regression techniques, aim to predict 3D joint locations directly from input images or video frames without intermediary steps. However, these methods often face depth ambiguity, where multiple plausible 3D poses can explain the same 2D observations. They also struggle with complex poses and occlusions [2, 3, 4, 5]. On the other hand, two-stage approaches, also known as 2D-to-3D lifting methods, typically consist of separate stages for joint detection and pose regression. The first stage detects 2D joint positions in the image, and the second stage uses these 2D detections to estimate the 3D joint positions. By incorporating an intermediate step for 2D joint detection, two-stage methods can mitigate challenges such as occlusions and depth ambiguity, resulting in more robust 3D pose estimations compared to their one-stage

counterparts. Moreover, they allow for the use of different 2D pose detectors and lifting networks, providing flexibility in designing and optimizing each component separately, thereby potentially leading to higher accuracy [2, 6, 8, 10].

Graph convolutional network (GCN)-based methods have recently demonstrated considerable promise in 3D human pose estimation [8, 11, 12, 14, 16], leveraging the inherent graph structure of the human body, where joints serve as nodes interconnected by edges representing skeletal connections. By capitalizing on this representation, GCN-based models can effectively capture spatial dependencies crucial for accurate pose estimation. While these methods have demonstrated effectiveness in capturing dependencies between body joints, they are, however, inherently limited in their ability to model interactions beyond immediate neighbors. To overcome this challenge, recent approaches have introduced high-order graph convolutions [9, 13, 15], which enable information propagation through multiple hops in the graph, allowing the model to gather insights from not only immediate neighbors but also nodes located farther away. This enhances the model’s capacity to capture global context and complex relationships between body joints. Another limitation of GCN-based methods is their inherent reliance on the adjacency matrix, which represents the connectivity between body joints in a graph, with non-zero entries indicating the presence of connections between neighboring joints. By modulating the adjacency matrix [14], we can incorporate additional information from nodes that are further apart in the graph, allowing the model to capture more complex dependencies and contextual cues.

To address the aforementioned limitations, we propose a novel graph convolutional network, dubbed Flex-GCN, which employs multi-hop neighbors through a flexible scaling factor that controls the balance between the information from immediate neighbors and the information from nodes that are at most two edges away in the graph. In addition to integrating an initial residual connection into the update rule of Flex-GCN, we also modulate the adjacency matrix to enable our model to consider not only the immediate connections between neighboring joints, but also the spatial relationships between distant joints that may contribute to the overall pose configuration. Our contributions are summarized as follows:

- We present a flexible graph convolutional network (Flex-GCN), which captures high-order dependencies essential for reducing uncertainty due to occlusion or depth ambiguity in 3D human pose estimation. We also theoretically demonstrate the training stability of Flex-GCN.
- We design a network architecture that includes flexible graph convolutional layers and a global response normalization layer.
- Experimental results and ablation studies demonstrate the competitive performance of Flex-GCN against strong baselines on two benchmark datasets.

2 Related Work

3D human pose estimation aims to estimate the 3D coordinates of the joints in the human body from images or videos. One-stage and two-stage approaches are two common strategies employed in this task. One-stage methods directly regress the 3D pose from input images [11, 12, 14, 15], while two-stage methods first predict intermediate representations, such as 2D joint locations, before lifting them to 3D space [2, 6, 8, 10]. Two-stage methods, often combined with robust 2D joint detectors, typically exhibit better performance,

particularly in addressing depth ambiguity challenges. Our proposed method falls under the category of two-stage approaches, with a network architecture design inspired by the ConvNeXt V2 framework [16], which leverages a global response normalization layer.

GCN-based methods for 3D human pose estimation offer an intuitive paradigm by representing the human body as a graph structure [6, 21, 22, 24, 26], where the joints of the body serve as nodes and the connections between them represent the bones. This approach leverages the inherent spatial relationships between body parts, allowing for the modeling of complex human poses through graph-based representations. Also, by analyzing the connectivity patterns within the skeletal graph, GCN-based methods can infer the positions of individual joints based on information propagated from neighboring nodes, facilitating robust and contextually informed pose predictions. For instance, SemGCN [22] integrates semantic information into the graph convolution, allowing the model to combine structural information from the graph with semantic features derived from the data. In GroupGCN [21], convolutional operations are performed within distinct groups, each of which has its own weight matrix and spatial aggregation kernel. Weight Unsharing [6] analyzes the trade-offs between weight sharing and unsharing in GCNs. Modulated GCN [24] combines weight modulation to learn unique modulation vectors for individual nodes and adjacency modulation to account for additional edges beyond the human skeleton connections. One major limitation of the standard GCN architecture is that it typically operates with one-hop neighbors, which can restrict the ability of GCN-based methods to capture long-range dependencies and complex interactions within the graph. In other words, these methods provide a relatively local perspective of the graph structure, potentially overlooking long-range interactions and intricate dependencies present in human body movements. To mitigate this issue, High-order GCN [25] incorporates high-order dependencies among body joints by considering neighbors located multiple hops away during the update of joint features. Similarly, multi-hop Modulated GCN (MM-GCN) [9] involves modulating and fusing features from multi-hop neighbors. Our proposed model differs from these GCN-based approaches in that we employ a new update rule for graph node feature propagation that seamlessly integrates both first- and second-order neighboring information, combined with an initial residual connection, with the aim of learning graph representations that capture more global information and dependencies, while maintaining the time and memory complexity of the standard GCN. We also leverage adjacency modulation to learn additional connections between body joints.

3 Method

3.1 Preliminaries and Problem Statement

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V} = \{1, \dots, N\}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and \mathbf{X} an $N \times F$ feature matrix of node attributes. We denote by \mathbf{A} an $N \times N$ adjacency matrix whose (i, j) -th entry is equal to 1 if i and j are neighboring nodes, and 0 otherwise. We also denote by $\hat{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$ the normalized adjacency matrix, where $\mathbf{D} = \text{diag}(\mathbf{A}\mathbf{1})$ is the diagonal degree matrix and $\mathbf{1}$ is a vector of all ones.

Let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ be a training set consisting of 2D joint positions $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^2$ and their associated ground-truth 3D joint positions $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^3$. The aim of 3D human pose estimation is to learn the parameters \mathbf{w} of a regression model $f_{\mathbf{w}} : \mathcal{X} \rightarrow \mathcal{Y}$ by finding a

minimizer of the following objective function

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^N l(f_{\mathbf{w}}(\mathbf{x}_i), \mathbf{y}_i), \quad (1)$$

where $l(f_{\mathbf{w}}(\mathbf{x}_i), \mathbf{y}_i)$ is an empirical regression loss function.

3.2 Flexible Graph Convolutional Network

Central to graph neural networks lies the fundamental concept of the feature propagation rule, which determines how information is transmitted among nodes in a graph. This rule entails updating node features by aggregating information from nearby and/or distant neighbors, followed by non-linear activation, to generate an updated node representation. To this end, we propose a flexible graph convolutional network (Flex-GCN) with the following layer-wise update rule for node feature propagation:

$$\mathbf{H}^{(\ell+1)} = \sigma \left(((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{X}\widetilde{\mathbf{W}}^{(\ell)} \right), \quad \ell = 0, \dots, L-1, \quad (2)$$

where $s \in (0, 1)$ is a positive scaling parameter, $\mathbf{W}^{(\ell)}$ and $\widetilde{\mathbf{W}}^{(\ell)}$ are learnable weight matrices, $\sigma(\cdot)$ is an element-wise activation function, $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times F_\ell}$ is the input feature matrix of the ℓ -th layer with F_ℓ feature maps. The input of the first layer is $\mathbf{H}^{(0)} = \mathbf{X}$.

The update rule of Flex-GCN is essentially comprised of three main components: (i) feature propagation that combines the features of the 1- and 2-hop neighbors of nodes (i.e., it aggregates information from immediate and high-order neighboring nodes), (ii) feature transformation that applies learnable weight matrices to the node representations to learn an efficient representation of the graph, and (iii) residual connection for ensuring that information from the initial feature matrix is preserved. The initial residual connection used in the proposed model allows information from the initial feature matrix to bypass the current layer and be directly added to the output of the current layer. This helps preserve important information that may be lost during the aggregation process, thereby improving the flow of information through the network. Note that the propagation operation/matrix $\mathbf{P} = ((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}}$ of our model is a weighted combination of the normalized adjacency matrix and its square. It allows Flex-GCN to capture information from nodes that are not only directly connected (1-hop), but also incorporates information from the neighbors of the neighbors (2-hop). The parameter s helps control the balance between the information from immediate neighbors and the information from nodes that are at most two edges away in the graph. This is particularly valuable for learning graph representations that capture more global information and dependencies.

Model Complexity. For simplicity, we assume the feature dimensions are the same across all layers, i.e., $F_\ell = F$ for all ℓ , with $F \ll N$. Multiplying the propagation matrix $((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}}$ with an embedding $\mathbf{H}^{(\ell)}$ costs $\mathcal{O}(\|\hat{\mathbf{A}}\|_0 F)$ in time, where $\|\hat{\mathbf{A}}\|_0$ denotes the number of non-zero entries of the sparse matrix $\hat{\mathbf{A}}$ (i.e., number of edges in the graph). Multiplying an embedding with a weight matrix costs $\mathcal{O}(NF^2)$. Also, multiplying the initial feature matrix by the residual connection weight matrix costs $\mathcal{O}(NF^2)$. Hence, the time complexity of an L -layer Flex-GCN is $\mathcal{O}(L\|\hat{\mathbf{A}}\|_0 F + LNF^2)$. For memory complexity, an L -layer Flex-GCN requires $\mathcal{O}(LNF + LF^2)$ in memory, where $\mathcal{O}(LNF)$ is for storing all embeddings and $\mathcal{O}(LF^2)$ is for storing all layer-wise weight matrices. Therefore, our proposed Flex-GCN model has the same time and memory complexity as that of GCN, albeit

Flex-GCN takes into account both immediate and distant graph nodes for improved learned node representations. It is important to note that there is no need to explicitly compute the square of the normalized adjacency matrix in the Flex-GCN model. Instead, we perform right-to-left multiplication of the normalized adjacency matrix with the embedding. This process avoids the computational cost associated with matrix exponentiation and simplifies the computation, making our model more efficient while achieving its objectives.

Numerical Stability. To demonstrate the numerical stability of the proposed Flex-GCN model, we start with a useful result in matrix analysis [13], which states that the spectral radius of the sum of two commuting matrices is bounded by the sum of the spectral radii of the individual matrices.

Lemma 1. *If two matrices \mathbf{M}_1 and \mathbf{M}_2 commute, i.e., $\mathbf{M}_1\mathbf{M}_2 = \mathbf{M}_2\mathbf{M}_1$, then*

$$\rho(\mathbf{M}_1 + \mathbf{M}_2) \leq \rho(\mathbf{M}_1) + \rho(\mathbf{M}_2), \quad (3)$$

where $\rho(\cdot)$ denotes matrix spectral radius (i.e., largest absolute value of all eigenvalues).

Since the eigenvalues of the normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \hat{\mathbf{A}}$ lie in the interval $[0, 2]$, it follows that $\rho(\hat{\mathbf{A}}) \leq 1$. Hence, we have the following result, which demonstrates the training stability of the proposed model, with information smoothly propagating through the graph layers without amplifying or dampening effects that could lead to instability.

Proposition 1. *The update rule of Flex-GCN is numerically stable.*

Proof. Recall that the propagation matrix of Flex-GCN is given by

$$\mathbf{P} = ((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}} = (1-s)\hat{\mathbf{A}} + s\hat{\mathbf{A}}^2. \quad (4)$$

Since the matrices $(1-s)\hat{\mathbf{A}}$ and $s\hat{\mathbf{A}}^2$ satisfy the assumptions of Lemma 1, we have

$$\rho((1-s)\hat{\mathbf{A}} + s\hat{\mathbf{A}}^2) \leq \rho((1-s)\hat{\mathbf{A}}) + \rho(s\hat{\mathbf{A}}^2) \leq 1, \quad (5)$$

because both $\rho(\hat{\mathbf{A}})$ and $\rho(\hat{\mathbf{A}}^2)$ are bounded by 1. Hence, the spectral radius of the propagation matrix is bounded by 1. Consequently, repeated layer-wise application of this propagation operator is stable.

Adjacency Modulation. We modulate the normalized adjacency matrix to capture not just the interactions between adjacent nodes, but also the relationships between distant nodes beyond the natural connections of body joints [24], yielding a modulated adjacency matrix $\check{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}$, where $\mathbf{Q} \in \mathbb{R}^{N \times N}$ is a learnable adjacency modulation matrix. Since the skeleton graph exhibits symmetry, we ensure symmetry in the learnable adjacency modulation matrix by averaging it with its transpose.

Model Architecture. The overall architecture of our proposed Flex-GCN model is illustrated in Figure 1. The input to the model is a 2D pose, typically obtained via an off-the-shelf 2D detector [11], which subsequently undergoes a flexible graph convolutional layer, followed by a GELU activation function. Following the architectural design of the ConvNeXt V2 block [2], our residual block consists of three flexible graph convolutional (Flex-GConv) layers. In each block, the first two convolutional layers are followed by layer normalization, while the third one is followed by GELU. This residual block is repeated four times. Then, a global response normalization (GRN) layer [16] is applied after the residual blocks with

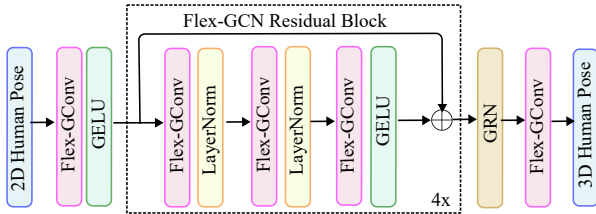


Figure 1: Network architecture of Flex-GCN for 3D human pose estimation.

the aim of increasing the contrast and selectivity of channels. Finally, the last flexible graph convolutional layer of the network generates the 3D pose.

Model Prediction. The output of the last flexible graph convolutional layer contains the final output node embeddings, $\hat{\mathbf{y}}_i$, $i = 1, \dots, N$, which are the predicted 3D pose coordinates.

Model Training. The parameters (i.e., weight matrices for different layers) of Flex-GCN are learned by minimizing the following loss function

$$\mathcal{L} = \frac{1}{N} \left[(1 - \alpha) \sum_{i=1}^N \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha \sum_{i=1}^N \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1 \right], \quad (6)$$

which is a weighted sum of the mean squared and mean absolute errors between the 3D ground truth coordinates \mathbf{y}_i and estimated 3D joint coordinates $\hat{\mathbf{y}}_i$ over N training body joints. The weighting factor $\alpha \in [0, 1]$ controls the contribution of each error term.

4 Experiments

4.1 Experimental Setup

Datasets. We assess Flex-GCN performance on Human3.6M [8] and MPI-INF-3DHP [9].

Evaluation Protocols and Metrics. We employ two standard evaluation protocols for training and testing on Human 3.6M designated as Protocol #1 and Protocol #2 [8], with associated metrics mean per-joint position error (MPJPE) and Procrustes-aligned mean per-joint position error (PA-MPJPE), respectively. For MPI-INF-3DHP, we use the Area Under Curve (AUC) and Percentage of Correct Keypoint (PCK) as assessment metrics.

Baselines. We compare the performance of our model with several state-of-the-art methods for 3D pose estimation, including Weight Unsharing [6], High-order GCN [25], Pose Grammar and Data Augmentation [19], Compositional GCN (CompGCN) [26], Higher-Order Implicit Fairing (HOIF-Net) [12], Multi-hop Modulated GCN (MM-GCN) [4], Group GCN [21], and Modulated GCN [22].

Implementation Details. All experiments are performed on a Linux machine with a single NVIDIA GeForce RTX A4500 GPU featuring 20GB of memory. We use PyTorch to implement our model and train it for 30 epochs using AMSGrad optimizer on detected 2D poses [11], as well as on ground truth 2D poses. The initial learning rate is set to 0.001, with a decay factor of 0.99 every four epochs, a batch size of 512, and $F = 384$. The scaling parameter $s = 0.2$ and the weighting factor $\alpha = 0.03$ are determined via grid search. To prevent overfitting, we apply dropout with a factor of 0.2 after each graph convolution layer.

4.2 Results and Analysis

Quantitative Results on Human3.6M. In Tables 1 and 2, we report the performance comparison of our Flex-GCN model and strong baselines for 3D pose estimation on Human3.6M using the detected 2D pose as input. In both tables, we present the results for all 15 actions, along with their average performance. Table 1 shows that Flex-GCN yields competitive performance, with average MPJPE and PA-MPJPE errors of 46.9mm and 38.6mm, respectively. Under Protocol #1, Table 1 reveals that Flex-GCN performs better than Modulated GCN [24] in 14 out of 15 actions, yielding 2.5mm error reduction on average, improving upon this best performing baseline by a relative improvement of 5.08%. Our model achieves better predictions than the best baseline on challenging actions like hard poses involving activities of self-occlusion such as ‘‘Eating’’, ‘‘Sitting’’ and ‘‘Smoking’’, showing relative error reductions of 1.53%, 7.47% and 8.24%, respectively, in terms of MPJPE. The presence of self-occlusions during activities can pose challenges for human pose estimation, as they restrict the model’s access to observable information. For instance, some activities like eating or smoking can lead to occlusions where a person’s hands and arms obstruct parts of their face and upper body besides, when sitting, a person’s legs and arms may obstruct other body parts such as the torso or feet.

Table 1: Comparison of our model and baseline methods in terms of MPJPE in millimeters, computed between the ground truth and estimated poses on Human3.6M under Protocol #1. The last column displays the average errors, with boldface numbers denoting the best performance and underlined numbers indicating the second-best performance.

Method	Action															
	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Liu <i>et al.</i> [8]	46.3	52.2	47.3	50.7	55.5	67.1	49.2	<u>46.0</u>	60.4	71.1	51.5	50.1	54.5	40.3	43.7	52.4
Zou <i>et al.</i> [24]	49.0	54.5	52.3	53.6	59.2	71.6	49.6	49.8	66.0	75.5	55.1	53.8	58.5	40.9	45.4	55.6
Xu <i>et al.</i> [19]	47.1	52.8	54.2	54.9	63.8	72.5	51.7	54.3	70.9	85.0	58.7	54.9	59.7	43.8	47.1	58.1
Zou <i>et al.</i> [24]	48.4	53.6	49.6	53.6	57.3	70.6	51.8	50.7	62.8	74.1	54.1	52.6	58.2	41.5	45.0	54.9
Quan <i>et al.</i> [16]	47.0	53.7	50.9	52.4	57.8	71.3	50.2	49.1	63.5	76.3	54.1	51.6	56.5	41.7	45.3	54.8
Zou <i>et al.</i> [24]	45.4	<u>49.2</u>	<u>45.7</u>	<u>49.4</u>	<u>50.4</u>	<u>58.2</u>	<u>47.9</u>	<u>46.0</u>	<u>57.5</u>	63.0	<u>49.7</u>	<u>46.6</u>	<u>52.2</u>	<u>38.9</u>	<u>40.8</u>	<u>49.4</u>
Lee <i>et al.</i> [9]	46.8	51.4	46.7	51.4	52.5	59.7	50.4	48.1	58.0	67.7	51.5	48.6	54.9	40.5	42.2	51.7
Zhang <i>et al.</i> [21]	<u>45.0</u>	50.9	49.0	49.8	52.2	60.9	49.1	46.8	61.2	70.2	51.8	48.6	54.6	39.6	41.2	51.6
Ours	40.2	45.8	45.0	46.8	48.6	54.0	42.4	42.1	53.2	<u>66.7</u>	45.6	45.4	48.8	38.4	40.1	46.9

Under Protocol #2, Table 2 shows that our model on average reduces the error by 1.28% compared to Modulated GCN [24], and achieves better results in 12 out of 15 actions, with same performance in the action ‘phone’. Also, our method outperforms Modulated GCN on the challenging actions of ‘‘Greeting’’, ‘‘Sitting’’ and ‘‘Smoking’’, yielding relative error reductions of 2%, 4.74% and 5.67%, respectively, in terms of PA-MPJPE. Moreover, our model performs better than Modulated GCN on the challenging ‘‘Photo’’ action, yielding a relative error reduction of 2%. In addition, Flex-GCN outperforms High-order GCN [23] by a relative improvement of 11.67% on average, as well as on all actions.

Qualitative Results. Figure 2 displays the visual results obtained by Flex-GCN on sample actions from the Human3.6M dataset. The effectiveness of our model is demonstrated by the close alignment between the predicted 3D poses and the ground truth, as shown in Figure 2. Compared to Modulated GCN, our model generates poses that closely resemble the ground truth, even in hard poses with self-occlusions.

Table 2: Comparison of our model and baseline methods in terms of PA-MPJPE, computed between the ground truth and estimated poses on Human3.6M under Protocol #2.

Method	Action															
	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	WalkT.	Avg.	
Liu <i>et al.</i> [16]	35.9	40.0	38.0	41.5	42.5	51.4	37.8	36.0	48.6	56.6	41.8	38.3	42.7	31.7	36.2	41.2
Zou <i>et al.</i> [23]	38.6	42.8	41.8	43.4	44.6	52.9	37.5	38.6	53.3	60.0	44.4	40.9	46.9	32.2	37.9	43.7
Xu <i>et al.</i> [14]	36.7	39.5	41.5	42.6	46.9	53.5	38.2	36.5	52.1	61.5	45.0	42.7	45.2	35.3	40.2	43.8
Zou <i>et al.</i> [24]	38.4	41.1	40.6	42.8	43.5	51.6	39.5	37.6	49.7	58.1	43.2	39.2	45.2	32.8	38.1	42.8
Quan <i>et al.</i> [15]	36.9	42.1	40.3	42.1	43.7	52.7	37.9	37.7	51.5	60.3	43.9	39.4	45.4	31.9	37.8	42.9
Zou <i>et al.</i> [24]	35.7	<u>38.6</u>	36.3	<u>40.5</u>	39.2	<u>44.5</u>	37.0	35.4	<u>46.4</u>	51.2	<u>40.5</u>	35.6	<u>41.7</u>	<u>30.7</u>	33.9	<u>39.1</u>
Lee <i>et al.</i> [9]	35.7	39.6	37.3	41.4	40.0	44.9	37.6	36.1	46.5	<u>54.1</u>	40.9	36.4	42.8	31.7	34.7	40.3
Zhang <i>et al.</i> [22]	<u>35.3</u>	39.3	38.4	40.8	41.4	45.7	<u>36.9</u>	<u>35.1</u>	48.9	55.2	41.2	36.3	42.6	30.9	<u>33.7</u>	40.1
Ours	34.1	38.0	<u>36.8</u>	39.7	39.2	43.6	33.4	34.5	44.2	57.1	38.3	<u>36.0</u>	41.0	29.9	33.1	38.6

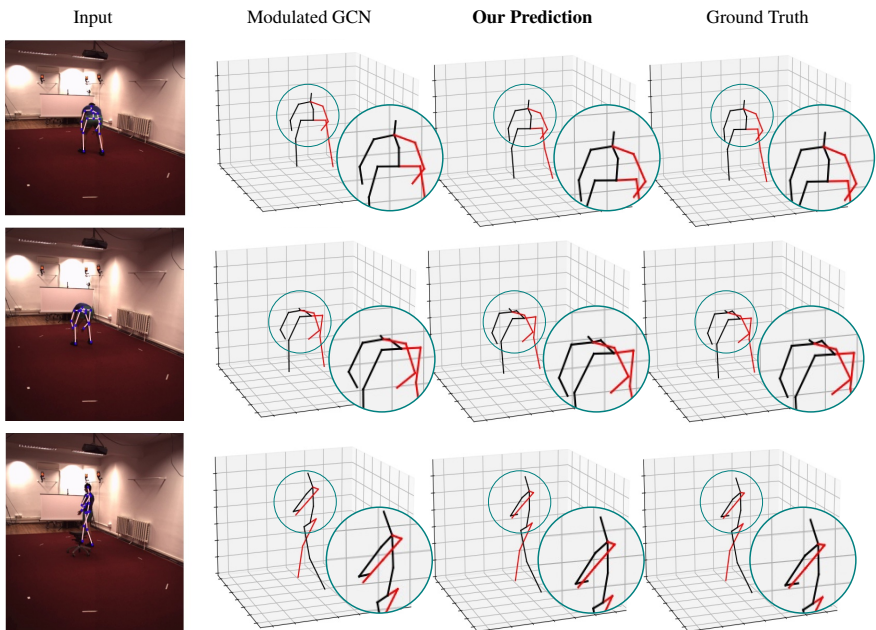


Figure 2: Visual comparison between Flex-GCN and Modulated GCN on sample actions from the Human3.6M dataset.

Cross-Dataset Results on MPI-INF-3DHP. In Table 3, We evaluate the generalization ability of our method by comparing it against strong baselines using a different dataset. Our model is trained on Human3.6M and evaluated on MPI-INF-3DHP. Results demonstrate that our model consistently outperforms the baselines, achieving relative improvements of 1.05% and 2.9% in terms of PCK and AUC metrics, respectively, in comparison with the best performing baselines. This highlights the strong generalization capability of our model to unseen scenarios and datasets.

Table 3: Results on MPI-INF-3DHP.

Method	PCK (\uparrow)	AUC (\uparrow)
Xu <i>et al.</i> [14]	80.1	45.8
Zeng <i>et al.</i> [20]	<u>82.1</u>	46.2
Lee <i>et al.</i> [9]	81.6	<u>50.3</u>
Zhang <i>et al.</i> [22]	81.1	49.9
Ours	85.2	51.8

Ground Truth Results and Runtime Analysis. Table 4 (left) shows that our model consistently yields better performance than the baselines under both Protocols #1 and #2 on the Human3.6M dataset using the ground truth 2D pose as input. This comparison highlights the efficacy of our model in leveraging ground truth 2D poses to generate more accurate 3D pose estimates. In addition to evaluating the accuracy and effectiveness of Flex-GCN, we conduct an analysis of its inference time to assess the efficiency of our approach in processing and generating outputs. The results of the inference time analysis are presented in Table 4 (right), where the computational performance of our Flex-GCN model is compared against that of strong baseline methods. Notably, our model demonstrates significantly improved inference time compared to the baseline methods, highlighting its computational efficiency.

Table 4: **Left:** Performance comparison of our model and baselines on Human3.6M using the ground truth 2D pose as input. **Right:** Runtime analysis.

Method	MPJPE (\downarrow)	PA-MPJPE (\downarrow)	Method	Inference Time
SemGCN [22]	42.14	33.53	High-Order GCN [25]	.013s
High-order GCN [25]	39.52	31.07	Weight Unsharing [6]	.032s
Modulated GCN [24]	38.25	30.06	MM-GCN [9]	.009s
Weight Unsharing [6]	37.83	30.09	Modulated GCN [24]	.010s
Ours	37.41	29.87	Ours	0.06s

4.3 Ablation Study

Effect of Batch and Filter Size. In Figure 3, we analyze the impact of varying batch and filter sizes on our model’s performance. This analysis is crucial as both parameters play a significant role in the training efficiency and the overall accuracy of the model. Batch size directly influences the stability and speed of the training process. Smaller batch sizes can lead to noisier gradient estimates but allow for more frequent updates, potentially improving generalization. Larger batch sizes, on the other hand, provide more stable gradient estimates but require more memory and can lead to slower convergence. Filter size, which determines the number of learnable parameters in each layer of the network, affects the model’s capacity to capture complex patterns. A larger filter size increases the model’s ability to learn intricate features but also raises the risk of overfitting. Conversely, a smaller filter size may lead to underfitting. Our analysis indicates that a batch size of 512 and a filter size of 384 result in the best performance. This combination yields the lowest MPJPE and PA-MPJPE values, respectively, signifying that the model is accurately estimating 3D human poses. The batch size of 512 strikes a balance between gradient stability and update frequency, while the filter size of 384 provides a sufficient number of parameters to learn detailed features without overfitting.

Effect of Residual Connection and Modulation Symmetry. We analyze the impact of the initial residual connection (IRC) in the layer-wise propagation rule on our model performance, and the results are reported in Table 5 (left). The inclusion of IRC shows significant improvements, with relative error reductions of 6.58% and 4.41% in terms of MPJPE and PA-MPJPE, respectively. This demonstrates the effectiveness of IRC in enhancing our model’s accuracy. By preserving and reinforcing the initial node features throughout the layers, IRC facilitates more stable and effective learning, ensuring that essential positional information is maintained and utilized in subsequent layers.

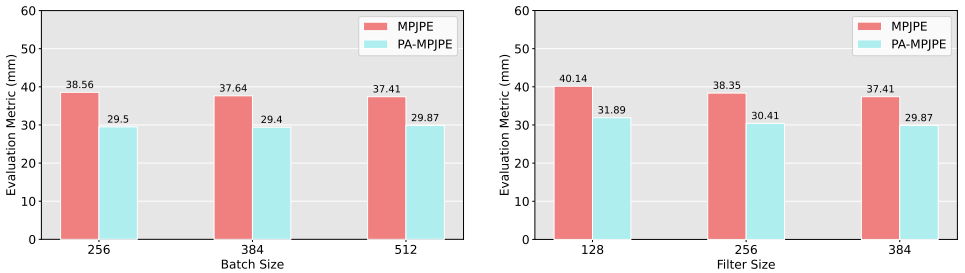


Figure 3: Performance of our proposed Flex-GCN model on the Human3.6M dataset using varying batch and filter sizes.

Furthermore, we assess the effect of symmetrizing the learnable adjacency modulation matrix on model performance. The results, presented in Table 5 (right), indicate that introducing symmetry into the adjacency modulation process yields tangible reductions in both MPJPE and PA-MPJPE. Specifically, the MPJPE error decreased by 0.58mm and the PA-MPJPE error decreased by 0.24mm, compared to the configuration without symmetry. This improvement highlights the advantage of leveraging skeleton graph symmetry to enhance the precision of our model’s estimations. By ensuring that the relational information between joints is consistently balanced, the symmetric adjacency modulation matrix enables more accurate and reliable pose estimations. This approach not only refines the positional accuracy of individual joints but also improves the overall coherence of the estimated poses, demonstrating the critical role of structured graph modulation in 3D human pose estimation.

Table 5: Effect of initial residual connection (IRC) and symmetry of modulation adjacency.

Method	MPJPE (\downarrow)	PA-MPJPE (\downarrow)	Method	MPJPE (\downarrow)	PA-MPJPE (\downarrow)
Without IRC	39.76	31.25	Without Symmetry	37.99	30.11
With IRC	37.41	29.87	With Symmetry	37.41	29.87

5 Conclusion

We introduced a simple yet efficient Flex-GCN model, which captures high-order dependencies essential for reducing uncertainty due to occlusion or depth ambiguity in 3D human pose estimation. We also theoretically demonstrated the training stability of Flex-GCN. Experimental results demonstrate that our model outperforms competitive baselines on standard datasets for 3D human pose estimation. Furthermore, our exploration of adjacency modulation enables Flex-GCN to incorporate richer contextual information beyond the natural connections of body joints, leading to enhanced performance in challenging scenarios. Through ablation studies, we have elucidated the contributions of various design choices, such as the initial residual connection and symmetry of modulation adjacency, highlighting their positive impact on model performance. For future work, we intend to apply our model to a broader range of computer vision and graph representation learning tasks.

Acknowledgments. This work was supported in part by the Discovery Grants Program of the Natural Sciences and Engineering Research Council of Canada under grant RGPIN-2024-04291.

References

- [1] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018.
- [2] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3D human pose estimation. In *Proc. IEEE International Conference on Computer Vision*, pages 2262–2271, 2019.
- [3] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7): 1325–1339, 2013.
- [4] Jae Yung Lee and I Gil Kim. Multi-hop modulated graph convolutional networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2022.
- [5] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3D human pose estimation. In *Proc. European Conference on Computer Vision*, pages 318–334, 2020.
- [6] Kenkun Liu, Zhiming Zou, and Wei Tang. Learning global pose features in graph convolutional networks for 3D human pose estimation. In *Proc. Asian Conference on Computer Vision*, 2020.
- [7] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 11976–11986, 2022.
- [8] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3D human pose estimation. In *Proc. IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.
- [9] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *Proc. International Conference on 3D Vision*, 2017.
- [10] Sunghoon Park, Jihye Hwang, and Nojun Kwak. 3D human pose estimation using convolutional neural networks with 2D pose information. In *Proc. European Conference on Computer Vision*, pages 156–169. Springer, 2016.
- [11] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.
- [12] Jianning Quan and A Ben Hamza. Higher-order implicit fairing networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2021.
- [13] Frigyes Riesz and Bela Sz.-Nagy. *Functional Analysis*. Dover Publications, 1990.

- [14] Xiao Sun, Jiayang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proc. IEEE International Conference on Computer Vision*, pages 2602–2611, 2017.
- [15] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proc. European Conference on Computer Vision*, pages 529–545, 2018.
- [16] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023.
- [17] Haiping Wu and Bin Xiao. 3D human pose estimation via explicit compositional depth maps. In *Proc. AAAI Conference on Artificial Intelligence*, volume 34, pages 12378–12385, 2020.
- [18] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3D human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 16105–16114, 2021.
- [19] Yuanlu Xu, Wenguan Wang, Tengyu Liu, Xiaobai Liu, Jianwen Xie, and Song-Chun Zhu. Monocular 3D pose estimation via pose grammar and data augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [20] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3D pose estimation. In *Proc. IEEE International Conference on Computer Vision*, pages 11436–11445, 2021.
- [21] Zijian Zhang. Group graph convolutional networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2022.
- [22] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.
- [23] Yun Zhao, Zejian Yuan, and Badong Chen. Accurate pedestrian detection by human pose regression. *IEEE Transactions on Image Processing*, 29:1591–1605, 2019.
- [24] Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3D human pose estimation. In *Proc. IEEE International Conference on Computer Vision*, pages 11477–11487, 2021.
- [25] Zhiming Zou, Kenkun Liu, Le Wang, and Wei Tang. High-order graph convolutional networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2020.
- [26] Zhiming Zou, Tianqi Liu, Dapeng Wu, and Wei Tang. Compositional graph convolutional networks for 3D human pose estimation. In *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2021.