Higher-Order Implicit Fairing Networks for 3D Human Pose Estimation

Jianning Quan
jianning.quan@mail.concordia.ca
A. Ben Hamza
hamza@ciise.concordia.ca

Concordia University – CIISE
Montreal, QC, Canada

Abstract

Estimating a 3D human pose has proven to be a challenging task, primarily because of the complexity of the human body joints, occlusions, and variability in lighting conditions. In this paper, we introduce a higher-order graph convolutional framework with initial residual connections for 2D-to-3D pose estimation. Using multi-hop neighborhoods for node feature aggregation, our model is able to capture the long-range dependencies between body joints. Moreover, our approach leverages residual connections, which are integrated by design in our network architecture, ensuring that the learned feature representations retain important information from the initial features of the input layer as the network depth increases. Experiments and ablations studies conducted on two standard benchmarks demonstrate the effectiveness of our model, achieving superior performance over strong baseline methods for 3D human pose estimation.

1 Introduction

The task of 3D human pose estimation is a fundamental problem in computer vision, robotics, and computer graphics. It refers to the process of predicting the positions of a person's joints (also known as keypoints or landmarks) in images or videos. Application domains of 3D human pose estimation are abundant, and range from activity recognition, surveillance and healthcare to games and sports.

Tremendous progress has been made in estimating 3D human pose from images or videos thanks to the rapid development of deep neural network solutions, which have been shown to achieve improved performance over classical approaches that use hand-crafted features. Most existing 3D pose estimation methods use an end-to-end pipeline [20] or a two-stage pipeline [24, 30]. The former employs a deep neural network to regress 3D keypoints from images in an end-to-end fashion, whereas the latter is comprised of two main stages, which are usually decoupled from each other. Two-stage approaches for 3D pose estimation have shown great promise [9, 10, 22, 25, 26, 27, 29, 35, 38], outperforming end-to-end models. This better performance is largely attributed to the fact that two-stage methods benefit from intermediate supervision provided, in part, by robust 2D pose detectors [26]. Martinez *el al.* [22] design a simple fully connected network with residual connections for estimating 3D poses from 2D joint detections, outperforming systems trained end-to-end from raw pixels.

In recent years, there has been a surge of interest in the adoption of graph convolution networks (GCNs) for 3D pose estimation [3, 37, 39], achieving state-of-the-art performance. Much of this interest stems from the fact that a 2D human skeleton can naturally be represented as a graph whose nodes are body joints and edges are connections between neighboring joints. Zhao *et al.* [37] propose SemGCN, a semantic graph convolutional network, which learns to capture semantic information encoded in a given graph (i.e. local and global relations between nodes), yielding improved performance in 3D pose estimation while using a much smaller number of parameters. While GCN is powerful for learning on graph-structured data, it suffers, however, from the oversmoothing problem [19], where the learned node representations become indistinguishable due to repeated graph convolutions as the network depth increases. Several attempts have been made toward remedying this issue of oversmoothing [4, 15, 33, 36]. Another issue with GCN is that its aggregation scheme uses one-hop neighbors, and hence lacks the ability to capture long-range dependencies. This issue can be mitigated by skipping connections during feature aggregation using, for example, the jumping knowledge networks [33] or by concatenating feature representations of multi-hop neighbors via sparsified neighborhood mixing (MixHop) [1], which leverages a graph convolutional layer that mixes powers of the adjacency matrix. Building on MixHop, Zou *et al.* [39] propose a high-order GCN for 3D pose estimation, with the goal of capturing long-range dependencies between body joints.

To address the above issues, we introduce a higher-order graph convolutional framework for 3D pose estimation via implicit fairing on graphs [8]. We follow the two-stage paradigm by employing a state-of-art 2D pose detector, followed by a lifting network for predicting the 3D pose locations from the 2D predictions. The aggregation scheme of the proposed approach leverages residual connections to help alleviate the oversmoothing problem, and uses multi-hop neighborhoods to capture long-range dependencies between body joints. The main contributions of this work can be summarized as follows:

- We derive an implicit fairing network (IF-Net) with initial residual connection by iteratively solving the implicit fairing equation on graphs via Jacobi method.
- We propose a higher-order implicit fairing network (HOIF-Net) for 3D human pose estimation by concatenating feature representations from multi-hop neighborhoods, with the aim to capture long-range dependencies.
- We demonstrate through experiments and ablation studies that our proposed model achieves state-of-the-art performance in comparison with strong baselines.

2 Related work

Graph Convolution Networks. GCNs have recently become the de facto model for learning representations on graphs. However, GCNs are prone to oversmooting as the network depth increases, and also fail to capture important dependencies between distant nodes. To circumvent these limitations, a plethora of GCN variants have been proposed, including jumping knowledge networks (JK-Nets) [33], graph convolutional networks with initial residual connection and identity mapping (GCNII) [4], and higher-order graph convolutional architectures via MixHop [1]. The latter learns neighborhood mixing relationships by repeatedly mixing feature representations of neighbors at various distances through powers of the graph adjacency matrix, while requiring no additional memory or computational complexity.

3D Human Pose Estimation. Most approaches to 3D human pose estimation can generally be classified into two main categories, namely single-stage and two-stage models, with the former using an end-to-end pipeline to predict 3D poses from images; and the latter using a two-stage pipeline, in which 2D joint locations are first extracted using a 2D pose detector and then a lifting network is employed to regress 3D poses from 2D detections. Our approach falls under the category of two-stage models [3, 6, 9, 10, 21, 22, 25, 26, 27, 29, 35, 37, 38, 39]. Zou *et al.* [39] design a high-order GCN model for 3D pose estimation based on MixHop in a bid to capture long-range dependencies between distant body joints using a network architecture comprised of a residual block repeated several times similar to the network design of Martinez *et al.* [22]. However, the model inherits the oversmoothing issue of GCNs, where repeated graph convolutions make learned node embeddings indistinguishable; thereby, resulting in performance drop. By contrast, our proposed network architecture has residual connections integrated by design, and hence is able to alleviate the oversmoothing problem. This is in line with existing approaches such as jumping knowledge networks [33] and graph convolutional networks with initial residual and identity mapping [4]. In addition, we use a scaled, learnable weight matrix with a layer-dependent scale factor in an effort to ensure that the weight decay adaptively increases as more layers are added [4].

3 Preliminaries and Problem Statement

Basic Notions. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \dots, N\}$ is the set of N nodes (e.g., body joints) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (e.g., connections between two body joints). Let $\mathbf{A}$ be an $N \times N$ adjacency matrix whose (i, j)-th entry is equal to the weight of the edge between neighboring nodes i and j, and 0 otherwise. We denote by $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ the adjacency matrix with self-added loops, where $\mathbf{I}$ is the identity matrix. We also denote by $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathsf{T}}$ an $N \times F$ feature matrix of node attributes, where $\mathbf{x}_i$ is an F-dimensional row vector for node i. We define the normalized Laplacian matrix as follows:

$$\mathbf{L} = \mathbf{I} - \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}, \tag{1}$$

where $\tilde{\mathbf{D}} = \operatorname{diag}(\tilde{\mathbf{A}}\mathbf{1})$ is the diagonal degree matrix, and $\mathbf{1}$ is an N-dimensional vector of all ones. The Laplacian matrix admits an eigendecomposition given by $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^{\mathsf{T}}$, where $\mathbf{U}$ is an orthonormal matrix whose columns constitute an orthonormal basis of eigenvectors and Λ is a diagonal matrix comprised of the corresponding eigenvalues.

Graph Convolutional Networks (GCNs). Given an input feature matrix $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times F_\ell}$ of the ℓ-th layer with F_ℓ feature maps, the output feature matrix $\mathbf{H}^{(\ell+1)}$ of GCN is obtained by applying the following layer-wise propagation rule:

$$\mathbf{H}^{(\ell+1)} = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(\ell)} \mathbf{W}^{(\ell)}), \quad \ell = 0, \ldots, L-1, \tag{2}$$

which is basically a node embedding transformation that projects $\mathbf{H}^{(\ell)}$ into a trainable weight matrix $\mathbf{W}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ with $F_{\ell+1}$ feature maps, followed by an activation function $\sigma(\cdot)$ such as $\operatorname{ReLU}(\cdot) = \max(0, \cdot)$. The input of the first layer is the initial feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$.

Jacobi Method. The Jacobi method [28] is an iterative approach for solving a matrix equation $\mathbf{M}\mathbf{x} = \mathbf{b}$, where the square matrix $\mathbf{M}$ has no zeros along its main diagonal, by first decomposing $\mathbf{M}$ into a diagonal component and an off-diagonal component, i.e.

$$\mathbf{M} = \operatorname{diag}(\mathbf{M}) + \operatorname{off}(\mathbf{M}). \tag{3}$$

Then, the solution of the matrix equation $\mathbf{M}\mathbf{x} = \mathbf{b}$ is obtained iteratively as follows:

$$\mathbf{x}^{(t+1)} = \operatorname{diag}(\mathbf{M})^{-1}(\mathbf{b} - \operatorname{off}(\mathbf{M})\mathbf{x}^{(t)}), \tag{4}$$

where $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t+1)}$ are the t-th and $(t+1)$-th iterations of $\mathbf{x}$, respectively.

Problem Statement. Let $\mathcal{D}_l = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ be a training set of 2D joint positions $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^{\mathsf{T}} \in \mathbb{R}^{N\times 2}$ and their associated 3D joint positions $\mathbf{Y} = (\mathbf{y}_1, \dots, \mathbf{y}_N)^{\mathsf{T}} \in \mathbb{R}^{N\times 3}$. The goal of 3D human pose estimation is to learn the parameters $\mathbf{w}$ of a regression model $f : \mathbf{X} \to \mathbf{Y}$ by minimizing the following loss function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i). \tag{5}$$

Since the 3D human pose estimation task is a regression problem, we train the model to minimize the mean squared error as a loss function.

4 Proposed Method

4.1 Implicit Fairing on Graphs

Applying a spectral graph filter with transfer function h on the graph signal $\mathbf{X}$ yields a filtered graph signal $\mathbf{H}$ given by

$$\mathbf{H} = h(\mathbf{L})\mathbf{X} = \mathbf{U}h(\Lambda)\mathbf{U}^{\mathsf{T}}\mathbf{X}. \tag{6}$$

Spectral graph filters are usually approximated using Chebyshev polynomials [7, 12, 31] or rational polynomials [17, 32]. The implicit fairing method, which uses implicit integration of a diffusion process for graph filtering, has shown to allow for both efficiency and stability [8]. The implicit fairing filter is an infinite impulse response filter whose transfer function is given by $h_s(\lambda) = 1/(1+s\lambda)$, where s is a positive parameter. Substituting h with h_s in Eq. (6), we obtain

$$\mathbf{H} = (\mathbf{I} + s\mathbf{L})^{-1}\mathbf{X}, \tag{7}$$

where $\mathbf{I} + s\mathbf{L}$ is a symmetric positive definite matrix (all its eigenvalues are positive), and hence admits an inverse. Therefore, performing graph filtering with implicit fairing is equivalent to solving the following sparse linear system:

$$(\mathbf{I} + s\mathbf{L})\mathbf{H} = \mathbf{X}. \tag{8}$$

4.2 Iterative Solution

The implicit fairing equation (8) can be solved iteratively using Jacobi's method, which uses matrix splitting. We can split the matrix $\mathbf{I} + s\mathbf{L}$ into the sum of a diagonal matrix and an off-diagonal matrix as follows:

$$\mathbf{I} + s\mathbf{L} = \operatorname{diag}(\mathbf{I} + s\mathbf{L}) + \operatorname{off}(\mathbf{I} + s\mathbf{L}), \tag{9}$$

where

$$\operatorname{diag}(\mathbf{I} + s\mathbf{L}) = (1+s)\mathbf{I} \quad \text{and} \quad \operatorname{off}(\mathbf{I} + s\mathbf{L}) = -s\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}.$$

Hence, the iterative solution of the implicit fairing equation is given by

$$\begin{aligned}\mathbf{H}^{(t+1)} &= -(\operatorname{diag}(\mathbf{I}+s\mathbf{L}))^{-1}\operatorname{off}(\mathbf{I}+s\mathbf{L})\mathbf{H}^{(t)} + (\operatorname{diag}(\mathbf{I}+s\mathbf{L}))^{-1}\mathbf{X} \\ &= (s/(1+s))\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(t)} + (1/(1+s))\mathbf{X},\end{aligned} \tag{10}$$

which can be rewritten as

$$\mathbf{H}^{(t+1)} = (1-\alpha)\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(t)} + \alpha\mathbf{X}, \tag{11}$$

where the hyperparameter $\alpha = 1/(1+s) \in (0,1)$, and $\mathbf{H}^{(t)}$ is the t-th iteration of $\mathbf{H}$.

4.3 Implicit Fairing Network

Inspired by the Jacobi iterative solution (11) of the implicit fairing equation, we propose a multi-layer implicit fairing network (IF-Net) with the following layer-wise propagation rule:

$$\mathbf{H}^{(\ell+1)} = \sigma(((1-\alpha)\mathbf{S}\mathbf{H}^{(\ell)} + \alpha\mathbf{X})\tilde{\mathbf{W}}^{(\ell)}), \tag{12}$$

where $\mathbf{S} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized adjacency matrix with self-added loops, and $\tilde{\mathbf{W}}^{(\ell)} = \beta_\ell \mathbf{W}^{(\ell)}$ is a scaled, learnable weight matrix with a layer-dependent scale factor defined as $\beta_\ell = \log(1+\beta/(1+\ell))$, which ensures that the decay of the weight matrix increases in tandem with the network depth [4].

4.4 Higher-Order Implicit Fairing Network

Using the feature diffusion rule of GCN is tantamount to applying a weighted sum of the features of neighboring nodes normalized by their degrees, which essentially performs Laplacian smoothing on the graph [19], and hence leads to oversmoothing. Also, the aggregation scheme of GCN uses 1-hop neighbors, and hence lacks the ability to capture long-range dependencies. To circumvent these issues, we define a higher-order implicit fairing network (HOIF-Net) with the following layer-wise propagation rule:

$$\mathbf{H}^{(\ell+1)} = \sigma(\big\Vert_{k=1}^{K} \tilde{\mathbf{H}}_k^{(\ell)}\tilde{\mathbf{W}}_k^{(\ell)}), \tag{13}$$

where

$$\tilde{\mathbf{H}}_k^{(\ell)} = (1-\alpha)\mathbf{S}^k\mathbf{H}^{(\ell)} + \alpha\mathbf{X}, \tag{14}$$

and $\mathbf{S}^k$ is the k-th power of the normalized adjacency matrix with self-added loops. Each (i,j)-th entry of $\mathbf{S}^k$ counts the number of walks of length k between nodes i and j. For example, the (i,j)-th entry of $\mathbf{S}^2$ gives the number of common neighbors of nodes i and j. The learnable weight matrix $\tilde{\mathbf{W}}_k^{(\ell)}$ is associated to the the node feature representation $\tilde{\mathbf{H}}_k^{(\ell)}$, and $\Vert$ denotes concatenation. For each k-hop neighborhood, the node feature representation $\tilde{\mathbf{H}}_k^{(\ell)}$ given by Eq. (14) is a weighted sum of the transformed feature matrix $\mathbf{S}^k\mathbf{H}^{(\ell)}$ for the ℓ-layer and the initial feature matrix $\mathbf{X}$. Intuitively, the transformation $\mathbf{S}^k\mathbf{H}^{(\ell)}$ yields a smooth hidden representation, and hence encourages similar predictions among k-hop neighboring nodes. The weighting factor α represents the weight assigned to the initial feature information that needs to be carried over, as the number of layers increase. Figure 1 shows an illustration of the layer-wise propagation rule of HOIF-Net when $K = 3$. Long-range dependencies between body joints are captured by high-order graph convolutions, which take into account distant neighbors when updating the learned node features. Note that HOIF-Net uses residual connections between the initial feature matrix and each hidden layer. Residual connections not only allow the model to carry over information from the initial node attributes, but also help facilitate training of multi-layer networks.

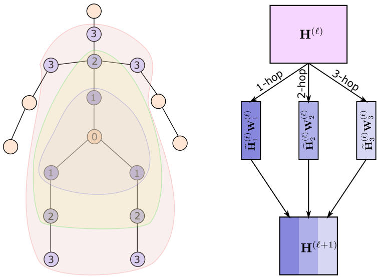
3
3
2
3
1
0
1
1
2
2
3
3
$\mathbf{H}^{(\ell)}$
1-hop
2-hop
3-hop
$\widetilde{\mathbf{H}}_1^{(\ell)}\mathbf{W}_1^{(\ell)}$
$\widetilde{\mathbf{H}}_2^{(\ell)}\mathbf{W}_2^{(\ell)}$
$\widetilde{\mathbf{H}}_3^{(\ell)}\mathbf{W}_3^{(\ell)}$
$\mathbf{H}^{(\ell+1)}$

Figure 1: Illustration of HOIF-Net feature concatenation for $K = 3$.

Model Architecture. The architecture of our proposed model for 3D human pose estimation is illustrated in Figure 2. The input consists of 2D keypoints generated via a 2D pose detector. The generated output of the proposed model consists of predicted 3D pose coordinates. We use higher-order graph convolutional layers defined by the layer-wise propagation rule of HOIF-Net to capture long-range structural information between body joints.

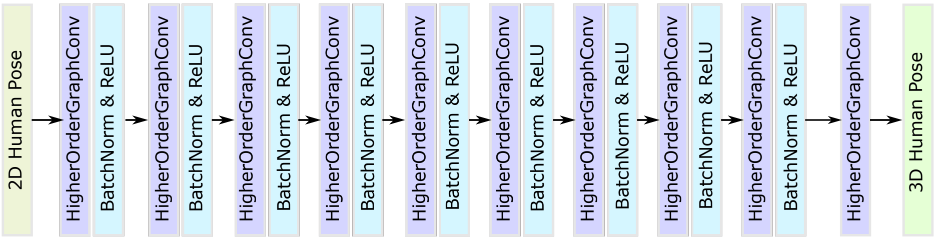

Figure 2: Overview of the proposed network architecture for 3D pose estimation. Our model takes 2D pose coordinates (17 joints) as input and generates 3D pose predictions (17 joints) as output. We use ten higher-order graph convolutional layers, each of which is followed by batch normalization and ReLU activation function, except the last convolutional layer.

Model Prediction. The output of the last higher-order graph convolutional layer of HOIF-Net contains the final output node embeddings, which are given by

$$\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N)^{\intercal} \in \mathbb{R}^{N \times 3}, \tag{15}$$

where $\hat{\mathbf{y}}_i$ is a three-dimensional raw vector of predicted 3D pose coordinates.

Model Training. The parameters (i.e. weight matrices for different layers) of the proposed HOIF-Net model for 3D human pose estimation are learned by minimizing the loss function

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2, \tag{16}$$

which is the mean squared error between the 3D ground truth poses $\mathbf{y}_i$ and estimated 3D joint poses $\hat{\mathbf{y}}_i$ over a training set consisting of N human poses.

5 Experiments

5.1 Experimental Setup

Datasets. We perform quantitative and qualitative evaluations on two standard, large-scale benchmark datasets: Human 3.6M and MPI-INF-3DHP. The Human 3.6M dataset [14] contains 3.6 million 3D human poses for 11 professional actors and corresponding images captured by a high-speed motion capture system with four different cameras. Each actor performs 15 actions (scenarios), including directions, discussion, eating, greeting, talking on the phone and so on. For data preprocessing, we apply standard normalization to the 2D and 3D poses before feeding the data to the model in line with previous work [22, 39]. For the MPI-INF-3DHP dataset [23], there are 8 actors performing 8 activities each. These activities range from walking and sitting to complex exercise poses and dynamic actions.

Evaluation Protocols and Metrics. For the Human 3.6M benchmark, there are two commonly used evaluation protocols, referred to as Protocol #1 and Protocol #2. Both protocols use 5 subjects (S1, S5, S6, S7, S8) for training and 2 subjects (S9, S11) for testing. Under Protocol #1, we report the mean per joint position error (MPJPE), which computes the average Euclidean distance between the predicted 3D joint positions and ground truth after the alignment of the root joint (central hip). Under Protocol #2, we report the Procrustes-aligned mean per joint position error (PA-MPJPE), where MPJPE is computed after rigid alignment of the prediction with respect to the ground truth. Both error metrics are measured in millimeters, and lower values indicate better performance. For MPI-INF-3DHP, we adopt two commonly-used evaluation metrics, namely Percentage of Correct Keypoints (PCK) under 150mm and the Area Under the Curve (AUC), following previous works [11, 25, 35]. Higher values of PCK and AUC indicate better performance.

Implementation Details. We train our model for 50 epochs using the Adam optimizer with a learning rate of 0.001. We set the decay factor to 0.96 per 100,000 steps, and the batch size to 64. We also set the hyperparameters α and β to 0.2 and 0.5, respectively, via grid search with cross-validation on the training set. To extract 2D keypoints from input images and following common practices in previous work [26, 39], we employ the cascaded pyramid network (CPN) [6], which uses bounding boxes obtained by Mask R-CNN [13]. For K-hop feature concatenation, we set the value of K to 3, as illustrated in Figure 1.

5.2 Results and Analysis

Quantitative Results. In Tables 1 and 2, we summarize the performance comparison results of our HOIF-Net model and various state-of-the-art methods for 3D pose estimation. As can be seen, our model performs the best in most of the actions and also on average under both Protocol #1 and Protocol #2, indicating that HOIF-Net is very competitive. Under Protocol #1, Table 1 shows that HOIF-Net performs better than high-order GCN [39] on 14 out of 15 actions, yielding an error reduction of approximately 1.44% on average over high-order GCN. Moreover, our model outperforms semGCN [37] by a relative improvement of 4.86%

on average. Under Protocol #2, Table 2 shows that our model performs better than high-order GCN with 1.83% error reduction on average, and also achieves better performance on 11 out of 15 actions.

Table 1: Performance comparison of our model and baseline methods using MPJPE (in millimeters) between the ground truth and estimated pose on Human3.6M under Protocol #1. The last column report the average errors, and boldface numbers indicate the best 3D pose estimation performance.

	Action															
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.* [22]	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
Sun *et al.* [30]	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Yang *et al.* [35]	51.5	58.9	**50.4**	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Fang *et al.* [9]	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
Hossain & Little [27]	48.4	**50.7**	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Pavlakos *et al.* [25]	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
Sharma *et al.* [29]	48.6	54.5	54.2	55.7	62.2	72.0	50.5	54.3	70.0	78.3	58.1	55.4	61.4	45.2	49.7	58.0
Zhao *et al.* [37]	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
Li *et al.* [18] (BH)	62.0	69.7	64.3	73.6	75.1	84.8	68.7	75.0	81.2	104.3	70.2	72.0	75.0	67.0	69.0	73.9
Banik *et al.* [2]	51.0	55.3	54.0	54.6	62.4	76.0	51.6	52.7	79.3	87.1	58.4	56.0	61.8	48.1	**44.1**	59.5
Xu *et al.* [34]	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 *et al.* [39]	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
Ours	**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**

Table 2: Performance comparison of our model and baseline methods using PA-MPJPE between the ground truth and estimated pose on Human3.6M under Protocol #2.

	Action															
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Pavlakos *et al.* [24]	47.5	50.5	48.3	49.3	50.7	55.2	46.1	48.0	61.1	78.1	51.1	48.3	52.9	41.5	46.4	51.9
Zhou *et al.* [38]	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1	**53.7**	65.5	51.6	50.4	54.8	55.9	55.3
Martinez *et al.* [22]	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
Sun *et al.* [30]	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
Fang *et al.* [9]	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
Hossain & Little [27]	**35.7**	**39.3**	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Lee *et al.* [16]	38.0	39.3	46.3	44.4	49.0	55.1	40.2	41.1	53.2	68.9	51.0	**39.1**	**33.9**	56.4	38.5	46.2
Li *et al.* [18] (BH)	38.5	41.7	39.6	45.2	45.8	**46.5**	37.8	42.7	52.4	62.9	45.3	40.9	45.3	38.6	38.4	44.3
Banik *et al.* [2]	38.4	43.1	42.9	44.0	47.8	56.0	39.3	39.8	61.8	67.1	46.1	43.4	48.4	40.7	**35.1**	46.4
Xu *et al.* [34]	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 *et al.* [39]	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
Ours	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**

Table 3 reports the quantitative comparison results of HOIF-Net and baseline methods on the MPI-INF-3DHP dataset. As can be seen, our method achieves the best performance on all evaluation metrics.

Qualitative Results. Figure 3 shows the qualitative results obtained by our model for various actions. Notice that the predictions made by HOIF-Net match perfectly the ground truth, indicating the effectiveness of our proposed approach in tackling the 2D-to-3D pose estimation problem.

Table 3: Performance comparison of our model and baseline methods on the MPI-INF-3DHP dataset using PCK and AUC as evaluation metrics. Higher values in boldface indicate the best performance.

Method	PCK	AUC
Yang *et al.* [35]	69.0	32.0
Pavlakos *et al.* [25]	71.9	35.3
Habibie *et al.* [11]	70.4	36.0
Ours	**72.8**	**36.5**

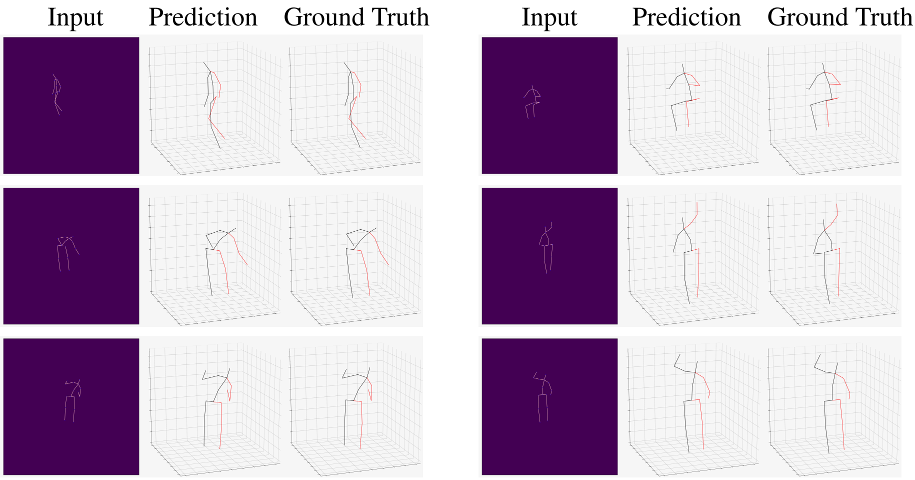

Figure 3: Qualitative results obtained by our model on the Human3.6M test set.

5.3 Ablation study

In our ablation experiments, we use the 2D ground truth as input to our model. We start by investigating the effect of the hyperparameters α and β on model performance. We conduct a sensitivity analysis to investigate how the performance of our model changes as we vary these two hyperparameters. In Figure 4 (left), we analyze the effect of α by plotting the error values vs. α for both protocols, where α varies from 0.1 to 0.5, and β is set to 1. We can see that our model achieves the lowest error values of MPJPE and PA-MPJPE when $\alpha = 0.12$ and $\alpha = 0.1$, respectively. In Figure 4 (right), we plot the error values vs. β for both protocols by varying the value of β from 0.1 to 1.5, and setting the value of α to 0.1. Notice that the best performance is generally achieved when $\beta = 0.7$

We also evaluate our method against SemGCN (Zhao *et al.* [37]) and high-order GCN (Zou *et al.* [39]), which are state-of-the-art GCN-based methods for 2D-to-3D pose estimation, and we report the results in Table 4. As can be seen, our approach outperforms both semGCN and High-order GCN under Protocols #1 and #2. Under Protocol #1, our HOIF-Net model outperforms semGCN and high-order GCN by 4.02 mm and 1.4 mm, corresponding to error reductions of 9.54% and 3.54%, respectively. Under Protocol #2, HOIF-Net outperforms semGCN and high-order GCN by 3.79 mm and 1.33 mm, corresponding to error reductions of 11.3% and 4.28%, respectively. In addition, our model offers comparable performance as high-order GCN, while using a much smaller number of filters (64 compared to

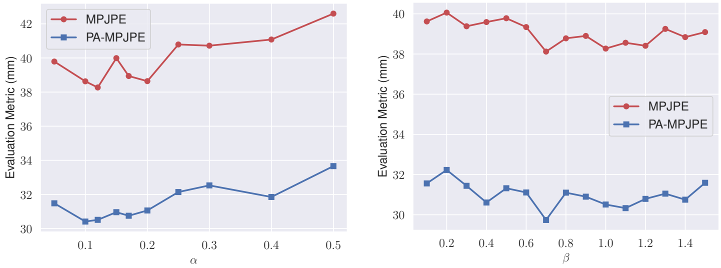

Figure 4: Parameter sensitivity analysis.

96) and also the number of learned parameters is reduced by more than half.

Table 4: Performance comparison of our model and GCN-based methods.

Method	Filters	Parameters	MPJPE	PA-MPJPE
SemGCN [37]	96	0.43M	42.14	33.53
High-order GCN [39]	96	1.20M	39.52	31.07
Ours	96	1.20M	**38.12**	**29.74**
Ours	64	**0.54M**	39.78	31.26

Figure 5 shows that the performance of the proposed HOIF-Net model on the Human3.6M dataset remains relatively stable as we increase the number of higher-order graph convolutional layers, demonstrating the robustness of our method against oversmoothing.

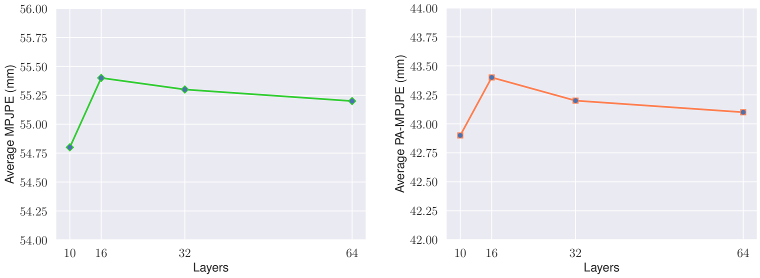

Figure 5: HOIF-Net's performance with increasing higher-order graph convolutional layers on the Human3.6M dataset.

6 Conclusion

In this paper, we proposed a higher-order implicit fairing network with initial residual connections for 3D human pose estimation, with the aim to alleviate the oversmoothing problem in graph convolutional networks, and also to capture long-range dependencies between body joints by enabling the model to aggregate multi-hop neighbors through feature concatenation. Empirical experiments and ablation studies showcase the merits of our model and demonstrate its competitive performance in comparison with state-of-the-art methods for 3D human pose estimation. For future work, we plan to apply the proposed framework to other downstream tasks such as semi-supervised node/graph classification and link prediction.

References

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *Proc. International Conference on Machine Learning*, pages 21–29, 2019.

[2] Soubarna Banik, Alejandro Mendoza Gracia, and Alois Knoll. 3D human pose regression using graph convolutional network. In *Proc. IEEE International Conference on Image Processing*, 2020.

[3] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks. In *Proc. IEEE International Conference on Computer Vision*, pages 2272–2281, 2019.

[4] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *Proc. International Conference on Machine Learning*, pages 1725–1735, 2020.

[5] 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 Pecognition*, pages 7103–7112, 2018.

[6] 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.

[7] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing*, pages 3844–3852, 2016.

[8] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proc. ACM SIGGRAPH*, pages 317–324, 1999.

[9] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3D pose estimation. In *Proc. AAAI Conference on Artificial Intelligence*, 2018.

[10] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[11] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2D features and intermediate 3D representations. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 10905–10914, 2019.

[12] D. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[13] K. He, G. Gkioxari, P. Dollár, , and R. Girshick. Mask R-CNN. In *Proc. IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[14] 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.

[15] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *International Conference on Learning Representations*, 2019.

[16] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proc. European conference on computer vision*, pages 123–141, 2018.

[17] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. CayleyNets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[18] Chen Li and Gim Hee Lee. Weakly supervised generative network for multiple 3D human pose hypotheses. In *Proc. British Machine Vision Conference*, 2020.

[19] Q. Li, Z. Han, and X.M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI Conference on Artificial Intelligence*, pages 3538–3545, 2018.

[20] Sijin Li and Antoni B Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *Proc. Asian Conference on Computer Vision*, pages 332–347, 2014.

[21] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. Comprehensive study of weight sharing in graph networks for 3D human pose estimation. In *Proc. European Conference on Computer Vision*, 2020.

[22] 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.

[23] 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 *International Conference on 3D Vision*, 2017.

[24] 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.

[25] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

[26] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019.

[27] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3D human pose estimation. In *Proc. European Conference on Computer Vision*, pages 68–84, 2018.

[28] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[29] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3D human pose estimation by generation and ordinal ranking. In *Proc. IEEE International Conference on Computer Vision*, pages 2325–2334, 2019.

[30] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proc. IEEE International Conference on Computer Vision*, pages 2602–2611, 2017.

[31] G. Taubin, T. Zhang, and G. Golub. Optimal surface smoothing as filter design. In *Proc. European Conference on Computer Vision*, 1996.

[32] Asiri Wijesinghe and Qing Wang. DFNets: Spectral CNNs for graphs with feedback-looped filters. In *Advances in Neural Information Processing Systems*, 2019.

[33] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proc. International Conference on Machine Learning*, 2018.

[34] 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.

[35] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3D human pose estimation in the wild by adversarial learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[36] Lingxiao Zhao and Leman Akoglu. PairNorm: Tackling oversmoothing in GNNs. In *International Conference on Learning Representations*, 2020.

[37] 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.

[38] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: a weakly-supervised approach. In *Proc. IEEE International Conference on Computer Vision*, pages 398–407, 2017.

[39] 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.