Spatio-Temporal MLP-Graph Network for 3D Human Pose Estimation

Tanvir Hassan
tanvirhassan970@gmail.com

A. Ben Hamza
hamza@ciise.concordia.ca

Concordia University
Montreal, QC, Canada

Abstract

Graph convolutional networks and their variants have shown significant promise in 3D human pose estimation. Despite their success, most of these methods only consider spatial correlations between body joints and do not take into account temporal correlations, thereby limiting their ability to capture relationships in the presence of occlusions and inherent ambiguity. To address this potential weakness, we propose a spatio-temporal network architecture composed of a joint-mixing multi-layer perceptron block that facilitates communication among different joints and a graph weighted Jacobi network block that enables communication among various feature channels. The major novelty of our approach lies in a new weighted Jacobi feature propagation rule obtained through graph filtering with implicit fairing. We leverage temporal information from the 2D pose sequences, and integrate weight modulation into the model to enable untangling of the feature transformations of distinct nodes. We also employ adjacency modulation with the aim of learning meaningful correlations beyond defined linkages between body joints by altering the graph topology through a learnable modulation matrix. Extensive experiments on two benchmark datasets demonstrate the effectiveness of our model, outperforming recent state-of-the-art methods for 3D human pose estimation.

1 Introduction

3D human pose estimation is a fundamental task in computer vision, with the aim of predicting the 3D pose of a human body from images or videos [25]. While significant strides have been made in recent years [24], accurately estimating the 3D human pose remains a challenging problem. This is largely attributed to the complex and articulated nature of the human body, as well as the difficulty of estimating 3D information from 2D images [28, 35].

Graph convolutional networks (GCNs) have recently emerged as a powerful framework for 3D human pose estimation [41]. Despite their promising results, GCN-based methods have several limitations. First, they use the same transformation matrix for all nodes in graph convolution, thereby limiting information exchange. To address this limitation, Liu *et al.* [24] introduce various weight unsharing mechanisms. One drawback of these mechanisms is that they result in a larger model size that scales with the number of body joints. Second, GCNs suffer from the oversmoothing problem [17], where the model may struggle to accurately distinguish between nodes and learn meaningful representations due to repeated

graph convolutions as the network depth increases. Third, to leverage temporal correlations, these methods require significant computational resources to process a larger number of input sequences such as a 243-frame sequence. Furthermore, GCNs may not be able to capture the global contextual information or long-range dependencies between nodes in the graph, which can limit their ability to learn more complex relationships and patterns in the data.

On the other hand, Transformer architectures, which utilize a multi-head self-attention mechanism to capture both spatial and temporal correlations from sequences of 2D poses [43], have proven effective at capturing long-range dependencies between body joints in the spatio-temporal domain. However, the complexity of the self-attention block increases quadratically with the number of input sequences, making the training and inference more computationally expensive. Taking this into account, Tolstikhin *et al.* [34] propose MLP-Mixer, which has shown competitive performance compared to more complex architectures such as Transformer networks. Compared to multi-layer perceptrons (MLPs), the MLP-Mixer model has been shown to be effective at modeling long-range dependencies in the input data. However, MLP-based models do not adequately capture the local information due largely to the lack of prior knowledge about the human skeleton topology.

In this paper, we address the aforementioned challenges by proposing a novel spatio-temporal graph neural network architecture, dubbed MLP-GraphWJ mixer, which leverages spatio-temporal correlations and also makes use of weight and adjacency modulation. The proposed framework employs a weighted Jacobi (WJ) feature propagation rule obtained via graph filtering with implicit fairing. In summary, we make the following key contributions:

- We propose a graph weighted Jacobi (GraphWJ) network, which employs a weighted Jacobi (WJ) feature propagation rule obtained via graph filtering with implicit fairing, and also leverages weight and adjacency modulation.

- We design a spatio-temporal network architecture by incorporating MLPs to capture global information and a GraphWJ network to capture local information between adjacent joints across different channels.

- We demonstrate through experiments and ablation studies that our proposed model outperforms strong baselines, attaining state-of-the-art performance in 3D human pose estimation, while retaining a small model size.

2   Related work

3D Human Pose Estimation.   The basic goal of 3D human pose estimation is to estimate the 3D coordinates of the joints in the human body from images or videos. Single-stage and two-stage methods are two commonly used approaches for 3D human pose estimation. Single-stage methods are based on a direct regression from the input image to the 3D pose estimation [18]. These methods typically use convolutional neural networks to extract features from the input image and then use a regression network to directly estimate the 3D pose. Two-stage methods, on the other hand, generally consist of two separate networks: an off-the-shelf 2D pose detection network to extract 2D keypoints and a 3D pose estimation network [3, 5, 7, 8, 12, 22, 29, 30, 53, 36, 41, 45]. Two-stage methods usually achieve higher accuracy than single-stage methods, especially for complex pose estimation tasks.

Spatio-Temporal Methods.   Current monocular 3D pose estimation methods can be classified into two mainstream types: single-frame or image-based and multi-frame or video-

based approaches. Single-frame-based methods aim to predict 3D pose from a single RGB image. In contrast, video-based methods take advantage of the temporal dependencies between frames in the video clip. Due to the ill-posed characteristic of generating accurate 3D poses from a single RGB image, a number of techniques [8, 12, 21, 37, 38, 43] have been developed that rely on temporal correlations to improve the robustness and accuracy of the resulting 3D poses. Liu *et al.* [21] develop graph attention blocks in conjunction with dilated temporal convolution that is capable of estimating 3D pose from consecutive 2D pose sequences. Zheng *et al.* [43] utilize a Transformer-based approach that is designed to capture both the correlations between human joints and their temporal dependencies. Zeng *et al.* [38] introduce a temporal aware dynamic graph convolution where the graph updates by physical skeleton topology and the features of nodes. However, most of these methods tend to be computationally demanding, and are inherently restricted in their ability to establish temporal connectivity. Moreover, most GCN-based approaches are constrained by the fact that they share a feature transformation for capturing the relationships between each node and its adjacent nodes in a graph convolution layer. Also, sharing the same feature transformation for all nodes hinders the efficient exchange of information between the body joints. Our proposed framework falls under the category of 2D-to-3D pose lifting. It differs from existing GCN-based approaches in that we use a new graph propagation rule combined with weight and adjacency modulation to learn additional connections between body joints by adjusting the graph topology through a learnable modulation matrix. We also design a network architecture that combines the strengths of MLPs and graph neural networks in order to improve accuracy in 3D human pose estimation, while maintaining simplicity and efficiency.

3 Proposed Method

3.1 Preliminaries and Problem Formulation

Basic Notions. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V} = \{1, \ldots, N\}$ is a set of nodes that correspond to body joints, $\mathcal{E}$ is the set of edges representing connections between two neighboring body joints, and $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\mathsf{T}}$ is an $N \times F$ feature matrix of node attributes whose i-th row $\mathbf{x}_i$ is an F-dimensional feature vector associated to node i. We denote by $\mathbf{A}$ an $N \times N$ adjacency matrix whose (i, j)-th entry is equal to 1 if there the edge between neighboring nodes i and j, and 0 otherwise. We also denote by $\hat{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ the normalized adjacency matrix, where $\mathbf{D} = \text{diag}(\mathbf{A1})$ is the diagonal degree matrix.

Weighted Jacobi Method. Given a matrix $\mathbf{B} \in \mathbb{R}^{N \times N}$ and a vector $\mathbf{x} \in \mathbb{R}^N$, the weighted Jacobi iteration [32] for solving a matrix equation $\mathbf{Bh} = \mathbf{x}$ is given by

$$\mathbf{h}^{(k+1)} = \omega \, \text{diag}(\mathbf{B})^{-1} \mathbf{x} + (\mathbf{I} - \omega \, \text{diag}(\mathbf{B})^{-1} \mathbf{B}) \mathbf{h}^{(k)}, \tag{1}$$

where ω is a relaxation factor, and $\mathbf{h}^{(k)}$ and $\mathbf{h}^{(k+1)}$ are the k-th and $(k+1)$-th iterations of the unknown $\mathbf{h}$, respectively.

Problem Statement. 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 is to learn a regression model $f : \mathcal{X} \to \mathcal{Y}$ by finding a minimizer of the following loss function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^N l(f(\mathbf{x}_i), \mathbf{y}_i), \tag{2}$$

where $l(f(\mathbf{x}_i), \mathbf{y}_i)$ is an empirical loss function defined by the learning task. Since human pose estimation is a regression task, we define $l(f(\mathbf{x}_i), \mathbf{y}_i)$ as a weighted sum (convex combination) of the ℓ_2 and ℓ_1 loss functions

$$l = (1 - \lambda) \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2 + \lambda \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_1, \tag{3}$$

where $\lambda \in [0, 1]$ is a weighting factor controlling the contribution of each term.

3.2 Graph Filtering with Implicit Fairing

In the context of graph filtering, the implicit fairing approach [6] is applied by defining a Laplacian operator on the graph, which captures the connectivity and structure of the graph. Specifically, graph filtering with implicit fairing can be performed by solving the sparse linear system $(\mathbf{I} + s\mathbf{L})\mathbf{H} = \mathbf{X}$, where $\mathbf{X}$ is the feature matrix of node attributes, $\mathbf{L} = \mathbf{I} - \hat{\mathbf{A}}$ is the normalized Laplacian matrix, $\mathbf{H}$ is the filtered graph signal, and s is a positive scalar. This sparse linear system can be efficiently solved using the weighted Jacobi method [32], which uses a weighting parameter ω to compute the k-th iteration as follows:

$$\begin{aligned} \mathbf{H}^{(k+1)} &= \omega(\text{diag}(\mathbf{I}+s\mathbf{L}))^{-1}\mathbf{X} + \big(\mathbf{I} - \omega(\text{diag}(\mathbf{I}+s\mathbf{L}))^{-1}(\mathbf{I}+s\mathbf{L})\big)\mathbf{H}^{(k)} \\ &= \mathbf{H}^{(k)} - \omega\mathbf{H}^{(k)} + (1-\alpha)\omega\hat{\mathbf{A}}\mathbf{H}^{(k)} + \alpha\omega\mathbf{X} \end{aligned} \tag{4}$$

where $\alpha = 1/(1+s)$, and ω can be chosen to optimize the convergence speed of the method.

3.3 Graph Weighted Jacobi Network

In graph neural networks, the aim of a layer-wise propagation rule is to encourage the network to learn useful representations at each layer that can be used to inform subsequent layers and ultimately improve the accuracy of the network's predictions. Drawing inspiration from the weighted Jacobi iterative solution for graph filtering with implicit fairing, we define a weighted Jacobi (WJ) layer-wise propagation rule as

$$\mathbf{H}^{(\ell+1)} = \sigma(\text{WJ}(\mathbf{H}^{(\ell)})), \quad \ell = 0, \dots, L-1 \tag{5}$$

where $\sigma(\cdot)$ is an activation function such as the Gaussian Error Linear Unit (GELU) [11] and L is the number of layers. The input of the first layer is the initial feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$. The weighted Jacobi operation on the input feature matrix $\mathbf{H}^{(\ell)}$ of the ℓ-th layer is given by

$$\text{WJ}(\mathbf{H}^{(\ell)}) = \mathbf{H}^{(\ell)}\mathbf{W}_1 - \Omega \odot (\mathbf{H}^{(\ell)}\mathbf{W}_2) + (1-\alpha)\Omega \odot (\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}_2) + \alpha\Omega \odot (\mathbf{X}\mathbf{W}_3), \tag{6}$$

where $\odot$ denotes element-wise matrix multiplication, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$ are learnable weight matrices, and Ω is a learnable weight modulation matrix. Notice that unlike the weighted Jacobi iteration, the proposed weighted Jacobi layer-wise propagation rule updates node features across layers, employs trainable weight matrices to learn an optimized graph representation, incorporates a learnable weight modulation matrix that functions similarly to the weighted parameter in the weighted Jacobi method, and applies a nonlinear activation function to capture the nonlinearity of the graph structure.

Adjacency Modulation. The graph structure has a limitation in that it cannot capture relationships between distant nodes. To tackle this issue, we use adjacency modulation [44], defined as $\check{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}$, where $\mathbf{Q}$ is an $N \times N$ learnable modulation matrix.

3.4 MLP-Graph Weighted Jacobi Mixer Model

Model Architecture. Inspired by the MLP-Mixer [34] and its recent variants for 3D human motion forecasting tasks [2, 19], the architecture of the proposed MLP-GraphWJ mixer consists of three main stages: **1)** skeleton embedding, **2)** MLP-GraphWJ mixer layer, and **3)** regression head. The overall architecture of the proposed model is illustrated in Figure 1, which shows that the joint-mixing layer aggregates information across different positions within each channel using MLPs, while the GraphWJ mixing layer is responsible for aggregating information across different channels of the input using the weighted Jacobi (WJ) feature propagation rule. The output of the final GraphWJ mixing layer is then passed on to the regression head network.
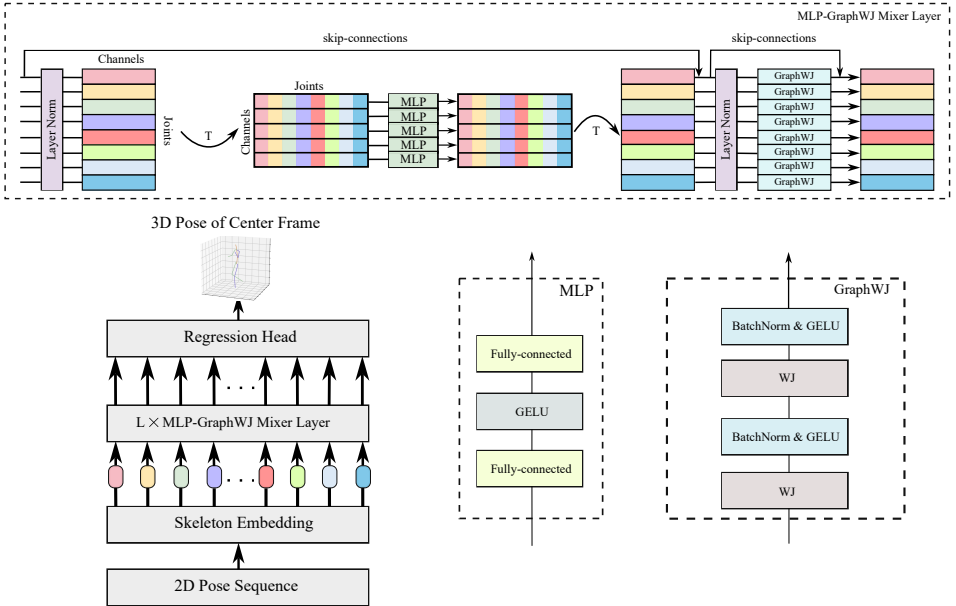

Figure 1: Schematic diagram of the proposed network architecture for 3D human pose estimation. The architecture is comprised of three main components: skeleton embedding, MLP-GraphWJ mixer layer, and a regression head. The MLP-GraphWJ mixer layer consists of a joints mixing MLP layer and a GraphWJ mixing layer. The architecture also includes additional components such as skip connections, dropout, layer normalization, and batch normalization.

1) Skeleton Embedding: To incorporate temporal information into our model, we take a 2D pose sequence as input. Given a 2D pose sequence $\mathbf{S} \in \mathbb{R}^{N \times 2 \times T}$ represented as a tensor, where T denotes the number of frames and N is the number of joints, we first reshape it into a matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{N \times 2T}$ by concatenating the 2D coordinates of all frames. Then, we pass it through a fully-connected layer, resulting in an $N \times F$ embedding matrix $\mathbf{X} = \tilde{\mathbf{S}}\mathbf{W}_4$, where $\mathbf{W}_4 \in \mathbb{R}^{2T \times F}$ is a learnable weight matrix and F is the embedding dimension.

2) MLP-GraphWJ Mixer Layer: MLP-based models are not well-suited for handling graph-structured data, as they simply connect all nodes without considering the graph structure. To

address this issue, we propose the MLP-GraphWJ mixer layer, which takes the advantages of both MLPs and graph neural networks in a single layer. Compared to the MLP-Mixer, our proposed MLP-GraphWJ mixer layer leverages graph neural networks to extract features of different channels, thereby helping to preserve domain-specific knowledge pertaining to human body configurations. Specifically, our MLP-GraphWJ mixer layer consists of two sub-layers: a joint-mixing MLP and a GraphWJ mixing layer. The joint-mixing MLP block allows communication between different joints, while the GraphWJ mixing layer allows communication between different channels. The joint-mixing MLP acts on the columns of the input feature matrix $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times F}$ (i.e., applied to its transpose). On the other hand, the GraphWJ mixing layer acts on the rows of its input feature. The joint-mixing MLP block contains two fully-connected layers. We also add a skip connection between the input and output. Hence, the output of the joint-mixing MLP is an $N \times F$ matrix given by

$$\mathbf{U}^{(\ell+1)} = \mathbf{H}^{(\ell)} + \left(\mathbf{W}_6 \sigma \left(\mathbf{W}_5 \left(\mathsf{LN} \left(\mathbf{H}^{(\ell)} \right)^{\mathsf{T}} \right) \right) \right)^{\mathsf{T}}, \tag{7}$$

where $\mathsf{LN}(\cdot)$ is layer normalization [■], $\mathbf{W}_5 \in \mathbb{R}^{N \times F}$ and $\mathbf{W}_6 \in \mathbb{R}^{F \times N}$ are learnable weight matrices. The input of the first layer is the $N \times F$ embedding matrix $\mathbf{H}^{(0)} = \mathbf{X}$.

On the other hand, our GraphWJ mixing layer consists of two weighted Jacobi (WJ) layers. The output $\mathbf{U}^{(\ell+1)}$ of the joint-mixing MLP layer is fed into the GraphWJ mixing layer, which acts on the rows of its input matrix. Hence, the outputs of the first and second WJ layers are given by

$$\mathbf{P}^{(\ell+1)} = \sigma \left(\mathsf{BN} \left(\mathsf{WJ} \left(\mathbf{U}^{(\ell+1)} \right) \right) \right) \in \mathbb{R}^{N \times R} \tag{8}$$

and

$$\mathbf{Q}^{(\ell+1)} = \sigma \left(\mathsf{BN} \left(\mathsf{WJ} \left(\mathbf{P}^{(\ell+1)} \right) \right) \right) \in \mathbb{R}^{N \times F}, \tag{9}$$

where $\mathsf{BN}(\cdot)$ is a batch normalization layer, and R and F are embedding dimensions. Batch normalization is similar to layer normalization, but instead of normalizing across the features of each input, it normalizes across a batch of inputs.

Finally, the output $\mathbf{Z}$ of the last MLP-GraphWJ mixing layer is obtained by adding a skip connection as follows:

$$\mathbf{Z} = \mathbf{U}^{(L)} + \mathbf{Q}^{(L)} \in \mathbb{R}^{N \times F} \tag{10}$$

3) Regression Head: The output $\mathbf{Z}$ of the last MLP-GraphWJ mixing layer is passed on to the regression head network comprised of a layer normalization, followed by a linear fully connected layer, yielding a prediction $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \dots, \hat{\mathbf{y}}_N)^{\mathsf{T}} \in \mathbb{R}^{N \times 3}$ of estimated 3D joint positions. This prediction is regressed to the ground-truth of the 3D pose for the center frame during model training.

Model Training.   In order to train the MLP-GraphWJ mixer model for 3D human pose estimation, the weight matrices for various layers are optimized by minimizing the following loss function

$$\mathcal{L} = \frac{1}{N} \left[(1 - \lambda) \sum_{i=1}^{N} \| \mathbf{y}_i - \hat{\mathbf{y}}_i \|_2^2 + \lambda \sum_{i=1}^{N} \| \mathbf{y}_i - \hat{\mathbf{y}}_i \|_1 \right], \tag{11}$$

which is a weighted combination of the mean squared and mean absolute errors between the estimated 3D joint positions $\hat{\mathbf{y}}_i$ and the ground-truth positions $\mathbf{y}_i$ over N training body joints.

4 Experiments

In this section, we evaluate the performance of our model against competitive baselines for 3D human pose estimation. More detailed descriptions of the datasets, additional experimental results and ablation studies are provided in the supplementary material. Code is available at: https://github.com/nies14/Spatio-Temporal-MLP-Graph

4.1 Experimental Setup

Datasets. We assess the performance of our model on two widely used benchmark datasets for 3D human pose estimation: Human3.6M [13] and MPI-INF-3DHP [26].

Evaluation Protocols and Metrics. For Human3.6M, we adopt two commonly used metrics, mean per joint position error (MPJPE) and Procrustes-aligned mean per joint position error (PA-MPJPE), which are measured in millimeters. A lower value of these metrics indicates better performance. For MPI-INF-3DHP, we evaluate our model using two standard metrics: Percentage of Correct Keypoints (PCK) within 150mm and Area Under the Curve (AUC), consistent with previous studies [10, 16, 29, 51, 56, 44]. Improved model performance is indicated by higher values of PCK and AUC.

Baselines. We evaluate the performance of our MLP-GraphWJ mixer model against various state-of-art methods, including semantic GCN [41], spatio-temporal GCN (ST-GCN) [3], Weight Unsharing [23], temporal convolutions and semi-supervised training [30], skeletal GNN [38], graph mixture density network (GraphMDN) [27], split-and-recombine network (SRNet) [37], graph attention spatio-temporal network (GAST-Net) [21], PoseFormer [43], modulated GCN (MGCN) [44], group graph convolutional networks (GroupGCN) [40], mesh transformer (METRO) [20], and pose augmentation (PoseAug) [9].

Implementation Details. We train the model using AMSGrad optimizer for 50 epochs, and the initial learning rate is set to 0.001 with a decay factor of 0.95 applied after each epoch and 0.5 after every 5 epochs. For 2D pose detections [4], we set the batch size to 256, the number of layers $L = 3$, the skeleton embedding layer hidden dimension and the MLP hidden dimension $F = 384$, and the GraphWJ mixing layer hidden dimension $R = 768$. We set the weighting factor $\lambda = 0.01$, $\alpha = 0.1$, and the total number of input frames $T = 243$ for both 2D detected poses and ground truth poses.

4.2 Results and Analysis

Quantitative Results. In Table 1, we report the performance comparison results of our MLP-GraphWJ mixer model and various state-of-art methods for 3D human pose estimation. As can be seen, our model demonstrates superior performance with detected 2D pose as an input across most actions and overall, as evidenced by both Protocol #1 and Protocol #2. These findings demonstrate the model's competitiveness, which is largely attributed to the fact that MLP-GraphWJ mixer can better exploit joint connections through the proposed graph propagation rule and also learns not only different modulation vectors for different body joints, but also additional connections between the joints. Under Protocol #1, Table 1 shows that using a single frame MLP-GraphWJ mixer performs better than MGCN [44] on 14 out of 15 actions by a relative improvement of 10.73% on average. Of significance is the fact that unlike our method, MGCN [44] employs a non-local layer. Despite this difference,

our model demonstrates superior performance compared to MGCN [44], highlighting the efficacy of our approach. Our model also performs better than Skeletal GNN [38], a temporal graph neural network method for hard 3D pose estimation, yielding an error reduction of approximately 3.92% on average. Under Protocol #2, our approach outperforms spatio-temporal GCN with a relative improvement (average) of 6.67% in terms of PA-MPJPE.

Table 1: Performance comparison of our model and baseline methods on Human3.6M under Protocol #1 and Protocol #2 using the detected 2D pose as input. The average errors are reported in the last column. Boldface numbers indicate the best performance, and the underlined numbers indicate the second-best performance. (†) - uses temporal information.

	Action															
Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Zhao et al. [41]	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
Quan et al. [51]	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
Liu et al. [23]	46.3	52.2	47.3	50.7	55.5	67.1	49.2	46.0	60.4	71.1	51.5	50.1	54.5	40.3	43.7	52.4
Lin et al. [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.0
Zhao et al. [42]	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
Lee et al. [14]	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 [40]	45.0	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
Gong et al. [9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.2
Zou et al. [44]	45.4	49.2	45.7	49.4	50.4	58.2	47.9	46.0	57.5	63.0	49.7	46.6	52.2	38.9	40.8	49.4
Cai et al. [3] (†)	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
Li et al. [19]	43.7	49.3	45.5	47.8	50.5	56.0	46.3	44.1	55.9	59.0	48.4	45.7	51.2	37.1	39.1	48.0
Pavllo et al. [50] (†)	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
Oikarinen et al. [27]	<u>40.0</u>	**43.2**	41.0	43.4	50.0	53.6	**40.1**	41.4	**52.6**	67.3	48.1	44.2	49.0	39.5	40.2	46.2
Zeng et al. [58] (†)	43.1	50.4	43.9	45.3	46.1	57.0	46.3	47.6	56.3	61.5	47.7	47.4	53.5	35.4	37.3	47.9
Zeng et al. [57] (†)	46.6	47.1	43.9	**41.6**	<u>45.8</u>	<u>49.6</u>	46.5	**40.0**	53.4	61.1	46.1	**42.6**	<u>43.1</u>	<u>31.5</u>	32.6	44.8
Liu et al. [21] (†)	43.3	46.1	<u>40.9</u>	44.6	46.6	54.0	44.1	42.9	55.3	<u>57.9</u>	45.8	43.4	47.3	**30.4**	**30.3**	44.9
Zheng et al. [43] (†)	41.5	44.8	**39.8**	<u>42.5</u>	46.5	51.6	<u>42.1</u>	42.0	<u>53.3</u>	60.7	<u>45.5</u>	43.3	46.1	31.8	<u>32.2</u>	<u>44.3</u>
Ours (†)	**38.9**	<u>44.5</u>	41.4	43.7	**45.0**	**48.7**	42.8	**39.5**	54.9	67.1	**42.5**	<u>43.1</u>	44.0	33.2	33.0	**44.1**
Protocol #2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Lee et al. [15] (†)	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
Quan et al. [51]	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
Liu et al. [23]	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
Lee et al. [14]	35.7	39.6	37.3	41.4	40.0	44.9	37.6	36.1	46.5	54.1	40.9	36.4	42.8	31.7	34.7	40.3
Zhang [40]	35.3	39.3	38.4	40.8	41.4	45.7	36.9	35.1	48.9	55.2	41.2	36.3	42.6	30.9	33.7	40.1
Zou et al. [44]	35.7	38.6	36.3	40.5	<u>39.2</u>	44.5	37.0	35.4	46.4	51.2	40.5	35.6	41.7	30.7	33.9	39.1
Gong et al. [9]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.1
Cai et al. [3] (†)	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
Lin et al. [20]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	36.7
Pavllo et al. [50] (†)	<u>34.1</u>	**36.1**	<u>34.4</u>	**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
Ours (†)	**33.0**	<u>36.8</u>	**34.3**	<u>37.5</u>	**36.4**	40.4	**34.1**	**31.9**	<u>45.4</u>	57.0	**35.6**	**34.8**	36.2	<u>26.5</u>	**26.9**	**36.4**

In Table 2, we report the quantitative comparison results of MLP-GraphWJ mixer using a single frame in comparison with several baselines on the MPI-INF-3DHP dataset. As can be seen, our method achieves significant improvements over the comparative methods. Our model outperforms the best performing baseline with relative improvements of 0.81% and 1.30% in terms of the PCK and AUC metrics, respectively. Although we train the model using only the Human3.6M dataset, our method outperforms others on MPI-INF-3DHP, indicating that our approach has strong generalization capabilities to unseen human poses.

Qualitative Results. Figure 2 shows some visualization results of the proposed MLP-

Table 2: Performance comparison of our model without pose refinement and baseline methods on the MPI-INF-3DHP dataset using PCK and AUC as evaluation metrics.

Method	PCK(↑)	AUC(↑)
Chen et al. [16]	67.9	-
Yang et al. [36]	69.0	32.0
Pavlakos et al. [29]	71.9	35.3
Habibie et al. [10]	70.4	36.0
Quan et al. [31]	72.8	36.5
Zeng et al. [37]	77.6	43.8
Zhang et al. [40]	81.1	49.9
Zeng et al. [38]	82.1	46.2
Zou et al. [44]	86.1	53.7
Ours	**86.8**	**54.4**

GraphWJ mixer model on Human3.6M. As can be seen, the 3D predictions on various actions made by our model are superior to those of MGCN [44] and more closely match the ground truth. This indicates the effectiveness of our approach. Notice that MGCN [44] struggles to accurately predict hand poses when there are overlapping joints or occlusions, whereas our model is able to predict them with a high degree of accuracy.
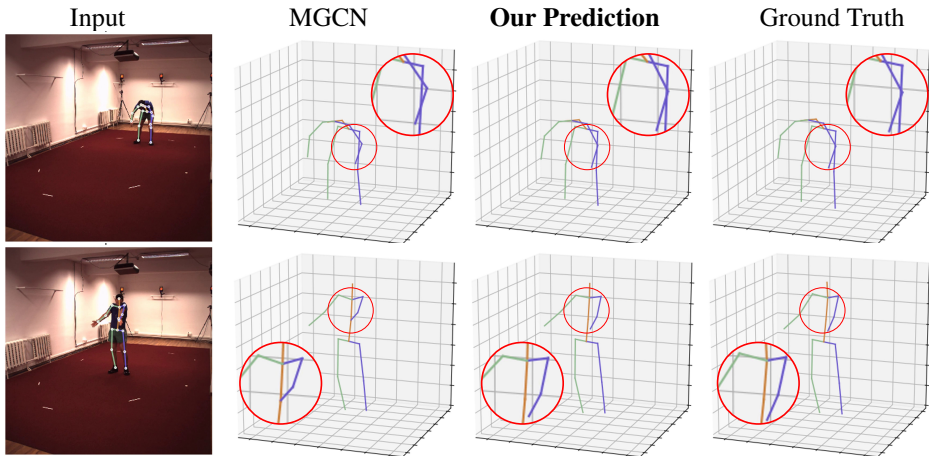

Figure 2: Qualitative comparison between our model and MGCN on Human3.6M actions.

4.3 Ablation Study

We conduct ablation experiments on the Human3.6M dataset under Protocol #1 using MPJPE as evaluation metric. Specifically, we investigate the effectiveness of each component in our network architecture. The results are presented in Table 3, where the first row corresponds to the performance of the MLP-Mixer baseline model [34] that does not include any GCN components. The remaining rows in the table display the results of replacing various components of the baseline model. We fix the number of parameters to be about 0.95M by merely changing the number of hidden dimensions of each model. Our proposed MLP-GraphWJ mixer

outperforms the baseline model by a margin of 1.9mm, demonstrating that the combined use of these components leads to more accurate 3D pose estimation.

Table 3: Effectiveness of each component used in our method without pose refinement on Human3.6M under Protocol#1 using detected 2D poses as input.

Joint-Mixing MLP	Channel-Mixing MLP	GraphWJ Mixing Layer	Standard GCN	Weighted Jacobi	MPJPE ($\downarrow$)
✓	✓	✗	✗	✗	53.1
✓	✗	✓	✓	✗	51.5
✓	✗	✓	✗	✓	**51.2**

Runtime Analysis. We report the model performance, the total number of parameters, and estimated floating-point operations (FLOPs) per frame with various input sequence lengths (T) in Table 4. We can see that increasing the sequence length of our model results in improved accuracy, while keeping the total number of learned parameters low.

Table 4: Comparison of our model and baselines in terms of total number of parameters, FLOPs, and MPJPE. The evaluation is performed without pose refinement on Human3.6M under Protocol#1 using detected 2D poses as input.

Method	Frames (T)	Params. (M)	FLOPs (M)	MPJPE($\downarrow$)
VideoPose [30]	27	8.56	17.09	48.8
PoseFormer [43]	9	9.58	150.0	49.9
Ray3D [39]	9	27.50	-	49.7
Ours	1	5.42	29.01	50.8
Ours	9	5.43	29.21	**48.7**

5 Conclusion

In this paper, we proposed a novel network architecture, named MLP-GraphWJ mixer, which is comprised of an MLP-mixer layer and a GraphWJ mixer layer. The MLP-mixer layer aggregates information across different positions within each channel, while the graph weighted Jacobi network layer aggregates information across different channels. Our contribution involves the incorporation of a feature propagation rule based on weighted Jacobi iteration, which is derived through the application of graph filtering with implicit fairing. The proposed model outperforms recent state-of-the-art techniques on two standard benchmark datasets for 3D human pose estimation, as demonstrated by our experimental results. Our approach achieves this improved performance while employing a model with a smaller parameter count. For future work, we aim to take high-order connectivity between joints into account by aggregating information from multi-hop neighbors.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Arij Bouazizi, Adrian Holzbock, Ulrich Kressel, Klaus Dietmayer, and Vasileios Belagiannis. MotionMixer: MLP-based 3D human body pose forecasting. In *Proc. International Joint Conference on Artificial Intelligence*, 2022.

[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. Conference on Computer Vision and Pattern Recognition*, pages 2272–2281, 2019.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Kehong Gong, Jianfeng Zhang, and Jiashi Feng. PoseAug: A differentiable pose augmentation framework for 3D human pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[10] 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.

[11] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3D human pose estimation with mixture density network. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 9887–9895, 2019.

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proc. AAAI Conference on Artificial Intelligence*, pages 3538–3545, 2018.

[18] 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.

[19] Wenhao Li, Hong Liu, Tianyu Guo, Hao Tang, and Runwei Ding. GraphMLP: A graph MLP-like architecture for 3D human pose estimation. *arXiv preprint arXiv:2206.06420*, 2022.

[20] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[21] Junfa Liu, Juan Rojas, Yihui Li, Zhijun Liang, Yisheng Guan, Ning Xi, and Haifei Zhu. A graph attention spatio-temporal convolutional network for 3D human pose estimation in video. In *Proc. IEEE International Conference on Robotics and Automation*, pages 3374–3380, 2021.

[22] 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.

[23] 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. Springer, 2020.

[24] Wu Liu, Qian Bao, Yu Sun, and Tao Mei. Recent advances of monocular 2D and 3D human pose estimation: A deep learning perspective. *ACM Computing Surveys*, 55(4): 1–41, 2022.

[25] 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.

[26] 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.

[27] Tuomas Oikarinen, Daniel Hannah, and Sohrob Kazerounian. GraphMDN: Leveraging graph structure and deep learning to solve inverse problems. In *Proc. IEEE International Joint Conference on Neural Networks*, pages 1–9, 2021.

[28] 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.

[29] 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.

[30] 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.

[31] Jianning Quan and A. Ben Hamza. Higher-order implicit fairing networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2021.

[32] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[33] 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.

[34] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. MLP-Mixer: An all-MLP architecture for vision. In *Advances in Neural Information Processing Systems*, pages 24261–24272, 2021.

[35] Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L. Yuille, and Wen Gao. Robust estimation of 3D human poses from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4328, 2014.

[36] 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.

[37] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Ching-Feng Lin. SRNet: Improving generalization in 3D human pose estimation with a split-and-recombine approach. In *Proc. European Conference on Computer Vision*, 2020.

[38] 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.

[39] Yu Zhan, Fenghai Li, Renliang Weng, and Wongun Choi. Ray3D: Ray-based 3D human pose estimation for monocular absolute 3D localization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 13116–13125, 2022.

[40] Zijian Zhang. Group graph convolutional networks for 3D human pose estimation. In *Proc. British Machine Vision Conference*, 2022.

[41] 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.

[42] Weixi Zhao, Weiqiang Wang, and Yunjie Tian. GraFormer: Graph-oriented transformer for 3D pose estimation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 20438–20447, 2022.

[43] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3D human pose estimation with spatial and temporal transformers. In *Proc. IEEE International Conference on Computer Vision*, 2021.

[44] 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.

[45] 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.