Graph Neural Networks For 3D Human Pose Estimation

Md. Tanvir Hassan

A Thesis
in
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Quality Systems Engineering) at
Concordia University
Montreal, QC, Canada

April 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Md. Tanvir Hassan**

Entitled: **Graph Neural Networks For 3D Human Pose Estimation**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Z. Patterson

_____ External Examiner
Dr. Name of External Examiner

_____ Examiner
Dr. M. Amayri

_____ Supervisor
Dr. A. Ben Hamza

Approved by _____
Dr. A. Ben Hamza, Chair
Department of Concordia Institute for Information Systems Engineering

April 5, 2023 _____
Dr. M. Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Graph Neural Networks For 3D Human Pose Estimation

Md. Tanvir Hassan

In human pose estimation methods based on graph convolutional architectures, the human skeleton is usually modeled as a graph whose nodes are body joints and edges are connections between neighboring joints. However, most of these methods tend to focus on learning relationships between body joints of the skeleton using first-order neighbors, ignoring higher-order neighbors and hence limiting their ability to exploit relationships between distant joints. In this thesis, we introduce a higher-order regular splitting graph network (RS-Net) for 2D-to-3D human pose estimation using matrix splitting in conjunction with weight and adjacency modulation. The core idea is to capture long-range dependencies between body joints using multi-hop neighborhoods and also to learn different modulation vectors for different body joints as well as a modulation matrix added to the adjacency matrix associated to the skeleton. This learnable modulation matrix helps adjust the graph structure by adding extra graph edges in an effort to learn additional connections between body joints. Instead of using a shared weight matrix for all neighboring body joints, the proposed RS-Net model applies weight unsharing before aggregating the feature vectors associated to the joints in order to capture the different relations between them. Experiments and ablations studies performed on two benchmark datasets demonstrate the effectiveness of our model, achieving superior performance over strong baselines for 3D human pose estimation.

The other contribution of this thesis consists of designing a spatio-temporal 3D human pose estimation model using multilayer perceptrons and graph neural networks. Despite the success of graph convolutional networks and their variants in 3D human pose estimation tasks, 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 issue, we propose a spatio-temporal network architecture composed of a joints-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. Extensive experiments on two benchmark datasets demonstrate the competitive performance of our model, outperforming recent state-of-the-art methods for 3D human pose estimation. In addition, we perform a runtime analysis and conduct a comprehensive ablation study to show the effect of the key components of our model.

Acknowledgments

I am deeply grateful to a number of individuals and organizations for their ongoing academic, financial, and emotional support during my Master's program. My sincere thanks go to my supervisor, Professor Abdessamad Ben Hamza, who not only is an excellent academic and mentor, but also one of the kindest and most giving people I have ever met. Professor Hamza encouraged me to pursue new ideas and challenges, provided valuable feedback and advice, and always kept me motivated and inspired. I am truly grateful for everything I have learned from him. I am also thankful to Hasib Zunair, Mahsa Mesgaran, Md Shakib Khan, and Zaedul Islam, my lab mates, who always provided great ideas and support. I have thoroughly enjoyed my experience at Concordia and am grateful to all of my instructors. Lastly, I am forever grateful to my parents, and my siblings for their unconditional love and support. They have always given me confidence, motivation, and hope, especially during difficult times. I want to especially thank my mother for her countless sacrifices and prayers.

Table of Contents

List of Figures

List of Tables

List of Acronyms

RS-Net Regular splitting graph network

MPJPE Mean per joint position error

PA-MPJPE Procrustes-aligned mean per joint position error

ISC Initial skip connection

GCNs Graph convolutional networks

MLPs Multi-layer perceptrons

FCNs Fully-connected networks

LSTM Long short-term memory

GraphWJ Graph weighted Jacobi

WJ Weighted Jacobi

RS-NetConv Regular splitting graph convolutional

HR-Net High-resolution network

CPN Cascaded pyramid network

PCK Percentage of correct keypoints

AUC Area under the curve

LN Layer normalization

BN Batch normalization

FLOPs Floating-point operations

CHAPTER 1

Introduction

In this chapter, we present the motivation behind this work, followed by the problem statement, objectives of the study, literature review, an overview of convolutional neural networks, graph convolution networks, and thesis contributions.

1.1 Framework and Motivation

The objective of 3D human pose estimation is to predict the positions of a person's joints in still images or videos. It is one of the most rapidly evolving computer vision technologies, with diverse real-world applications ranging from activity recognition and pedestrian behavior analysis [7] to sports and safety surveillance in assisted living retirement homes. In healthcare, for instance, potential benefits of human pose estimation include posture correction during exercise and rehabilitation of the limbs, thereby helping people adopt a healthy lifestyle.

Existing 3D human pose estimation methods can be broadly categorized into two main streams: single-stage [8] and two-stage approaches [9, 10]. Single-stage methods typically use a deep neural network to regress 3D keypoints from images in an end-to-end manner. On the other hand, two-stage approaches, also referred to as lifting methods, consist of two decoupled stages. In the first stage, 2D keypoints are extracted from an image using an off-the-shelf 2D pose detector such as the cascaded pyramid network [11] or the high-resolution network [12]. In the second stage, the extracted 2D keypoints are fed into a regression model to predict 3D poses [13–18]. These keypoints include the shoulders, knees, ankles, wrists, pelvis, hips, head, and others on the human skeleton. Two-stage approaches generally outperform the single-stage methods thanks,

in part, to recent advances in 2D pose detectors, particularly the high-resolution representation learning networks that learn not only semantically strong representations, but are also spatially precise [12]. For example, Martinez *et al.* [13] introduce a simple two-stage approach to 3D human pose estimation by designing a multilayer neural network with two blocks comprised of batch normalization, dropout, and a rectified linear unit activation function. This multilayer network also uses residual connections to facilitate model training and improve generalization performance. Pavllo *et al.* [19] use dilated temporal convolutions to leverage temporal correlations in 2D pose sequences for video data analysis.

1.2 Objectives

In this thesis, we propose deep learning approaches for 3D human pose estimation.

- We propose a higher-order regular splitting graph network for 3D human pose estimation using matrix splitting in conjunction with weight and adjacency modulation. 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.
- We also propose a novel spatio-temporal network architecture, which we call MLP-GraphWJ mixer, for 3D human pose estimation by incorporating multi-layer perceptrons (MLPs) to capture global information and a graph weighted Jacobi network to capture local information between adjacent joints across different channels.

1.3 Literature Review

Both graph convolutional networks and 3D human pose estimation have received a flurry of research activity over the past few years. Here, we only review the techniques most closely related to ours. Like much previous work discussed next, we approach the problem of 3D human pose estimation using a two-stage pipeline.

Graph Convolution Networks. GCNs and their variants have recently become the method of choice in graph representation learning, achieving state-of-the-art performance in numerous downstream tasks [20–23], including 3D human pose estimation [1,24,25]. However, GCNs apply graph convolutions in the one-hop neighborhood of each node, and hence fail to capture long-range relationships between body joints. This weakness can be mitigated using higher-order graph convolution filters [26] and concatenating the features of body joints from multi-hop neighborhoods

with the aim of improving model performance in 3D human pose estimation [3, 27]. To capture higher-order information in the graph, Wu *et al.* [28] also propose a simple graph convolution by removing the nonlinear activation functions between the layers of GCNs and collapsing the resulting function into a single linear transformation using the normalized adjacency matrix powers.

Transformer and MLP-based Architectures. Transformer-based models have shown promising results in 3D human pose estimation and are an active area of research [29–33]. A Transformer encodes 2D joint positions into a series of feature vectors using a self-attention mechanism, which allows the model to capture long-range dependencies between different joints and to attend to the most relevant joints for predicting the 3D joint positions. For example, Zheng *et al.* [30] introduce PoseFormer, a spatio-temporal approach for 3D human pose estimation in videos that combines spatial and temporal transformers. This approach uses two separate transformers, one for modeling spatial information and the other for modeling temporal information. The spatial transformer focuses on modeling the 2D spatial relationships between the joints of the human body, while the temporal transformer models the temporal dependencies between frames. However, Poseformer only estimates human poses from the central frame of a video, which may not provide sufficient context for accurate pose estimation in complex scenarios. While Transformers have shown great potential in 3D human pose estimation, they typically require large amounts of labeled data to train effectively and are designed to process sequential data. Also, as with any spatio-temporal method, the quality of the input video can significantly impact the accuracy of the model's pose estimations. In contrast, GCNs are specifically designed for processing graph-structured data, more efficient on sparse data, produce interpretable feature representations, and require less training data to achieve good performance.

Motivated by the good performance of the MLP-mixer model [34] in image classification tasks, Wenhao *el at.* [35] propose GraphMLP, a neural network architecture comprised of multilayer perceptrons (MLPs) and GCNs, showing competitive performance in 3D human pose estimation. GraphMLP integrates the graph structure of the human body into an MLP model, which facilitates both local and global spatial interactions. It employs a GCN block to aggregate local information between neighboring joints and a prediction head to estimate the 3D joint positions.

3D Human Pose Estimation. The basic goal of 3D human pose estimation is to predict the locations of a human body joints in images or videos. To achieve this goal, various methods have been proposed, which can learn to categorize human poses. Most of these methods can be classified into one-stage approaches that regress 3D keypoints from images using deep neural networks in an end-to-end manner [8] or two-stage approaches that employ an off-the-shelf 2D pose detector to extract 2D keypoints and then feed them into a regression model to predict 3D

poses [2–5, 14–19, 36, 37]. Fully-connected networks (FCNs) have been shown to be effective at regressing 3D poses from 2D keypoints [13]. Pavllo *et al.* [19] takes this one step further by using dilated temporal convolutions which add gaps (or dilations) between the time steps that the convolutional kernel is applied, in order to exploit the temporal correlations in 2D pose sequences, allowing FCNs to be applied to video data.

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 [4, 30, 38–41] have been developed that rely on temporal correlations to improve the robustness and accuracy of the resulting 3D poses. Hossain *et al.* [38] introduce a recurrent neural network that incorporates Long Short-Term Memory (LSTM) to take advantage of temporal correlations in the input sequence. Liu *et al.* [39] 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.* [30] utilize a Transformer-based approach that is designed to capture both the correlations between human joints and their temporal dependencies. Zeng *et al.* [41] introduce a temporal aware dynamic graph convolution where the graph updates by physical skeleton topology, and through the features of nodes. Most of the state-of-art methods tend to be computationally demanding and utilize dilated temporal convolutions to capture global dependencies, however, these methods are inherently restricted in their ability to establish temporal connectivity. Moreover, most GCN-based approaches have been 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. This weight-sharing may not be able to fully represent the diverse range of relational patterns present in a graph.

Our proposed graph neural network falls under the category of 2D-to-3D lifting. While GCNs have proven powerful at learning discriminative node representations on graph-structured data, they usually extract first-order neighborhood patterns for each joint, ignoring higher-order neighborhood information and hence limiting their ability to exploit relationships between distant joints. Moreover, GCNs share the same feature transformation for each node, hampering the efficiency of information exchange between body joints. Our work differs from existing approaches in that we use higher-order neighborhoods in combination with weight and adjacency modulation in order to not only capture long-range dependencies between body joints but also learn additional connections between body joints by adjusting the graph structure via a learnable modulation matrix. In

addition, we design a variant of the ConvNeXt block and integrate it into our model architecture with the goal of improving accuracy in human pose estimation, while maintaining the simplicity and efficiency of standard convolutional networks. Moreover, we propose 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.

1.4 Overview and Contributions

The organization of this thesis is as follows:

- Chapter 1 begins with the motivations and goals for this research, followed by the problem statement, the objective of this study, and a literature review with a brief discussion of some algorithms relevant to deep learning in 3D human pose estimation.

- In Chapter 2, we propose a higher-order regular splitting graph network for 3D human pose estimation using matrix splitting in conjunction with weight and adjacency modulation along with a new objective function for training our proposed graph network by leveraging the regularizer of the elastic net regression. In addition, we design a variant of the ConvNeXT residual block and integrate it into our graph network architecture. We demonstrate through experiments and ablation studies that our proposed model achieves state-of-the-art performance in comparison with strong baselines.

- In Chapter 3, 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 to improve accuracy and model generalization capability. In addition, We design a novel spatio-temporal network architecture, which we call MLP-GraphWJ mixer, for 3D human pose estimation by incorporating multi-layer perceptrons (MLPs) to capture global information and a graph weighted Jacobi network to capture local information between adjacent joints across different channels. Extensive experiments on two benchmark datasets demonstrate the effectiveness of our model, outperforming recent state-of-the-art methods for 3D human pose estimation.

- Chapter 4 presents a summary of the contributions of this thesis, its limitations, and outlines several directions for future research in this area of study.

CHAPTER 2

Regular Splitting Graph Network for 3D Human Pose Estimation

In this chapter, we introduce a higher-order regular splitting graph network (RS-Net) for 2D-to-3D human pose estimation using matrix splitting in conjunction with weight and adjacency modulation. The core idea is to capture long-range dependencies between body joints using multi-hop neighborhoods and also to learn different modulation vectors for different body joints as well as a modulation matrix added to the adjacency matrix associated to the skeleton. This learnable modulation matrix helps adjust the graph structure by adding extra graph edges in an effort to learn additional connections between body joints. Instead of using a shared weight matrix for all neighboring body joints, the proposed RS-Net model applies weight unsharing before aggregating the feature vectors associated to the joints in order to capture the different relations between them. Experiments and ablations studies performed on two benchmark datasets demonstrate the effectiveness of our model, achieving superior performance over recent state-of-the-art methods for 3D human pose estimation.

2.1 Introduction

Recently, graph convolutional networks (GCNs) and their variants have emerged as powerful methods for 2D-to-3D human pose estimation [1–3, 25, 27, 42] due largely to 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. For example, Zhao *et al.* [2] propose a semantic GCN archi-

tecture to capture local and global node relationships that are learned through end-to-end training, resulting in improved 3D pose estimation performance. While graph neural networks, particularly GCNs, have shown great promise in effectively tackling the 3D human pose estimation problem, they suffer, however, from a number of issues. First, GCNs focus primarily on learning relationships between body joints using first-order neighbors, ignoring higher-order neighbors; thereby limiting their ability to exploit relationships between distant joints. This challenge can be mitigated using higher-order graph neural networks [26], which have proven effective at capturing long-range dependencies between body joints [3,27]. Second, GCNs share the transformation matrix in the graph convolutional filter for all nodes, hindering the efficiency of information exchange between nodes, especially for a multi-layer network. To overcome this limitation, Liu *et al.* [25] introduce various weight unsharing mechanisms and apply different feature transformations to graph nodes before aggregating the associated features. The downside of these mechanisms is that they increase the model size by a factor equal of the number of body joints. This challenge can be alleviated by incorporating both weight and affinity modulation into the shared weight matrix and adjacency matrix, respectively [1] in order to help improve model generalization.

Another recent line of work leverages Transformer architectures, which employ a multi-head self-attention mechanism, to capture spatial and temporal information from 2D pose sequences [30, 43]. While Transformer-based architectures are able to encode long-range dependencies between body joints in the spatial and temporal domains, they often require, however, large-scale training datasets to achieve comparable performance in comparison with their convolutional networks counterparts, particularly on visual recognition tasks. This can make training and inference computationally expensive. Moreover, the attention mechanism used in Transformers involves computing an attention score between every pair of tokens in the input sequence, which can be computationally expensive, especially for longer sequences. More recently, Zhuang *et al.* [44] have proposed ConvNeXt architecture, competing favorably with Transformers in terms of accuracy and scalability, while maintaining the simplicity and efficiency of standard convolutional networks. Similar to the Transformer block and unlike the ResNet block, the ConvNeXt block is comprised of convolutional layers, followed by layer normalization and a Gaussian error linear unit activation function [44].

To address the aforementioned issues, we introduce a higher-order regular splitting graph network, dubbed RS-Net, for 3D human pose estimation by leveraging regular matrix splitting together with weight and adjacency modulation. The layer-wise propagation rule of the proposed method is inspired by the iterative solution of a sparse linear system via regular splitting. We follow the two-stage approach for 3D human pose estimation by first applying a state-of-art 2D

pose detector to obtain 2D pose predictions, followed by a lifting network for predicting the 3D pose locations from the 2D predictions. The key contributions of this work can be summarized as follows:

- We propose a higher-order regular splitting graph network for 3D human pose estimation using matrix splitting in conjunction with weight and adjacency modulation.
- We introduce a new objective function for training our proposed graph network by leveraging the regularizer of the elastic net regression.
- We design a variant of the ConvNeXT residual block and integrate it into our graph network architecture.
- We demonstrate through experiments and ablation studies that our proposed model achieves state-of-the-art performance in comparison with strong baselines.

The rest of this chapter is structured as follows. In Section 2.2, we summarize the basic notation and concepts, and then provide a problem formulation. In Section 2.3, we formulate the learning task at hand and then describe the main building blocks of the proposed graph network architecture, including a generalization to higher-order settings. In Section 2.4, we present empirical results comparing our model with state-of-the-art approaches for 3D pose estimation on a large-scale standard benchmark.

2.2 Preliminaries

Basic Notions. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. In human pose estimation, nodes correspond to body joints and edges represent connections between two body joints. We denote by $\mathbf{A} = (\mathbf{A}_{ij})$ an $N \times N$ adjacency matrix (binary or real-valued) whose (i, j)-th entry $\mathbf{A}_{ij}$ is equal to the weight of the edge between neighboring nodes i and j, and 0 otherwise. Two neighboring nodes i and j are denoted as $i \sim j$, indicating that they are connected by an edge. We denote by $\mathcal{N}_i = \{j \in \mathcal{V} : i \sim j\}$ the neighborhood of node i. We also denote by $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^\intercal$ an $N \times F$ feature matrix of node attributes, where $\mathbf{x}_i$ is an F-dimensional row vector for node i.

Spectral Graph Theory. The normalized Laplacian matrix is defined as

$$\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{I} - \hat{\mathbf{A}}, \tag{2.1}$$

where $\mathbf{D} = \mathrm{diag}(\mathbf{A1})$ is the diagonal degree matrix, $\mathbf{1}$ is an N-dimensional vector of all ones, and $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the normalized adjacency matrix. Since the normalized Laplacian matrix is symmetric positive semi-definite, it admits an eigendecomposition given by $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\intercal}$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ is an orthonormal matrix whose columns constitute an orthonormal basis of eigenvectors and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix comprised of the corresponding eigenvalues such that $0 = \lambda_1 \leq \cdots \leq \lambda_N \leq 2$ in increasing order [45]. Hence, the eigenvalues of the normalized adjacency matrix lie in the interval $[-1, 1]$, indicating that the spectral radius (i.e. the highest absolute value of all eigenvalues) $\rho(\hat{\mathbf{A}})$ is less than 1

Regular Matrix Splitting. Let $\mathbf{S}$ be an $N \times N$ matrix. The decomposition $\mathbf{S} = \mathbf{B} - \mathbf{C}$ is called a regular splitting of $\mathbf{S}$ if $\mathbf{B}$ is nonsingular and both $\mathbf{B}^{-1}$ and $\mathbf{C}$ are nonnegative matrices [46]. Using this matrix splitting, the solution of the matrix equation $\mathbf{S}\mathbf{x} = \mathbf{r}$, where $\mathbf{r}$ is a given vector, can be obtained iteratively as follows

$$\mathbf{x}^{(t+1)} = \mathbf{B}^{-1}\mathbf{C}\mathbf{x}^{(t)} + \mathbf{B}^{-1}\mathbf{r}, \tag{2.2}$$

where $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(t+1)}$ are the t-th and $(t+1)$-th iterations of $\mathbf{x}$, respectively. This iterative method is convergent if and only if the spectral radius of the iteration matrix $\mathbf{B}^{-1}\mathbf{C}$ is less than 1. It can also be shown that given a regular splitting, $\rho(\mathbf{B}^{-1}\mathbf{C}) < 1$ if and only if $\mathbf{S}$ is nonsingular and its inverse is nonnegative [46].

2.3 Proposed Method

In this section, we first start by defining the learning task at hand, including the objective function. Then, we present the main components of the proposed higher-order regular splitting graph network with weight and adjacency modulation for 3D human pose estimation.

2.3.1 Problem Statement

Let $\mathbf{D}_l = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ be a training set of 2D joint positions $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^{\top} \in \mathbb{R}^{N\times 2}$ and their associated 3D joint positions $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^{\top} \in \mathbb{R}^{N\times 3}$. An example of a 2D human pose graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which comprises $|\mathcal{V}| = 16$ nodes (joints) and $|\mathcal{E}| = 15$ edges, is illustrated in Figure 2.1. The skeletal graph encompasses 16 keypoints or joints distributed throughout the body, including shoulders, knees, ankles, wrists, pelvis, hips, head, among others. The pelvis joint is typically chosen as the root joint.

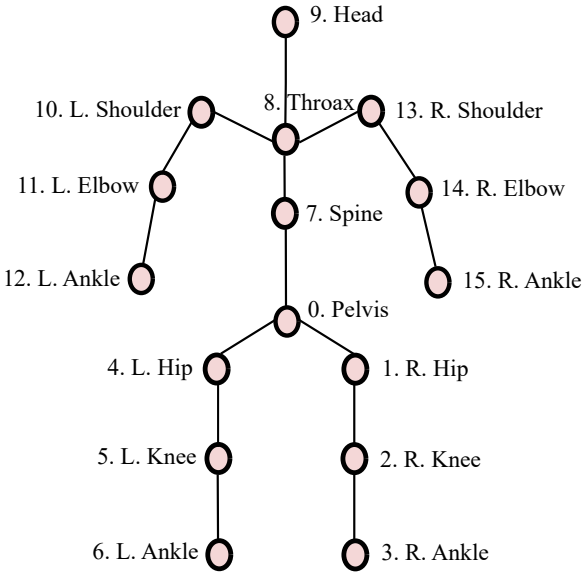

Figure 2.1: Example of a 2D human pose skeletal graph.

The goal of 3D human pose estimation is to learn the parameters w of a regression model $f : \mathbf{X} \rightarrow \mathbf{Y}$ by finding a minimizer of the following loss function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i), \tag{2.3}$$

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

$$\mathcal{L}(f(\mathbf{x}_i), \mathbf{y}_i) = (1 - \alpha) \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2 + \alpha \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_1, \tag{2.4}$$

where $\alpha \in [0, 1]$ is a weighting factor controlling the contribution of each term. It is worth pointing out that our proposed loss function (3.2) is inspired by the regularizer used in the elastic net regression technique [47], which is a hybrid of ridge regression and lasso regularization.

2.3.2 Spectral Graph Filtering

The goal of spectral graph filtering is to use polynomial or rational polynomial filters defined as functions of the graph Laplacian in order to attenuate high-frequency noise corrupting the graph signal. Since the normalized Laplacian matrix is diagonalizable, applying a spectral graph filter with transfer function h on the graph signal $\mathbf{X} \in \mathbb{R}^{N \times F}$ yields

$$\mathbf{H} = h(\mathbf{L})\mathbf{X} = \mathbf{U}h(\mathbf{\Lambda})\mathbf{U}^\intercal\mathbf{X} = \mathbf{U}\,\mathrm{diag}(h(\lambda_i))\mathbf{U}^\intercal\mathbf{X}, \tag{2.5}$$

where $\mathbf{H}$ is the filtered graph signal. However, computing all the eigenvalue/eigenvectors of the Laplacian matrix is notoriously expensive, particularly for very large graphs. To circumvent this issue, spectral graph filters are usually approximated using Chebyshev polynomials [48–50] or rational polynomials [51–53].

2.3.3 Implicit Fairing Filter

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 [27, 54]. Substituting h with h_s in Eq. (2.5), we obtain

$$\mathbf{H} = (\mathbf{I} + s\mathbf{L})^{-1}\mathbf{X}, \tag{2.6}$$

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

$$(\mathbf{I} + s\mathbf{L})\mathbf{H} = \mathbf{X}, \tag{2.7}$$

which can be efficiently solved using iterative methods [46].

2.3.4 Regular Splitting and Iterative Solution

Regular Splitting. For notational simplicity, we denote $\mathbf{L}_s = \mathbf{I} + s\mathbf{L}$, which we refer to as the implicit fairing matrix. Using regular splitting, we can split the matrix $\mathbf{L}_s$ as follows:

$$\mathbf{L}_s = (1 + s)\mathbf{I} - s\hat{\mathbf{A}} = \mathbf{B} - \mathbf{C}, \tag{2.8}$$

where

$$\mathbf{B} = (1 + s)\mathbf{I} \quad \text{and} \quad \mathbf{C} = s\hat{\mathbf{A}} = s\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}.$$

Note that $\mathbf{B}$ is a scaled identity matrix and $\mathbf{C}$ is a scaled normalized adjacency matrix. It should be noted that for both matrices, the scaling is uniform (i.e. constant scaling factors). Since $\hat{\mathbf{A}}$ is a nonnegative matrix and its spectral radius is less than 1, it follows that $\rho(s\hat{\mathbf{A}}) < s + 1$. Therefore, the implicit fairing matrix $\mathbf{L}_s$ is an M-matrix, and consequently its inverse is a nonnegative matrix. In words, an M-matrix can be defined as a matrix with positive diagonal elements, nonpositive off-diagonal elements and a nonnegative inverse.

Iterative Solution. Using regular splitting, the implicit fairing equation (3.3) can be solved iteratively as follows:

$$\begin{aligned} \mathbf{H}^{(t+1)} &= \mathbf{B}^{-1}\mathbf{C}\mathbf{H}^{(t+1)} + \mathbf{B}^{-1}\mathbf{X} \\ &= (s/(1 + s))\hat{\mathbf{A}}\mathbf{H}^{(t)} + (1/(1 + s))\mathbf{X}, \end{aligned} \tag{2.9}$$

Since the spectral radius of the normalized adjacency matrix $\hat{\mathbf{A}}$ is smaller than 1, it follows that the spectral radius of the iteration matrix $\mathbf{B}^{-1}\mathbf{C}$ is less than $s/(1+s)$, which is in turn smaller than 1. Therefore, the iterative method is convergent. This convergence property can also be demonstrated by noting that $\mathbf{L}_s$ is nonsingular and its inverse is nonnegative; thereby $\mathbf{B}^{-1}\mathbf{C} < 1$.

We can rewrite the iterative solution given by Eq. (2.9) in matrix form as follows

$$\mathbf{H}^{(t+1)} = \hat{\mathbf{A}}\mathbf{H}^{(t)}\mathbf{W}_s + \mathbf{X}\widetilde{\mathbf{W}}_s, \tag{2.10}$$

where $\mathbf{W}_s = \mathrm{diag}(s/(1+s))$ and $\widetilde{\mathbf{W}}_s = \mathrm{diag}(1/(1+s))$ are $F \times F$ diagonal matrices, each of which has equal diagonal entries, and $\mathbf{H}^{(t)}$ is the t-th iteration of $\mathbf{H}$.

Theoretical Properties. In the regular splitting $\mathbf{L}_s = \mathbf{B} - \mathbf{C}$ given by Eq. (2.8), both $\mathbf{L}_s$ and $\mathbf{B}$ are nonsingular because $\mathbf{L}_s$ is a symmetric positive definite matrix and $\mathbf{B}$ is a scaled identity matrix. Hence, the following properties hold:

- The matrices $\mathbf{B}^{-1}\mathbf{C}$ and $\mathbf{L}_s^{-1}\mathbf{C}$ commute, i.e. $\mathbf{B}^{-1}\mathbf{C}\mathbf{L}_s^{-1} = \mathbf{L}_s^{-1}\mathbf{C}\mathbf{B}^{-1}$.
- The matrices $\mathbf{B}^{-1}\mathbf{C}$ and $\mathbf{L}_s^{-1}\mathbf{C}$ have the same eigenvectors.
- If μ_i and τ_i are the eigenvalues of $\mathbf{B}^{-1}\mathbf{C}$ and $\mathbf{L}_s^{-1}\mathbf{C}$, respectively, then $\mu_i = \tau_i/(1+\tau_i)$.
- The regular splitting is convergent if and only if $\tau_i > -1/2$ for all $i = 1, \ldots, N$.
- Since both $\mathbf{B}^{-1}\mathbf{C}$ and $\mathbf{L}_s^{-1}\mathbf{C}$ are nonnegative matrices, the regular splitting is convergent and

$$\rho(\mathbf{B}^{-1}\mathbf{C}) = \frac{\rho(\mathbf{L}_s^{-1}\mathbf{C})}{1+\rho(\mathbf{L}_s^{-1}\mathbf{C})}.$$

Detailed proofs of these properties for a regular splitting of any matrix can be found in [55].

2.3.5 Regular Splitting Graph Network

In order to learn new feature representations for the input feature matrix of node attributes over multiple layers, we draw inspiration from the iterative solution given by Eq. (2.10) to define a multi-layer graph convolutional network with skip connections as follows:

$$\mathbf{H}^{(\ell+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}),\ \ell = 0, \ldots, L-1 \tag{2.11}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ and $\widetilde{\mathbf{W}}^{(\ell)} \in \mathbb{R}^{F \times F_{\ell+1}}$ are learnable weight matrices, $\sigma(\cdot)$ is an element-wise nonlinear activation function such as the Gaussian Error Linear Unit (GELU), $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N \times F_\ell}$ is the input feature matrix of the ℓ-th layer and $\mathbf{H}^{(\ell+1)} \in \mathbb{R}^{N \times F_{\ell+1}}$ is the output feature matrix.

The input of the first layer is the initial feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$. Notice that the key difference between (2.10) and (2.11) is that the latter defines a representation updating rule for propagating node features layer-wise using trainable weight matrices for learning an efficient representation of the graph, followed by an activation function to introduce non-linearity into the network in a bid to enhance its expressive power. This propagation rule is essentially comprised of feature propagation and feature transformation. The skip connections used in the proposed model allow 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.

The i-th row of the output feature matrix can be expressed as follows

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j \in \mathcal{N}_i} \hat{a}_{ij} \mathbf{h}_j^{(\ell)} \mathbf{W}^{(\ell)} + \mathbf{x}_i \widetilde{\mathbf{W}}^{(\ell)}\right), \tag{2.12}$$

where $\hat{a}_{ij}$ is the (i, j)-th entry of the normalized adjacency matrix and $\mathbf{h}_j^{(\ell)}$ is the neighboring feature vector of node i in the input feature matrix. In words, the feature vector of each node i is updated by transforming (i.e. embedding) the feature vectors of its neighboring nodes via the same projection matrix (i.e. shared weight matrix) $\mathbf{W}^{(\ell)}$, followed by aggregating the transformed feature vectors using a sum aggregator and then adding them to the transformed initial feature vector. Using a shared weight matrix is, however, suboptimal for articulated body modeling due largely to the fact the relations between different body joints are different [25]. To address this limitation, Liu *et al.* [25] introduce various weight unsharing mechanisms in an effort to capture the different relations between body joints, and hence improve human pose estimation performance. The basic idea is to use different weight matrices to transform the features vectors of the neighboring nodes before applying the sum aggregator:

$$\mathbf{h}_i^{\ell+1} = \sigma\left(\sum_{j \in \mathcal{N}_i} \hat{a}_{ij} \mathbf{h}_j^{(\ell)} \mathbf{W}_j^{(\ell)} + \mathbf{x}_i \widetilde{\mathbf{W}}^{(\ell)}\right), \tag{2.13}$$

where $\mathbf{W}_j^{(\ell)}$ is the weight matrix for feature vector $\mathbf{h}_j^{(\ell)}$ at the ℓ-th layer. This weight unsharing mechanism is referred to as pre-aggregation because weight unsharing is applied before feature vectors' aggregation. In addition, the pre-aggregation method performs the best in 3D human pose estimation [25].

Weight Modulation. While weight unsharing has proven affective at capturing the different relations between body joints, it also increases the model size by a factor equal to the number of joints. To tackle this issue, we use weight modulation [1] in lieu of weight unsharing. Weight modulation

employs a shared weight matrix, but learns a different modulation vector for each neighboring node j according to the following update rule

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\sum_{j\in\mathcal{N}_i} \hat{a}_{ij}\mathbf{h}_j^{(\ell)}\left(\mathbf{W}^{(\ell)} \odot \mathbf{m}_j^{(\ell)}\right) + \mathbf{x}_i\widetilde{\mathbf{W}}^{(\ell)}\right), \tag{2.14}$$

where $\mathbf{m}_j^{(\ell)} \in \mathbb{R}^{F_{\ell+1}}$ is a learnable modulation (row) vector for each neighboring node j and $\odot$ denotes element-wise multiplication.

Hence, the layer-wise propagation rule with weight modulation can be written in matrix form as follows

$$\mathbf{H}^{(\ell+1)} = \sigma\Big(\hat{\mathbf{A}}((\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}) \odot \mathbf{M}^{(\ell)}) + \mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}\Big), \tag{2.15}$$

where $\mathbf{M}^{(\ell)} \in \mathbb{R}^{N\times F_{\ell+1}}$ is a weight modulation matrix whose j-th row is the modulation vector $\mathbf{m}_j^{(\ell)}$.

Adjacency Modulation. Following [1], we modulate the normalized adjacency matrix in order to capture not only the relationships between neighboring nodes, but also the distant nodes (e.g. arms and legs of a human skeleton)

$$\check{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}, \tag{2.16}$$

where $\mathbf{Q} \in \mathbb{R}^{N\times N}$ is a learnable adjacency modulation matrix. Since we consider undirected graphs (e.g. the human skeleton graph), we symmetrize the adjacency modulation matrix $\mathbf{Q}$ by adding it to its transpose and dividing by 2. Therefore, the layer-wise propagation rule of the regular splitting graph network with weight and adjacency modulation is given by

$$\mathbf{H}^{(\ell+1)} = \sigma\Big(\check{\mathbf{A}}((\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}) \odot \mathbf{M}^{(\ell)}) + \mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}\Big). \tag{2.17}$$

The proposed layer-wise propagation rule is illustrated in Figure 2.2, where each block consists of a skip connection and a higher-order graph convolution with weight and adjacency modulation. The idea of skip connection is to carry over information from the initial feature matrix.

2.3.6 Higher-Order Regular Splitting Graph Network

In order to capture high-order connection information and long-range dependencies, we use k-hop neighbors to define a higher-order regular splitting network with the following layer-wise propagation rule:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\Big\|_{k=1}^{K} (\tilde{\mathbf{H}}_k^{(\ell)} + \mathbf{X}\widetilde{\mathbf{W}}_k^{(\ell)})\right) \tag{2.18}$$

where

$$\tilde{\mathbf{H}}_k^{(\ell)} = \check{\mathbf{A}}^k((\mathbf{H}^{(\ell)}\mathbf{W}_k^{(\ell)}) \odot \mathbf{M}_k^{(\ell)}) \tag{2.19}$$

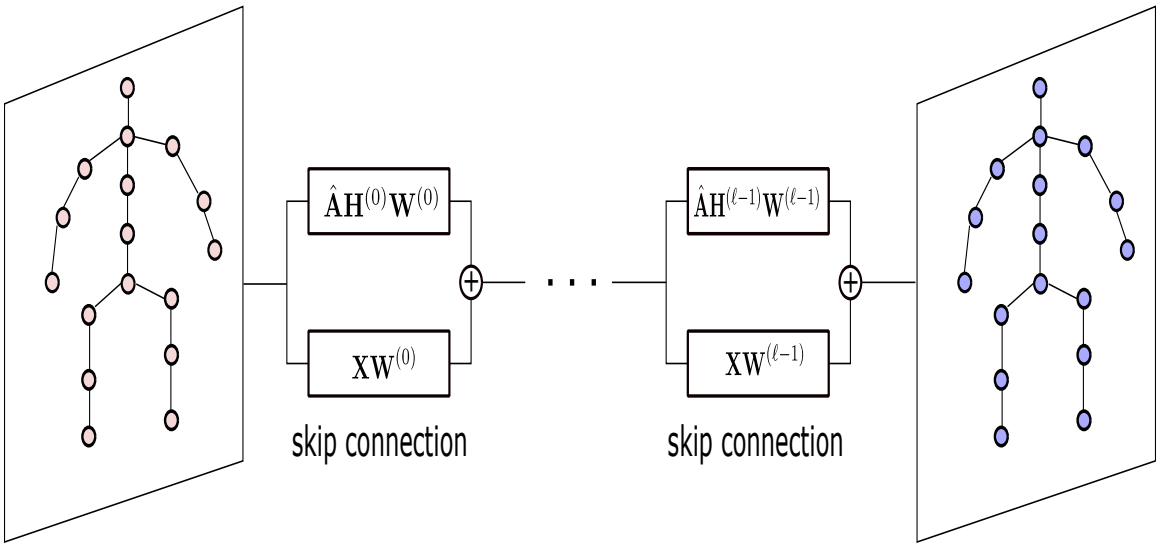

Figure 2.2: Illustration of the layer-wise propagation rule for the proposed RS-Net model. Each block is comprised of a skip connection and a higher-order graph convolution with weight and adjacency modulation.

and $\check{\mathbf{A}}^k$ is the k-th power of the normalized adjacency matrix with adjacency modulation. The learnable weight and modulation matrices $\mathbf{W}_k^{(\ell)}$ and $\mathbf{M}_k^{(\ell)}$ are associated with the k-hop neighborhood, and $\|$ denotes concatenation. For each k-hop neighborhood, the node representation is updated by aggregating information from its neighboring nodes using weight and adjacency modulation as well as carrying over information from the initial node features via skip connection. Then, high-order features are concatenated, as illustrated in Figure 3.1, followed by applying a non-linear transformation. Notice how additional edges, shown as dashed lines, are created as a result of adding a learnable modulation matrix to the normalized adjacency matrix.

Model Architecture. Figure 3.2 depicts the architecture of our proposed RS-Net model for 3D human pose estimation. The input consists of 2D keypoints, which are obtained via a 2D pose detector. We use higher-order regular splitting graph convolutional layers defined by the layer-wise propagation rule of RS-Net to capture long-range connections between body joints. Inspired by the architectural design of the ConvNeXt block [44], we adopt a residual block comprised of two higher-order regular splitting graph convolutional (RS-NetConv) layers. The first convolutional layer followed by layer normalization, while the second convolutional layer is followed by a GELU activation function, as illustrated in Figure 3.2. We also employ a non-local layer [56] before the last convolutional layer and we repeat each residual block four times.

Model Prediction. The output of the last higher-order graph convolutional layer of HOIF-Net

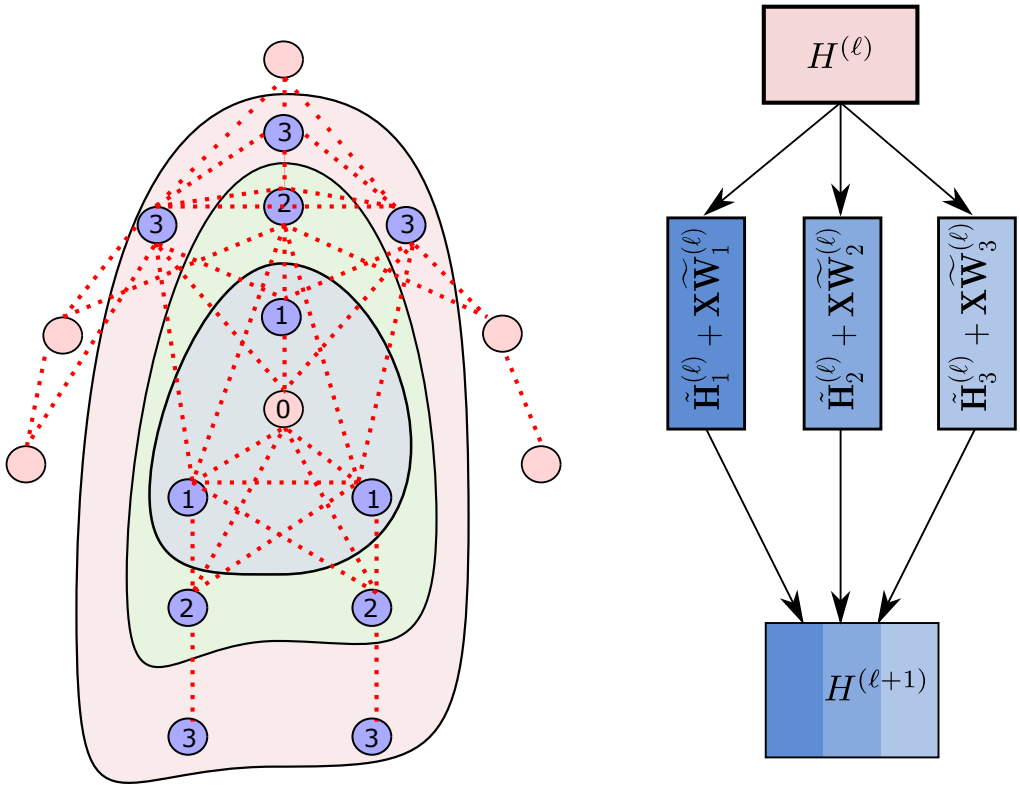

Figure 2.3: Illustration of RS-Net feature concatenation for $K = 3$ with weight and adjacency modulation. Dashed lines represent extra edges added to the human skeleton via the learnable matrix in adjacency modulation.

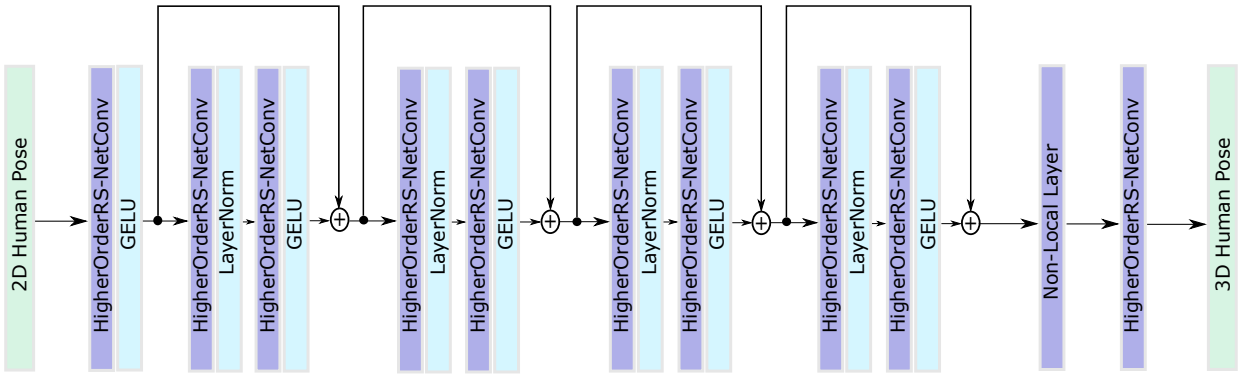

Figure 2.4: Overview of the proposed network architecture for 3D pose estimation. Our model takes 2D pose coordinates (16 or 17 joints) as input and generates 3D pose predictions (16 or 17 joints) as output. We use ten higher-order graph convolutional layers with four residual blocks. In each residual block, the first convolutional layer is followed by layer normalization, while the second convolutional layer is followed by a GELU activation function, except for the last convolutional layer which is preceded by a non-local layer.

contains the final output node embeddings, which are given by

$$\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N)^\intercal \in \mathbb{R}^{N \times 3}, \tag{2.20}$$

where $\hat{\mathbf{y}}_i$ is a three-dimensional row 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

$$\mathcal{L} = \frac{1}{N} \left[(1-\alpha) \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1 \right], \tag{2.21}$$

which is a weighted sum of the mean square and mean absolute errors 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.

2.4 Experiments

In this section, we conduct experiments on real-world datasets to evaluate the performance of the proposed model in comparison with competitive baselines for 3D human pose estimation.

2.4.1 Experimental Setup

Datasets. We evaluate our approach on two large-scale benchmark datasets: Human 3.6M and MPI-INF-3DHP. Human 3.6M is the most widely-used dataset in 3D human pose estimation [6], comprised of 3.6 million 3D human poses for 5 female and 6 male actors as well as their corresponding images captured from four synchronized cameras at 50 Hz. A total of 15 actions are performed by each actor in an indoor environment. These actions include directions, discussion, eating, greeting, talking on the phone, and so on, as shown in Figure 2.5. Following [3, 13], we apply normalization to the 2D and 3D poses before feeding the data into the model. For the MPI-INF-3DHP dataset [57], there are 8 actors performing 8 actions from 14 camera views, covering a greater diversity of poses. This dataset includes a test set of 6 subjects with confined indoor and complex outdoor scenes.

Evaluation Protocols and Metrics. We adopt different metrics to evaluate the performance of our model in comparison with strong baselines for 3D human pose estimation. For the Human 3.6M dataset, we employ two widely-used metrics: mean per joint position error (MPJPE) and Procrustes-aligned mean per joint position error (PA-MPJPE). Both metrics are measured in millimeters, and lower values indicate better performance. MPJPE, also referred to as Protocol #1, computes the average Euclidean distance between the predicted 3D joint positions and ground

Figure 2.5: Various types of actions performed by actors in the Human 3.6M dataset.

truth after the alignment of the root joint (central hip). PA-MPJPE, also known as Protocol #2, is computed after rigid alignment of the prediction with respect to the ground truth. Both protocols use 5 subjects (S1, S5, S6, S7, S8) for training and 2 subjects (S9, S11) for testing. For the MPI-INF-3DHP datset, we also employ two commonly-used evaluation metrics: Percentage of Correct Keypoints (PCK) under 150mm and the Area Under the Curve (AUC) in line with previous works [14, 17, 27, 58–60]. Higher values of PCK and AUC indicate better performance.

Baseline Methods. We evaluate the performance of our RS-Net model against various state-of-the-art pose estimation methods, including Semantic GCN [2], High-order GCN [3], Weight Unsharing [25], Compositional GCN [58], and Modulated GCN [1].

Implementation Details. Following the 2D-to-3D lifting approach [1, 4, 19, 43], we employ the high-resolution network (HR-Net) [12] as 2D detector and train/test our model using the detector's

output. We use PyTorch to implement our model, and all experiments are conducted on a single NVIDIA GeForce RTX 3070 GPU with 8G memory. We train our model for 30 epochs using AMSGrad, a variant of ADAM optimizer, which employs the maximum of past squared gradients in lieu of the exponential average to update the parameters. For 2D pose detections, we set the batch size to 512 and the filter size to 96. We also set the initial learning rate to 0.005 and the decay factor to 0.90 per 4 epochs. The weighting factor α is set to 0.1. For the 2D ground truth, we set the batch size to 128 and the filter size to 64. 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 K-hop feature concatenation, we set the value of K to 3. Following [4], we incorporate a non-local layer [56] and a pose refinement network to improve the performance. We also decouple self-connections from the modulated normalized adjacency matrix [25]. In addition, we apply horizontal flip augmentation [1, 43]. Furthermore, to prevent overfitting we add dropout with a factor of 0.2 after each graph convolutional layer.

2.4.2 Results and Analysis

Quantitative Results. In Table 3.1, we report the performance comparison results of our RS-Net model and various state-of-the-art methods for 3D human pose estimation. As can be seen, our model yields the best performance in most of the actions and also on average under both Protocol #1 and Protocol #2, indicating that our RS-Net is very competitive. This is largely attributed to the fact that RS-Net can better exploit high-order connections through multi-hop neighborhoods and also learns not only different modulation vectors for different body joints, but also additional connections between the joints. Under Protocol #1, Table 3.1 shows that RS-Net performs better than ModulatedGCN [1] on 13 out of 15 actions by a relative improvement of 4.86% on average. It also performs better than high-order GCN [3] on all actions, yielding an error reduction of approximately 15.47% on average. Moreover, our model outperforms SemGCN [2] by a relative improvement of 18.40% on average.

Under Protocol #2, Table 3.2 shows that RS-Net outperforms ModulatedGCN [1] on 11 out of 15 actions, as well as on average. Our model also performs better than high-order GCN [3] with a 11.67% error reduction on average, achieving better performance on all 15 actions, and indicating the importance of weight and adjacency modulation. Another insight from Tables 3.1 and 3.2 is that our model outperforms GCN with weight unsharing [25] on all actions under Protocol #1 and Protocol #2, while using a fewer number of learnable parameters. This indicates the usefulness of not only higher-order structural information, but also weight and adjacency modulation in boosting human pose estimation performance.

Table 2.1: 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 average errors are reported in the last column. Boldface numbers indicate the best performance, whereas the underlined numbers indicate the second best performance.

	Action															
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.* [13]	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.* [10]	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.* [14]	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.* [15]	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 [16]	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.* [17]	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.* [18]	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.* [2]	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.* [61]	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.* [62]	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.* [63]	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.* [3]	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
Quan *et al.* [27]	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 *et al.* [58]	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
Liu *et al.* [25]	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
Zou *et al.* [1]	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
Ours	**41.0**	**46.8**	**44.0**	**48.4**	**47.5**	50.7	**45.4**	**42.3**	**53.6**	65.8	**45.6**	**45.2**	48.9	39.7	**40.6**	**47.0**

Table 2.2: 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.
Zhou *et al.* [64]	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
Pavlakos *et al.* [9]	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
Martinez *et al.* [13]	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.* [10]	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.* [15]	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 [16]	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
Li *et al.* [61]	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.* [62]	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.* [63]	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.* [3]	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
Quan *et al.* [27]	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 *et al.* [58]	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
Liu *et al.* [25]	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 *et al.* [1]	35.7	38.6	36.3	**40.5**	39.2	44.5	37.0	35.4	46.4	**51.2**	40.5	**35.6**	41.7	**30.7**	33.9	39.1
Ours	**34.2**	**38.2**	**35.6**	40.8	**38.5**	**41.8**	**36.0**	**34.0**	**43.9**	56.2	**38.0**	36.3	**40.2**	31.2	**33.3**	**38.6**

In Table 3.3, we report the quantitative comparison results of RS-Net and several baselines on the MPI-INF-3DHP dataset. As can be seen, our method achieves significant improvements over the comparative methods. In particular, our model outperforms the best baseline with relative improvements of 7.94% and 15.90% in terms of the PCK and AUC metrics, respectively. Overall, our model consistently outperforms the baseline methods in terms of all evaluation metrics on both datasets, indicating its effectiveness in 3D human pose estimation.

Table 2.3: 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, while the best baselines are underlined.

Method	PCK(↑)	AUC(↑)
Chen *et al.* [60]	67.9	-
Yang *et al.* [14]	69.0	32.0
Pavlakos *et al.* [17]	71.9	35.3
Habibie *et al.* [59]	70.4	36.0
Quan *et al.* [27]	72.8	36.5
Zou *et al.* [58]	<u>79.3</u>	<u>45.9</u>
Ours	**85.6**	**53.2**

Qualitative Results. Figure 3.3 shows the qualitative results obtained by the proposed RS-Net model for various actions. As can be seen, the predictions made by our model are better than ModulatedGCN and match more closely the ground truth, indicating the effectiveness of RS-Net approach in tackling the 2D-to-3D human pose estimation problem. Notice that ModulatedGCN fails to properly predict the hand poses when there are occlusions. In comparison, our model is able to reliably predict the hand poses.

2.4.3 Ablation study

In order to verify the impact of the various components on the effectiveness of the proposed RS-Net model, we conduct ablation experiments on the Human3.6M dataset under Protocol #1 using MPJPE as evaluation metric.

Effect of Skip Connection. We start by investigating the impact of the initial skip connection on model performance. Results reported in Table 2.4 show that skip connection helps improve the performance of our model, yielding relative error reductions of .58% and .74% in terms of MPJPE and PA-MPJPE, respectively. While these improvements may not sound significant, they, however, add up because the evaluation metrics are measured in millimeters.

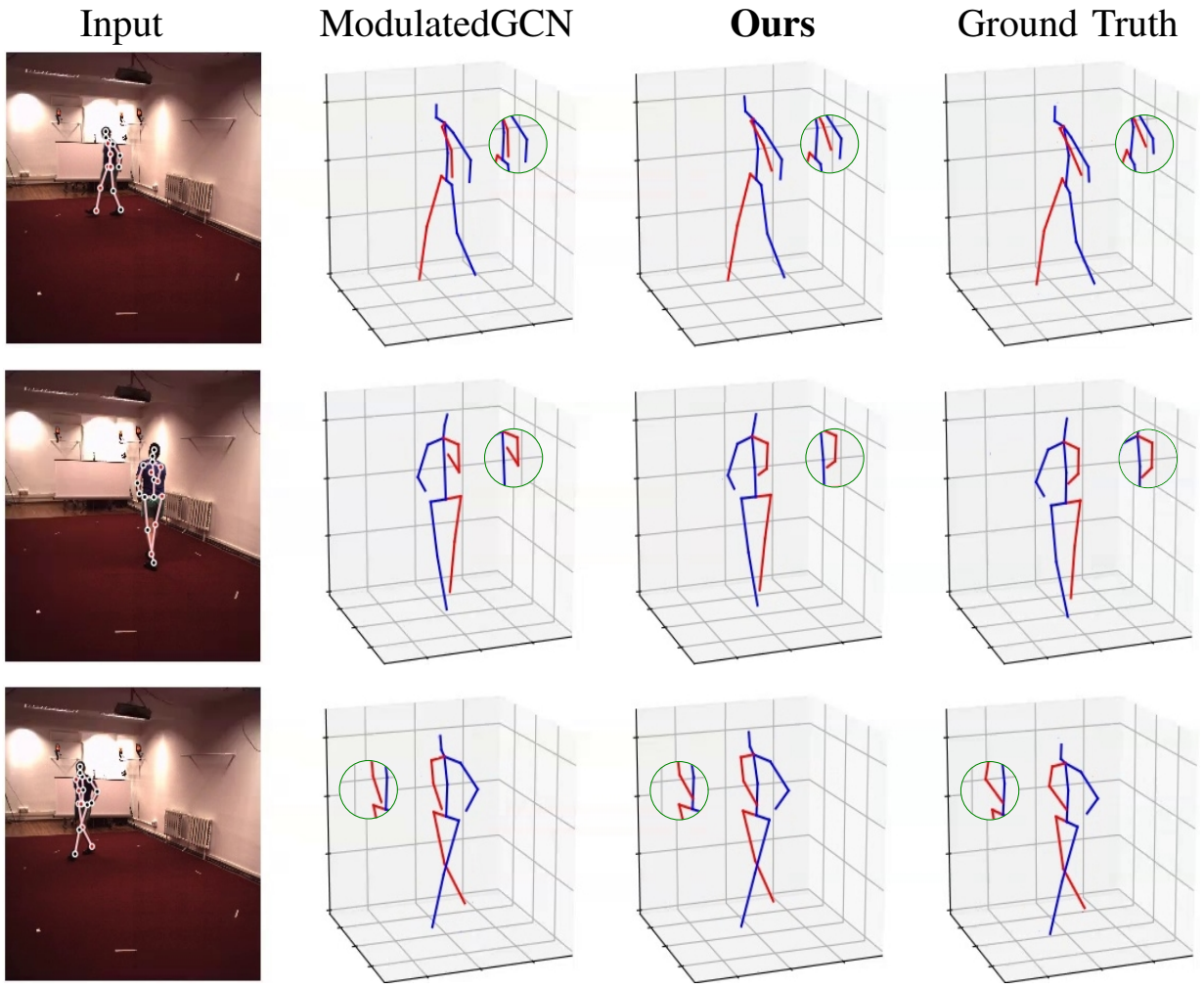

Figure 2.6: Qualitative comparison between our model and ModulatedGCN on the Human 3.6M dataset for different actions. The green circle indicates the locations where our model yields better results.

Table 2.4: Effectiveness of initial skip connection (ISC). Boldface numbers indicate the best performance.

Method	Filters	Param.	MPJPE(↓)	PA-MPJPE(↓)
w/o ISC	64	0.7M	51.7	40.4
w/ ISC	48	0.7M	**51.4**	**40.1**

Effect of Batch/Filter Size. We also investigate the effect of using different batch and filter sizes on the performance of our model. We report the results in Figure 2.7, which shows that the best performance is achieved using a batch size of 128. Similarly, filter sizes of 96 and 64 yield the best performance in terms of MPJPE and PA-MPJPE, respectively.

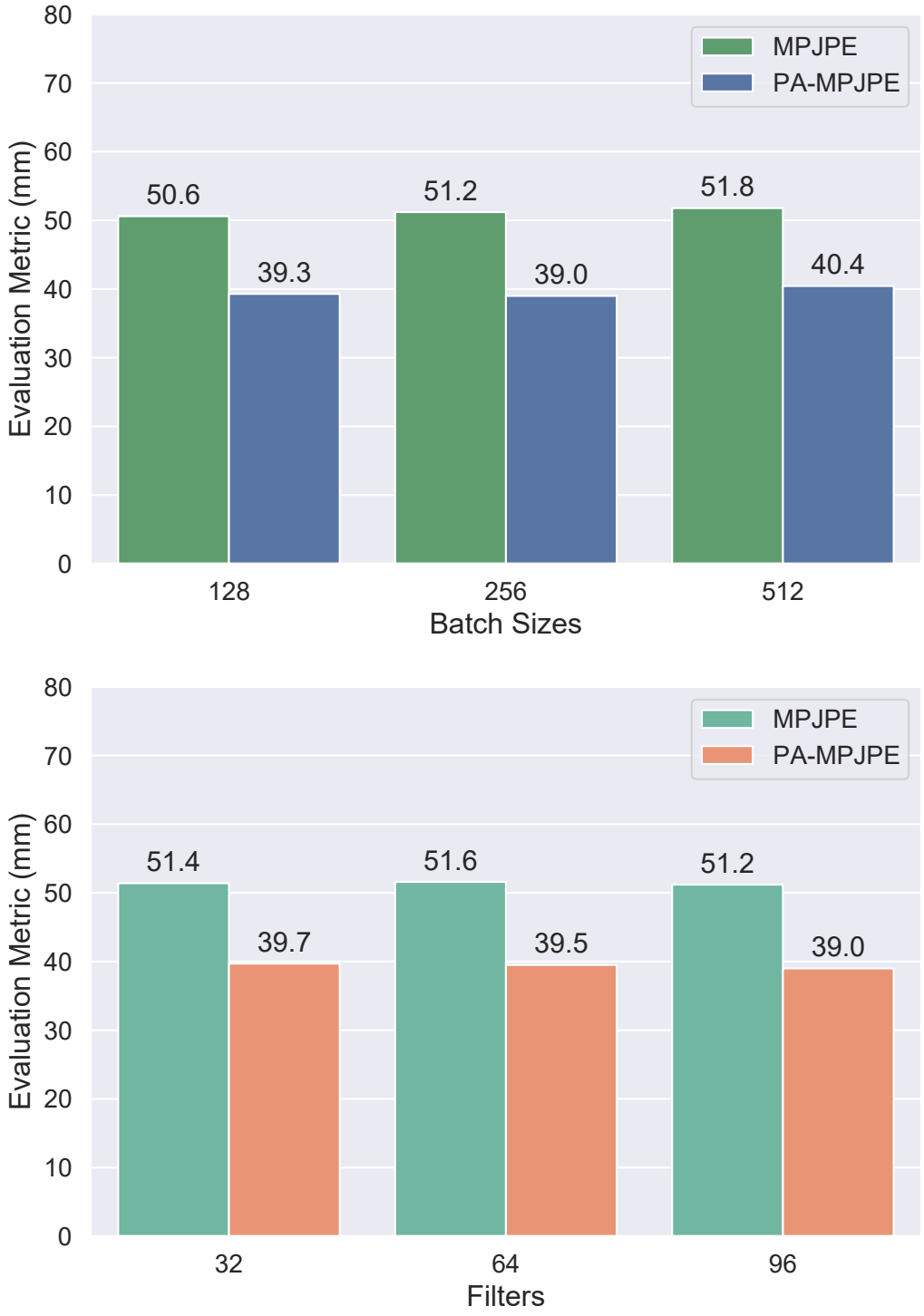

Figure 2.7: Performance of our proposed RS-Net model on the Human3.6M dataset using various batch and filter sizes.

Effect of Pose Refinement. Following [4], we use a pose refinement network, which is comprised of two fully connected layers. Pose refinement helps improve the estimation accuracy of

3D joint locations. Through experimentation, we find that using a batch size of 512 with pose refinement yields improvements around .52 mm in MPJPE and .32 mm in PA-MPJPE compared to a batch size of 128. Figure 2.8 shows the performance of our model with and without pose refinement under Protocol #1 (top) and Protocol #2 (bottom). As can be seen, lower errors are obtained when integrating pose refinement into our model, particularly under Protocol #1 for various human actions. In the case of the Sitting Down action, for example, pose refinement yields an error reduction 5.32% in terms of MPJPE.

Effect of Residual Block Design. In Table 2.5, we report the comparison results between two residual block designs: the first design employs blocks consisting of convolutional layers followed by batch normalization (BatchNorm) and a ReLU activation function, while the second design uses blocks comprised of convolutional layers followed by layer normalization (LayerNorm) and a GELU activation function, which is a smoother version of ReLU and is commonly used in Transformers based approaches. As can be seen, using the ConvNext architectural block design, we obtain relative performance gains of 1.67% and 1.28% in terms of MPJPE and PA-MPJPE, respectively.

Table 2.5: Effect of residual block design of the performance of our model. We use filters of size 96. Lower values in boldface indicate the best performance.

Method	MPJPE(↓)	PA-MPJPE(↓)
Ours w/ BatchNorm and ReLU	47.8	39.1
Ours w/ LayerNorm and GELU	**47.0**	**38.6**

We also compare our model to ModulatedGCN [1], Weight Unsharing [25], SemGCN [2], and High-order GCN [3] using ground truth keypoints, and we report the results in Table 2.6. As can be seen, our model consistently performs better than these baselines under both Protocols #1 and #2. Under Protocol #1, our RS-Net model outperforms ModulatedGCN, Weight Unsharing, High-order GCN and SemGCN by .15 mm, .55 mm, 2.24 mm and 3.50 mm, which correspond to relative error reductions of .40%, 1.45%, 5.67%, and 8.58%, respectively. Under Protocol #2, our RS-Net model performs better than ModulatedGCN, Weight Unsharing, High-order GCN, and SemGCN by .66 mm, 1.02 mm, 2 mm and 2.39 mm, which translate into relative improvements of 2.22%, 3.39%, 6.44% and 7.60%, respectively.

In order to gain further insight into the importance of pose refinement, we train our model with pose refinement on the Human3.6M dataset using 2D poses from three different 2D pose detectors, including cascaded pyramid network (CPN) [11], Detectron [65] and high-resolution network (HR-

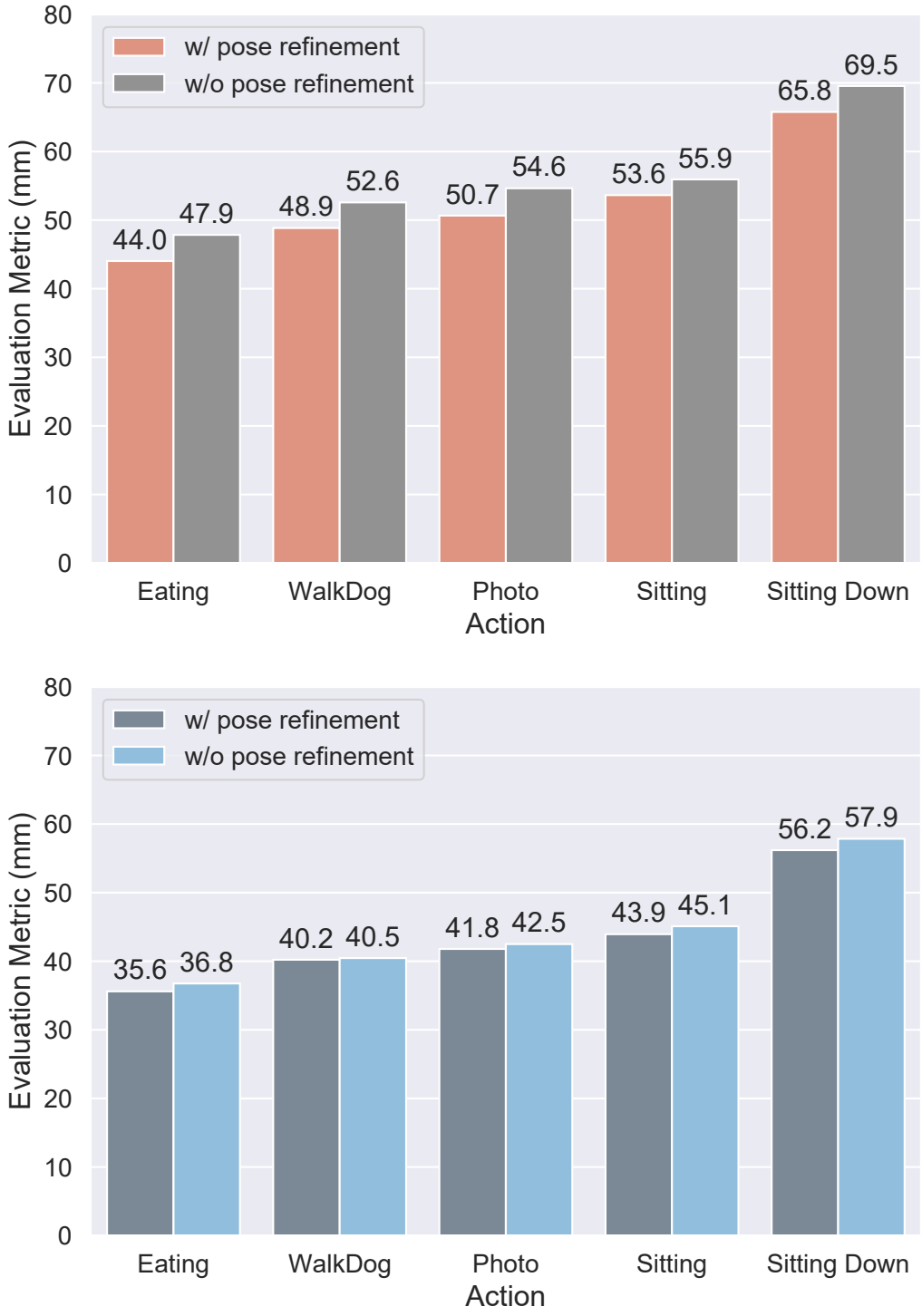

Figure 2.8: Performance of our model with and without pose refinement using MPJPE (top) and PA-MPJPE (bottom).

Net) [12]. As shown in Figure 2.9, the best performance is achieved using the HR-Net detector in terms of both MPJPE and PA-MPJPE.

Table 2.6: Performance comparison of our model and other GCN-based methods without pose refinement using ground truth keypoints. Boldface numbers indicate the best performance.

Method	Filters	Param.	MPJPE(↓)	PA-MPJPE(↓)
SemGCN [2]	128	0.43M	40.78	31.46
High-order GCN [3]	96	1.20M	39.52	31.07
Weight Unsharing [25]	128	4.22M	37.83	30.09
ModulatedGCN [1]	256	1.10M	37.43	29.73
Ours	64	1.77M	**37.28**	**29.07**

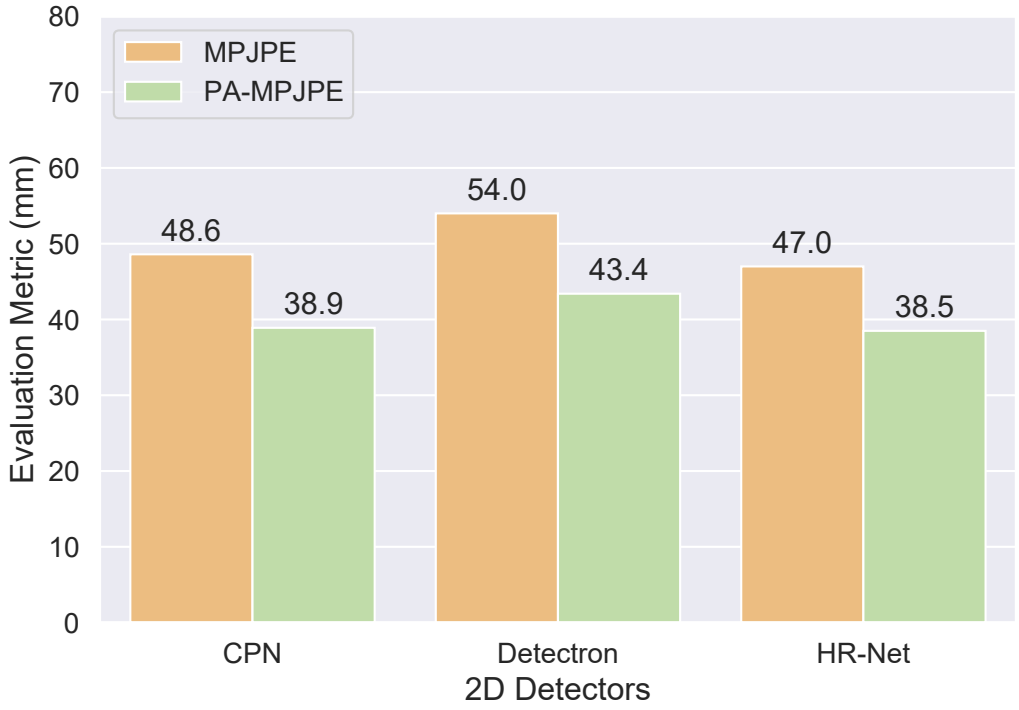

Figure 2.9: Performance of our model with pose refinement using different 2D detectors.

CHAPTER 3

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

In this chapter, we introduce a spatio-temporal network architecture composed of a joints-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 via graph filtering via implicit fairing. We leverage temporal information from the 2D pose sequences and show that temporal correlations can be modeled effectively in a straightforward manner with a minimal increase in computational cost, even for longer pose sequences. We also integrate weight modulation into the model to enable untangling of the feature transformations of distinct nodes. Moreover, we employ adjacency modulation in an effort to learn 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. In addition, we perform a runtime analysis and conduct a comprehensive ablation study to show the effect of the key components of our model.

3.1 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 monocular images or videos [13]. It is a challenging problem due

in large part to the complex and articulated nature of the human body, as well as the difficulty of estimating 3D information from 2D images [9,66], which are often adversely affected by occlusion and lighting. Its real-world applications range from activity recognition and augmented reality to gaming, robotics, and human-computer interaction. It is also used in physical therapy and rehabilitation to help track patients progress and monitor their movements during exercises [67].

Existing methods for 3D human pose estimation can broadly categorized into two main approaches: single-stage and two-stage. Single-stage approaches involve the direct prediction of 3D keypoints from images using deep neural networks, while two-stage approaches, also known as lifting methods, consist of two separate stages. The initial stage of the two-stage approaches involves using a pre-trained 2D pose detector, such as the cascaded pyramid network [11] or the high-resolution network [12], to extract 2D keypoints from the input image. In the second stage, a regression model is used to predict 3D human poses from these 2D keypoints. The superiority of two-stage methods over single-stage approaches can be attributed, in part, to the advancements in 2D pose detection, particularly the high-resolution representation learning networks that provide meaningful and spatially accurate representations [12].

Most of the existing methods for 3D human pose estimation rely solely on spatial correlations, which can make it challenging to infer a reliable 3D pose in cases of occlusion or inherent ambiguity. Cai *et al.* [4] propose local-to-global network architecture which forms a spatial-temporal graph based on the 2D pose sequence and human skeleton topology to predict 3D pose. Liu *et al.* [39] propose graph attention blocks that take the advantage of dilated temporal convolution to predict 3D pose from consecutive 2D pose sequences. Despite their promising results, these methods have their limitations. First, they use the same transformation matrix for all nodes in graph convolution, limiting information exchange, especially for multi-layer networks. To address this limitation, Liu *et al.* [25] 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. Zou *et al.* [1] propose weight and adjacency modulation to tackle this issue. Second, GCNs suffer from oversmoothing problem [68], where the model may struggle to accurately distinguish between nodes and learn meaningful representations due to repeated graph convolutions as the network depth increases. Chen *et al.* [23] solve this problem with initial residual connection and identity mapping. Third, to leverage temporal correlations, they require significant computational resources to process a larger number of input sequences such as 243 frames. To overcome this limitation, Li *et al.* [69] propose a skeleton embedding module that can effectively process a larger number of input sequences without significantly increasing the model size. Furthermore, GCNs may not be able to capture more global contextual information or long-range dependencies be-

tween nodes in the graph, which can limit their ability to learn more complex relationships and patterns in the data.

Another recent line of research employs Transformer architectures, which utilize a multi-head self-attention mechanism to capture both spatial-temporal correlations from sequences of 2D poses [30]. While these architectures can effectively capture long-range dependencies between body joints in spatio-temporal domains, the complexity of the self-attention block increases quadratically with the number of input sequences, which can make 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), which use fully-connected layers to model interactions between features, the MLP-Mixer model has been shown to be effective at modeling long-range dependencies in the input data. This model is comprised of two main components: a token-mixing layer and a channel-mixing layer. The former enables effective communication among distinct spatial locations, facilitating the extraction of global features. The latter enables communication between different feature channels, thereby facilitating the extraction of local features. This combination of token- and channel-mixing layers is intended to improve the network ability to learn complex patterns in the input data. However, such ML-based models do not adequately capture the local information due largely to the lack of prior knowledge about the human skeleton topology.

In this chapter, 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. One of the key benefits of our model is that it presents a simple and competitive alternative to existing approaches that do not use self-attention mechanisms, while outperforming previous work and retaining a small model size, as illustrated in Figure 3.1. Our model accepts a sequence of 2D joints shaped as joints $\times$ channels as input and preserves this dimensionality throughout the network. As shown in Figure 3.2, the MLP-GraphWJ mixer architecture consists of a series of layers, each of which has two components: a joints-mixing MLP layer and a GraphWJ mixing layer. The joints-mixing MLP layer facilitates communication among different joints, while the GraphWJ mixing layer enables communication among various feature channels, with the former responsible for capturing global information and the latter for capturing local information between adjacent joints across different channels. Moreover, our approach effectively merges temporal information within the feature channels, while incurring minimal computational cost in terms of

sequence length. In summary, this work makes the following key contributions:

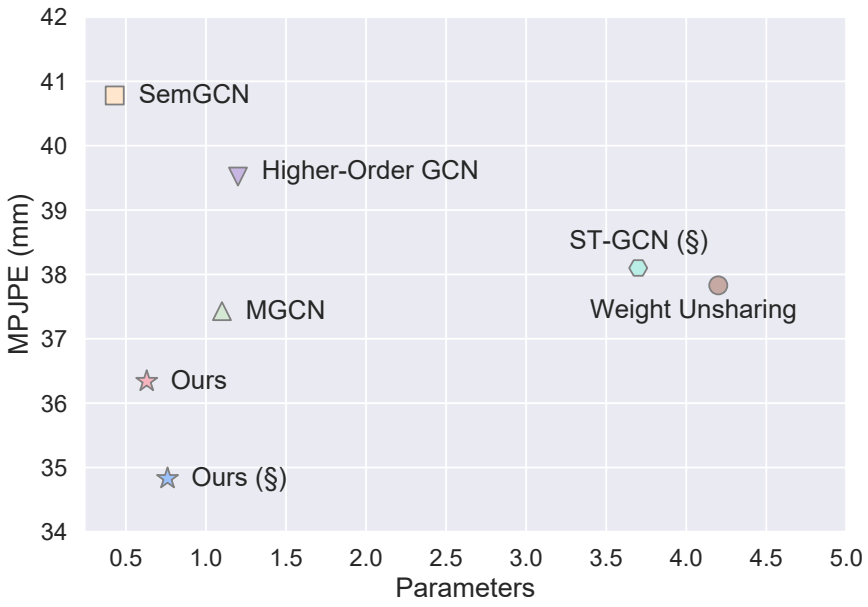

Figure 3.1: Performance and model size comparison between our proposed model and state-of-the-art approaches for 3D human pose estimation, including MGCN [1], SemGCN [2], High-Order GCN [3], ST-GCN [4], and Weight Unsharing [5]. Lower Mean Per Joint Position Error (MPJPE) values indicate better performance. Evaluation conducted on a single frame of Human3.6M [6] dataset with ground truth 2D joints as input. (§) - uses a pose refinement network.

- 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 to improve accuracy and model generalization capability.
- We design a novel spatio-temporal network architecture, which we call MLP-GraphWJ mixer, for 3D human pose estimation by incorporating multi-layer perceptrons (MLPs) to capture global information and a graph weighted Jacobi 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.

The outline of this chapter is as follows. In Section 2, we review related work on 3D pose estimation. In Section 3, we formulate the learning task at hand and then describe the main building

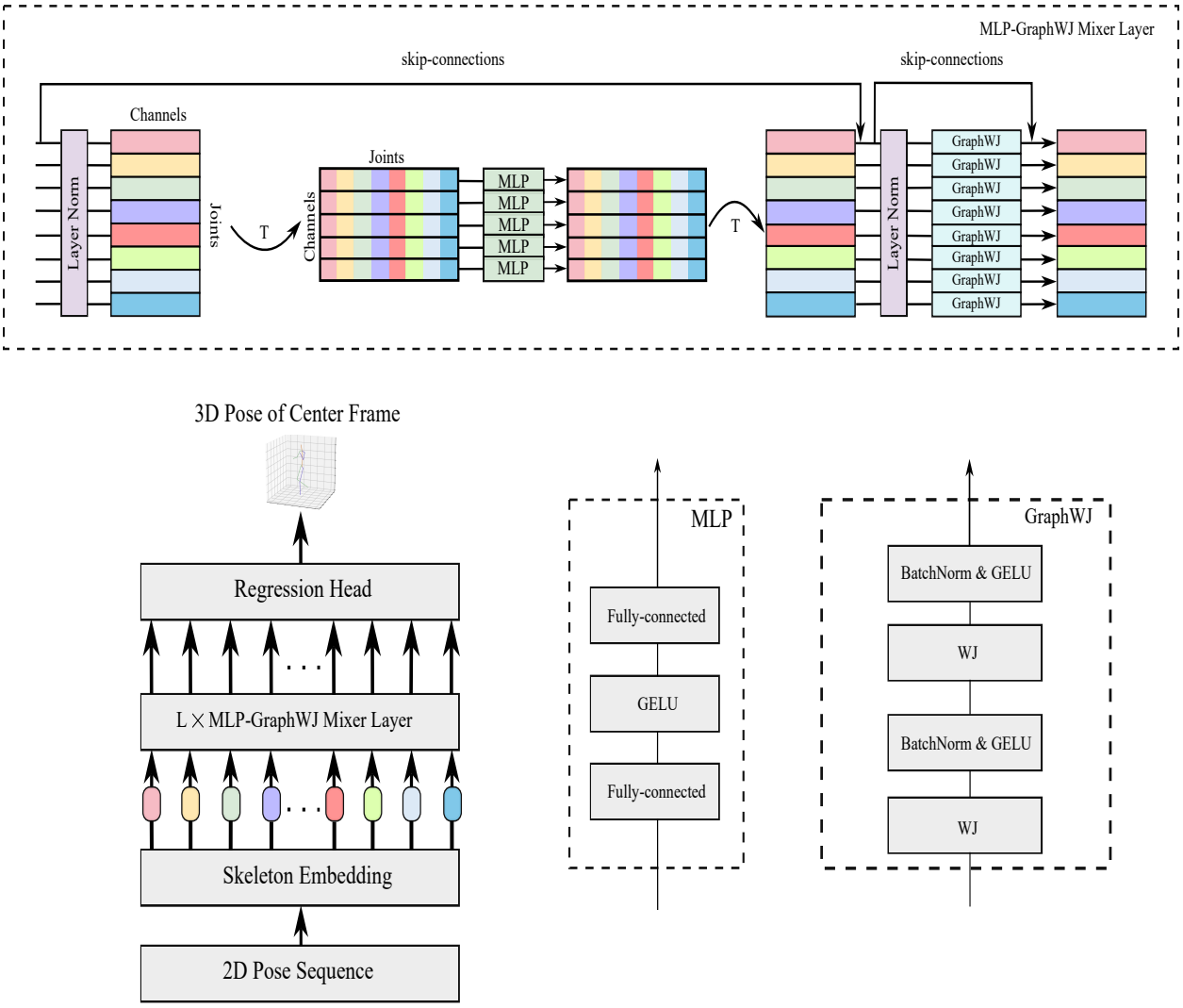

Figure 3.2: Schematic diagram of the proposed network architecture for 3D human pose estimation. The architecture is comprised of three main components: skeleton embedding, MLP-GraphWJ mixer layer, and a regression head. The MLP-GraphWJ mixer layer consists of a joints mixing MLP layer and a GraphWJ mixing layer. The architecture also includes additional components such as skip connections, dropout, layer normalization, and batch normalization. The 2D poses (16 or 17 joints) are fed as input to our model, which then produces 3D pose predictions as output.

blocks of the proposed spatio-temporal graph network architecture for 3D human pose estimation. In Section 4, we present empirical results comparing our model with state-of-the-art approaches on two standard benchmarks. Finally, we conclude in Section 5 by summarizing our key contributions and pointing out future work directions.

3.2 Proposed Method

In this section, we start by defining the learning task. Then, we present the main components of the proposed MLP-GraphWJ mixer model for 3D human pose estimation, including a weighted Jacobi (WJ) feature propagation rule obtained via graph filtering with implicit fairing.

3.2.1 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 $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^\intercal$ 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. These attributes describe the nodes characteristics or properties such as node embeddings or any other relevant information. We denote by $\mathbf{A}$ an $N \times N$ adjacency matrix whose (i, j)-th entry a_{ij} 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 $\mathbf{D} = \mathrm{diag}(\mathbf{A}\mathbf{1})$ is the diagonal degree matrix and $\mathbf{1}$ is an N-dimensional vector of all ones

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 of 3D human pose estimation is to learn the parameters $\mathbf{w}$ of a regression model $f : \mathcal{X} \to \mathcal{Y}$ by finding a minimizer of the following loss function

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} l(f(\mathbf{x}_i), \mathbf{y}_i), \tag{3.1}$$

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

$$l(f(\mathbf{x}_i), \mathbf{y}_i) = (1 - \lambda) \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2 + \lambda \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_1, \tag{3.2}$$

where $\lambda \in [0, 1]$ is a weighting factor controlling the contribution of each term. It is noteworthy that the proposed loss function draws inspiration from the regularizer employed in elastic net regression method, which is a blend of lasso and ridge regularization. Elastic net regression combines the ℓ_1 regularization of Lasso regression with the ℓ_2 regularization of Ridge regression, and is designed to address some of the limitations of Lasso and Ridge regression by allowing for both variable selection and shrinkage of coefficients.

3.2.2 Graph Filtering with Implicit Fairing

Graph filtering refers to the process of applying a filtering operation to signals defined on a graph. The aim of this filtering operation is to smooth or enhance the signal while preserving the underlying structure of the graph.

A popular graph filtering operation is graph filtering with implicit fairing, which is a technique used in computer graphics to smooth surfaces while preserving important features such as edges and boundaries [54]. In the context of graph filtering, the implicit fairing approach is applied by defining a Laplacian operator on the graph, which captures the connectivity and structure of the graph. More specifically, graph filtering with implicit fairing can be performed by solving the following sparse linear system:

$$(\mathbf{I} + s\mathbf{L})\mathbf{H} = \mathbf{X}, \tag{3.3}$$

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 iterative methods [46], such as the weighted Jacobi method, which is a variant of the Jacobi method that adds a weighting parameter to the iterative equation in order to improve the convergence speed of the method. More specifically, the weighted Jacobi iteration uses a parameter ω to compute the k-th iteration as follows

$$\begin{aligned}
\mathbf{H}^{(k+1)} &= \omega(\mathrm{diag}(\mathbf{I} + s\mathbf{L}))^{-1}\mathbf{X} \\
&\quad + \big(\mathbf{I} - \omega(\mathrm{diag}(\mathbf{I} + s\mathbf{L}))^{-1}(\mathbf{I} + s\mathbf{L})\big)\mathbf{H}^{(k)} \\
&= \alpha\omega\mathbf{X} + \big(\mathbf{I} - \alpha\omega\mathbf{I} - (1-\alpha)\omega(\mathbf{I} - \hat{\mathbf{A}})\big)\mathbf{H}^{(k)} \\
&= \mathbf{H}^{(k)} - \omega\mathbf{H}^{(k)} + (1-\alpha)\omega\hat{\mathbf{A}}\mathbf{H}^{(k)} + \alpha\omega\mathbf{X}
\end{aligned} \tag{3.4}$$

where $\alpha = 1/(1+s)$.

The weighting parameter ω can be chosen to optimize the convergence speed of the method. In general, larger values of ω will lead to faster convergence, but may also increase the risk of numerical instability or oscillation in the iteration process.

3.2.3 Graph Weighted Jacobi Network

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

$$\mathbf{H}^{(\ell+1)} = \sigma(\mathrm{WJ}\big(\mathbf{H}^{(\ell)}\big)), \quad \ell = 0, \ldots, L-1 \tag{3.5}$$

where $\sigma(\cdot)$ is a non-linear activation function such as the Gaussian Error Linear Unit (GELU) [70] and L is the number of network layers. The weighted Jacobi operation on the input feature matrix

$\mathbf{H}^{(\ell)}$ of the ℓ-th layer is given by

$$\begin{aligned} \mathrm{WJ}(\mathbf{H}^{(\ell)}) &= \mathbf{H}^{(\ell)}\mathbf{W}_1 - \boldsymbol{\Omega} \odot (\mathbf{H}^{(\ell)}\mathbf{W}_2) \\ &+ (1-\alpha)\boldsymbol{\Omega} \odot (\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}_2) \\ &+ \alpha\boldsymbol{\Omega} \odot (\mathbf{X}\mathbf{W}_3), \end{aligned} \tag{3.6}$$

where $\odot$ denotes element-wise matrix multiplication, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$ are learnable weight matrices, and $\boldsymbol{\Omega}$ is a learnable weight modulation matrix.

Adjacency Modulation. The graph structure, derived from the topology of the human skeleton, is modeled as an undirected graph. However, this graph structure has a limitation in that it cannot capture relationships between distant nodes. To tackle this issue, we use adjacency modulation [1] to modulate the normalized adjacency matrix $\hat{\mathbf{A}}$ as follows

$$\check{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}, \tag{3.7}$$

where $\mathbf{Q}$ is an $N \times N$ learnable adjacency modulation matrix. This adjacency modulation enables us to capture some meaningful relations apart from the predetermined connection between distant nodes such as the hip and ankle of the human skeleton. As we are dealing with undirected graphs, such as the human skeleton graph, the adjacency modulation matrix $\mathbf{Q}$ needs to be symmetrized. To this end, we take the sum of the adjacency modulation matrix and its transpose, and subsequently divide the result by 2.

3.2.4 MLP-Graph Weighted Jacobi Mixer Model

Model Architecture. Inspired by the MLP-Mixer [34] and its recent variants for 3D human pose estimation and human motion forecasting tasks [69, 71], the architecture of the proposed MLP-GraphWJ mixer consists of three main stages: skeleton embedding, MLP-GraphWJ mixer layer, and regression head. The overall architecture of the proposed model is illustrated in Figure 3.2, which shows that the joints-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.

Each joints-mixing block consists of two fully connected layers, followed by a layer normalization operation and a GELU activation function. The first fully connected layer takes the entire sequence of input joints and produces an intermediate representation. The second fully connected layer then takes this intermediate representation and produces the final output for the entire sequence. The

layer normalization operation is used to improve the stability and convergence of the training process, by normalizing the output of the second fully connected layer across the sequence dimension. Also, each GraphWJ mixing block consists of two WJ layers, followed by batch normalization and a GELU activation function.

1) Skeleton Embedding: Following [69], our skeleton embedding module takes the detected 2D joints as input and treats each joint as a distinct token. The next step involves learning more complex representations of each joint using a fully-connected layer, which captures the input data in a more concise and informative manner. To incorporate temporal information, the architecture adopts a video representation approach inspired by [69]. For a 2D pose sequence $\mathbf{S} \in \mathbb{R}^{T\times N\times 2}$, where T represents the number of frames and N represents the number of joints, the features of each joint for all frames are merged into $\tilde{\mathbf{S}} \in \mathbb{R}^{N\times 2T}$ and passed through a fully-connected layer. Hence, the skeleton embedding layer can be defined as follows:

$$\mathbf{X} = \tilde{\mathbf{S}}\mathbf{W}_4 \in \mathbb{R}^{N\times F}, \tag{3.8}$$

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 preserve domain-specific knowledge pertaining to human body configurations. More specifically, our MLP-GraphWJ mixer layer consists of two sub-layers: a joints-mixing MLP and a GraphWJ mixing layer. The joints-mixing MLP block allows communication between different joints, while the GraphWJ mixing layer allows communication between different channels. The joints-mixing MLP acts on the columns of the input feature matrix $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times F}$ (i.e., applied to the transpose of $\mathbf{H}^{(\ell)}$). On the other hand, the GraphWJ mixing layer acts on the rows of its input feature. The joints-mixing MLP block contains two fully-connected layers. We also add a skip connection between the input and output. Hence, the output of the joints-mixing MLP is an $N \times F$ matrix given by

$$\mathbf{U}^{(\ell+1)} = \mathbf{H}^{(\ell)} + \left(\mathbf{W}_6\sigma\left(\mathbf{W}_5\left(\mathrm{LN}\left(\mathbf{H}^{(\ell)}\right)^{\mathsf{T}}\right)\right)\right)^{\mathsf{T}}, \tag{3.9}$$

where $\mathrm{LN}(\cdot)$ is layer normalization [72], $\mathbf{W}_5 \in \mathbb{R}^{N\times F}$ and $\mathbf{W}_6 \in \mathbb{R}^{F\times N}$ are learnable weight matrices.

On the other hand, our GraphWJ mixing layer consists of two weighed Jacobi (WJ) layers. The output $\mathbf{U}^{(\ell+1)}$ of the joints-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

$$\mathbf{P}^{(\ell+1)} = \sigma\Big(\mathrm{BN}\Big(\mathrm{WJ}\Big(\mathbf{U}^{(\ell+1)}\Big)\Big)\Big) \in \mathbb{R}^{N\times R} \tag{3.10}$$

and

$$\mathbf{Q}^{(\ell+1)} = \sigma\Big(\mathrm{BN}\Big(\mathrm{WJ}\Big(\mathbf{P}^{(\ell+1)}\Big)\Big)\Big) \in \mathbb{R}^{N\times F}, \tag{3.11}$$

where $\mathrm{BN}(\cdot)$ is a batch normalization layer. 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:

$$\mathbf{Z} = \mathbf{U}^{(L)} + \mathbf{Q}^{(L)} \in \mathbb{R}^{N\times F} \tag{3.12}$$

3) Regression Head: The output $\mathbf{Z}$ of the last MLP-GraphWJ mixing layer is passed on to the regression head network which consists of a layer normalization followed by a fully-connected layer, yielding a prediction $\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N)^\intercal \in \mathbb{R}^{N\times 3}$ of estimated 3D joint positions.

Model Training. 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

$$\mathcal{L} = \frac{1}{N}\left[(1-\lambda)\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1\right]. \tag{3.13}$$

The loss function, computed over a training set comprised of N joints, is a weighted combination of the mean square and mean absolute errors between the estimated 3D joint positions $\hat{\mathbf{y}}_i$ and the ground-truth positions $\mathbf{y}_i$.

3.3 Experiments

3.3.1 Experimental Setup

Datasets. Our proposed approach is evaluated on two popular and extensively utilized datasets in the domain of 3D human pose estimation: Human 3.6M and MPI-INF-3DHP. The Human 3.6M dataset, which is regarded as the benchmark for evaluating 3D human pose estimation [6], contains 3.6 million 3D human poses for 5 female and 6 male actors, along with their corresponding images captured by four cameras organized into 15 scenarios, such as greeting, posing, walking like a dog, waiting, etc. During training, we use 5 subjects (S1, S5, S6, S7, S8), and during testing, we use 2

subjects (S9, S11) from the dataset. Prior to feeding the data into our model, we normalize both the 2D and 3D poses according to the standard normalization method used in [3, 13].

The MPI-INF-3DHP dataset [57] contains 1.3 million frames and features 8 actors performing 8 actions, providing a wider range of poses. This dataset includes a test set with 6 subjects in both indoor and complex outdoor scenes, enabling the evaluation of the model's generalization ability to unseen environments. We use the test set from this dataset to evaluate the performance of our proposed model.

Evaluation Protocols and Metrics. Our model's performance is evaluated using various metrics compared to strong baselines in the 3D human pose estimation task. For the Human 3.6M dataset, 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. Additionally, for the MPI-INF-3DHP dataset, 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 [1, 14, 17, 27, 59, 60]. Improved model performance is indicated by higher values of PCK and AUC.

Baseline Methods. We evaluate the performance of our MLP-GraphWJ mixer model against various state-of-art methods, including ST-GCN [4], Semantic GCN [2], High-Order GCN [3], Weight Unsharing [25], and MGCN [1].

Implementation Details. Our approach utilizes the 2D-to-3D lifting technique following [73, 74], in which we adopt the high-resolution network (HR-Net) [12] as our 2D detector and train/test our model with the detector's output. Our model is implemented in PyTorch, and we conduct all experiments on a single NVIDIA GeForce RTX 3070 GPU with 8G memory. The task of predicting 3D poses from 2D detections is more complex compared to doing the same from 2D ground truth, due to the added uncertainty that needs to be managed in the 2D space. Therefore, following previous work [1, 75], we use different configurations for them. To update the parameters, we employ AMSGrad, a variant of the ADAM optimizer, which uses the maximum of past squared gradients instead of the exponential average. Our model is trained for 50 epochs using this optimizer. For 2D pose detections, 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 also set the initial learning rate to 0.005 and the decay factor to 0.90 per 4 epochs. For the 2D ground truth, we set the batch size to 256, the number of layers $L = 3$, the skeleton embedding layer hidden dimension and the MLP hidden dimension $F = 128$, and the GraphWJ mixing layer hidden dimension $R = 256$ 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. We set the weighting factor λ to 0.1, α to 0.1 and the total number of input frames to 243 for both 2D detected poses and ground truth poses. Following [1, 4], we incorporate a pose refinement network to improve the performance. In the ablation study, the pose refinement network is excluded. In addition, we apply horizontal flip augmentation both in training and testing following [1, 19, 30]. Furthermore, to prevent overfitting we add dropout with a factor of 0.2 after each graph weighted Jacobi layer.

3.3.2 Results and Analysis

Quantitative Results. In Table 3.1, we report the performance comparison results of our MLP-GraphWJ mixer model and various state-of-art methods for 3D human pose estimation. Based on the results, our proposed 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 suggest that our MLP-GraphWJ mixer is highly competitive. This 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 3.1 shows that using a single frame MLP-GraphWJ mixer performs better than MGCN [1] on 14 out of 15 actions by a relative improvement of 10.73% on average. Of significance is the fact that MGCN [1] employs a non-local layer, unlike our method. Despite this difference, our model demonstrates superior performance compared to MGCN [1], highlighting the efficacy of our approach. Our method also performs better than Skeletal GCN [41] which is the recent state-of-art method based on temporal GCN, yielding an error reduction of approximately 3.92% on average.

Table 3.2 shows the results of our MLP-GraphWJ mixer model compared to various state-of-the-art methods for 3D human pose estimation when using ground truth keypoints as input. The findings indicate that our proposed model outperforms Graphmdn [76] on 12 out of 15 actions with an average error reduction of approximately 2.42% under Protocol #1. Moreover, our model shows better performance compared to MGCN [1], High-Order GCN [3], SemGCN [2], and weight unsharing [25] on average, while having a lower number of learnable parameters and inference time. These results highlight the effectiveness of our proposed method.

In Table 3.3, we report the quantitative comparison results of MLP-GraphWJ mixer using a single frame and several baselines on the MPI-INF-3DHP dataset. As can be seen, our method achieves significant improvements over the comparative methods. In particular, our model outperforms the best baseline with relative improvements of .81% and 1.30% in terms of the PCK and AUC metrics, respectively. Although we train the model using only the Human3.6M, our method

Table 3.1: Performance comparison of our model and baseline methods on Human3.6M under protocol #1& protocol #2 using the detected 2D pose as input. The average errors are reported in the last column. Boldface numbers indicate the best performance, whereas the underlined numbers indicate the second-best performance. (ϒ) - 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.* [2]	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.* [27]	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.* [25]	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.* [77]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	54.0
Zhao *et al.* [78]	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.* [75]	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 *et al.* [79]	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.* [73]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	50.2
Zou *et al.* [1]	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.* [4] (ϒ)	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.* [69]	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.* [19] (ϒ)	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.* [76]	40.0	**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.* [41] (ϒ)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.7
Zeng *et al.* [40] (ϒ)	46.6	47.1	43.9	**41.6**	45.8	49.6	46.5	40.0	53.4	61.1	46.1	**42.6**	43.1	31.5	32.6	44.8
Liu *et al.* [39] (ϒ)	43.3	46.1	40.9	44.6	46.6	54.0	44.1	42.9	55.3	57.9	45.8	43.4	47.3	**30.4**	**30.3**	44.9
Zheng *et al.* [80] (ϒ)	41.5	44.8	**39.8**	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Ours (ϒ)	**38.9**	44.5	41.4	43.7	**45.0**	**48.7**	42.8	**39.5**	54.9	67.1	**42.5**	43.1	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.* [81] (ϒ)	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.* [27]	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.* [25]	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.* [75]	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 *et al.* [79]	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.* [1]	35.7	38.6	36.3	40.5	39.2	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.* [73]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.1
Cai *et al.* [4] (ϒ)	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.* [77]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	36.7
Pavllo *et al.* [19] (ϒ)	34.1	**36.1**	34.4	**37.2**	**36.4**	42.2	34.4	33.6	**45.0**	52.5	37.4	33.8	37.8	**25.6**	27.3	36.5
Ours (ϒ)	**33.0**	36.8	**34.3**	37.5	**36.4**	**40.4**	**34.1**	**31.9**	45.4	57.0	**35.6**	**34.8**	**36.2**	26.5	**26.9**	**36.4**

outperforms others on MPI-INF-3DHP, indicating that our approach has strong generalization capabilities to unseen datasets.

Qualitative Results. Figure 3.3 depicts some visualization results of the proposed MLP-GraphWJ mixer model on the Human3.6M dataset. As depicted, the 3D predictions on various actions made by our model are superior to those of MGCN [1] and more closely match the ground truth. This implies that the MLP-GraphWJ mixer approach is more effective. It is worth noting that MGCN [1] struggles to accurately predict hand poses when there are overlapping or occlu-

Table 3.2: Performance comparison of our model and baseline methods on Human3.6M under protocol #1 using the ground truth 2D pose as input. Boldface numbers indicate the best performance, whereas the underlined numbers indicate the second-best performance. (ϒ) - uses temporal information.

	Action															
Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.* [13]	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Pavlakos *et al.* [17]	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
Hossain *et al.* [38] (ϒ)	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.	44.1
Cai *et al.* [4] (ϒ)	32.9	38.7	32.9	37.0	37.3	44.8	38.7	36.1	41.0	45.6	36.8	37.7	37.7	29.5	31.6	37.2
Liu *et al.* [25]	36.8	40.3	33.0	36.3	37.5	45.0	39.7	34.9	40.3	47.7	37.4	38.5	38.6	29.6	32.0	37.8
Pavllo *et al.* [19] (ϒ)	35.2	40.2	32.7	35.7	38.2	45.5	40.6	36.1	48.8	47.3	37.8	39.7	38.7	27.8	29.5	37.8
Zou *et al.* [1]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.4
Oikarinen *et al.* [76]	33.9	39.9	33.0	35.4	36.8	44.4	38.9	33.0	41.0	50.0	36.4	38.3	37.8	28.2	31.5	37.2
Lee *et al.* [75]	34.6	39.6	31.3	34.7	33.9	40.3	39.5	32.2	**35.4**	43.5	34.0	35.0	36.9	29.7	31.4	35.6
Zhang *et al.* [79]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	35.3
Zhao *et al.* [78]	32.0	38.0	**30.0**	34.4	34.7	43.3	**35.2**	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
Zhan *et al.* [82] (ϒ)	**31.2**	35.7	31.4	33.6	35.0	37.5	37.2	30.9	42.5	**41.3**	34.6	36.5	32.0	27.7	28.9	34.4
Ours (ϒ)	31.6	**35.6**	31.5	**31.0**	**32.1**	**35.1**	36.3	**30.1**	38.8	41.4	**32.6**	**34.6**	**31.4**	**25.5**	**25.8**	**32.9**

Table 3.3: Performance comparison of our model without pose refinement 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(↑)
Chen *et al.* [60]	67.9	-
Yang *et al.* [14]	69.0	32.0
Pavlakos *et al.* [17]	71.9	35.3
Habibie *et al.* [59]	70.4	36.0
Quan *et al.* [27]	72.8	36.5
Zeng *et al.* [40]	77.6	43.8
Zhang *et al.* [79]	81.1	49.9
Zeng *et al.* [41]	82.1	46.2
Zou *et al.* [1]	86.1	53.7
Ours	**86.8**	**54.4**

sions, whereas our model is able to predict them with reliability. Moreover, we also show the performance of our method in the in-the-wild images in Figure 3.4.

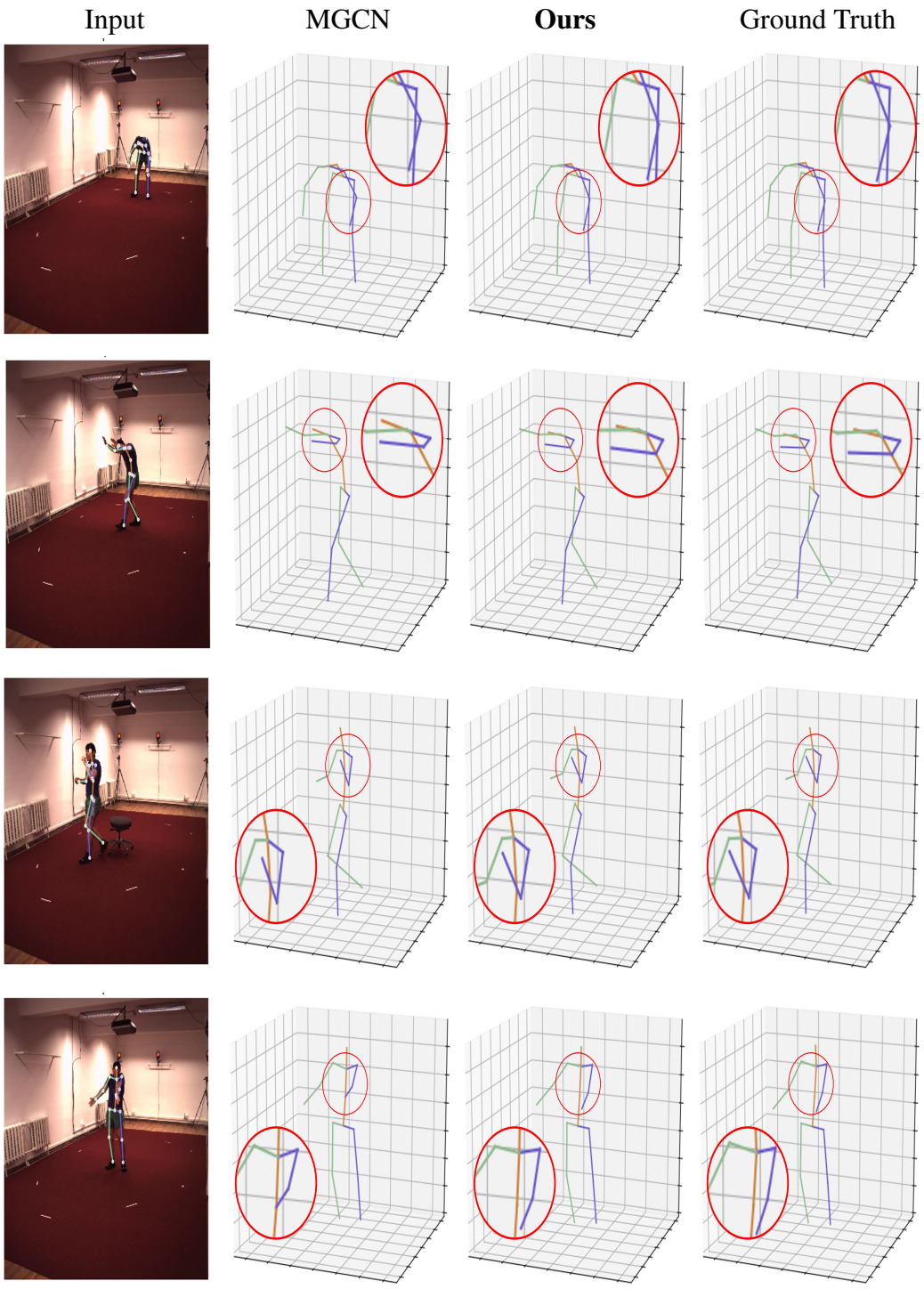

Figure 3.3: Qualitative comparison between our model and MGCN on the Human 3.6M dataset for different actions. The red circle indicates the locations where our model yields better results.

3.3.3 Ablation study

In order to verify the impact of the various components on the effectiveness of the proposed MLP-GraphWJ mixer model, we conduct ablation experiments on the Human3.6M dataset under Proto-

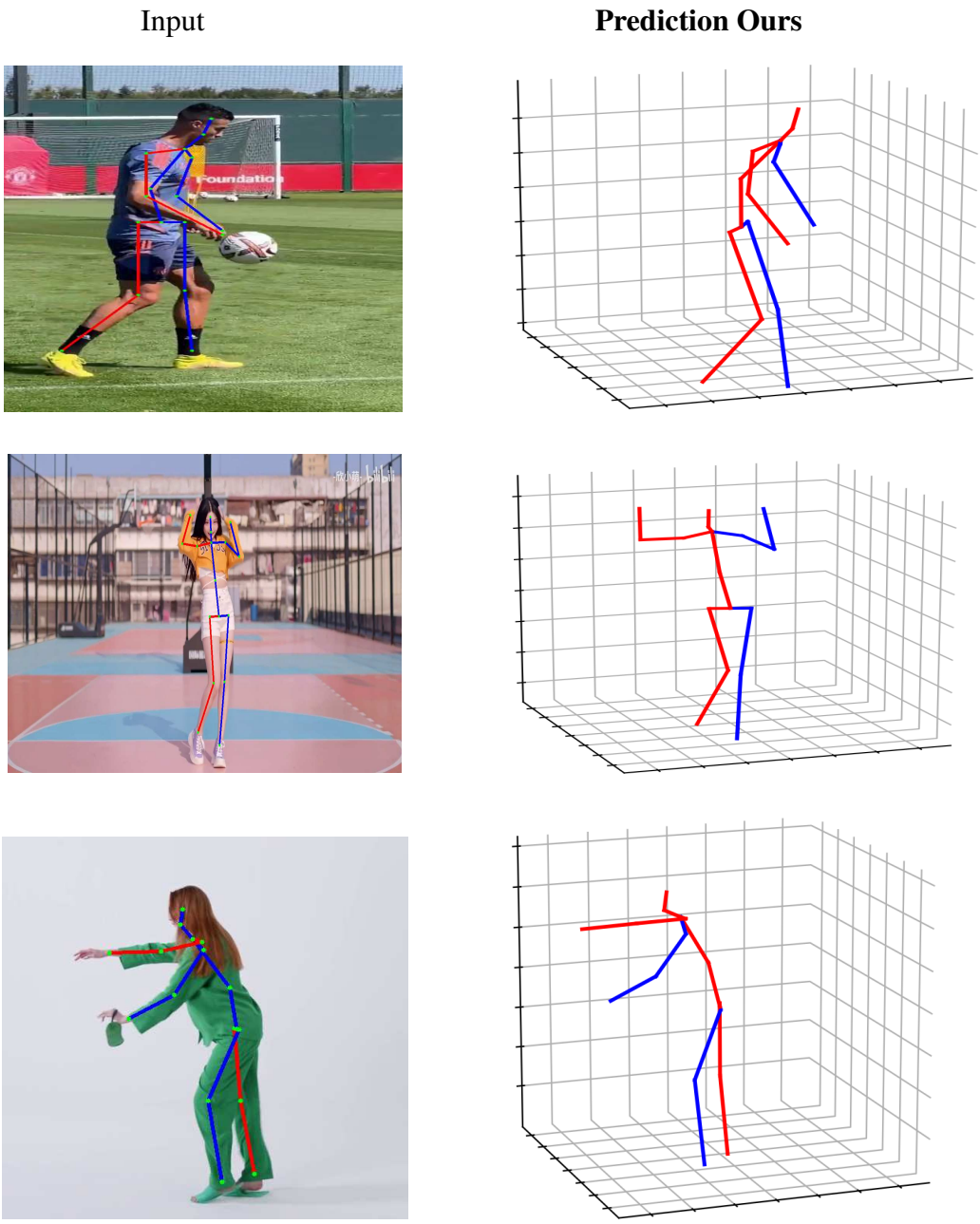

Figure 3.4: Qualitative results of our method on in-the-wild images.

col #1 using MPJPE as the evaluation metric.

Hyper-Parameter Sensitivity Analysis. We start by investigating the impact of the different hyper-parameters on model performance. Results are reported in Table 3.4. It can be observed that the expanding ratio of 2 ($F = 384$, $R = 768$) performs better than the commonly used ratio of 4 in vision Transformers and MLPs. The value of the skeleton embedding hidden dimension F affects the model ability to capture patterns. After increasing F from 128 to 384 and R from 256 to 768,

the MPJPE decreases from 47.5 mm to 45.3 mm. However, the number of trainable parameters increases from 0.65M to 5.48M. The best results are obtained using $F = 384$, and $R = 768$. Using three MLP-GraphWJ mixer layers yields the best performance, while increasing or decreasing the number of layers negatively impacts performance.

Table 3.4: Ablation study on various configurations of our approach without pose refinement on Human3.6M under protocol#1 using detected 2D pose as input. L is the number of MLP-GraphWJ mixer layers, F is the hidden dimension of skeleton embedding and joints mixing MLP and R is the hidden dimension of GraphWJ mixing layer. The number of input frames is set to $T = 81$. Boldface numbers indicate the best performance.

L	F	R	Params. (M)	MPJPE (↓)
3	128	256	0.65	47.5
3	256	256	1.28	47.7
3	256	512	2.47	47.9
3	256	1024	4.86	47.3
3	384	384	2.80	46.8
3	384	768	5.48	**45.3**
3	384	1536	10.83	46.1
1	384	768	1.87	48.3
2	384	384	3.68	46.6
4	384	768	7.29	46.6

Effect of Model Components. We also investigate the effectiveness of each component in our network architecture. The results are presented in Table 3.5, with the first row representing the performance of the baseline model (MLP-Mixer [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 clearly 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.

In order to bypass the influence of 2D pose detectors and gain further insight into the importance of our network architecture and graph propagation rule, we train our model on the Human3.6M dataset using 2D ground truth poses by maintaining the expanding ratio of 2 ($F = 128$, $R = 256$) and we report the results in Table 3.6. Our method demonstrates superior performance compared to recent state-of-art methods based on a single frame, despite utilizing fewer trainable parameters.

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 3.5: Effectiveness of each component used in our method without pose refinement on Human3.6M under protocol#1 using detected 2D poses as input. Boldface number indicates the best performance.

Skeleton Embed.	Joint-Mixing MLP	Channel-Mixing MLP	GraphWJ Mixing Layer	Vanilla GCN	Weighted Jacobi	MPJPE (↓)
✓	✓	✓	×	×	×	53.1
✓	✓	×	✓	✓	×	51.5
✓	✓	×	✓	×	✓	**51.2**

Table 3.6: Performance comparison of our model and baseline methods without pose refinement using ground-truth keypoints. Boldface numbers indicate the best performance.

Method	Filters	Params (M)	MPJPE (↓)	PA-MPJPE(↓)	Infer. Time
SemGCN [2]	128	0.43	40.78	31.46	.012s
High-Order GCN [3]	96	1.20	39.52	31.07	.013s
Weight Unsharing [25]	128	4.22	37.83	30.09	.032s
MGCN [1]	256	1.10	37.43	29.73	.008s
Ours	-	0.63	**36.34**	**28.97**	.005s

Table 3.7. Increasing the sequence length of our model leads to improved accuracy without a significant increase in the total number of parameters. This is because the number of frames only impacts the skeleton embedding layer, which does not require a large number of parameters. We also compare our method with some recent state-of-art methods in Table 3.7. Our model demonstrates a 2.40% reduction in MPJPE error and a 43.78% decrease in trainable parameters while using the same number of frames when compared to transformer-based methods such as Poseformer [30], highlighting the effectiveness of our approach.

Improvements on Hard Poses. Hard poses, which are characterized by high prediction errors, are specific to the model being used. These poses often have certain inherent characteristics, such as overlapping and self-occlusion. The way in which such cases are dealt with, however, may vary across different models [2, 40, 41]. Our proposed method aims to address this challenge by learning to capture the complex relationships between the joints via the joints mixing MLP layer and GraphWJ mixing layer. Our method yields better performance on hard poses such as Directions, Sitting Down, Photo, Purchase, etc. compared to the recent state-of-art methods [1, 2, 41] based on GCN, as shown in Table 1. In addition, we test our model on the top 5% hardest poses following [40, 41]. As shown in Figure 3.5, our model performs better than others.

Table 3.7: Comparison of our model and baselines in terms of total number of parameters, MPJPE, FLOPs. The evaluation is performed without pose refinement on Human3.6M under protocol#1 using detected 2D poses as input. Boldface numbers indicate the best performance. (§) - uses a pose refinement network.

Method	Frames (T)	Params. (M)	FLOPs (M)	MPJPE(↓)
Videopose [19]	27	8.56	17.09	48.8
ST-GCN (§) [4]	7	5.18	469.81	48.8
Poseformer [80]	9	9.58	150.0	49.9
Ray3D [80]	9	27.50	-	49.7
Ours	1	5.42	29.01	50.8
Ours	9	5.43	29.21	**48.7**

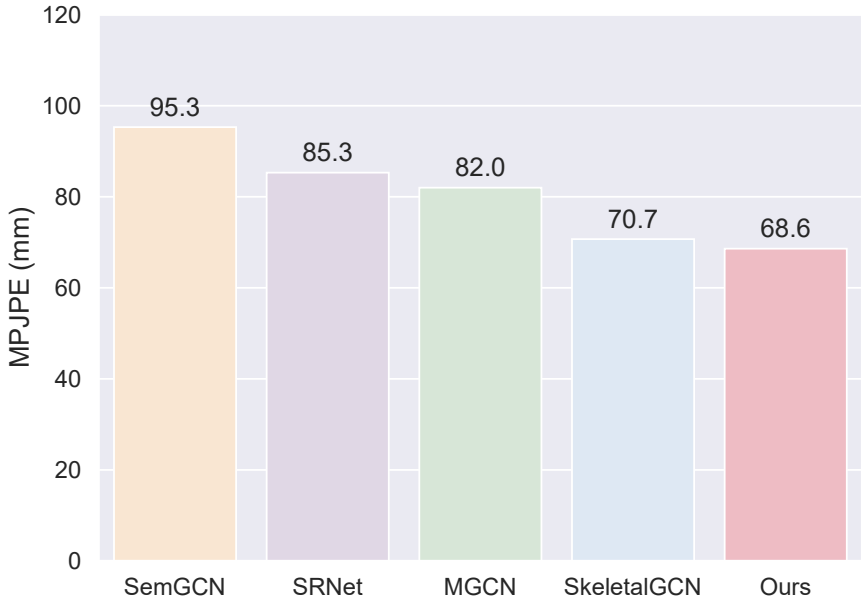

Figure 3.5: Comparison of our model and baselines on the 5% hardest poses under Protocol #1.

CHAPTER 4

Conclusions and Future Work

In this thesis, novel deep neural network architectures for 3D human pose estimation have been presented. To achieve this, a higher-order graph convolutional network was proposed, which draws inspiration from the concept of iterative solution of the implicit fairing equation using regular matrix splitting. The network takes 2D poses as input and uses a two-stage paradigm consisting of an off-the-shelf 2D pose detector and a graph convolutional network to predict the 3D poses of the human body. The proposed approach employs a residual connection-based aggregation scheme to address the oversmoothing problem and multi-hop neighborhoods to capture long-range dependencies between body joints. We also proposed a novel spatio-temporal network architecture, for 3D human pose estimation by incorporating multi-layer perceptrons (MLPs) to capture global information and a graph weighted Jacobi network to capture local information between adjacent joints across different channels. To evaluate the efficacy of both proposed approaches for 3D human pose estimation, quantitative and qualitative evaluations were performed on a large-scale dataset. The average Euclidean distance between the predicted 3D joint positions and the ground truth was used to assess the results after the alignment of the root joint. Additionally, the Procrustes-aligned mean per joint position error was also evaluated. Finally, in Section 4.1, the concluding outcomes of the associated research work in each of the previous chapters are discussed along with the contributions made. Moreover, Section 4.2 addresses the limitations of the proposed approach, while Section 4.3 provides suggestions for potential research directions related to this thesis.

4.1 Contributions of the Thesis

4.1.1 Regular Splitting Graph Network for 3D Human Pose Estimation

In Chapter 2, we introduced an effective higher-order graph network with initial skip connection for 3D human pose estimation using regular matrix splitting in conjunction with weight and adjacency modulation. The aim is to capture not only the long-range dependencies between body joints, but also the different relations between neighboring joints and distant ones. In our proposed model architecture, we designed a variant of the ConvNeXt residual block, comprised of convolutional layers, followed by layer normalization and a GELU activation function. Experimental results on two standard benchmark datasets demonstrate that our model can outperform qualitatively and quantitatively several recent state-of-the-art methods for 3D human pose estimation.

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

In Chapter 3, 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. We introduced a weighted Jacobi feature propagation rule obtained via graph filtering via implicit fairing, and we integrated both weight and adjacency modulation into the model. We also showed that leveraging temporal information for input sequences of larger lengths can be achieved with only a slight increase in computational cost. Our proposed method for 3D human pose estimation outperforms recent state-of-the-art techniques on two widely-used benchmark datasets, as demonstrated by our experimental results. Moreover, our approach achieves this improved performance while employing a model with a smaller parameter count.

4.2 Limitations

Although the proposed methods show improvements in the robustness and accuracy of 3D human pose estimation tasks, they also have certain limitations. Despite its capability to capture long-range dependencies between body joints as well as different relations between neighboring joints and distant ones, the proposed HigherOrderRS-Net framework has a relatively large number of trainable parameters compared to state-of-the-art methods. In addition, our Spatio-Temporal MLP-Graph relies on MLP to capture global information, which can be computationally expensive, especially for large datasets or deep networks, making it challenging to scale the model to real-

time applications where low latency is crucial. Furthermore, both methods are not end-to-end solutions for regressing 3D keypoints from images or videos, as they consist of two distinct stages that are typically decoupled. To address this issue, one potential solution is to leverage extensively trained, deep models.

4.3 Future Work

Several interesting research directions, motivated by this thesis, are discussed below:

4.3.1 RS-Net for Exploiting Temporal Information

In order to further improve the accuracy of 3D pose estimation, we are planning to incorporate temporal information into our model. This will be achieved by constructing a spatiotemporal graph on skeleton sequences, which will enable us to capture both spatial and temporal relationships between body joints. By incorporating this temporal information, we hope to make our model more robust to variations in the input data, such as changes in body position and movement over time. This will enable us to better estimate the 3D poses of human subjects, even in situations where the input data is noisy or incomplete. Additionally, by leveraging both spatial and temporal relationships between body joints, we aim to improve the generalization performance of our model, enabling it to perform well on a wide range of 3D pose estimation tasks in diverse settings. Overall, we believe that the incorporation of temporal information will be a key step in advancing the state-of-the-art in 3D pose estimation

4.3.2 MLP-Graph with Multi-hop Neighbors

We aim to develop a method that takes into account the high-order connectivity between joints. To achieve this, we plan to aggregate information from multi-hop neighbors, which will allow us to capture more complex relationships between body joints. By incorporating high-order connectivity into our model, we hope to improve the accuracy of 3D pose estimation and enable it to perform better in challenging situations such as overlapping or occlusions. Additionally, we believe that our approach may have broader applications beyond 3D pose estimation, and we are interested in exploring its potential in other downstream computer vision and learning tasks. For example, our method may be useful for tasks such as action recognition, human-object interaction detection, and scene understanding, where capturing local and global relationships between objects and entities is crucial for achieving high performance. By applying our method to these tasks, we hope to gain

a better understanding of its strengths and limitations and to identify potential avenues for future research.

References

[1] Z. Zou and W. Tang, "Modulated graph convolutional network for 3D human pose estimation," in *Proc. International Conference on Computer Vision*, pp. 11477–11487, 2021.

[2] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3D human pose regression," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 3425–3435, 2019.

[3] Z. Zou, K. Liu, L. Wang, and W. Tang, "High-order graph convolutional networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2020.

[4] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 2272–2281, 2019.

[5] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang, "Comprehensive study of weight sharing in graph networks for 3D human pose estimation," in *Proc. European Conference on Computer Vision*, 2020.

[6] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, 2013.

[7] Y. Zhao, Z. Yuan, and B. Chen, "Accurate pedestrian detection by human pose regression," *IEEE Transactions on Image Processing*, vol. 29, pp. 1591–1605, 2020.

[8] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," in *Proc. Asian Conference on Computer Vision*, pp. 332–347, 2014.

[9] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 7025–7034, 2017.

[10] X. Sun, J. Shang, S. Liang, and Y. Wei, "Compositional human pose regression," in *Proc. International Conference on Computer Vision*, pp. 2602–2611, 2017.

[11] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 7103–7112, 2018.

[12] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2019.

[13] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *Proc. International Conference on Computer Vision*, pp. 2640–2649, 2017.

[14] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3D human pose estimation in the wild by adversarial learning," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 5255–5264, 2018.

[15] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3D pose estimation," in *Proc. AAAI Conference on Artificial Intelligence*, 2018.

[16] M. Rayat Imtiaz Hossain and J. J. Little, "Exploiting temporal information for 3D human pose estimation," in *Proc. European Conference on Computer Vision*, pp. 68–84, 2018.

[17] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3D human pose estimation," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 7307–7316, 2018.

[18] S. Sharma, P. T. Varigonda, P. Bindal, A. Sharma, and A. Jain, "Monocular 3D human pose estimation by generation and ordinal ranking," in *Proc. International Conference on Computer Vision*, pp. 2325–2334, 2019.

[19] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 7753–7762, 2019.

[20] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 29, pp. 3844–3852, 2016.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[23] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and deep graph convolutional networks," in *Proc. International Conference on Machine Learning*, pp. 1725–1735, 2020.

[24] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3D human pose regression," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2019.

[25] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang, "A comprehensive study of weight sharing in graph networks for 3D human pose estimation," in *Proc. European Conference on Computer Vision*, pp. 318–334, Springer, 2020.

[26] S. Abu-El-Haija, B. Perozzi, A. Kapoor, N. Alipourfard, K. Lerman, H. Harutyunyan, G. Ver Steeg, and A. Galstyan, "MixHop: Higher-order graph convolutional architectures via sparsified neighborhood mixing," in *Proc. International Conference on Machine Learning*, pp. 21–29, 2019.

[27] J. Quan and A. B. Hamza, "Higher-order implicit fairing networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2021.

[28] F. Wu, T. Zhang, A. de Souza Jr., C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. International Conference on Machine Learning*, 2019.

[29] W. Zhao, Y. Tian, Q. Ye, J. Jiao, and W. Wang, "GraFormer: Graph convolution transformer for 3D pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20438–20447, 2021.

[30] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3D human pose estimation with spatial and temporal transformers," in *Proc. International Conference on Computer Vision*, 2021.

[31] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan, "MixSTE: Seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20438–20447, 2022.

[32] J. Zhang, Y. Chen, and Z. Tu, "Uncertainty-aware 3D human pose estimation from monocular video," in *Proc. ACM International Conference on Multimedia*, pp. 5102–5113, 2022.

[33] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021.

[34] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, "MLP-Mixer: An all-MLP architecture for vision," in *Advances in neural information processing systems*, pp. 24261–24272, 2021.

[35] W. Li, H. Liu, T. Guo, H. Tang, and R. Ding, "GraphMLP: A graph MLP-like architecture for 3D human pose estimation," *arXiv preprint arXiv:2206.06420*, 2022.

[36] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3D hand shape and pose estimation from a single RGB image," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 10833–10842, 2019.

[37] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3D human pose estimation," in *Proc. International Conference on Computer Vision*, pp. 2262–2271, 2019.

[38] M. R. I. Hossain and J. J. Little, "Exploiting temporal information for 3D human pose estimation," in *Proc. European Conference on Computer Vision*, pp. 68–84, 2018.

[39] J. Liu, J. Rojas, Y. Li, Z. Liang, Y. Guan, N. Xi, and H. Zhu, "A graph attention spatio-temporal convolutional network for 3D human pose estimation in video," in *Proc. IEEE International Conference on Robotics and Automation*, pp. 3374–3380, 2021.

[40] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, and S. C.-F. Lin, "SRNet: Improving generalization in 3D human pose estimation with a split-and-recombine approach," in *Proc. European Conference on Computer Vision*, 2020.

[41] A. Zeng, X. Sun, L. Yang, N. Zhao, M. Liu, and Q. Xu, "Learning skeletal graph neural networks for hard 3D pose estimation," in *Proc. IEEE International Conference on Computer Vision*, pp. 11436–11445, 2021.

[42] G. Hua, H. Liu, W. Li, Q. Zhang, R. Ding, and X. Xu, "Weakly-supervised 3D human pose estimation with cross-view u-shaped graph convolutional network," *IEEE Transactions on Multimedia*, 2022.

[43] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool, "MHFormer: Multi-hypothesis transformer for 3D human pose estimation," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2022.

[44] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2022.

[45] F. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997.

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

[47] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B*, vol. 60, no. 1, pp. 301–320, 2005.

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

[49] D. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.

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

[51] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "CayleyNets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2018.

[52] F. M. Bianchi, D. Grattarola, C. Alippi, and L. Livi, "Graph neural networks with convolutional ARMA filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 3496–3507, 2022.

[53] A. Wijesinghe and Q. Wang, "DFNets: Spectral CNNs for graphs with feedback-looped filters," in *Advances in Neural Information Processing Systems*, 2019.

[54] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. ACM SIGGRAPH*, pp. 317–324, 1999.

[55] Z. Woźnicki, "Matrix splitting principles," *International Journal of Mathematics and Mathematical Sciences*, vol. 28, pp. 251–284, 2001.

[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

[57] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D human pose estimation in the wild using improved CNN supervision," in *Proc. International Conference on 3D Vision*, 2017.

[58] Z. Zou, T. Liu, D. Wu, and W. Tang, "Compositional graph convolutional networks for 3D human pose estimation," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 1–8, 2021.

[59] I. Habibie, W. Xu, D. Mehta, G. Pons-Moll, and C. Theobalt, "In the wild human pose estimation using explicit 2D features and intermediate 3D representations," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 10905–10914, 2019.

[60] C. Li and G. H. Lee, "Generating multiple hypotheses for 3D human pose estimation with mixture density network," in *Proc. Conference on Computer Vision and Pattern Recognition*, pp. 9887–9895, 2019.

[61] C. Li and G. H. Lee, "Weakly supervised generative network for multiple 3D human pose hypotheses," in *Proc. British Machine Vision Conference*, 2020.

[62] S. Banik, A. M. Gracia, and A. Knoll, "3D human pose regression using graph convolutional network," in *Proc. IEEE International Conference on Image Processing*, 2020.

[63] Y. Xu, W. Wang, T. Liu, X. Liu, J. Xie, and S.-C. Zhu, "Monocular 3D pose estimation via pose grammar and data augmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[64] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3D human pose estimation in the wild: a weakly-supervised approach," in *Proc. International Conference on Computer Vision*, pp. 398–407, 2017.

[65] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." https://github.com/facebookresearch/detectron2, 2019.

[66] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, "Robust estimation of 3D human poses from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4321–4328, 2014.

[67] Y. Li, C. Wang, Y. Cao, B. Liu, J. Tan, and Y. Luo, "Human pose estimation based in-home lower body rehabilitation system," in *Proc. IEEE International Joint Conference on Neural Networks*, 2020.

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

[69] W. Li, H. Liu, T. Guo, H. Tang, and R. Ding, "GraphMLP: A graph MLP-like architecture for 3D human pose estimation," *arXiv preprint arXiv:2206.06420*, 2022.

[70] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[71] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer, and V. Belagiannis, "MotionMixer: MLP-based 3D human body pose forecasting," in *Proc. International Joint Conference on Artificial Intelligence*, 2022.

[72] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[73] K. Gong, J. Zhang, and J. Feng, "Poseaug: A differentiable pose augmentation framework for 3D human pose estimation," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2021.

[74] K. Gong, B. Li, J. Zhang, T. Wang, J. Huang, M. B. Mi, J. Feng, and X. Wang, "PoseTriplet: co-evolving 3D human pose estimation, imitation, and hallucination under self-supervision," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11017–11027, 2022.

[75] I. G. K. Jae Yung Lee, "Multi-hop modulated graph convolutional networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2022.

[76] T. Oikarinen, D. Hannah, and S. Kazerounian, "GraphMDN: Leveraging graph structure and deep learning to solve inverse problems," in *Proc. IEEE International Joint Conference on Neural Networks*, pp. 1–9, 2021.

[77] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1954–1963, 2021.

[78] W. Zhao, W. Wang, and Y. Tian, "GraFormer: Graph-oriented transformer for 3D pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 20438–20447, 2022.

[79] Z. Zhang, "Group graph convolutional networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2022.

[80] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, "3D human pose estimation with spatial and temporal transformers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11656–11665, 2021.

[81] K. Lee, I. Lee, and S. Lee, "Propagating lstm: 3d pose estimation based on joint interdependency," in *Proc. European Conference on Computer Vision*, pp. 123–141, 2018.

[82] Y. Zhan, F. Li, R. Weng, and W. Choi, "Ray3D: ray-based 3D human pose estimation for monocular absolute 3D localization," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13116–13125, 2022.