Graph Representation Learning for 3D Human Pose Estimation

Zaedul Islam

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

August 2023

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Zaedul Islam

Entitled: Graph Representation Learning 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 the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. N. Bouguila — Chair

Dr. N. Bouguila — Examiner

Dr. M. Amayri — Examiner

Dr. A. Ben Hamza — Thesis Supervisor(s)

_____ Thesis Supervisor(s)

Approved by _____

Dr. Jun Yan — Chair of Department or Graduate Program Director

Dr. M. Debbabi — Dean of Faculty of Engineering and Computer Science

Abstract

Graph Representation Learning for 3D Human Pose Estimation

Zaedul Islam

Graph convolutional networks (GCNs) have proven to be an effective approach for 3D human pose estimation. By naturally modeling the skeleton structure of the human body as a graph, GCNs are able to capture the spatial relationships between joints and learn an efficient representation of the underlying pose. However, most GCN-based methods use a shared weight matrix, making it challenging to accurately capture the different and complex relationships between joints. In this thesis, we introduce an iterative graph filtering framework for 3D human pose estimation, which aims to predict the 3D joint positions given a set of 2D joint locations in images. Our approach builds upon the idea of iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method. Motivated by this iterative solution, we design a Gauss-Seidel network architecture, which makes use of weight and adjacency modulation, skip connection, and a pure convolutional block with layer normalization. Adjacency modulation facilitates the learning of edges that go beyond the inherent connections of body joints, resulting in an adjusted graph structure that reflects the human skeleton, while skip connections help maintain crucial information from the input layer's initial features as the network depth increases. Our experimental results demonstrate that our approach outperforms the baseline methods on standard benchmark datasets.

This thesis makes another significant contribution by designing a spatio-temporal 3D human pose estimation model. Accurate 3D human pose estimation is a challenging task due to occlusion and depth ambiguity. To address these issues, we introduce a novel approach called Multi-hop Graph Transformer Network, which combines the strengths of multi-head self-attention and multi-hop graph convolutional networks with disentangled neighborhoods to capture spatio-temporal dependencies and handle long-range interactions. The proposed network architecture consists of two main blocks: a graph attention block composed of stacked layers of multi-head self-attention and graph convolution with learnable adjacency matrix, and a multi-hop graph convolutional block comprised of multi-hop convolutional and dilated convolutional layers. Extensive experiments demonstrate the effectiveness and generalization ability of our model, achieving state-of-the-art performance on benchmark datasets while maintaining a compact model size.

Acknowledgments

I am immensely grateful to several individuals and organizations who have provided unwavering academic, financial, and emotional support during my Master's program. My heartfelt appreciation goes to Professor Abdessamad Ben Hamza, my exceptional supervisor, whose academic prowess and mentoring have been complemented by his extraordinary kindness and generosity. Professor Hamza consistently encouraged me to explore new ideas, embrace challenges, and offered invaluable feedback and advice that kept me motivated and inspired throughout my journey. I truly cherish the wealth of knowledge I have gained under his guidance. Additionally, I extend my thanks to my lab mates, Hasib Zunair, Md Shakib Khan, and Md. Tanvir Hassan, whose constant support has been instrumental in enriching my experience. Their great ideas and encouragement have played a significant role in shaping my academic pursuits. Reflecting on my time at Concordia, I am deeply appreciative of all my instructors, who have contributed to my growth and learning in profound ways. Their dedication to education has been truly inspiring. Finally, I owe immeasurable gratitude to my parents and sibling for their unconditional love and unwavering support. Their belief in my potential, endless encouragement, and prayers have been the pillars of strength, especially during the most challenging times. I want to extend a special thank you to my mother, whose selfless sacrifices and prayers have been an endless source of motivation and hope. The support I have received from all these remarkable individuals and institutions has been transformative, and I am filled with gratitude for their role in my academic and personal development.

Table of Contents

List of Figures

List of Tables

List of Acronyms

DNNs Deep Neural Networks

CNN Convolutional neural network

GS-Net Gauss-Seidel graph neural network

MGT-Net Multi-hop graph transformer network

GCNs Graph convolutional networks

GS-NetConv Gauss-Seidel graph convolution

ResNets Residual networks

MPJPE Mean per joint position error

PA-MPJPE Procrustes-aligned mean per joint position error

PCK Prcentage of correct keypoints

AUC Area under the curve

LayerNorm Layer normalization

BatchNorm Batch normalization

ISC Initial skip connection

PRN Pose refinement network

LAM-GConv Graph convolutional with learnable adjacency matrix

MSA Multi-head self-attention

DCLs Dilated convolutional layes

FLOPs Floating-point operations

RELU Rectified Linear Unit

GELU Gaussian Error Linear Unit

CHAPTER 1

Introduction

In this chapter, we present the underlying motivation for this work, followed by a concise problem statement, a comprehensive review of the relevant literature, and thesis contributions. The literature review section provides an overview of 3D human pose estimation, graph convolutional networks, transformer based methods, and spatial-temporal based methods.

1.1 Framework, Motivation and Background

The goal of 3D human pose estimation is to predict the 3D locations of human body joints in the camera coordinate system from images or videos, with the aim of providing a way to interpret human movements and actions in computer vision applications, such as action recognition [10], human-computer interaction, sports performance analysis, and pedestrian behavior analysis [11]. Pose estimation can also be used in various healthcare applications, such as monitoring the physical therapy progress of patients or detecting abnormalities in movement patterns, and assisted living in retirement homes [12].

Despite significant progress in recent years [13], 3D human pose estimation remains a challenging task. This is largely attributed to two main challenges: (i) self-occlusions that occur when a body part is obscured by another part, and hence making it difficult for the model to accurately estimate the position of the occluded body part; and (ii) depth ambiguity that arises due to occlusions, self-occlusions, and variations in body shape, where there can be multiple 3D poses that correspond to the same 2D projection of a person in an image.

Recent approaches to 3D human pose estimation have focused on improving the accuracy and

robustness of existing methods, as well as addressing the challenges of occlusion and depth ambiguity. These approaches can be grouped into two main categories: one-stage and two-stage methods. One-stage methods [14–21], also known as direct regression methods, aim to directly predict the 3D joint locations from an input image or video without requiring any intermediate predictions. However, these methods usually suffer from depth ambiguity, which arises because the 3D pose estimation problem is inherently under-constrained, meaning that there are multiple possible 3D poses that can explain the same 2D observations. Also, they do not perform well when dealing with complex poses or occlusions. On the other hand, two-stage methods [4,7,22–37], also known as indirect regression methods, first predict intermediate representations such as 2D joint locations and then use them to predict the 3D joint locations. These methods are usually more accurate than one-stage methods, particularly when combined with robust 2D joint detectors, since they can better handle the depth ambiguity and occlusions. Pavllo *et al.* [4] demonstrate the utilization of dilated temporal convolutions to effectively leverage temporal correlations within 2D pose sequences for the purpose of video data analysis.

Residual Connections. Residual connections are a crucial architectural innovation in deep learning models, especially in deep neural networks with numerous layers. They were first introduced in the context of residual networks (ResNets) and have since become a fundamental building block in various state-of-the-art architectures. In light of this insight, incorporating residual connections into our architecture also proves beneficial. Traditional deep neural networks can encounter difficulties in training as the number of layers increases. By introducing residual connections, deeper networks become easier to optimize, as they enable the model to retain essential information from earlier layers and build upon it in subsequent layers. This facilitates the training process and allows the network to explore more complex and meaningful representations. In addition, residual connections enable the reuse of learned features from previous layers in subsequent ones. Moreover, residual connections effectively allow a neural network to learn identity mapping, i.e., passing the input directly to the output. This is crucial when the optimal mapping between input and output is close to an identity mapping. Without residual connections, the network would need to learn this mapping from scratch, which can be inefficient and challenging, especially in very deep architectures.

Layer Normalization. Layer Normalization is a technique used in deep learning to normalize the inputs of each layer in a neural network. It is an alternative to other normalization techniques, like Batch Normalization. The main objective of Layer Normalization is to address the issue of internal covariate shift, which refers to the change in the distribution of intermediate activations that occurs during training. Layer Normalization normalizes the inputs of each layer independently, regardless

of the batch size. For each feature in the layer, the mean and variance are computed across all the samples within that specific layer. This normalization is applied individually to each training sample. The normalization process is given by the following formula:

$$\text{LayerNorm}(\mathbf{x}) = \alpha \odot \frac{\mathbf{x} - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{1.1}$$

where where $\odot$ is an element-wise product (i.e., the Hadamard product), μ and σ are the mean and variance of $\mathbf{x}$, α and β are learned scaling factors and bias terms.

Dropout. Dropout is a regularization technique used in neural networks to mitigate the problem of overfitting. Overfitting occurs when a neural network becomes too specialized in learning from the training data and fails to generalize well to unseen or new data. In a typical neural network, each neuron in a layer connects to neurons in the previous and subsequent layers, forming a dense network of connections. During training, dropout randomly "deactivates" or "turns off" a certain fraction of neurons in the network with a probability p. This means that those neurons and their connections are temporarily ignored for the current forward and backward pass. Therefore, we also apply dropout layers to our models.

1.2 Objectives

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

- We propose a Gauss-Seidel graph neural network (GS-Net) whose layer-wise propagation rule is obtained by iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method. Our framework follows the two-stage paradigm for 3D human pose estimation, where GS-Net is employed as a lifting network to predict the 3D pose locations from 2D predictions.

- We also propose a multi-hop graph transformer network (MGT-Net), which leverages multi-head self-attention, multi-hop graph convolutions with disentangled neighborhoods, and dilated convolutions to effectively capture both long-range dependencies and local-global contextual information.

1.3 Literature Review

The basic goal of 3D human pose estimation is to determine the 3D positions of the joints of a person in a given image or video. These joints can include the head, torso, arms, legs, and

hands, and the accurate estimation of 3D human pose can provide valuable insights into human movement. Several lines of work are related to ours. In this section, we provide an overview of 3D human pose estimation methods, a discussion of approaches based on graph convolutional neworks (GCNs), Transformers, and spatial-temporal methods.

Graph Convolutional Network based Methods. In recent years, GCNs have emerged as a powerful approach for 3D human pose estimation, leveraging the inherent graph structure of human skeletons to capture spatial dependencies between body joints. Early works [30–37] primarily focused on exploiting graph representations to model human body structures. This arises from the fact that the human skeleton can naturally be represented as a graph, with nodes corresponding to joints and edges capturing the connections between the joints. Zhao *et al.* [30] propose SemGCN, a semantic graph convolutional network that has the ability to learn and encode semantic information such as both local and global node relationships, which may not be explicitly represented in the graph structure. Zou *et al.* [31] design a high-order graph convolutional network to capture high-order dependencies between body joints by considering neighbors that are multiple hops away when updating the joint features in an effort to reduce the uncertainty that arises from occlusion or depth ambiguity. In [34], a modulated GCN comprised of weight modulation and adjacency modulation is proposed, where weight modulation enables the learning of unique modulation vectors for individual nodes and adjacency modulation modifies the graph structure to account for additional edges beyond the human skeleton that can be modeled. Building upon modulated GCN, Lee *et al.* [36] design multi-hop modulated GCN, an architecture in which the features of multi-hop neighbors are modulated and fused by a learnable matrix designed to assign higher weights to features of neighbors that are closer in hop distance. Zhang *et al.* [37] introduce GroupGCN, which is inspired by the concept of group convolution in convolutional neural networks. The GroupGCN architecture includes group convolution that is used to ensure that each group has its own weight matrix and spatial aggregation kernel, and group interaction that allows features to interact between groups and take into account global information for better performance.

Transformer-based Methods. Inspired by the success of Transformers in natural language processing and computer vision tasks [38, 39], Transformer-based methods for 3D human pose estimation have recently emerged as a promising approach for capturing long-range dependencies and global context between body joints in video sequences by leveraging self-attention mechanisms. Self-attention allows each body joint to interact with all other joints in the sequence, capturing their relative importance and relevance. Zhao *et al.* [40] present GraFormer, a model built on the Transformer architecture to model the relations between the different joints. It is comprised of a graph attention module and a Chebyshev graph convolutional module. However, Chebyshev graph

convolutions approximate the spectral graph convolution by truncating the Chebyshev polynomial expansion. This approximation introduces errors, which can lead to the mixing of information from distant body joints into local joint features. Such errors can negatively impact the accuracy of pose estimation, especially for complex poses. Zheng *et al.* [3] introduce PoseFormer, a Transformer-based model composed of a spatial transformer module that employs spatial self-attention layers, considering the positional information of 2D joints, to generate a latent feature representation for each frame, and a temporal transformer module that analyzes global dependencies across frames. Since PoseFormer is a pure Transformer-based model that directly models the spatial and temporal aspects of the input video sequence, it does not explicitly consider the graph structure of the human body, and hence it may not fully exploit the inherent graph relationships that exist between body joints, potentially leading to suboptimal performance. More recently, Cai *et al.* propose HTNet [9], a human topology aware network comprised of a local joint-level connection module based on GCNs to model physical connections between adjacent joints at the joint level, an intra-part constraint module to provide constraints for intra-part joints at the part level, and a global body-level interaction module based on multi-head self-attention to extract global features among inter-part joints at the body level. However, one key limitation of HTNet is the challenge in designing an optimal connection structure for its three modules. Also, the series structure, where the modules are connected sequentially, can increase the model size.

3D Human Pose Estimation. As mentioned earlier, there are two general approaches for 3D human pose estimation: one-stage and two-stage. In the one-stage approach [15, 17, 20, 41–43], 3D human pose estimation is performed in a single step without any intermediate processing. This approach typically involves training a deep learning model such as a convolutional neural network or a graph neural network to directly predict the 3D pose of the person in the image or video. The network takes the input image or video frame as input and outputs the 3D pose as a set of joint coordinates. The inherent ambiguity of 2D-to-3D mapping can, however, lead to multiple possible 3D pose solutions for a given 2D input. For example, Toshev *et al.* [44] introduce the application of Deep Neural Networks (DNNs) to human pose estimation, leveraging a DNN-based regression approach to joint coordinates and a cascaded regressor system that provides the advantage of holistic pose reasoning and context awareness. Pavlakos *et al.* [17] uses a coarse-to-fine method to predict the 3D pose, where the coarse predictions guide the fine-grained predictions. This method works by first generating a coarse 3D volumetric representation of the human body using a 3D convolutional neural network (CNN). The coarse volumetric representation is then used to guide the fine-grained prediction of the 3D pose by training a 2D CNN to regress the 2D joint locations, followed by a 3D refinement network that refines the predicted 3D pose by taking

into account the coarse volumetric representation. Another common approach to 3D human pose estimation is multi-view fusion, which leverages multiple camera views and fuses 2D poses to generate 3D pose estimates. Qiu *et al.* [45] present an novel approach for estimating 3D human poses from multiple calibrated cameras, utilizing a CNN-based multi-view feature fusion method to enhance 2D pose estimation accuracy and a recursive pictorial structure model to estimate 3D poses from the multi-view 2D poses. He *et al.* [46] introduce the epipolar transformer, enabling 2D pose detectors to utilize 3D-aware features by fusing features along epipolar lines of neighboring views, along with a recursive pictorial structure model to reconstruct the 3D pose from multi-view 2D poses. Liu *et al.* [47] present a dual consecutive network for multi-frame person pose estimation, leveraging a pose temporal merger and a pose residual fusion module to extract contextual information from adjacent frames, resulting in localized search ranges and improved accuracy in locating keypoints. Also, Liu *et al.* [48] explore the multi-frame human pose estimation task by emphasizing the effective utilization of temporal contexts through feature alignment and complementary information mining, achieved by introducing a hierarchical coarse-to-fine network that progressively aligns supporting frame features with the key frame feature. While 3D pose estimation typically focuses on predicting the positions of skeleton joints, the task of 3D human body mesh recovery [49] from monocular images aims to reconstruct the complete mesh representation of the human body, encompassing both pose and shape.

In the two-stage approach [4, 7, 22–31, 50], 3D human pose estimation is performed in two steps: first, the 2D keypoints of the person are detected in the input image or video frame using an off-the-shelf 2D pose detector [51, 52], and then the 3D pose is estimated from the 2D keypoints using a 2D-to-3D lifting network. For instance, Martinez *et al.* [22] propose a simple multilayer neural network architecture with a building block comprised primarily of a linear layer, followed by batch normalization, dropout, a Rectified Linear Unit (RELU) activation function, as well as a residual connection to help improve generalization performance and reduce training time. Li *et al.* [50] introduce multi-hypothesis Transformer, a novel three-stage framework based on the Transformer architecture, designed to address the ambiguous inverse problem of 3D human pose estimation from monocular videos by generating multiple pose hypotheses in the spatial domain and facilitating communication between them in both independent and mutual manners in the temporal domain. Our approach falls under the category of two-stage methods, and we employ the proposed GS-Net model as a lifting network. We first estimate the 2D joint coordinates in the input image using a 2D pose estimator, and then we use these 2D joint locations as input to GS-Net to predict 3D joint coordinates. A key advantage of using GS-Net as our lifting network is that we can leverage the key components (i.e., skip connection, weight and adjacency modulation, and

ConvNeXt) present in the model architecture to effectively estimate the 3D joint positions.

Spatial-temporal based Methods. Recent approaches have utilized temporal information from video to produce more accurate predictions of 3D human pose estimation. To mitigate the challenges of depth ambiguity and visual jitters in static images, Hossain *et al.* [53] introduce a recurrent neural network composed of Long Short-Term Memory (LSTM) units with shortcut connections to exploit temporal information from human pose sequences. Their normalized LSTM network first encodes 2D poses into a fixed feature vector and decodes it into a 3D pose. However, encoding the spatial configuration of 2D poses into a one-dimensional vector ignores the representation of the spatial arrangement of these poses. Moreover, Pavlo *et al.* [4] propose a fully convolutional based model based on dilated temporal convolutions to estimate 3D poses over 2D keypoint sequences. Other recent methods incorporate both spatial configuration constraints and temporal information to estimate 3D poses. For instance, Cai *et al.* [5] exploit graph pooling and graph unsampling to learn multi-scale feature representations pertaining to different semantic meanings. However, their local-to-global network architectures are limited to embedding fixed-length spatial-temporal sequences, making it less efficient at capturing global context information across frames. Building upon [4], Liu *et al.* [54] utilize attention mechanisms to model local and global spatial information and employ the dilated temporal model to process varying skeleton sequences.

Our approach differs from these GCN-based approaches in that instead of using GCN as a 2D-to-3D lifting network, we design a new graph neural network with skip connections, together with weight and adjacency modulation. We leverage skip connections, which are integrated by design in our network architecture, allowing the network to reuse lower-level features in higher-level layers, thereby helping the model to learn more complex representations. In addition, we employ a variant of the ConvNeXt block in our network architecture. ConvNeXt has demonstrated competitive accuracy and scalability [55] compared to Transformers, while retaining the simplicity and efficiency of standard convolutional neural networks. We also propose MGT-Net, which, in comparison to the aforementioned methods, combines the strengths of multi-head self-attention and multi-hop graph convolutions with disentangled neighborhoods. By leveraging both local and global contextual information and capturing long-range dependencies effectively, MGT-Net aims to improve the accuracy and robustness of 3D human pose estimation in challenging scenarios. Moreover, we integrate dilated graph convolutions into our network architecture to enhance the model's receptive field, enabling it to capture larger contextual information and better understand the dependencies between body joints across different scales and distances.

1.4 Overview and Contributions

This thesis is structured as follows:

- Chapter 1 begins with a comprehensive exploration of the motivations and goals driving this study. Subsequently, the problem statement is articulated, followed by a clear definition of the study's objective. Additionally, a detailed literature review is conducted, offering valuable insights into relevant algorithms employed in the domain of deep learning for 3D human pose estimation.

- In Chapter 2, we propose a novel Gauss-Seidel graph neural network (GS-Net) [56] whose layer-wise propagation rule is obtained by iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method. In addition, we design a network architecture comprised of weight and adjacency modulation, skip connection, and a variant of the ConvNeXt residual block. We demonstrate through extensive experiments and ablation studies the effectiveness and generalization ability of the proposed GS-Net model, achieving competitive performance on two benchmark datasets.

- In Chapter 3, we propose a multi-hop graph transformer network (MGT-Net), which leverages multi-head self-attention, multi-hop graph convolutions with disentangled neighborhoods, and dilated convolutions to effectively capture both long-range dependencies and local-global contextual information. In addition, we design a network architecture composed of a graph attention block and a multi-hop graph convolutional block to capture spatial dependencies and model the complex interactions among body joints. We also demonstrate through extensive experiments and ablation studies the effectiveness and generalization ability of the proposed MGT-Net model, achieving competitive performance in 3D human pose estimation on two benchmark datasets while retaining a small model size.

- Chapter 4 encapsulates the thesis's key contributions, addresses its limitations, and delineates promising avenues for future research in this area of study.

CHAPTER 2

Iterative Graph Filtering Network for 3D Human Pose Estimation

In this chapter, we introduce an iterative graph filtering framework for 3D human pose estimation, which aims to predict the 3D joint positions given a set of 2D joint locations in images. Our approach builds upon the idea of iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method. Motivated by this iterative solution, we design a Gauss-Seidel network (GS-Net) architecture, which makes use of weight and adjacency modulation, skip connection, and a pure convolutional block with layer normalization. Adjacency modulation facilitates the learning of edges that go beyond the inherent connections of body joints, resulting in an adjusted graph structure that reflects the human skeleton, while skip connections help maintain crucial information from the input layer's initial features as the network depth increases. We evaluate our proposed model on two standard benchmark datasets, and compare it with a comprehensive set of strong baseline methods for 3D human pose estimation. Our experimental results demonstrate that our approach outperforms the baseline methods on both datasets, achieving state-of-the-art performance. Furthermore, we conduct ablation studies to analyze the contributions of different components of our model architecture and show that the skip connection and adjacency modulation help improve the model performance.

2.1 Introduction

In traditional 3D human pose estimation methods [22], each joint in the human skeleton is localized independently, and the spatial relationship between the joints is not considered. This can lead to incorrect poses, especially in cases where joints occlude each other. To tackle this potential weakness, graph convolutional networks (GCNs) have recently shown promising results in 3D human pose estimation [30], yielding improved performance over traditional methods. By naturally modeling the skeleton structure of the human body as a graph, where each joint is represented as a node and the edges represent the relationships between the joints, GCNs are able to capture the spatial relationships between joints and learn an efficient representation of the underlying pose. One of the main benefits of using GCNs is that they can capture the dependencies between the joints. The position of each joint is dependent on the position of other joints in the body, and GCNs can model these dependencies explicitly. This allows the model to reason about the body as a connected system and to make accurate predictions even when some joints are occluded. Despite their promising results, GCNs suffer from several shortcomings [32]. First, they use a shared weight matrix (i.e., the same set of weights is used for all nodes in the graph) to determine the importance of neighboring joints when computing the representation of a particular joint. However, the shared weight matrix can be limited in its ability to capture complex relationships between joints in a human body, and may not be able to capture the nuances of joint relationships. This can limit the expressiveness of the model, making it difficult to learn more complex relationships between joints, thereby resulting in suboptimal performance [34]. Second, GCNs without skip connections are limited by the receptive field of their filters, which only allow them to capture information from the local neighborhood of each joint in the graph. Skip connections allow for the flow of information to bypass certain layers in the network and improve the modeling of long-range dependencies in the graph without sacrificing performance. Third, GCNs that use only local information can lead to over-smoothing [57–60], where the learned node representations become too similar, especially when the network depth increases.

More recently, Transformer-based models [40, 61, 62] have been shown to be an effective approach for 3D human pose estimation due to their ability to capture long-range dependencies between the joints. The basic idea is to treat the 3D poses of human bodies as a sequence of tokens, each of which represents a joint in the human body, and the goal is to predict the 3D coordinates of each joint. However, there are some challenges associated with using Transformers for 3D human pose estimation, such as the need for large amounts of training data and the computational cost of processing long sequences of joint tokens. To mitigate some of these challenges, Zhuang *et al.* [55] introduce ConvNeXt, an architecture that solely utilizes CNN modules and performs com-

petitively with Transformers in both accuracy and scalability, yielding superior performance over the Swin Transformer [63]. Unlike the Swin Transformer, specialized modules like shifted window attention or relative position biases are not needed for ConvNeXt. The ConvNeXt residual block is fully convolutional, and uses a layer normalization, followed by the Gaussian Error Linear Unit activation function. This motivates us to explore and eventually design a variant of the ConvNeXt block for skeleton-based representation learning.

In this chapter, we propose a Gauss-Seidel graph neural network (GS-Net), an architecture that incorporates a skip connection, weight and adjacency modulation, and a variant of the ConvNeXt residual block while upholding the flexibility of models intended for broad applications, including 3D human pose estimation, which is the primary focus of this work. Using an iterative graph filtering approach for solving a linear system of sparse equations, we start by deriving the layer-wise propagation rule of the proposed network via the Gauss-Seidel iterative method. Our framework follows the two-stage paradigm for 3D human pose estimation, where GS-Net is employed as a lifting network to predict the 3D pose locations from 2D predictions. Inspired by the regularized elastic net regression method, we train our model using a flexible loss function defined as a weighted sum of the mean squared and mean absolute errors between the 3D ground truth coordinates and estimated 3D joint coordinates over a training set consisting of human poses. Experiments on two large-scale datasets verify the effectiveness of our model using both quantitative and qualitative evaluations. Moreover, our ablation studies highlight the potential for further improvements through the incorporation of skip connections and adjacency modulation. Our contributions can be summarized as follows:

- We propose a novel Gauss-Seidel graph neural network (GS-Net) whose layer-wise propagation rule is obtained by iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method.
- We design a network architecture comprised of weight and adjacency modulation, skip connection, and a variant of the ConvNeXt residual block.
- We demonstrate through extensive experiments and ablation studies the effectiveness and generalization ability of the proposed GS-Net model, achieving competitive performance on two benchmark datasets.

The remainder of this chapter is organized as follows. Section 2.2 presents the methodology, including the problem formulation, propagation rule, model architecture, and model training and prediction. The experimental setup and results are presented in Section 2.3.

2.2 Proposed Method

2.2.1 Preliminaries and Problem Statement

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. The graph structure is encoded by an $N \times N$ adjacency matrix $\mathbf{A}$ whose (i, j)-th entry is equal to 1 if there the edge between neighboring nodes i and j, and 0 otherwise. We 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.

Gauss-Seidel Method. Let $\mathbf{\Phi}$ be an $N \times N$ matrix whose diagonal entries are all nonzero, and consider the matrix decomposition $\mathbf{\Phi} = \mathbf{\Lambda} - (\mathbf{E} + \mathbf{F})$, where $\mathbf{\Lambda}$ is the diagonal matrix of $\mathbf{\Phi}$, $\mathbf{E}$ is its strictly lower triangular part, and $\mathbf{F}$ is its strictly upper triangular part. Given a vector $\mathbf{x} \in \mathbb{R}^N$, the Gauss-Seidel iteration [64] for solving a matrix equation $\mathbf{\Phi}\mathbf{h} = \mathbf{x}$ is given by

$$\mathbf{h}^{(k+1)} = (\mathbf{\Lambda} - \mathbf{E})^{-1}\mathbf{F}\mathbf{h}^{(k)} + (\mathbf{\Lambda} - \mathbf{E})^{-1}\mathbf{x}, \tag{2.1}$$

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

Problem Formulation. 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

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} l(f(\mathbf{x}_i), \mathbf{y}_i), \tag{2.2}$$

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

$$l = (1 - \alpha) \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_2^2 + \alpha \sum_{i=1}^{N} \|\mathbf{y}_i - f(\mathbf{x}_i)\|_1, \tag{2.3}$$

where $\alpha \in [0, 1]$ is a weighting factor that controls the mixing amount between the ℓ_2 and ℓ_1 penalties. It is worth pointing out that the proposed loss function draws inspiration from the penalty function used in the elastic net regression model [65], which is a weighted combination of lasso

and ridge regularization. When $\alpha = 0$, the penalty function is equivalent to lasso regression, and when $\alpha = 1$, it is equivalent to ridge regression. A key advantage of elastic net regression is that it reduces the impact of irrelevant predictors by shrinking their coefficients towards zero, unlike ridge regression which only reduces the size of the coefficients.

2.2.2 Iterative Graph Filtering

Graph filtering refers to the process of enhancing signals defined on graphs while preserving the underlying graph structure. It typically employs graph Laplacian regularization, which aims to incorporate the underlying graph structure into optimization problems to achieve desirable properties in the solutions while encouraging the filtered signal to be smooth and to vary smoothly across the graph. More specifically, the goal of graph filtering with Laplacian regularization is to minimize the following objective function

$$\mathcal{J}(\mathbf{H}) = \frac{1}{2}\|\mathbf{H} - \mathbf{X}\|_F^2 + \frac{\beta}{2}\operatorname{tr}(\mathbf{H}^\intercal \mathbf{L}\mathbf{H}), \tag{2.4}$$

where $\mathbf{X}$ is the initial feature matrix, $\mathbf{H}$ is the filtered feature matrix, $\mathbf{L} = \mathbf{I} - \hat{\mathbf{A}}$ is the normalized Laplacian matrix, $\beta \in (0, 1)$ is a regularization parameter, and $\|\cdot\|_F$ and $\operatorname{tr}(\cdot)$ denote the Frobenius norm and trace operator, respectively. The first term on the right-hand side represents a fitting constraint or data fidelity, where a filtered feature matrix is expected to exhibit limited deviation from the initial feature matrix. The second term represents a smoothness constraint, which requires an effective filtered feature matrix to have consistent behavior in the vicinity of neighboring graph nodes (i.e., adjacent nodes should have similar node features). The trade-off parameter β controls how much emphasis to put on data fidelity versus smoothness. A larger value of β corresponds to a stronger regularization effect and a greater emphasis on smoothness of the filtered feature matrix at the expense of fitting the initial data matrix more closely, while a smaller value places more emphasis on the data fidelity term and results in a better fit to the initial feature matrix.
Taking the derivative of $\mathcal{J}$ and set it to zero, we have

$$\frac{\partial \mathcal{J}}{\partial \mathbf{H}} = \mathbf{H} - \mathbf{X} + \beta\mathbf{L}\mathbf{H} = \mathbf{0}, \tag{2.5}$$

which yields a system of sparse equations given by

$$(\mathbf{I} + \beta\mathbf{L})\mathbf{H} = \mathbf{X}, \tag{2.6}$$

where β controls the smoothness of the filtered graph signal. It is important to point out that the matrix $\mathbf{I} + \beta\mathbf{L}$ is symmetric positive definite with minimal eigenvalue equal to 1 and maximal eigenvalue bounded from above by $1 + 2\beta$.

Iterative Solution. Using the matrix decomposition for the Gauss-Seidel method, we can express $\mathbf{I} + \beta\mathbf{L}$ as the sum of a diagonal matrix, a strictly lower triangular matrix, and a strictly upper triangular matrix as follows:

$$\mathbf{I} + \beta\mathbf{L} = (1+\beta)\mathbf{I} - \beta\hat{\mathbf{A}} = \mathbf{\Lambda}_\beta - (\hat{\mathbf{A}}_\beta^\intercal + \hat{\mathbf{A}}_\beta), \tag{2.7}$$

where $\mathbf{\Lambda}_\beta = (1+\beta)\mathbf{I}$ is a diagonal matrix (i.e., uniformly scaled identity matrix), and $\hat{\mathbf{A}}_\beta$ denotes the upper triangular matrix of $\beta\hat{\mathbf{A}}$. Since the uniformly scaled and normalized adjacency matrix $\beta\hat{\mathbf{A}}$ is symmetric with zero diagonal entries, its lower triangular part is $\hat{\mathbf{A}}_\beta^\intercal$. Hence, the Gauss-Seidel iterative solution of $(\mathbf{I} + \beta\mathbf{L})\mathbf{H} = \mathbf{X}$ is given by

$$\mathbf{H}^{(k+1)} = (\mathbf{\Lambda}_\beta - \hat{\mathbf{A}}_\beta^\intercal)^{-1}\hat{\mathbf{A}}_\beta\mathbf{H}^{(k)} + (\mathbf{\Lambda}_\beta - \hat{\mathbf{A}}_\beta^\intercal)^{-1}\mathbf{X}, \tag{2.8}$$

Since matrix inversion can be computationally expensive, especially for large matrices, a common approach to approximate the inverse of a matrix is to use a truncated series approximation such as the Neumann series expansion, which involves summing a finite number of terms of an infinite series, providing a good approximation of the inverse.

Lemma 1 *Let* $\mathbf{S}$ *be an* $N \times N$ *matrix with spectral radius* $\rho(\mathbf{S}) < 1$*. Then,* $\mathbf{I} - \mathbf{S}$ *is invertible and we have*

$$(\mathbf{I} - \mathbf{S})^{-1} = \sum_{i=1}^{\infty} \mathbf{S}^i. \tag{2.9}$$

Recall that the eigenvalues of a lower or upper triangular matrix are its diagonal entries. Hence, the spectral radius of the lower triangular matrix $\hat{\mathbf{A}}_\beta^\intercal - \beta\mathbf{I}$ is equal to β. Since $\beta \in (0, 1)$, we can approximate the inverse of $\mathbf{\Lambda}_\beta - \hat{\mathbf{A}}_\beta^\intercal$ using Lemma 1 with first order approximation as follows:

$$(\mathbf{\Lambda}_\beta - \hat{\mathbf{A}}_\beta^\intercal)^{-1} = (\mathbf{I} - (\hat{\mathbf{A}}_\beta^\intercal - \beta\mathbf{I}))^{-1} \approx (1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal. \tag{2.10}$$

Therefore, the Gauss-Seidel iterative solution becomes

$$\mathbf{H}^{(k+1)} = ((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)\hat{\mathbf{A}}_\beta\mathbf{H}^{(k)} + ((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)\mathbf{X}. \tag{2.11}$$

2.2.3 Gauss-Seidel Network

Propagation rules in graph neural networks such as GCNs generally define how information is passed between nodes in a graph during the forward pass of the network. These rules are used to update the feature representations of nodes in the graph based on information from their neighbors, and are a critical component in the learning process of graph neural networks. Motivated by the

Gauss-Seidel iterative solution (2.11) of graph filtering with Laplacian regularization, we propose a Gauss-Seidel network (GS-Net) with the following layer-wise propagation rule:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)\hat{\mathbf{A}}_\beta\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} + ((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)\mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}\right), \tag{2.12}$$

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 $\mathbf{H}^{(0)} = \mathbf{X}$.

Note that the second term on the right-hand side of the propagation rule is basically a skip connection that provides a effective way to pass information directly from the initial feature matrix to the next network layer, without any nonlinear transformation in between. This allows the information to be preserved and can help improve the flow of information through the network, enabling the model to learn richer, more expressive feature representations.

Weight Modulation. One of the main shortcomings of GCNs is that they share the weight matrix, meaning that they treat all nodes in the graph equally, without considering their individual characteristics. This can lead to over-smoothing, where the learned representations of nodes become too similar, reducing the ability of the model to distinguish between different nodes in the graph. To circumvent this limitation, we leverage weight modulation [34], which employs a common weight matrix, but a learnable weight modulation vector is introduced for each node, which scales the weight matrix according to the local topology of the node. This enables each node to have a unique and adaptive representation, which can capture more fine-grained information about the node's local structure and its position in the graph. Hence, the layer-wise propagation rule with weight modulation can be written as

$$\begin{aligned}\mathbf{H}^{(\ell+1)} = \sigma\Big(&((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)\hat{\mathbf{A}}_\beta(\mathbf{M}^{(\ell)} \odot (\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)})) \\ &+ ((1-\beta)\mathbf{I} + \hat{\mathbf{A}}_\beta^\intercal)(\mathbf{M}^{(\ell)} \odot (\mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}))\Big),\end{aligned} \tag{2.13}$$

where $\mathbf{M}^{(\ell)} \in \mathbb{R}^{N \times F_{\ell+1}}$ is a learnable weight modulation matrix, and $\odot$ denotes element-wise multiplication.

Adjacency Modulation. We modulate the normalized adjacency matrix to capture not just the interactions between adjacent nodes, but also the relationships between distant nodes beyond the natural connections of body joints [34].

$$\breve{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}, \tag{2.14}$$

where $\mathbf{Q}^{(\ell)} \in \mathbb{R}^{N \times N}$ is a learnable adjacency modulation matrix. Since the skeleton graph is symmetric, we symmetrize the adjacency modulation matrix by taking the average of the matrix and its transpose, i.e., $(\mathbf{Q} + \mathbf{Q}^T)/2$. Therefore, the layer-wise propagation rule of the Gauss-Seidel graph network with weight and adjacency modulation can be written as

$$\begin{aligned} \mathbf{H}^{(\ell+1)} = \sigma\Big(& ((1-\beta)\mathbf{I} + \check{\mathbf{A}}_\beta^\intercal)\check{\mathbf{A}}_\beta(\mathbf{M}^{(\ell)} \odot (\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)})). \\ & + ((1-\beta)\mathbf{I} + \check{\mathbf{A}}_\beta^\intercal)(\mathbf{M}^{(\ell)} \odot (\mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}))\Big). \end{aligned} \tag{2.15}$$

In adjacency modulation, a weight modulation vector is introduced for each node, which is learned during training and is used to modulate the weights of the adjacency matrix. This enables the adjacency matrix to be dynamically adjusted based on the local node features, which can lead to better performance on 3D human pose estimation.

Model Architecture. Figure 2.1 illustrates the architecture of our proposed GS-Net model for 3D human pose estimation. The input to the model consists of 2D keypoints, obtained via an off-the-shelf 2D detector [51]. Inspired by the architectural design of the ConvNeXt block [55], our residual block consists of two graph convolutional (GS-NetConv) layers. The first convolutional layer is followed by layer normalization, while the second convolutional layer is followed by a GELU activation function, which is a smoother version of ReLU and is commonly used in Transformers based approaches. This residual block is repeated four times. We also employ a non-local layer [66] before the last convolutional layer. The last layer of the network generates the 3D pose.

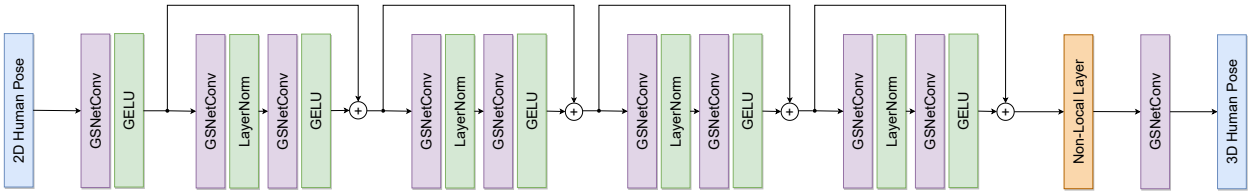


Figure 2.1: Network architecture of the proposed GS-Net model for 3D human pose estimation. Our model accepts 2D pose coordinates (16 or 17 joints) as input and generates 3D pose predictions (16 or 17 joints) as output. We use ten Gauss-Seidel graph convolutional layers with four residual blocks. In 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.

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

$$\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N)^\intercal \in \mathbb{R}^{N \times 3}, \tag{2.16}$$

where $\hat{\mathbf{y}}_i$ is a 3D row vector of predicted 3D pose coordinates. This predicted set of 3D joint coordinates can be visualized in a 3D space, allowing for interactive manipulation and analysis of the pose.

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

$$\mathcal{L} = \frac{1}{N}\left[(1-\alpha)\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1\right], \tag{2.17}$$

which is a weighted sum of the mean squared and mean absolute errors between the 3D ground truth coordinates $\mathbf{y}_i$ and estimated 3D joint coordinates $\hat{\mathbf{y}}_i$ over a training set consisting of N human poses. For the mean squared error, the squared differences between the predicted and ground truth coordinates are averaged, meaning that larger errors have a greater impact on the overall score. In other words, the mean squared error is more sensitive to outliers and penalizes larger errors more heavily than the mean absolute error, which is more robust to outliers and treats all errors equally. The weighting factor α balances the contribution of each loss term. When $\alpha = 0$, our loss function reduces to the mean squared error (i.e., ridge regression) and when $\alpha = 1$, it reduces to the mean absolute error (i.e., lasso regression).

2.3 Experiments

In this section, we present the results of our proposed framework against competing baselines for 3D human pose estimation. We begin by outlining the experimental setup, followed by providing an overview of evaluation protocols and implementation details. Then, we present both quantitative and qualitative results on two benchmark datasets using various evaluation metrics. We also conduct ablation studies on the significance of various components in our model in an effort to provide valuable insight into the effectiveness of the model. The source code is available at: https://github.com/zaedulislam/GS-Net.

2.3.1 Experimental Setup

Datasets. We comprehensively evaluate our model on Human 3.6M [1] and MPI-INF-3DHP [2], which are standard large-scale benchmark datasets for 3D human pose estimation.

Human3.6M is a large-scale dataset comprised of 3.6 million images captured at 50Hz by 4 synchronized cameras in different positions and perspectives. A total of 11 professional actors (6 men

and 5 women) perform 15 actions (Directions, Discussion, Eating, Greeting, Phoning, Posing, Purchases, Sitting, Sitting Down, Smoking, Photo, Waiting, Walk Dog, Walking, and Walk Together) in an interior setting, as depicted in Figure 2.2. A motion capture system captures the annotations of precise 3D body joint coordinates, while projection with known intrinsic and extrinsic camera parameters yields the 2D poses. Annotated 3D joints are available for 7 subjects. The dataset is split into two sets: a training set and a test set. The training set contains data from five of the actors (S1, S5, S6, S7, S8), while the test set contains data from the remaining two actors (S9 and S11). These training and test sets are balanced, meaning that they contain an equal number of samples for each activity and for each subject. For data preprocessing [22, 30, 31, 67], we apply standard normalization to the 2D and 3D poses before feeding the data to the model. To achieve zero-centering, the hip joint is adopted as the root joint of the 3D poses.

MPI-INF-3DHP is a benchmark dataset for 3D human pose estimation from monocular RGB images, and comprises both indoor environments with limited space and complex outdoor scenes, as illustrated in Figure 2.3. A total of 8 actors (4 men and 4 women) were captured on camera from 14 camera views, each performing 8 sets of activities that encompassed a wider range of pose categories than the Human3.6M dataset. The activities varied from simple movements such as walking and sitting to more challenging exercises and dynamic actions. The duration of each activity set is approximately one minute, and the actors were dressed in two different sets of clothing that were rotated across the activity sets. One clothing set consisted of casual wear suitable for everyday use, while the other set was plain-colored to facilitate easy augmentation. The dataset also includes ground-truth annotations of the 3D joint positions.

Evaluation Protocols and Metrics. For the Human 3.6M benchmark, there are two standard evaluation protocols used for training and testing [22], referred to as Protocol #1 and Protocol #2. Under Protocol #1, we report mean per-joint position error (MPJPE), which computes the average Euclidean distance between the predicted and ground-truth 3D positions of each joint after aligning the root joint (i.e., hip joint). Another commonly used metric for evaluating the accuracy of 3D human pose estimation models is the Procrustes-aligned mean per-joint position error (PA-MPJPE), where MPJPE is computed after rigid alignment of the prediction with respect to the ground truth. The PA-MPJPE metric first applies Procrustes analysis to align the predicted and ground-truth joint positions to a common coordinate system. This alignment is performed by scaling, rotating, and translating the predicted joint positions to minimize the sum of squared distances between the predicted and ground-truth joint positions. Once the joint positions are aligned, the PA-MPJPE is calculated by computing the mean of the Euclidean distances between the aligned predicted and ground-truth joint positions for each joint. Both MPJPE and PA-MPJPE

Figure 2.2: Examples of actions performed by different actors in the Human3.6M dataset [1].

Figure 2.3: Examples of activities in the MPI-INF-3DHP dataset [2].

are measured in millimeter (mm), and lower error values imply better performance. Both Protocol #1 and Protocol #2 use five subjects (S1, S5, S6, S7 and S8) for training and two subjects (S9 and S11) for testing. All camera views are trained with a single model for all actions. For the MPI-

INF-3DHP dataset, we use Percentage of Correct Keypoint (PCK) with a threshold of 150mm and Area Under Curve (AUC) for a range of PCK thresholds as evaluation metrics [19, 24, 68–71]. Both the PCK and AUC metrics provide a measure of how well the predicted joint positions align with the ground-truth joint positions within a certain distance threshold. Higher PCK and AUC scores indicate better performance.

Baseline Methods. We compare the performance of our model against various state-of-the-art GCN-based approaches for estimating 3D poses, including SemGCN [30], a GCN-based model that uses a multi-task learning approach to jointly optimize for 3D joint positions and body joint angles; CompGCN [35], a hierarchical composition approach that uses a multi-level attention mechanism to adaptively weight the contributions of different body parts at different levels of the hierarchy; High-order GCN [31], a baseline method that employs high-order GCNs to model the complex interactions between body joints; Weight Unsharing [32], a comprehensive study that analyzes the trade-offs between sharing weights across different body parts versus having separate weights for each body part in GCNs; MM-GCN [36], a multi-hop GCN-based method that incorporates modulated attention mechanisms to capture the interactions between different body joints across multiple hops; GroupGCN [37], a decoupling GCN for 3D human pose estimation consisting of group convolution and group interaction; and Modulated GCN [34], a GCN-based architecture that employs weight and adjacency modulation mechanisms to capture the complex relationships between different body parts in the human body.

Implementation Details. We implement our model in PyTorch, and conduct all experiments on a single NVIDIA GeForce RTX 3070 GPU with 8GB of memory. For both 2D ground truth and 2D pose detections [51], we train our model for 30 epochs using the AMSGrad optimizer, and we set the initial learning rate to 0.005, the decay factor to 0.65 per 4 epochs, the batch size to 512, and the number of channels to 384. We set the hyperparameter β to 0.2, which was computed via grid search with cross-validation on the training set. We also set the weighting factor α to 0.01. We apply dropout with a factor of 0.2 after each graph convolution layer to prevent overfitting. We also integrate a non-local layer [66] and a pose refinement module [23]. Incorporating an additional pose refinement network, comprised of two fully-connected layers, helps improve performance. However, in the ablation study we exclude the pose refinement network and the non-local layer to ensure a fair comparison.

2.3.2 Results and Analysis

Quantitative Results on Human3.6M. In Tables 2.1 and 2.2, we summarize the performance comparison results between our GS-Net model and various state-of-the-art methods for 3D pose

estimation on Human3.6M. In both tables, we report results for all 15 actions, as well as the average performance. As can be seen in Table 2.1, our method achieves on average 47.1mm and 38.7mm in terms of MPJPE and PA-MPJPE, respectively, outperforming all the baselines. Under Protocol #1, Table 2.1 reveals that our GS-Net model performs better than Modulated GCN [34] in 13 out of 15 actions, yielding 2.3mm error reduction on average, improving upon this best performing baseline by a relative improvement of 4.65%, while maintaining a fairly small number of learnable parameters. Our method also consistently performs better in almost all actions and outperforms SemGCN [30] by a significant relative improvement of 18.23% on average. Moreover, our model achieves better predictions than the best baseline on challenging actions (i.e., hard poses) that involve self-occlusion such as "Eating", "Sitting" and "Smoking", yielding relative error reductions of 5.9%, 5.22% and 7.04%, respectively, in terms of MPJPE. These self-occlusions can make human pose estimation more challenging because they limit the visible information available to the model. For instance, when eating or smoking, the hands and arms of a person can occlude parts of their face and upper body. Also, when a person is sitting, their legs and arms can occlude other body parts, such as the torso or feet.

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 last column report the average errors. Boldface numbers indicate the best 3D pose estimation 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.* [22]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Sun *et al.* [18]	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.* [19]	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.* [72]	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 [53]	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.* [69]	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.* [73]	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.* [30]	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.* [74]	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.* [75]	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.* [76]	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.* [31]	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.* [33]	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.* [35]	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.* [32]	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.* [34]	45.4	<u>49.2</u>	<u>45.7</u>	<u>49.4</u>	<u>50.4</u>	58.2	<u>47.9</u>	<u>46.0</u>	<u>57.5</u>	<u>63.0</u>	<u>49.7</u>	<u>46.6</u>	<u>52.2</u>	<u>38.9</u>	**40.8**	<u>49.4</u>
Lee *et al.* [36]	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.* [37]	<u>45.0</u>	50.9	49.0	49.8	52.2	60.9	49.1	46.8	61.2	70.2	51.8	48.6	54.6	39.6	41.2	51.6
Ours	**41.1**	**46.6**	**43.0**	**48.0**	**48.6**	<u>52.4</u>	**44.6**	**41.9**	**54.5**	65.9	**46.2**	**46.1**	**48.2**	**38.6**	<u>40.9</u>	**47.1**

Under Protocol #2, Table 2.2 shows that our model on average reduces the error by 1.83% compared to Modulated GCN [34], and achieves better results in 12 out of 15 actions. Also, our method outperforms Modulated GCN on the challenging actions of "Eating", "Sitting" and "Smoking", yielding relative error reductions of 3.58%, 4.74% and 5.68%, respectively, in terms of PA-MPJPE. Moreover, our model performs better than Modulated GCN on the challenging "Photo" action, yielding a relative error reduction of 4.72%. In addition, GS-Net outperforms High-order GCN [31] by a relative improvement of 4.86% on average, as well as on all actions.

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.
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
Zhou *et al.* [77]	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1	53.7	65.5	51.6	50.4	54.8	55.9	55.3
Martinez *et al.* [22]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Sun *et al.* [18]	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.* [72]	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 [53]	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Lee *et al.* [21]	38.0	39.3	46.3	44.4	49.0	55.1	40.2	41.1	53.2	68.9	51.0	39.1	**33.9**	56.4	38.5	46.2
Li *et al.* [74]	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.* [75]	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.* [76]	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.* [31]	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.* [33]	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.* [35]	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.* [32]	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.* [34]	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
Lee *et al.* [36]	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.* [37]	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
Ours	**34.5**	**38.4**	**35.0**	40.9	**38.9**	**42.4**	35.9	**33.9**	**44.2**	55.9	**38.2**	36.7	40.6	**30.4**	**33.8**	**38.7**

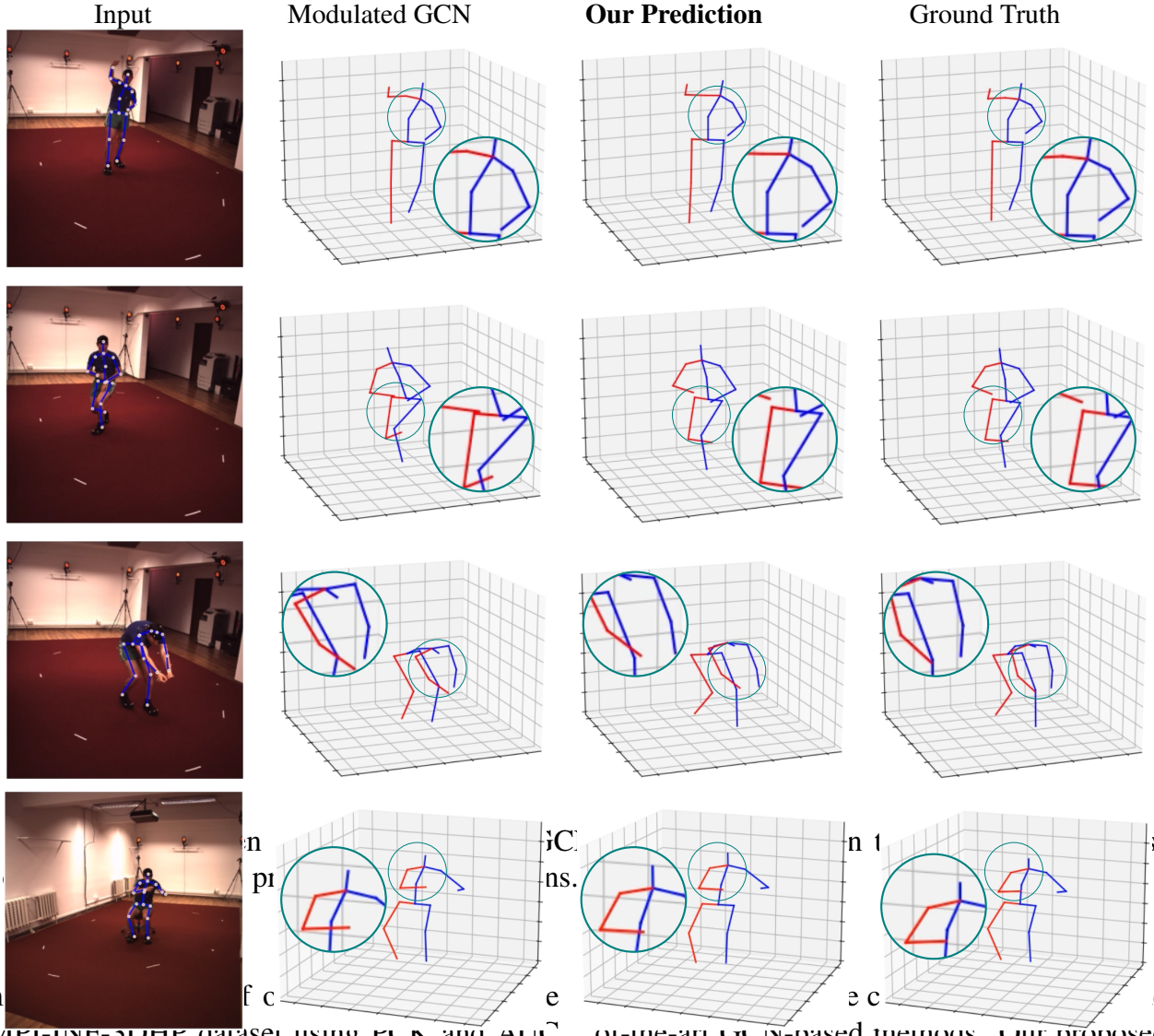
Cross-Dataset Results on MPI-INF-3DHP. In Table 3.2, we compare our method against strong baselines to test its generalization ability across different datasets. We train our model on the Human3.6M dataset and test it on the MPI-INF-3DHP dataset. The results show that our approach achieves the highest PCK and AUC scores, consistently yielding superior performance over the baseline methods in both indoor and outdoor scenes. Compared to the best performing baseline, our model yields relative improvements of 2.68% and 5.37% in terms of the PCK and AUC metrics, respectively. Although our model is only trained with indoor scenes on Human3.6M, it produces satisfactory results with outdoor settings. This verifies the strong generalization ability of our approach to unseen scenarios and datasets.

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, whereas the underlined numbers indicate the second best performance.

Method	PCK (↑)	AUC (↑)
Martinez *et al.* [22]	42.5	17.0
Mehta *et al.* [2]	64.7	31.7
Li *et al.* [78]	67.9	-
Yang *et al.* [19]	69.0	32.0
Zhou *et al.* [77]	69.2	32.5
Habibie *et al.* [68]	70.4	36.0
Pavlakos *et al.* [69]	71.9	35.3
Wang *et al.* [79]	71.9	35.8
Quan *et al.* [33]	72.8	36.5
Ci *et al.* [24]	74.0	36.7
Zhou *et al.* [80]	75.3	38.0
Zeng *et al.* [6]	77.6	43.8
Liu *et al.* [32]	79.3	47.6
Zhou *et al.* [35]	79.3	45.9
Xu *et al.* [67]	80.1	45.8
Zeng *et al.* [70]	<u>82.1</u>	46.2
Lee *et al.* [36]	81.6	<u>50.3</u>
Zhang *et al.* [37]	81.1	49.9
Ours	**84.3**	**53.0**

Qualitative Results. In Figure 3.5, we show the visual results obtained by GS-Net on the Human3.6M dataset for various actions. The effectiveness of our proposed approach in addressing the 2D-to-3D pose estimation task is demonstrated by the close match between the inferred 3D poses and the ground truth, as evidenced by the accurate results produced by our model from input images. Compared to Modulated GCN, our model yields pose predictions that are closer to the ground truth, even when dealing with challenging actions that involve self-occlusion.

Quantitative Results using Ground Truth. We also compare our model with GCN-based methods, including SemGCN [30], High-order GCN [31], HOIF-Net [33], Weight Unsharing [32], and Modulated GCN [34] using ground truth. The results are reported in Table 2.4, which shows that our model consistently yields better performance than GCN-based approaches under both Protocols #1 and #2. Under Protocol #1, our model outperforms SemGCN, High-order GCN, HOIF-Net, Modulated GCN, and Weight Unsharing by 4.73mm, 2.11mm, 0.71mm, 0.84mm, and 0.42mm, respectively, resulting in relative error reductions of 11.22%, 5.34%, 1.86%, 2.20%, and 1.11%. Under Protocol #2, our model also outperforms SemGCN, High-order GCN, HOIF-Net, Modulated



ure 4: Visual [illegible] set. Compared
dulated GCN, [illegible] ns.

le 3: Perform [illegible] el and other sta
thods on the MPI-INF-3DHP dataset using PCK and AUC [illegible] of-the-art GCN-based methods. Our proposed GS-Net meth
evaluation met [illegible] by boldface nu
t performance, [illegible] mm).
ond best perfor

Meth	(↑)	MPJPE (↓)
Marti		33.53
Mehta *et al.* [17]		31.07
Li *et al.* [61]	67.9	29.74
Yang *et al.* [10]	69.0	32.0
Zhou *et al.* [60]	69.2	32.5
Habibie *et al.* [49]	70.4	36.0
Pavlakos *et al.* [50]	71.9	35.3
Wang *et al.* [62]	71.9	35.8
Quan *et al.* [26]	72.8	36.5
Ci *et al.* [15]	74.0	36.7
Zhou *et al.* [63]	75.3	38.0
Zeng *et al.* [64]	77.6	43.8
Liu *et al.* [25]	79.3	47.6
Zhou *et al.* [28]	79.3	45.9
Xu *et al.* [48]	80.1	45.8
Zeng *et al.* [51]	82.1	46.2
Lee *et al.* [29]	81.6	50.3
Zhang *et al.* [30]	81.1	49.9

CN	]	
Modulated GCN [27]	38.25	30.06
Weight Unsharing [25]	37.83	30.09
Ours	**37.41**	**28.94**

Figure 2.4: Visual comparison between GS-Net, Modulated GCN and ground truth on the Human3.6M test set. Compared to Modulated GCN, our model is able to produce better predictions.

GCN, and Weight Unsharing by 4.59mm, 2.13mm, 0.8mm, 1.12mm, and 1.15mm, with relative error reductions of 13.69%, 6.86%, 2.69%, 3.73%, and 3.82%, respectively.

4.3 Ablation Study

In order to better understand the impact of the design choi made for our network architecture, we perform ablation exp ments on Human3.6M under controlled settings. We report results using 2D ground truth for training and testing to eli nate the added uncertainty from 2D pose detectors. These ab tion studies demonstrate the efficacy of the key components our model.

Table 2.4: Performance comparison of our model and other state-of-the-art GCN-based methods. Our proposed GS-Net method achieves the best performance, as indicated by boldface numbers. All errors are measured in millimeters (mm).

Method	MPJPE (↓)	PA-MPJPE (↓)
SemGCN [30]	42.14	33.53
High-order GCN [31]	39.52	31.07
HOIF-Net [33]	38.12	29.74
Modulated GCN [34]	38.25	30.06
Weight Unsharing [32]	37.83	30.09
Ours	**37.41**	**28.94**

2.3.3 Ablation Study

In order to better understand the impact of the design choices made for our network architecture, we perform ablation experiments on Human3.6M under controlled settings. We report the results using 2D ground truth for training and testing to eliminate the added uncertainty from 2D pose detectors. These ablation studies demonstrate the efficacy of the key components of our model.

Impact of Skip Connection. We analyze how the weighted initial skip connection in the layer-wise propagation rule affects the model performance, and we report the results in Table 2.5. We can see that our model benefits from the weighted initial skip connection, yielding relative error reductions of 3.43% and 2.91% in terms of MPJPE and PA-MPJPE, respectively.

Table 2.5: Effectiveness of initial skip connection (ISC). Boldface numbers indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o ISC	38.74	29.79
w ISC	**37.41**	**28.94**

Impact of Residual Block Design. We compare two residual block designs and report the results in the Table 2.6. The first design employs blocks consisting of convolutional layers, followed by batch normalization (BatchNorm) and a ReLU activation function. By contrast, the second design uses blocks comprised of convolutional layers, followed by layer normalization (LayerNorm) and a GELU activation function. We incorporate the latter design into the proposed architecture, resulting in improved performance compared to the first design. The results show that our model with ConvNeXt block achieves a 0.63% decrease in error in terms of the MPJPE metric and yields comparable performance in terms of the PA-MPJPE metric. In this part of the ablation study,

we supply 2D keypoints obtained from the 2D pose detector as input to our model to examine the model reliability. We also include the pose refinement module and the non-local layer in the network.

Table 2.6: Effect of residual block design on the performance of our model. Lower values in boldface indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
Ours w/ BatchNorm and ReLU	47.40	**38.62**
Ours w/ LayerNorm and GELU	**47.10**	38.65

Impact of Pose Refinement. We also scrutinize the effectiveness of the pose refinement network. The results in Table 2.7 show that on average the MPJPE and PA-MPJPE errors are reduced by 3.74mm and 1mm, respectively, demonstrating the advantage of using pose refinement in achieving better performance under both protocols. To reinforce our claim, we report our findings in Figure 2.5 that shows the performance comparison with or without the pose refinement model under Protocol #1 (top) and Protocol #2 (bottom) for various challenging actions such as "Eating" and "Photo". For example, relative error reductions of 10.62% and 5.98% are achieved for the "Eating" action in terms of MPJPE and PA-MPJPE, respectively.

Table 2.7: Effectiveness of the pose refinement network (PRN). Boldface numbers indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o PRN	37.41	28.94
w PRN	**33.67**	**27.94**

Impact of Symmetrizing Adjacency Modulation. As reported in Table 2.8, symmetrizing the modulated adjacency modulation matrix helps reduce the MPJPE and PA-MPJPE errors by 1.25mm and 0.48mm, respectively.

Table 2.8: Effectiveness of symmetrizing adjacency modulation. Boldface numbers indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o Symmetry	38.66	29.42
w Symmetry	**37.41**	**28.94**

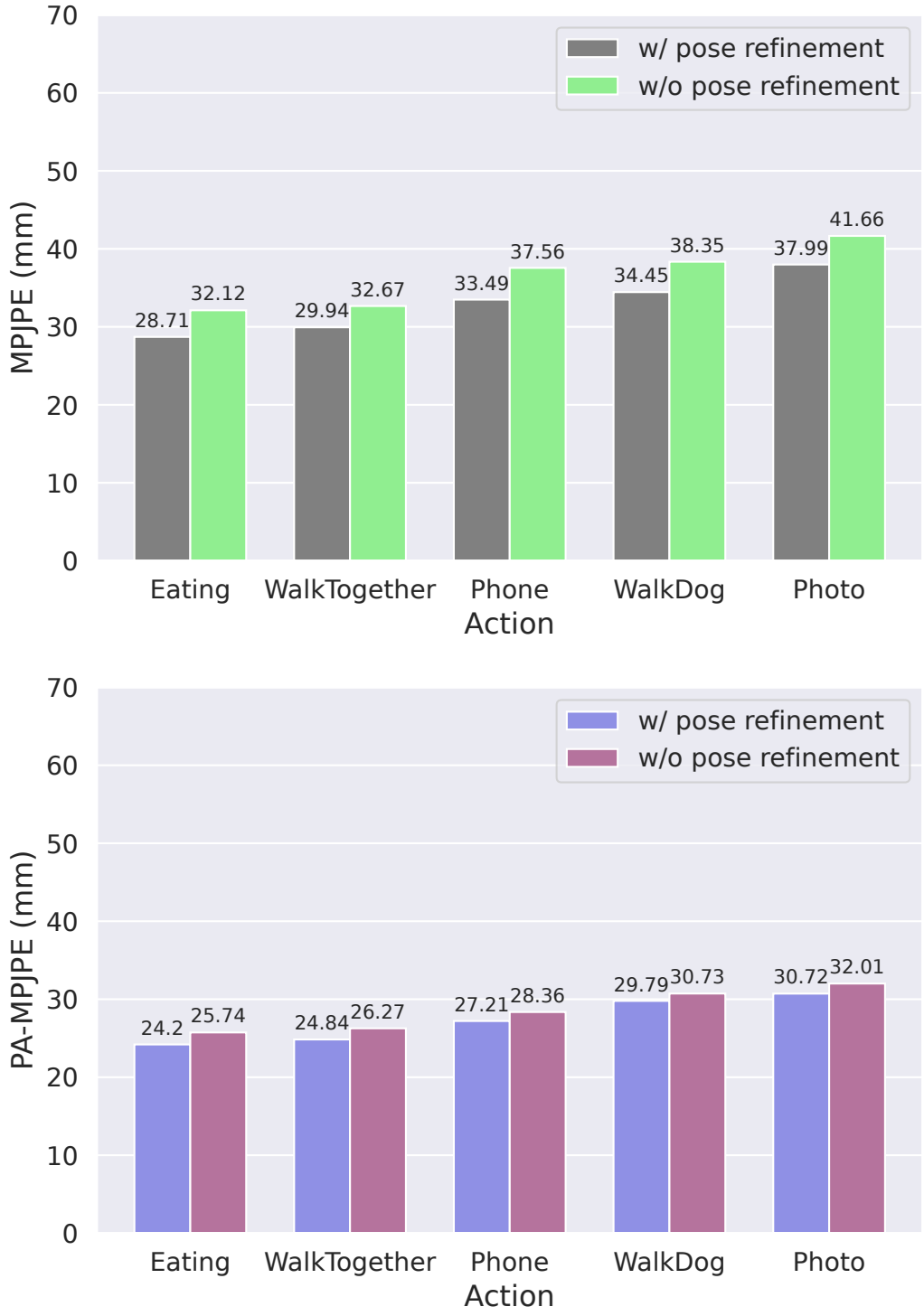


Figure 2.5: Performance comparison of our model with and without pose refinement using MPJPE (top) and PA-MPJPE (bottom). When coupled with a pose refinement network, GS-Net performs consistently better on challenging actions.

Impact of Loss Functions. The results in Table 2.9 validate our design decision to adopt a weighted loss function comprised of ℓ_1 and ℓ_2 penalty terms. It is evident that using the weighted sum of both penalty terms results in better performance in terms of MPJPE and PA-MPJPE. This better performance is largely attributed to the fact that using a weighted sum of both loss functions helps balance the trade-offs between robustness and sensitivity to smaller errors. In other words, we can take advantage of the robustness of ℓ_1 loss while still providing some level of sensitivity to large errors.

Table 2.9: Effectiveness of the loss functions. Boldface numbers indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
Only ℓ_1 loss	37.43	29.29
Only ℓ_2 loss	37.62	29.25
Weighted sum of ℓ_1 and ℓ_2 losses	**37.41**	**28.94**

Impact of Batch/Filter Size. In Figure 2.6, we analyze the effect of varying batch and filter sizes on our model performance. We can see that a batch size of 512 and a filter size of 384 yield the best results in terms of MPJPE and PA-MPJPE, respectively.

Hyperparameter Sensitivity Analysis. We also examine the influence of the Laplacian regularization hyperparameter β on model performance by plotting the error metrics vs. β for a range of values in the interval $(0, 1)$. This hyperparameter controls the strength of the Laplacian regularization term in the objective function. It determines how much emphasis is placed on the regularization term relative to the data fidelity term in the objective function. A larger value of the regularization parameter corresponds to a stronger regularization effect, places more emphasis on smoothness in the filtered feature vectors of neighboring graph nodes, but at the expense of how well the filtered feature vectors match the initial feature vectors. Conversely, a smaller value of β places more emphasis on the data fidelity term and results in a smaller error between the initial and filtered feature vectors. As shown in Figure 2.7, the best results are typically obtained when the regularization parameter is small. We can observe that our model achieves the lowest error values of MPJPE and PA-MPJPE when $\beta = 0.2$, respectively.

2.3.4 Runtime Analysis

We also analyze the model's inference time, which is a crucial factor in determining the efficiency of the performance of our proposed approach. By examining the inference time, we aim to understand the speed at which our model processes and generates the output. The inference time results are reported in Table 3.11, which shows that our model significantly outperforms strong baselines.

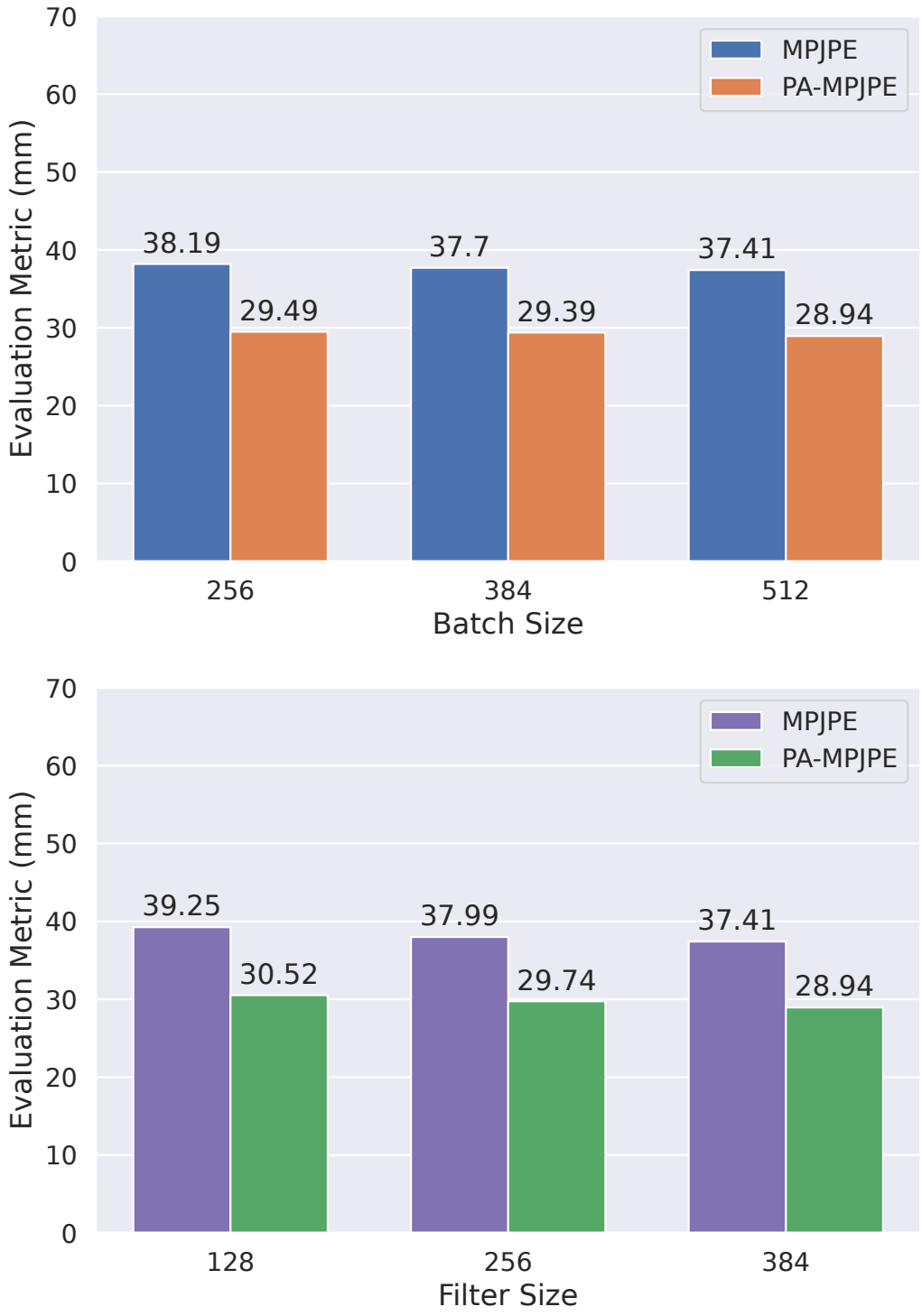


Figure 2.6: Performance of our proposed GS-Net model on the Human3.6M dataset using varying batch and filter sizes.

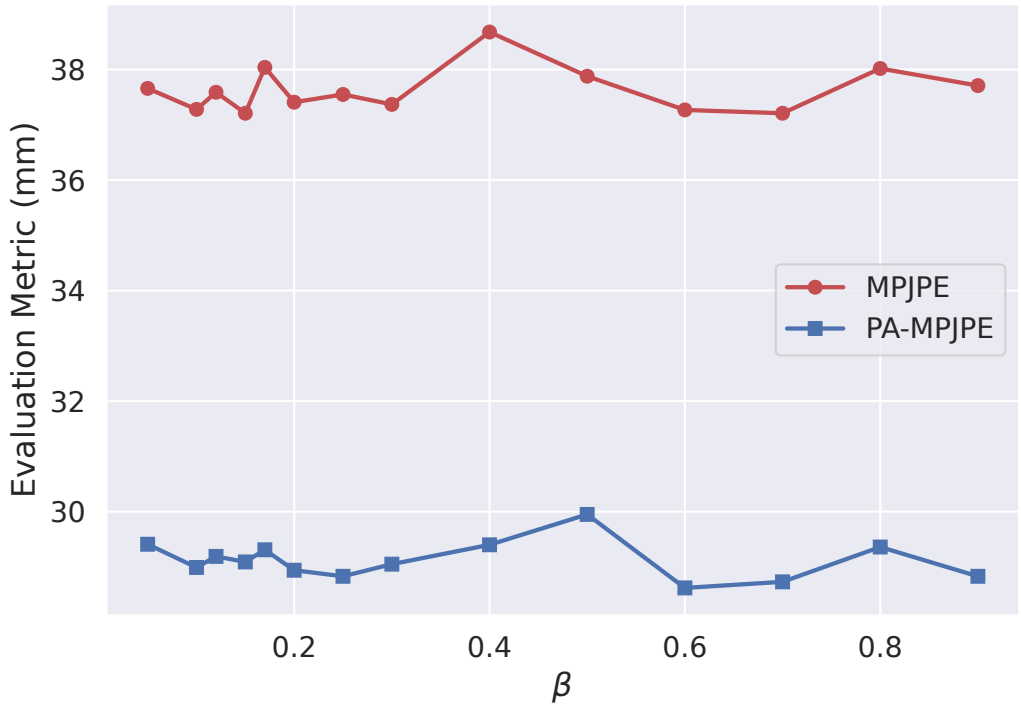


Figure 2.7: Sensitivity analysis of our model to the choice of the Laplacian regularization hyper-parameter β. Smaller values of β generally result in loweer MPJPE and PA-MPJPE errors.

Table 2.10: Runtime analysis of our model in comparison with competing baselines.

Method	Inference Time
SemGCN [30]	.012s
High-Order GCN [31]	.013s
HOIF-Net [33]	.016s
Weight Unsharing [32]	.032s
MM-GCN [36]	.009s
Modulated GCN [34]	.010s
Ours	.003s

CHAPTER 3

Multi-hop Graph Transformer Network for 3D Human Pose Estimation

In this chapter, we introduce a novel approach called Multi-hop Graph Transformer Network (MGT-Net), which combines the strengths of multi-head self-attention and multi-hop graph convolutional networks with disentangled neighborhoods to capture spatio-temporal dependencies and handle long-range interactions. Disentangling the neighborhoods helps remove redundant dependencies between further and closer neighborhoods, allowing the model to effectively capture long-range dependencies between graph nodes. The proposed network architecture consists of two main blocks: a graph attention block composed of stacked layers of multi-head self-attention and graph convolution with learnable adjacency matrix, and a multi-hop graph convolutional block comprised of multi-hop convolutional and dilated convolutional layers. The combination of multi-head self-attention and multi-hop graph convolutional layers enables the model to capture both local and global dependencies, while the integration of dilated convolutional layers enhances the model's ability to handle spatial details required for accurate localization of the human body joints. Extensive experiments demonstrate the effectiveness and generalization ability of our model, achieving state-of-the-art performance on benchmark datasets while maintaining a compact model size.

3.1 Introduction

3D human pose estimation is a fundamental and challenging task in computer vision, with a myriad of applications spanning action recognition [10, 81], autonomous driving [82], sports analyt-

ics [83], and healthcare diagnostics [84]. At its core, it involves the prediction of 3D coordinates of human body joints from images or videos with the goal of understanding human movements and interactions. In healthcare, for instance, 3D human pose estimation can be used to monitor patient rehabilitation and assess physical therapy exercises, thereby enabling healthcare professionals to offer tailored treatments, track recovery, and enhance the quality of care provided to patients.

With the advent of deep learning, several deep neural networks [14–21, 41–43] have been devised for 3D human pose estimation, classified into one-stage and two-stage approaches. One-stage methods aim to directly regress the 3D pose from input images, whereas two-stage methods first predict intermediate representations, such as 2D joint locations, using 2D pose detectors before lifting them to 3D space. Two-stage methods typically exhibit better performance compared to one-stage approaches, particularly when combined with robust 2D joint detectors, as they improve the accuracy of 3D pose estimation while addressing depth ambiguity challenges. Despite notable progress [13], several challenges persist in 3D human pose estimation. Self-occlusions often occur when body parts obscure each other, making it difficult for the model to accurately estimate the position of occluded joints. Moreover, depth ambiguity arises due to variations in body shape, occlusions, and self-occlusions, leading to multiple 3D pose possibilities for the same 2D projections. Overcoming these challenges remains critical for improving the robustness and accuracy of 3D human pose estimation methods.

In recent years, graph convolutional network (GCN)-based methods [30, 37, 85] and approaches built on the Transformer architecture [3, 40] have become more prevalent in 3D human pose estimation. While effective, GCN-based methods are limited in their ability to capture dependencies between body joints that are beyond immediate neighbors. The standard GCN architecture, which relies on the adjacency matrix to propagate information, considers only the direct connections between nodes in the graph, resulting in a relatively local view of the graph structure. As a consequence, these methods may not fully exploit the long-range interactions and complex dependencies that exist in human body movements. To address this challenge, various approaches [31, 33] employ high-order graph convolutions, which leverage higher powers of the adjacency matrix, thereby allowing information to be propagated through multiple hops in the graph. By using higher powers of the adjacency matrix, the model can gather information from not only the immediate neighbors but also nodes that are further away in the graph, enhancing its ability to consider global context and complex relationships between body joints. Using higher powers can, however, lead to the biased weighting problem [86]. This issue arises due to the nature of undirected graphs, which can contain cyclic walks. In such graphs, edge weights can be influenced by the number of hops or steps required to traverse from one node to another. As a result, the weights tend to be biased

towards nodes that are closer in terms of hop count, and this bias can lead to an overemphasis on local connections while neglecting long-range dependencies. To mitigate this issue, we incorporate disentangled neighborhoods in our proposed framework so that our model can capture information from nodes that are further away without being overly influenced by local connections. This helps enhance the ability of our model to capture both short-range and long-range dependencies effectively. Also, GCN-based methods are constrained by the limited receptive fields. To address this limitation, we incorporate dilated graph convolutions into our model architecture. By adjusting the dilation rate of the convolutional kernels, our model can capture information from a wider region of the graph, effectively incorporating global contextual information into the feature representation.

On the other hand, Transformer-based methods [3, 40] leverage self-attention mechanisms [38] to process a sequence of 2D joint locations across frames in a video, enabling them to efficiently model dependencies between different body joints in the sequence. By attending to relevant joints in the sequence, Transformers can effectively capture the temporal and spatial relationships [3] between body joints, leading to improved pose estimation accuracy. Although self-attention mechanisms enable the body joints to interact by capturing global visual information through long-range dependencies and contextual information, they often neglect the inherent graph structure information among the joints (i.e., the adjacency relation of joints). By neglecting the graph structure, Transformer-based methods might overlook the specific dependencies between joints, potentially leading to suboptimal predictions. To overcome this limitation, we combine the strengths of Transformer-based methods with GCNs by incorporating multi-hop graph convolutions with the aim of explicitly taking advantage of the graph structure information, allowing our model to better capture the relationships between body joints from various hop distances. We achieve this integration through a two-pathway design, where one pathway handles the self-attention mechanisms of the Transformer, capturing global visual information, and the other pathway performs graph convolutions to consider the graph structure information and interactions between joints. Specifically, we present a novel approach, dubbed Multi-hop Graph Transformer Network (MGT-Net), whose architecture is comprised of two main building blocks: a graph attention block and a multi-hop graph convolutional block. The graph attention block is composed of a multi-head self-attention layer that allows each body joint in the skeleton graph to attend to its neighboring joints and learn relationships between them, and a graph convolutional layer with learnable adjacency matrix. By learning the adjacency matrix, our model can adaptively adjust the connections between the body joints, allowing it to capture complex dependencies in the graph. The multi-hop graph convolutional block, on the other hand, consists of multi-hop graph convolutional layers with disentangled neighborhoods that enable the model to capture dependencies between nodes at varying hop dis-

tances and dilated convolutional layers to enhance the model's receptive field without increasing the number of learnable parameters. One of the key strengths of our proposed model is its simplicity, which stands in contrast to many existing spatio-temporal approaches for 3D human pose estimation. Despite its straightforward design, our model achieves competitive performance, surpassing strong baseline methods in accuracy, as demonstrated in Figure 3.1. This is particularly noteworthy as our model maintains a compact model size, making it computationally efficient. In summary, we make the following key contributions:

- We propose a multi-hop graph transformer network (MGT-Net), which leverages multi-head self-attention, multi-hop graph convolutions with disentangled neighborhoods, and dilated convolutions to effectively capture both long-range dependencies and local-global contextual information.
- We design a network architecture composed of a graph attention block and a multi-hop graph convolutional block to capture spatial dependencies and model the complex interactions among body joints.
- We demonstrate through extensive experiments and ablation studies the effectiveness and generalization ability of the proposed MGT-Net model, achieving competitive performance in 3D human pose estimation on two benchmark datasets while retaining a small model size.

The remainder of this chapter is structured as follows. In Section 3.2, we present our proposed framework, which encompasses the problem formulation, the main building blocks of our network architecture, and model training. In Section 3.3, we present empirical results comparing our model with state-of-the-art approaches on two standard benchmarks.

3.2 Proposed Method

In this section, we begin by defining the learning task of 3D human pose estimation, which involves predicting the accurate 3D joint positions of a human body given 2D poses. We then delve into GCNs and their high-order extensions, followed by introducing a multi-hop GCN with disentangled neighborhoods, which is an extension of GCN that addresses the challenges of biased weighting and redundant dependencies between different neighborhoods. The aim is to enhance the modeling capabilities of GCNs by incorporating multi-scale aggregation and disentangling neighborhood features, thereby enabling more comprehensive and effective representation of joint relationships in the human skeleton. Subsequently, we present the main building blocks of the proposed network architecture.

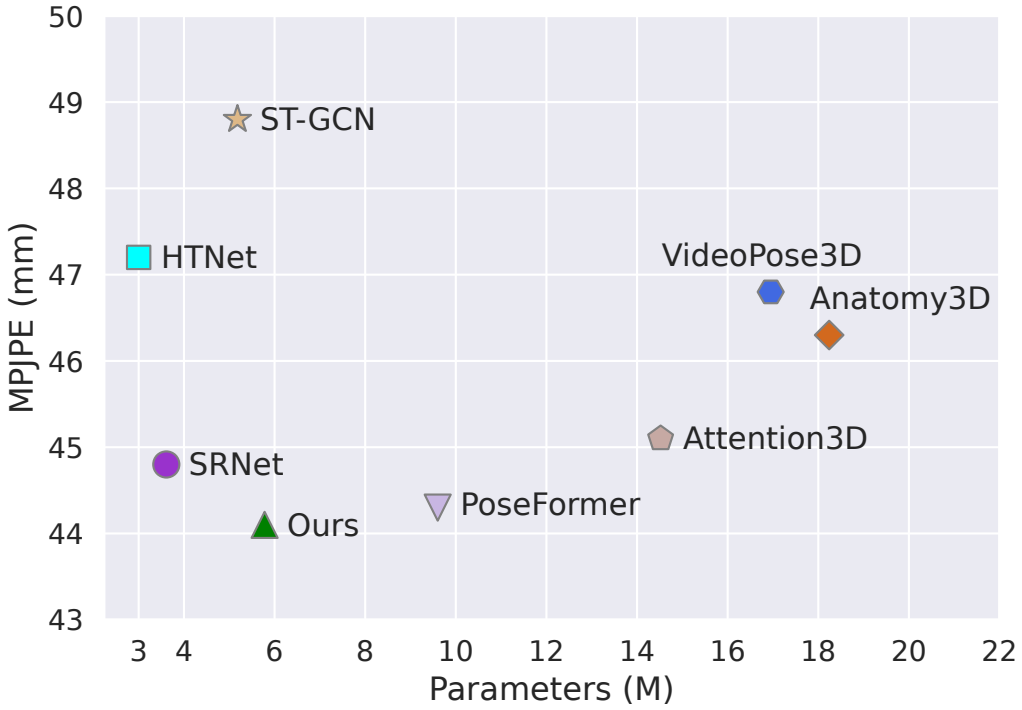


Figure 3.1: Performance and model size comparison between our model and state-of-the-art temporal methods for 3D human pose estimation, including PoseFormer [3], VideoPose3D [4], ST-GCN [5], SRNet [6], Attention3D [7], Anatomy3D [8], and HTNet [9]. Lower Mean Per Joint Position Error (MPJPE) values indicate better performance. Evaluation is conducted on the Human3.6M dataset using detected 2D joints as input.

3.2.1 Preliminaries and Problem Statement

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 $\mathbf{X}$ is an $N \times F$ feature matrix of node attributes whose i-th row is an F-dimensional feature vector associated to node i. The graph structure is encoded by an $N \times N$ adjacency matrix $\mathbf{A}$ whose (i, j)-th entry is equal to 1 if there the edge between neighboring nodes i and j, and 0 otherwise. We denote by $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ the normalized adjacency matrix with self-added loops, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{I}$ is the identity matrix, $\tilde{\mathbf{D}} = \mathrm{diag}(\tilde{\mathbf{A}}\mathbf{1})$ is the diagonal degree matrix, and $\mathbf{1}$ is an N-dimensional vector of all ones.

Problem Formulation. 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

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

where $\alpha \in [0, 1]$ is a weighting factor that controls the mixing amount between the ℓ_2 and ℓ_1 penalties. It is worth pointing out that the proposed loss function draws inspiration from the penalty function used in the elastic net regression model [65], which is a weighted combination of lasso and ridge regularization. When $\alpha = 0$, the penalty function is equivalent to lasso regression, and when $\alpha = 1$, it is equivalent to ridge regression. A key advantage of elastic net regression is that it reduces the impact of irrelevant predictors by shrinking their coefficients towards zero, unlike ridge regression which only reduces the size of the coefficients.

3.2.2 Multi-hop Graph Convolutional Networks

Graph convolutional networks and their variants have emerged as a promising approach for 3D human pose estimation, addressing the challenges of modeling complex spatial relationships and capturing contextual dependencies between body joints. By representing the human skeleton as a graph and leveraging graph convolution operations, these networks can effectively learn joint interactions and infer accurate 3D pose estimations.

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

$$\mathbf{H}^{(\ell+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}), \quad \ell = 0, \ldots, L-1, \tag{3.3}$$

where $\hat{\mathbf{A}}$ is the normalized adjacency matrix with self-added loops, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{F_\ell \times F_{\ell+1}}$ is a trainable weight matrix with $F_{\ell+1}$ feature maps, and L is the number of layers. The GCN propagation rule can be interpreted as follows: The input node features $\mathbf{H}^{(\ell)}$ are transformed by the trainable weight matrix $\mathbf{W}^{(\ell)}$. Then, the transformed features are multiplied by the normalized adjacency matrix $\hat{\mathbf{A}}$ to aggregate information from neighboring nodes. The normalized adjacency matrix represents the

connections between nodes and is scaled by the degree of each node to ensure stability and effective information propagation. Finally, an activation function $\sigma(\cdot)$ such as $\text{ReLU}(\cdot) = \max(0, \cdot)$ is applied to introduce non-linearity to the aggregated features. The input of the first layer is the initial feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$. By applying the propagation rule iteratively for multiple layers, the GCN can capture and propagate information through the graph, enabling it to learn meaningful node representations.

Revisiting High-order GCNs. The aggregation scheme of GCN is limited to using immediate neighboring nodes as illustrated in Figure 3.2 (left), which means it only considers direct connections between nodes. Hence, it fails to capture long-range dependencies that may exist between nodes that are further apart in the graph. To address this limitation, high-order GCNs are often employed [31, 33], as they extend the aggregation scheme to K-hop neighbors, allowing for the capture of long-range dependencies between nodes in the graph via the following update rule:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\sum_{k=0}^{K} \hat{\mathbf{A}}^k \mathbf{H}^{(\ell)} \mathbf{W}_k^{(\ell)}\right), \tag{3.4}$$

where $\hat{\mathbf{A}}^k$ is the k-th power of the normalized adjacency matrix whose (i, j)-th element counts the number of walks of length k between nodes i and j, and $\mathbf{W}_k^{(\ell)}$ is a learnable weight matrix associated with $\hat{\mathbf{A}}^k$. The update rule sums the contributions from different powers of the normalized adjacency matrix, ranging from $k = 0$ to $k = K$, where K represents the maximum number of hops considered. This summation allows the model to capture information from neighboring nodes up to K hops away. By considering multiple hops, high-order GCNs can incorporate information from more distant nodes and enhance their ability to capture global context and relationships within the graph structure.

While high-order GCNs are effective at capturing long-range dependencies, they are, however, prone to the biased weighting problem. This issue arises because undirected graphs can have cyclic walks, which can result in edge weights being biased towards closer nodes compared to further nodes, leading to preferential emphasis on local connections and neglecting long-range dependencies. Also, as the value of k increases, the influence of each node's features spreads to increasingly distant neighbors. This can result in oversmoothing, where the node representations become overly similar and lose local discriminative information. In addition, computing higher powers of the normalized adjacency matrix can be computationally expensive, especially for large graphs. The computational complexity grows exponentially with the power of the normalized adjacency matrix, limiting the scalability of the approach. To address these limitations, we employ a multi-hop GCN aggregation scheme that leverages the k-adjacency matrix with the aim of removing redundant dependencies between node features from different neighborhoods [86]. In

contrast to the k-th power of the normalized adjacency matrix that focuses on the number of walks of length k, the k-adjacency matrix emphasizes the direct connections within k hops, as depicted in Figure 3.2 (right).

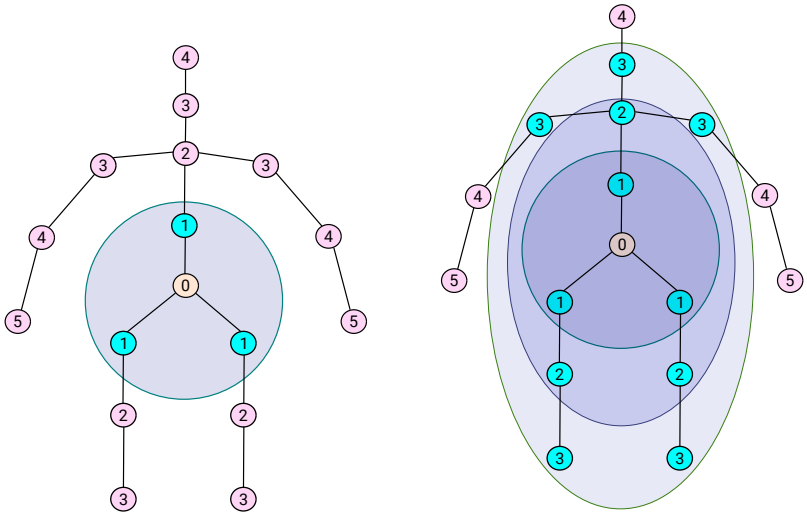


Figure 3.2: Visual comparison between the standard graph convolution, which only considers the 1-hop neighbors, and the multi-hop graph convolution, which takes into account neighbors at different distances. The node label $k \in \{0, \ldots, 5\}$ indicates that the corresponding body joint is a k-hop neighbor of the pelvis (i.e., root node denoted by 0).

Multi-hop GCNs with Disentangled Neighborhoods. We aim to address the biased weighting problem and capture long-range dependencies more effectively. To this end, we leverage the k-adjacency matrix $\tilde{\mathbf{A}}_k$ whose (i, j)-th element is given by

$$[\tilde{\mathbf{A}}_k]_{ij} = \begin{cases} 1 & \text{if } d_{ij} = k, \\ 1 & \text{if } i = j, \\ 0 & \text{otherwise}, \end{cases} \tag{3.5}$$

where d_{ij} denotes the shortest distance in the number of hops between nodes i and j. It is important to note that incorporating self-loops in the k-adjacency matrix allows each joint to have a connection with itself, ensuring that the identity information of each joint is preserved. This is particularly important when dealing with nodes that do not have k-hop neighbors. The self-loop ensures that each joint's features are taken into account during the aggregation process, even if it does not have any direct neighbors within the specified hop distance. Note that $\tilde{\mathbf{A}}_k$ can be seen as an extension of the standard adjacency matrix $\tilde{\mathbf{A}}$ with self-added loops to incorporate relationships beyond immediate neighbors. While $\tilde{\mathbf{A}}$ captures the connections between directly connected nodes, the k-adjacency matrix $\tilde{\mathbf{A}}_k$ considers relationships up to a distance of k hops, thereby capturing the relationships between neighboring joints up to a distance of k hops.

Using the k-adjacency matrix, the layer-wise propagation rule of the multi-hop GCN can be defined as follows:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\sum_{k=0}^{K} \hat{\mathbf{A}}_k \mathbf{H}^{(\ell)} \hat{\mathbf{W}}_k^{(\ell)}\right), \tag{3.6}$$

where $\hat{\mathbf{A}}_k = \tilde{\mathbf{D}}_k^{-\frac{1}{2}} \tilde{\mathbf{A}}_k \tilde{\mathbf{D}}_k^{-\frac{1}{2}}$ is the normalized k-adjacency matrix, and $\tilde{\mathbf{D}}_k = \mathrm{diag}(\tilde{\mathbf{A}}_k \mathbf{1})$ is the associated diagonal degree matrix. The multiplication $\hat{\mathbf{A}}_k \mathbf{H}^{(\ell)} \hat{\mathbf{W}}_k^{(\ell)}$ performs the graph convolution operation, where the normalized k-adjacency matrix $\hat{\mathbf{A}}_k$ is applied to the input features $\mathbf{H}^{(\ell)}$ and $\hat{\mathbf{W}}_k^{(\ell)}$ is is the associated weight matrix. The result is then passed through the activation function $\sigma(\cdot)$ to introduce non-linearity. The updated feature matrix $\mathbf{H}^{(\ell+1)}$ is obtained by aggregating information from neighboring nodes up to K hops away, using the weighted sum of the convolved features. In fact, each term $\hat{\mathbf{A}}_k \mathbf{H}^{(\ell)}$ represents the aggregation of information from nodes within k hops of each node, weighted by the normalized k-adjacency matrix. This allows the model to capture multi-hop dependencies and incorporate information from distant nodes in the graph during the convolutional operation, thereby capturing both local and long-range dependencies in the graph structure. By disentangling the features under multi-hop aggregation, we ensure that the weights assigned to further neighborhoods are not biased due to their distance from the central node [86]. This disentanglement process allows our model to effectively capture graph-wide joint relationships on human skeletons, leading to more accurate modeling of long-range dependencies and improving the performance of GCNs in capturing complex relationships between body joints.

The key advantage of using the k-adjacency matrix is that it allows for flexible modeling of long-range relationships between body joints. Also, one important characteristic of the k-adjacency matrices is that the relationships they represent have low correlations with each other. This means that the relationships captured by the k-adjacency matrices of different hops are distinct and provide unique information. As a result, the k-adjacency matrix structure enables the modeling of diverse and long-range relationships between body joints, even in cases where direct neighbors might not exist or are limited within the specified hop distance. By considering further neighborhoods, the k-adjacency matrix provides a broader view of the graph structure and captures more distant dependencies between nodes. It allows for modeling long-range relationships and capturing complex interactions among nodes that may not be apparent in the standard adjacency matrix with self-added loops.

By incorporating additional hops with larger values of k, the aggregation rule of the multi-hop GCN aggregates information in an additive manner, allowing for effective modeling of long-range dependencies, as the model can capture relationships between joints that are further apart. In terms of sparsity, the k-adjacency matrix is generally more sparse than the k-th power of the normalized

adjacency matrix, especially for large values of k as shown in Figure 3.3. In general, as k increases, the number of non-zero entries in the k-th power of the adjacency matrix tends to increase, potentially leading to a denser representation. Hence, the sparsity property of the k-adjacency matrix is beneficial as it reduces the computational complexity and memory requirements of the model, resulting in more efficient representations.

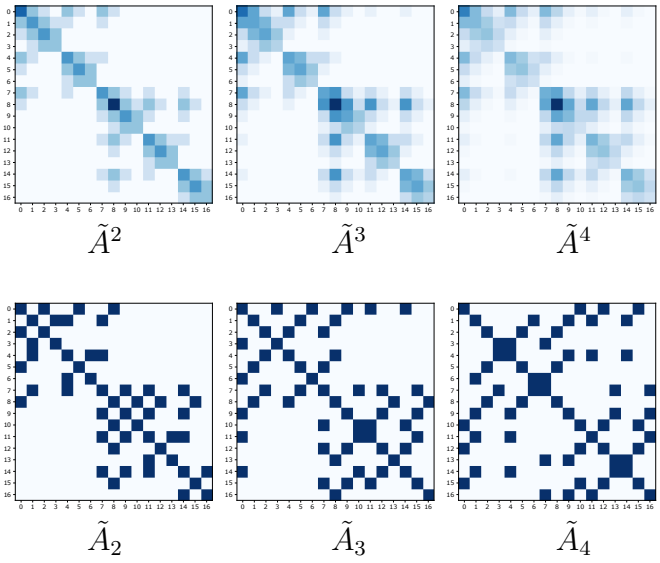


Figure 3.3: Comparing the sparsity of the k-th power of the adjacency matrix (top row) and the k-adjacency matrix (bottom row). As the value of k increases, the k-th power representation tends to become denser, while the k-adjacency matrix maintains higher sparsity. The sparsity of the k-adjacency matrix makes it an efficient choice for capturing long-range dependencies in the multi-hop GCN with disentangled neighborhoods, reducing computational complexity and memory usage.

3.2.3 Multi-hop Graph Transformer Network

The overall framework of our proposed MGT-Net is shown in Figure 3.4. The network architecture is comprised of three main components: skeleton embedding, a graph attention block and a multi-hop graph convolutional block. The input is a sequence of 2D human poses, which are obtained using an off-the-shelf 2D detector [43], and the output is a 3D human pose. The multi-head self-attention mechanism enables our model to capture long-range dependencies and encode global context information, while multi-hop graph convolutions leverage the graph structure of body joints to exchange information between non-neighboring joints, allowing for the modeling of higher-order relationships.

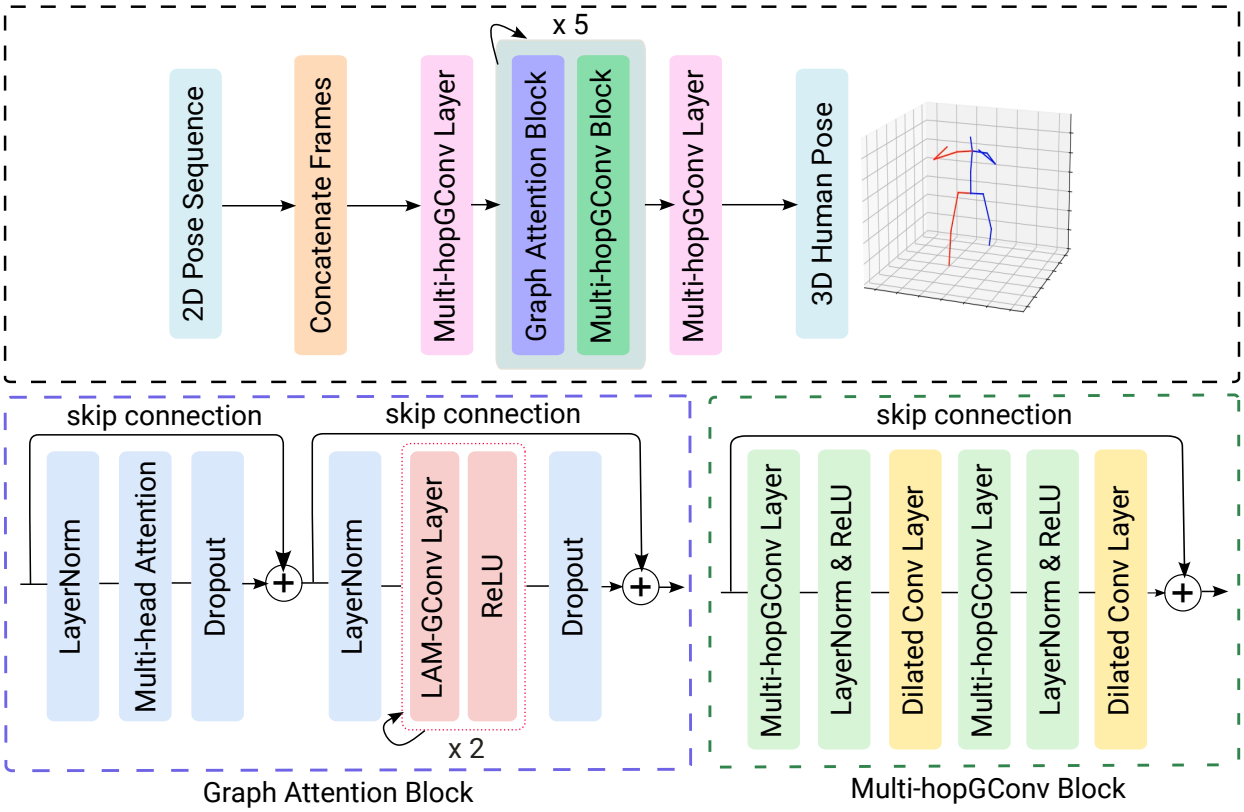


Figure 3.4: Network architecture of the proposed MGT-Net for 3D human pose estimation. Our model takes a sequence of 2D pose coordinates as input and generates 3D pose predictions as output. The core building blocks of the network are a graph attention block and a multi-hop graph convolutional block, which are stacked together. We use a total of five layers for these stacks. In the graph attention block, the multi-head attention layer is followed by two consecutive graph convolutional layers with learnable adjacency matrix (LAM-GConv). The multi-hop graph convolutional block is composed of two subblocks, each of which comprises a multi-hopGConv layer, followed by a dilated convolutional layer.

Skeleton Embedding

In order to incorporate temporal information into our model, we take a 2D pose sequence as input. Specifically, given a 2D pose sequence $\mathbf{S} \in \mathbb{R}^{N\times 2\times T}$ represented as a tensor, where T denotes the number of frames and N is the number of joints, we first reshape it into a matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{N\times 2T}$ by concatenating the 2D coordinates of all frames along the time axis. This allows us to consider the 2D coordinates of all frames for each joint as a continuous sequence, effectively capturing the temporal dependencies and changes of the joints over time. Then, we pass it through a multi-hop graph convolution layer, resulting in an $N \times F$ embedding matrix $\mathbf{X}$ of joint attributes whose i-th row is an F-dimensional feature vector associated to the i-th joint of the skeleton graph. This skeleton embedding matrix serves as a fundamental representation that incorporates both spatial

and temporal information from the 2D pose sequence. It forms the basis for subsequent components of our network architecture, facilitating the integration of both spatial and temporal context into the model.

Graph Attention Block

The graph attention block consists of stacked layers of multi-head self-attention and graph convolution, leveraging a learnable adjacency matrix. This block allows our model to capture both global and local dependencies within the graph structure. The multi-head self-attention mechanism enables the model to weigh the importance of different nodes, while the graph convolutional layers help propagate information across neighboring nodes.

Multi-head Self Attention. At their core, Transformers [38] rely on self-attention, which allows the model to weigh the importance of different tokens within a given input sequence when making predictions. Self-attention operates on the embedding matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ consisting of N feature vectors, each of which has an embedding dimension F. The input matrix $\mathbf{X}$ is first linearly projected into a query matrix $\mathbf{Q}$, a key matrix $\mathbf{K}$ and a value matrix $\mathbf{V}$ as follows:

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_k, \mathbf{V} = \mathbf{X}\mathbf{W}_v, \tag{3.7}$$

where $\mathbf{W}_q \in \mathbb{R}^{F \times d_k}$, $\mathbf{W}_k \in \mathbb{R}^{F \times d_k}$ and $\mathbf{W}_v \in \mathbb{R}^{F \times d_k}$ are learnable weight matrices, and d_k is the projection dimension. Then, the self-attention (SA) output is the weighted sum of the value vectors for each token

$$\mathrm{SA}(\mathbf{X}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_k}}\right)\mathbf{V} \in \mathbb{R}^{N \times d_k}, \tag{3.8}$$

where the weights are attention scores computed as the dot product between the query and key vectors, scaled by the square root of the projection dimension, and followed by softmax applied row-wise. These attention weights determine the importance of each value vector, enabling the model to capture contextual dependencies. Scaling the dot product between the query and key vectors by $\sqrt{d_k}$ helps stabilize the attention weights and prevents them from becoming too large, which can lead to numerical instability during training. This ensures that the attention weights are more balanced and can better capture the relationships between different feature vectors.

In practice, Transformers employ multi-head attention to capture different types of relationships and patterns in the input sequence. This involves performing multiple self-attention operations in parallel, each with its own set of weight matrices, allowing the model to capture different types of dependencies and attending to different parts of the input sequence. For simplicity, we assume $d_k = F/h$, where h is the number of attention heads (i.e., self-attention operations). The outputs

of h heads are concatenated and linearly transformed using an $F \times F$ learnable weight matrix $\mathbf{W}_o$ to obtain the output of the multi-head self-attention (MSA) as follows:

$$\text{MSA}(\mathbf{X}) = \text{Concat}(\mathbf{Y}_1, \ldots, \mathbf{Y}_h)\mathbf{W}_o, \tag{3.9}$$

where $\mathbf{Y}_i = \text{SA}_i(\mathbf{X}) \in \mathbb{R}^{N \times \frac{F}{h}}$ is the output of the i-th attention head. Hence, the final output of concatenated attention heads is $\mathbf{Y} = \text{MSA}(\mathbf{X}) \in \mathbb{R}^{N \times F}$.

Taking inspiration from the architectural design of Graformer [40], our graph attention block is composed of a multi-head attention layer and two graph convolutional layers. Each convolutional layer is equipped with a learnable adjacency matrix (LAM-GConv), which enables the model to adaptively learn the relationships between joints, making the graph convolution operation more flexible and effective in capturing local interactions. The combination of the multi-head attention layer and the LAM-GConv layers in our graph attention block allows for the integration of both global and local information, enhancing the model's ability to exploit the relationships among joints and capture important features for 3D human pose estimation. The output $\mathbf{Y}$ of the multi-head self-attention is passed through a LAM-GConv layer, which applies graph convolution to aggregate information from neighboring joints based on the learnable adjacency matrix. This step helps refine and update the joint features by considering their relationships with adjacent joints. We also apply a layer normalization layer (LayerNorm) to normalize the features across the joint dimension and a dropout layer to prevent overfitting and improve generalization. Moreover, a skip connection is employed to facilitate information flow, enabling the model to preserve important features from earlier stages of processing and pass them directly to subsequent layers.

Multi-hop Graph Convolutional Block

The multi-hop graph convolutional block combines multi-hop convolutional and dilated convolutional layers. This block facilitates the modeling of long-range dependencies and captures spatial relationships across different hops in the graph. A key benefit of using the dilated convolutional layer is that it can capture contextual information from a wider context thanks, in large part, to its larger receptive field. This is particularly useful in tasks where long-range dependencies and global patterns are important, such as 3D human pose estimation.

Multi-hop Graph Convolution. The multi-hop graph convolutional layer exchanges information between non-neighboring nodes, enabling the model to capture broader context and larger receptive fields, incorporating the influence of distant joints on the pose estimation. This is particularly important for understanding the global structure and relationships among different parts of the graph, even when they are not directly connected in the graph representation. Also, the ability

to aggregate information from different hops allows the model to effectively learn and represent the hierarchical structure of the pose, capturing both local details and global patterns. Moreover, by focusing on distinct and non-redundant neighborhoods, the multi-hop graph convolutional layer with disentangled neighborhoods can better discern meaningful information and discard irrelevant signals.

Dilated Convolution. In traditional convolutional networks, the receptive field is determined by the kernel size, which directly affects the size of the local neighborhood that each convolutional operation considers. In contrast, dilated convolutional networks utilize dilated kernels to increase the receptive field of the convolutional layer [87]. Specifically, given a 2D input $\mathbf{X}$ and a kernel filter $\mathbf{w}$ of size $(2m+1) \times (2m+1)$, the dilated convolution, denoted by $*_d$, is defined as

$$(\mathbf{X} *_d \mathbf{w})(i,j) = \sum_{r,s=-m}^{m} \mathbf{X}(i + d \times r, j + d \times s)\mathbf{w}(r,s), \tag{3.10}$$

where d is a dilation rate parameter, which controls the spacing between the kernel elements. By increasing the dilation rate, the kernel can effectively "expand" and cover a larger region of the input feature map. This expansion leads to an increased receptive field without the need for larger kernel sizes.

By applying a dilated convolutional layer after a multi-hop graph convolutional layer, we can leverage the strengths of both operations. The multi-hop graph convolutional layer captures local relationships and structural dependencies among nodes, while the dilated convolutional layer further enhances the receptive field by incorporating larger contextual information without increasing the number of parameters. This is crucial for 3D human pose estimation, as it allows the model to consider the relationships and dependencies between body joints across a broader region, providing a more comprehensive understanding of the pose. Moreover, dilated convolutions are effective at capturing long-range dependencies by considering information from distant nodes of the graph, and also at handling occlusion scenarios where body joints may be temporarily hidden or obscured.

3.2.4 Model Training

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

$$\mathcal{L} = \frac{1}{N}\left[(1-\alpha)\sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1\right], \tag{3.11}$$

which is a weighted sum of the mean squared and mean absolute errors between the 3D ground truth coordinates $\mathbf{y}_i$ and estimated 3D joint coordinates $\hat{\mathbf{y}}_i$ over a training set consisting of N

human poses. These estimated 3D poses are generated by the last multi-hop graph convolutional layer of MGT-Net, as illustrated in Figure 3.4. For the mean squared error, the squared differences between the predicted and ground truth coordinates are averaged, meaning that larger errors have a greater impact on the overall score. In other words, the mean squared error is more sensitive to outliers and penalizes larger errors more heavily than the mean absolute error, which is more robust to outliers and treats all errors equally. The weighting factor α balances the contribution of each loss term. When $\alpha = 0$, our loss function reduces to the mean squared error (i.e., ridge regression) and when $\alpha = 1$, it reduces to the mean absolute error (i.e., lasso regression). The value of the weighting factor is determined by performing a grid search over a range of possible values for α. In our experiments, the best performance on the validation set is achieved with $\alpha = 0.01$.

3.3 Experiments

In this section, we present a comprehensive evaluation of the proposed method by comparing it with competing baselines. We provide details of the experimental setup, including dataset descriptions, evaluation metrics, baselines for comparison, and implementation details to ensure result reproducibility. We also present quantitative and qualitative results, and include ablation studies to assess the model's performance.

3.3.1 Experimental Setup

Datasets. We conduct a comprehensive evaluation of our model on two standard benchmark datasets for 3D human pose estimation: Human3.6M [1] and MPI-INF-3DHP [2]. These datasets provide robust evaluation scenarios for assessing the model performance.

Human3.6M is a large-scale dataset comprised of 3.6 million images that are captured at a rate of 50 Hz using four synchronized cameras positioned at different locations and angles. The dataset features performances by 11 actors (6 men and 5 women) engaged in 15 distinct actions (Directions, Discussion, Eating, Greeting, Phoning, Posing, Purchases, Sitting, Sitting Down, Smoking, Photo, Waiting, Walk Dog, Walking, and Walk Together) within an indoor environment. Precise annotations of 3D body joint coordinates are obtained through a motion capture system, while 2D poses are derived through projection using known intrinsic and extrinsic camera parameters. The annotations for 3D joints are available for 7 of the subjects. The dataset is partitioned into a training set and a test set. The training set encompasses data from five actors (S1, S5, S6, S7, S8), while the test set comprises data from the remaining two actors (S9 and S11). Both sets are balanced, ensuring an equal number of samples for each activity and subject. Prior to feeding the data into

the model, standard normalization techniques [22, 30, 31, 67] are applied to normalize the 2D and 3D poses. The root joint of the 3D poses is designated as the hip joint to achieve zero-centering.

MPI-INF-3DHP is a benchmark dataset for estimating 3D human pose using monocular RGB images. It encompasses a diverse range of environments, including indoor spaces with limited room and complex outdoor scenes. The dataset features recordings of 8 actors (comprising 4 men and 4 women) from 14 different camera views. These actors engage in 8 sets of activities, which encompass a broader spectrum of pose categories compared to the Human3.6M dataset. The activities encompass a variety of movements, ranging from simple actions like walking and sitting to more challenging exercises and dynamic motions. Each activity set lasts approximately one minute, and the actors wear two distinct sets of clothing that are alternated across the activity sets. One clothing set consists of everyday casual wear, while the other set has plain colors to facilitate easy augmentation. Moreover, the dataset provides ground-truth annotations of the 3D joint positions.

Evaluation Protocols and Metrics. We adopt two standard evaluation protocols used for training and testing [22], referred to as Protocol #1 and Protocol #2, for the Human3.6M benchmark. Protocol #1 utilizes the mean per-joint position error (MPJPE) metric, which calculates the average Euclidean distance between the predicted and ground truth 3D positions of each joint after aligning the root joint (hip joint). Another commonly used metric is the Procrustes-aligned mean per-joint position error (PA-MPJPE), which applies Procrustes analysis to align the predicted and ground truth joint positions to a shared coordinate system. This alignment involves scaling, rotating, and translating the predicted joint positions to minimize the sum of squared distances between the predicted and ground truth joint positions. Subsequently, PA-MPJPE is computed by determining the mean Euclidean distances between the aligned predicted and ground-truth joint positions for each joint. Both MPJPE and PA-MPJPE are measured in millimeters (mm), and lower error values indicate better performance. Protocol #1 and Protocol #2 employ five subjects (S1, S5, S6, S7, and S8) for training and two subjects (S9 and S11) for testing, using a single model for all camera views and actions. In the case of the MPI-INF-3DHP dataset, evaluation metrics include the Percentage of Correct Keypoint (PCK) using a threshold of 150mm and the Area Under Curve (AUC) for a range of PCK thresholds [69]. Both PCK and AUC provide measures of how well the predicted joint positions align with the ground-truth joint positions within a specified distance threshold, with higher PCK and AUC scores indicating superior performance.

Baseline Methods. We evaluate the performance of our MGT-Net against various state-of-the-art methods, including SemGCN [30], MöbiusGCN [85], GroupGCN [37], MM-GCN [88], MGCN [34], GraFormer [40], PoseFormer [3], VideoPose3D [4], ST-GCN [5], SRNet [6], Attention3D [7], Anatomy3D [8], GAST-Net [54], Skeletal-GNN [70], and HTNet [9].

Implementation Details. We implement our model in PyTorch and conduct all experiments on a single NVIDIA RTX A4500 with 20GB memory. For both 2D ground truth and 2D pose detections [51], we train our model using the AMSGrad optimizer for 30 epochs with an initial learning rate of 0.005, and a decay factor of 0.90 per 4 epochs. We set the batch size to 128, the number of layers $L = 5$, and adopt $h = 4$ heads for self-attention. For 2D pose detections [51], we set the middle feature dimension $F = 256$, whereas for the 2D ground truth, we set $F = 128$. We set the weighting factor $\alpha = 0.01$, and the total number of input frames $T = 243$ for both 2D detected poses and ground truth poses. To prevent overfitting, we apply dropout with a factor of 0.2 after each multi-head attention and LAM-GConv layers in the graph attention block. After each Multi-hopGConv and dilated convolutional layer in the Multi-hopGConv block, we also apply dropout with a factor of 0.1. Moreover, we incorporate a pose refinement module [5], which consists of two fully-connected layers and is designed to refine the predicted poses.

3.3.2 Results and Analysis

Quantitative Results on Human3.6M. We report the performance comparison results for all 15 actions, along with the average performance, in Table 3.1. Our MGT-Net outperforms several state-of-the-art methods when using the detected 2D pose as input. As can be seen, our method achieves on average 44.1mm and 36.2mm in terms of MPJPE and PA-MPJPE, respectively, outperforming all the baselines. These findings demonstrate the model's competitiveness, which is largely attributed to the fact that the MGT-Net can better exploit joint connections through the multi-hop graph propagation rule and learn not only the global information from all the nodes, but also the explicit adjacency structure of nodes. Under Protocol #1, Table 1 reveals that our MGT-Net model performs better than GAST-Net [54] in 11 out of 15 actions, yielding 0.8mm error reduction on average, improving upon this best performing temporal baseline by a relative improvement of 1.78%, while maintaining a fairly small number of learnable parameters. In comparison to MGCN [34], our method consistently achieves superior performance across 14 out of 15 actions, demonstrating an average error reduction of 5.3mm. This improvement represents a significant relative enhancement of 10.73% over the strong MGCN baseline.

Under Protocol #2, Table 3.1 shows that our model, on average, reduces the error by 0.82% compared to Anatomy3D [8], which is the best spatio-temporal baseline, and achieves better results in 10 out of 15 actions. Also, our method outperforms GroupGCN in all 15 actions, yielding a relative error reduction of 9.73% in terms of PA-MPJPE.

Cross-Dataset Results on MPI-INF-3DHP. In Table 3.2, we report the quantitative comparison results of the MGT-Net using a single frame against strong baselines on the MPI-INF-3DHP

Table 3.1: Performance comparison of our model and baseline methods on Human3.6M under Protocol #1 and Protocol #2 using the detected 2D pose as input. MPJPE and PA-MPJPE errors are in millimeters. The average errors are reported in the last column. Boldface numbers indicate the best performance, whereas the underlined numbers indicate the second best performance. T denotes the number of input frames used in each spatio-temporal method.

	Action															
Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
SemGCN [30]	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
MöbiusGCN [85]	46.7	60.7	47.3	50.7	64.1	61.5	46.2	45.3	67.1	80.4	54.6	51.4	55.4	43.2	48.6	52.1
GroupGCN [37]	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
MM-GCN [88]	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
MGCN [34]	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
GraFormer [40]	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
Hossain *et al.* [53] ($T = 5$)	44.2	46.7	52.3	49.3	59.9	59.4	47.5	46.2	59.9	65.6	55.8	50.4	52.3	43.5	45.1	51.9
Skeletal-GNN [70] ($T = 9$)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	45.7
ST-GCN [5] ($T = 7$)	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
HTNet [9] ($T = 9$)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	47.2
Lin *et al.* [89] ($T = 50$)	42.5	44.8	42.6	44.2	48.5	57.1	42.6	41.4	56.5	64.5	47.4	43.0	48.1	33.0	35.1	46.6
VideoPose3D [4] ($T = 243$)	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
Attention3D [7] ($T = 243$)	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
SRNet [6] ($T = 243$)	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
GAST-Net [54] ($T = 243$)	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
Anatomy3D [8] ($T = 243$)	42.5	45.4	42.3	45.2	49.1	56.1	43.8	44.9	56.3	64.3	47.9	43.6	48.1	34.3	35.2	46.6
PoseFormer [3] ($T = 81$)	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 ($T = 243$)	**38.7**	**43.9**	42.3	43.8	**44.8**	**48.1**	42.4	41.2	**52.6**	63.8	**43.5**	42.7	44.7	34.1	34.5	**44.1**
Protocol #2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Li *et al.* [90]	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
High-order GCN [31]	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
HOIF-Net [33]	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
MM-GCN [88]	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
GroupGCN [37]	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
MM-GCN [88]	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
MGCN [34]	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
Hossain *et al.* [53] ($T = 5$)	36.9	37.9	42.8	40.3	46.8	46.7	37.7	36.5	48.9	52.6	45.6	39.6	43.5	35.2	38.5	42.0
ST-GCN [5] ($T = 7$)	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.* [89] ($T = 50$)	**32.5**	**35.3**	34.3	**36.2**	37.8	43.0	**33.0**	**32.2**	45.7	51.8	38.4	**32.8**	37.5	25.8	28.9	36.8
VideoPose3D [4] ($T = 243$)	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
Anatomy3D [8] ($T = 243$)	33.6	36.0	34.4	36.6	37.5	42.6	33.5	33.8	44.4	51.0	38.3	33.6	37.7	26.7	28.2	36.5
GAST-Net [54] ($T = 243$)	34.9	37.5	34.9	38.3	37.4	44.0	34.4	34.6	45.1	**48.0**	49.3	34.8	37.7	26.2	**27.1**	36.9
Ours ($T = 243$)	33.0	36.1	**34.1**	37.4	**36.2**	**40.4**	33.6	32.4	**44.1**	54.4	**36.5**	34.5	**36.2**	26.4	27.4	**36.2**

dataset. We train our model on the Human3.6M dataset and evaluate its performance on the MPI-INF-3DHP dataset to test its generalization ability across different datasets. The results demonstrate that our approach consistently outperforms the baseline methods in both indoor and outdoor scenes, achieving the highest PCK and AUC scores. In comparison to the best performing baseline, our model exhibits relative improvements of 4.26% and 7.75% in terms of the PCK and AUC metrics, respectively. Despite being trained solely on indoor scenes from the Human3.6M dataset, our model delivers satisfactory results when applied to outdoor settings. This demonstrates the

robust generalization capability of our approach, extending its performance to unseen scenarios and datasets. In addition, the improvements in both PCK and AUC metrics demonstrate that our proposed model excels in both joint localization accuracy and overall pose estimation quality.

Table 3.2: Performance comparison of our model without pose refinement and baseline methods on the MPI-INF-3DHP dataset using PCK and AUC as evaluation metrics.

Method	PCK (↑)	AUC (↑)
Chen *et al.* [78]	67.9	-
Yang *et al.* [19]	69.0	32.0
Pavlakos *et al.* [69]	71.9	35.3
Habibie *et al.* [91]	70.4	36.0
HOIF-Net [33]	72.8	36.5
SRNet [6]	77.6	43.8
GraphSH [67]	80.1	45.8
GroupGCN [37]	81.1	49.9
MM-GCN [88]	81.6	<u>50.3</u>
Skeletal-GNN [70]	<u>82.1</u>	46.2
Ours	**85.6**	**54.2**

Qualitative Results. In Figure 3.5, we present qualitative results obtained by our proposed MGT-Net on the Human3.6M dataset. The visual comparison showcases the performance of our model in 2D-to-3D human pose estimation, particularly in challenging scenarios. It is evident from the results that our MGT-Net outperforms the baseline method MGCN [34] and achieves closer alignment with the ground truth poses. Our model demonstrates effectiveness in accurately estimating 3D human poses, even in challenging cases where occlusions are present. Notably, MGCN [34] struggles to accurately predict poses in difficult instances such as "Photo", "Sitting", "SitDown", and "WalkDog" due to the presence of occlusions. In contrast, our MGT-Net consistently delivers reliable pose predictions in these challenging scenarios.

Comparison with Spatio-Temporal Methods using Ground Truth. Testing with 2D ground truth as input, instead of 2D detected poses, allows for a more controlled evaluation of the performance of the model. By using the ground truth 2D poses, we can eliminate any errors or inaccuracies that may be present in the 2D pose detector. This provides an indication of how well the model can utilize the available 2D pose information to estimate accurate 3D poses, without being affected by any errors introduced during the 2D pose detection process. In Table 3.3, we report the comparison results between MGT-Net and various spatio-temporal baselines using 2D ground truth keypoints as input. The findings demonstrate that our model surpasses SRNet [6] in 13 out

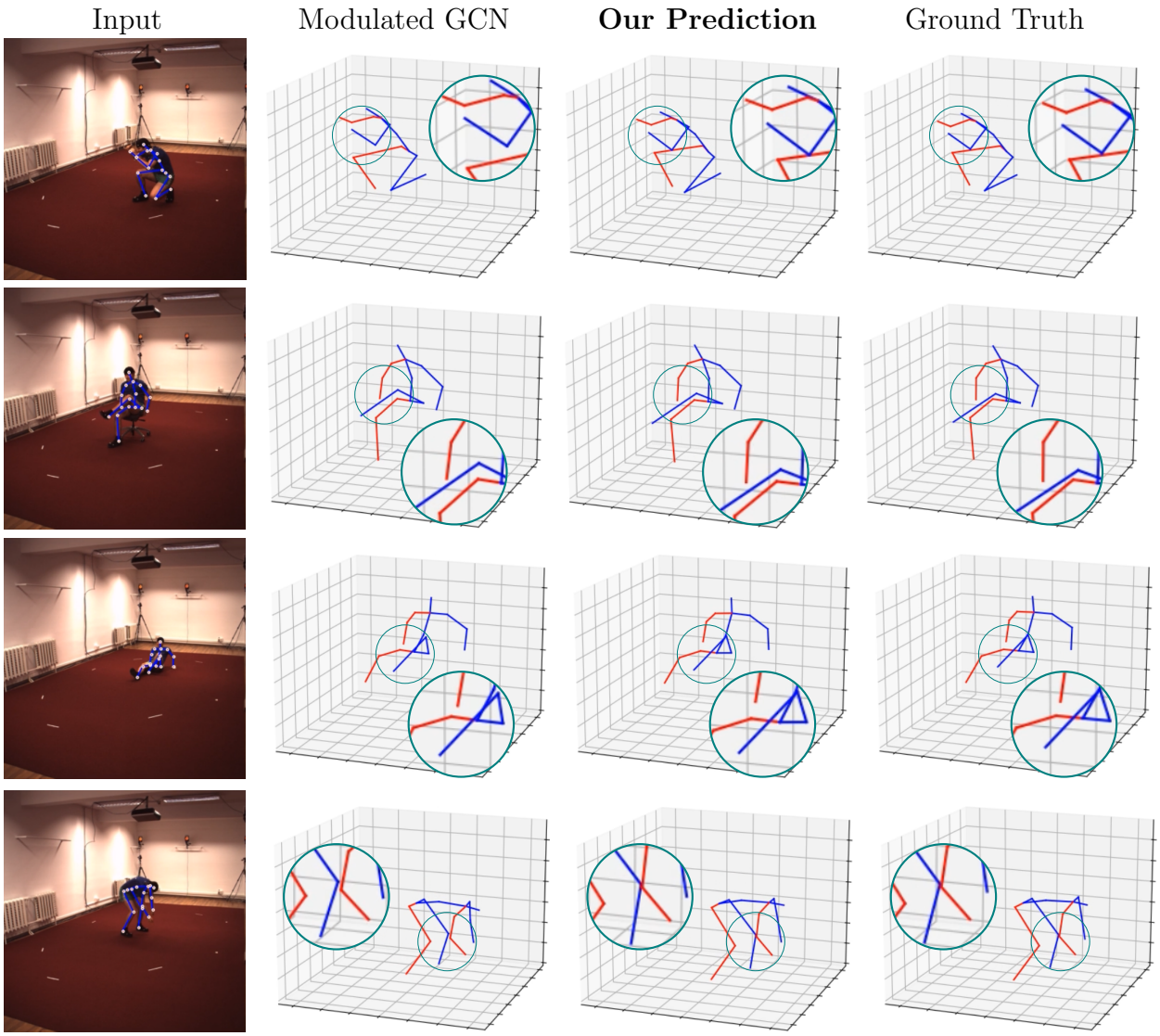


Figure 3.5: Visual comparison between MGT-Net, MGCN and ground truth on the Human3.6M test set. Compared to MGCN, our model is able to produce better predictions.

of 15 actions, resulting in an average reduction in error of approximately 4.38% under Protocol #1. Furthermore, our model exhibits superior performance compared to Anatomy3D [8], Attention3D [7], GAST-Net [54], and PoseFormer [3], on average, while also having fewer learnable parameters. The consistent outperformance of MGT-Net across multiple actions and its ability to surpass several state-of-the-art approaches demonstrate its effectiveness in 3D human pose estimation tasks.

1

Improvements on Hard Poses. Hard poses are characterized by high prediction errors and often exhibit certain inherent characteristics, such as depth ambiguity and self-occlusion [6, 30, 70]. For example, accurately estimating the 3D pose of a person sitting in a crossed-leg position can pose

Table 3.3: Performance comparison of our model without pose refinement and spatio-temporal baseline methods on Human3.6M under Protocol #1 using the ground truth 2D pose as input. T denotes the number of input frames used in each spatio-temporal method.

	Action															
Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Hossain *et al.* [53] ($T = 5$)	35.2	40.8	37.2	37.4	43.2	44.0	38.9	35.6	42.3	44.6	39.7	39.7	40.2	32.8	35.5	39.2
Ray3D [92] ($T = 9$)	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
ST-GCN [5] ($T = 7$)	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
Lin *et al.* [89] ($T = 50$)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.8
PoseFormer [3] ($T = 81$)	30.0	33.6	29.9	31.0	**30.2**	**33.3**	34.8	31.4	37.8	38.6	31.7	31.5	**29.0**	23.3	**21.1**	31.3
Attention3D [7] ($T = 243$)	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	40.7	41.4	33.0	33.8	26.6	26.9	34.7
SRNet [6] ($T = 243$)	34.8	**32.1**	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
GAST-Net [54] ($T = 243$)	30.5	33.1	27.6	31.0	31.8	37.0	**33.2**	30.0	35.7	**37.7**	31.4	**29.8**	31.7	24.0	25.7	31.4
VideoPose3D [4] ($T = 243$)	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
Anatomy3D [8] ($T = 243$)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	32.3
Ours ($T = 243$)	**29.2**	32.9	**27.5**	**30.5**	30.8	37.1	33.5	**27.8**	**34.7**	39.5	**30.1**	30.3	29.8	**21.9**	23.1	**30.6**

difficulties due to the intricate interactions among various body parts. Our method aims to address this challenge by learning to capture the complex relationships between the joints via the graph attention block and the multi-hop graph convolutional block with disentangled neighborhoods. As reported in Table 3.3, MGT-Net outperforms SRNet [6] by reducing prediction errors in the actions "Directions", "Eating", "Purchases", "Sitting", and "Sitting Down" by 5.6mm, 1.0mm, 2.7mm, 4.2mm, and 1.0mm, respectively, which correspond to relative improvements of 16.10%, 3.51%, 8.85%, 10.80%, and 2.47% under Protocol #1. The average improvement on these hard poses is 8.35%. Also, the visualization results shown in Figure 3.5 offer a comparative analysis with MGCN [34] on hard poses, demonstrating superior performance of our model. Hence, these quantitative and qualitative results demonstrate the effectiveness of our model in dealing with hard poses.

Comparison with Spatial Methods using Ground Truth. Table 3.4 presents a performance comparison between our method and various spatial baselines on the Human3.6M dataset under Protocol #1 using the ground truth 2D pose as input. The results demonstrate that our method consistently outperforms the baseline methods across various actions. In terms of the average MPJPE error, our model achieves a relative improvement of 0.57% over GraFormer [40], reducing the error from 35.2mm to 35.0mm. Moreover, our model outperforms MM-GCN [88] by 0.6mm on average, resulting in a relative improvement of 1.69%. Furthermore, our model consistently yields better performance than GroupGCN [37], MöbiusGCN [85], MGCN [34], and GraphMDN [93] in terms of the average MPJPE error. These findings further demonstrate the effectiveness of our method in accurately estimating 3D human poses, surpassing strong spatial baselines while achieving better overall performance.

Table 3.4: Performance comparison of our model without pose refinement and spatial baseline methods on Human3.6M under Protocol #1 using the ground truth 2D pose as input.

	Action															
Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez [22]	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
SemGCN [30]	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
GraphSH [67]	35.8	38.1	31.0	35.3	35.8	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	30.7	35.8
High-order GCN [31]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.5
HOIF-Net [33]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	39.2
Liu [32]	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
MGCN [34]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38.3
GraphMDN [93]	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
MöbiusGCN [85]	**31.2**	46.9	32.5	**31.7**	41.4	44.9	**33.9**	**30.9**	49.2	55.7	35.9	36.1	37.5	29.1	33.1	36.2
MM-GCN [88]	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
GroupGCN [37]	32.5	**36.4**	30.7	33.2	34.9	**40.0**	37.8	33.1	38.3	47.8	34.4	36.2	35.1	28.4	**29.2**	35.2
GraFormer [40]	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
Ours	31.9	38.6	31.2	33.8	35.0	40.9	37.6	32.3	38.5	**42.6**	34.8	35.4	**34.8**	28.2	29.6	**35.0**

3.3.3 Ablation Studies

We conduct ablation studies on the Human3.6M dataset with the aim of analyzing the impact of different design choices in our network architecture. We use MPJPE and PA-MPJPE as evaluation metrics. Evaluation of our ablation experiments is performed without pose refinement, as we aim to assess the performance of our base model without additional refinement steps. To ensure a fair comparison, we train and test our model using 2D ground truth poses, eliminating any potential uncertainties introduced by 2D pose detectors. In our ablation experiments, we systematically vary specific parameters and/or components of our model and assess their influence on the overall performance. By conducting these controlled experiments, we can gain deeper insights into the importance of the key components of our model. Unless indicated otherwise, we set the number of input frames $T = 243$. This choice allows us to evaluate the model's performance with a significant temporal context.

Hyperparameter Sensitivity Analysis. Our findings regarding the impact of various hyperparameters on model performance are summarized in Table 3.5. From the results, it can be observed that a batch size of $B = 128$ yields the best performance. The value of the hidden dimension F has an impact on the model's performance in terms of of capturing patterns. Increasing F from 96 to 128 leads to a decrease in both MPJPE and PA-MPJPE from 32.9mm and 26.0mm to 30.6mm and 24.7mm, respectively, along with a reasonable increase in the number of model parameters. However, further increasing F to 256 results in a significant increase in the number of learnable parameters (from 1.65M to 5.78M) with a noticeable degradation in performance under both protocols. Among different values of the number of attention heads, $h = 4$ produces the best results

in terms of MPJPE (30.6mm) and PA-MPJPE (24.7mm) compared to $h = 2$ and $h = 8$. As for the number of layers, starting with $L = 3$ and increasing it to $L = 5$ yields the best results. Therefore, the best performance on Human3.6M with ground truth 2D pose as input is achieved using hyperparameter values of $L = 3$, $F = 128$, $B = 128$, and $h = 4$.

Table 3.5: Ablation study on various configurations of our model: L is the number of MGT-Net layers, F is the hidden dimension of skeleton embedding, B is the batch size, and h is the number of attention heads.

L	F	B	h	Params (M)	MPJPE (↓)	PA-MPJPE (↓)
5	96	128	4	1.02	32.9	26.0
5	128	128	4	1.65	**30.6**	**24.7**
5	256	128	4	5.78	31.2	25.3
3	128	128	4	1.12	31.8	26.0
4	128	128	4	1.38	31.2	25.3
6	128	128	4	1.91	32.0	25.4
7	128	128	4	2.18	31.9	25.0
5	128	128	2	1.65	32.5	25.7
5	128	128	8	1.65	31.3	25.3
5	128	64	4	1.65	30.6	25.2
5	128	256	4	1.65	31.6	25.2
5	128	512	4	1.65	30.9	25.4

Impact of Input Sequence Length. In Table 3.6, we report the MPJPE and PA-MPJPE results of our method using different input sequence lengths. We can observe that increasing the number of frames leads to better results. This is expected since temporal correlations help address challenges like depth ambiguity and self-occlusions, which are typically not easy to handle by single frame 3D pose estimation methods. It is also worth noting that the MPJPE and PA-MPJPE errors for $T = 1$ are 35.0mm and 27.5mm, respectively. As T increases, the errors decrease and the best results are obtained when $T = 243$ for both protocols. We also report the number of model parameters for different input sequence lengths, and we can see that as T increases, there is only a moderate increase in the number of learnable parameters. Specifically, when T is set to 1 (i.e., single input frame), the total number of model parameters is 1.46M. However, as we progressively increase T to 243, which represents a substantial number of input frames, the model parameters is only slightly increased to 1.65M, while achieving 12.57% and 10.18% relative error reductions in MPJPE and PA-MPJPE, respectively. This slight increase in model size is largely attributed to the fact that the number of frames mostly impacts the first Multi-hopGConv layer, which does not require a large number of learnable parameters. Hence, increasing the input sequence length yields

improved results without significantly increasing the number of parameters.

Table 3.6: Ablation study on the number of input frames (T).

T	1	3	27	81	121	243
MPJPE (↓)	35.0	34.3	33.2	33.2	31.7	**30.6**
PA-MPJPE (↓)	27.5	27.7	26.9	26.8	25.2	**24.7**
Params (M)	1.46	1.46	1.48	1.52	1.55	1.65

Impact of Number of Hops. Table 3.7 provides an analysis of the performance of our model with varying numbers of hops (k) in the multi-hop graph convolutional layer with disentangled neighborhoods. We observe that as the number of hops increases, the model's performance improves. Specifically, the 1-hop model outperforms the 0-hop model, resulting in a reduction of 0.2mm in both MPJPE and PA-MPJPE errors. Furthermore, the 2-hop model achieves even better performance than the 1-hop model, with a relative error reduction of 3.17% in terms of MPJPE. This demonstrates the effectiveness of the multi-hop graph convolutional layer in capturing long-range interdependencies among body joints, enabling the model to better understand the spatial relationships and dependencies within the human pose. The results highlight the importance of incorporating multi-hop connections to enhance the model's ability to capture complex structural dependencies, leading to improved pose estimation accuracy.

Table 3.7: Ablation study on the number of hops. The embedding dimension is set to $F = 128$.

# hops (k)	Params (M)	MPJPE (↓)	PA-MPJPE (↓)
0-hop	1.19	31.8	25.1
1-hop	1.42	31.6	24.9
2-hop	1.65	**30.6**	**24.7**

Impact of Dilated Convolutional Layer. We conducted an ablation experiment to analyze the impact of incorporating dilated convolutional layers (DCLs) after each Multi-hop GConv layer in the multi-hop graph convolutional block. The results, as shown in Table 3.8, demonstrate that the inclusion of DCLs significantly improves the performance of the model while maintaining computational efficiency. Notably, we observe a reduction of 1.5mm in MPJPE and 0.6mm in PA-MPJPE errors without a substantial increase in the number of learnable parameters. This improvement can be attributed, in part, to the ability of dilated convolutions to effectively increase the receptive field without adding more parameters or computational cost. By adjusting the dilation rate, the convolutional kernel can capture information from a wider region of the input, allowing the model

to capture larger contextual information. This capability is particularly important in 3D pose estimation, where understanding the relationship between body joints across different scales and distances is crucial. By effectively capturing long-range dependencies and global context, dilated convolutions contribute to a more accurate estimation of 3D human poses, especially in scenarios involving complex poses.

Table 3.8: Ablation study on dilated convolutional layer (DCL). The embedding dimension is set to $F = 128$.

Method	Params (M)	MPJPE (↓)	PA-MPJPE (↓)
w/o DCL	1.48	32.1	25.3
Ours	1.65	**30.6**	**24.7**

Impact of Pose Refinement. We performed an analysis of the pose refinement network (PRN) and present the results in Table 3.9. The findings show that, on average, incorporating PRN leads to a reduction of 2.5mm in MPJPE and 0.8mm in PA-MPJPE errors when $T = 1$. Similarly, when $T = 243$, the inclusion of PRN results in an average reduction of 2.1mm in MPJPE and 0.9mm in PA-MPJPE errors. These results demonstrate that incorporating pose refinement improves the overall performance of our model under both protocols.

Table 3.9: Impact of the pose refinement network (PRN) on the performance of our model.

Our model	Params (M)	MPJPE (↓)	PA-MPJPE (↓)
w/o PRN ($T = 1$)	1.46	35.0	27.5
w PRN ($T = 1$)	1.46	**32.5**	**26.7**
w/o PRN ($T = 243$)	1.65	30.6	24.7
w PRN ($T = 243$)	1.65	**28.5**	**23.8**

Figure 3.6 showcases the performance of our model with and without pose refinement using $T = 243$ under Protocol #1 (top) and Protocol #2 (bottom), specifically for challenging actions such as “Directions”, “Eating”, ‘Photo”, “Posing”, and “Sitting”. Notably, we observe a 12.77% reduction in relative error for the "Photo" action in terms of MPJPE, and a 5.56% reduction in relative error in terms of PA-MPJPE. These results highlight the effectiveness of pose refinement in improving the accuracy of pose estimation, particularly for challenging poses and actions.

Impact of Graph Convolutional Layers. The multi-hopGConv layer is a key component of our model, allowing for the capture of long-range dependencies among body joints. We evaluate the performance of our model by replacing this layer with different types of graph convolutional

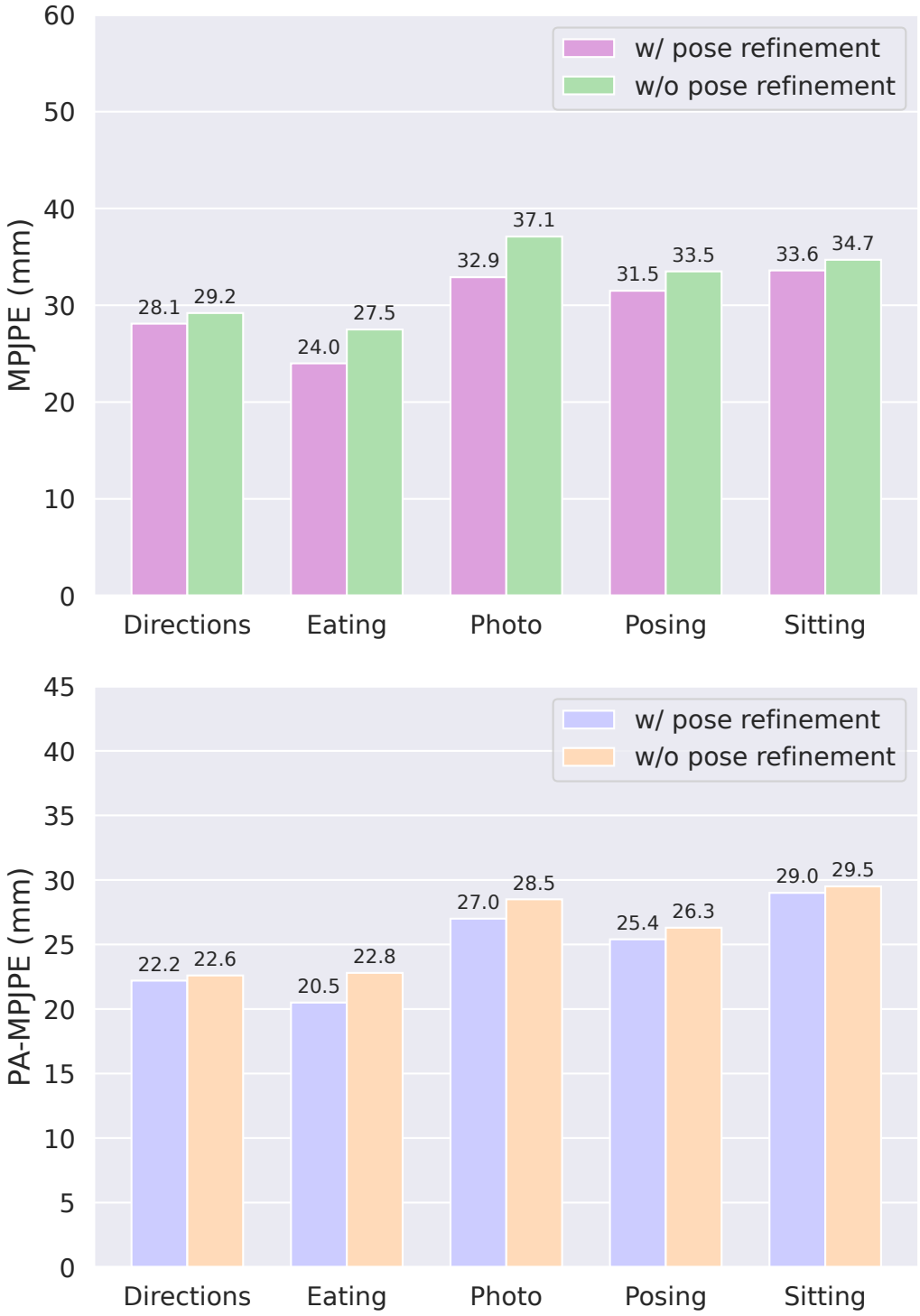


Figure 3.6: Performance of our model with and without pose refinement using MPJPE (top) and PA-MPJPE (bottom). When coupled with a pose refinement network, MGT-Net performs consistently better on challenging actions.

layers. Specifically, we investigate and compare the results of our model with three variants: Semantic Graph Convolutional (SemGConv) layer [30], Modulated Graph Convolutional (MGConv) layer [34], and Chebyshev Graph Convolutional (ChebGConv) layer [40]. The SemGConv layer incorporates semantic information into the graph convolutional operation, while the MGConv layer introduces modulation to the graph convolution operation, allowing the model to adaptively adjust the importance of different nodes in the graph. The ChebGConv layer, on the other hand, utilizes Chebyshev polynomials to capture localized spectral information of the graph structure. We follow the original implementation for all these variants and consider 1-hop neighbors for both SemGConv and MGConv, and 2-hop neighbors for ChebGConv. We also set the embedding dimension to $F = 128$. The results are reported in Table 3.10, which shows that the multi-hopGConv layer with 2-hop neighbors achieves the best results under both Protocol #1 (30.6mm) and Protocol #2 (24.7mm), indicating the importance of capturing long-range dependencies and considering a wider context in the pose estimation task. In addition, our model maintains a relatively low number of learnable parameters, demonstrating the efficiency of the proposed approach.

Table 3.10: Ablation study on various types of graph convolutional layers. The embedding dimension is set to $F = 128$.

Method	k	Params (M)	MPJPE (↓)	PA-MPJPE (↓)
MGConv [34]	1	1.45	35.5	27.4
SemGConv [30]	1	1.42	33.5	26.1
ChebGConv [40]	2	1.65	31.8	25.0
Ours	2	1.65	**30.6**	**24.7**

3.3.4 Model Efficiency

In Table 3.11, we present a comprehensive analysis of the computational efficiency and performance of our model compared to state-of-the-art baselines. We assess the trade-off between the model's computational cost and its performance in terms of the number of input frames (T), total number of parameters, estimated number of floating-point operations (FLOPs), and MPJPE. The evaluation is conducted on the Human3.6M dataset using both detected 2D poses and ground truth as inputs. Notably, our model achieves a balance between computational efficiency and accuracy, outperforming strong baselines while maintaining a relatively low computational cost. For the detected 2D poses, our model achieves an MPJPE of 44.1, demonstrating superior performance compared to the baselines. Furthermore, when provided with ground truth 2D poses as input,

our model achieves a significantly reduced MPJPE of 30.6, showcasing the effectiveness of our approach.

Table 3.11: Efficiency of our model in comparison with baselines in terms of the number of input frames (T), total number of parameters, FLOPs, and MPJPE. Evaluation is performed on Human3.6M using both the detected 2D poses and ground-truth as inputs.

Method	T	Params (M)	FLOPs (M)	MPJPE ($\downarrow$)
ST-GCN [5]	7	5.18	469.81	48.8
PoseFormer [3]	81	9.60	1358	44.3
CrossFormer [94]	27	9.93	515	46.5
Anatomy3D [8]	81	45.53	88.9	44.6
Ours	243	5.78	84.41	**44.1**
Ours (GT)	243	1.65	21.23	**30.6**

CHAPTER 4

Conclusions and Future Work

In this thesis, we presented novel deep neural network architectures for 3D human pose estimation. In pursuit of this objective, we propose a graph neural network whose layer-wise propagation rule derives inspiration from iteratively solving graph filtering with Laplacian regularization via the Gauss-Seidel iterative method. The proposed approach adopts a two-stage paradigm, combining an off-the-shelf 2D pose detector with a graph convolutional network. By doing so, the network achieves the objective of accurately estimating the 3D poses of individuals based on their 2D pose representations. The proposed network architecture incorporates weight and adjacency modulation, skip connection, and a variant of the ConvNeXt residual block. We also proposed a novel network architecture for 3D human pose estimation that leverages multi-head self-attention, multi-hop graph convolutions with disentangled neighborhoods, and dilated convolutions. Disentangling the neighborhoods helps remove redundant dependencies between further and closer neighborhoods, allowing the model to effectively capture long-range dependencies between graph nodes. The combination of multi-head self-attention and multi-hop graph convolutional layers enables the model to capture both local and global dependencies, while the integration of dilated convolutional layers enhances the model's ability to handle spatial details required for accurate localization of the human body joints. We performed quantitative and qualitative evaluations on two large-scale datasets to assess the efficacy of both proposed approaches for 3D human pose estimation. Finally, in Section 4.1, we discuss the concluding outcomes of the associated research work in each of the previous chapters, along with the contributions made. Furthermore, we address the limitations of the proposed approach in Section 4.2, and present suggestions for potential research directions related to this thesis in Section 4.3.

4.1 Contributions of the Thesis

4.1.1 Iterative Graph Filtering Network for 3D Human Pose Estimation

In Chapter 2, we presented a simple yet effective Gauss-Seidel graph neural network together with weight and adjacency modulation. The layer-wise propagation rule of our proposed framework is inspired by the iterative solution of graph filtering with Laplacian regularization using the Gauss-Seidel method. Our network architecture leverages the ConvNeXt residual block, which makes the network more computationally efficient and reduces the risk of overfitting, where the model learns to fit the training data too closely. Empirical experiments show that our model achieves state-of-the-art performance on two benchmark datasets, and can serve as a strong baseline for 3D human pose estimation. We also conducted extensive ablation studies to analyze the impact of different design choices on the model performance.

4.1.2 Multi-hop Graph Transformer Network for 3D Human Pose Estimation

In Chapter 3, we proposed MGT-Net, a spatio-temporal model for 3D human pose estimation. Our approach combines multi-head self-attention and multi-hop graph convolutions with disentangled neighborhoods, enabling the model to effectively capture both long-range dependencies and local-global contextual information. By leveraging multi-head self-attention, our model is able to focus on different aspects of the input data simultaneously, allowing it to capture intricate spatial relationships between body joints. On the other hand, the integration of multi-hop graph convolutions with disentangled neighborhoods enables our model to aggregate information from neighboring nodes at different hops, capturing both local and global contextual cues in the graph structure. To further enhance the model's understanding of spatial relationships and dependencies, we incorporated dilated graph convolutions into our network architecture. This integration extends the model's receptive field, allowing it to capture larger contextual information and better comprehend the relationships between body joints at varying scales and distances. By leveraging dilated convolutions, we avoided the need for additional learnable parameters, ensuring model efficiency while still achieving improved performance. Through extensive experiments and ablation studies, we demonstrated the effectiveness of our proposed model and its superiority over existing methods in terms of performance and computational efficiency. We also showed that MGT-Net not only improves the accuracy of 3D human pose estimation, but also addresses challenges with hard poses such as occlusion and depth ambiguity.

4.2 Limitations

While 3D human pose estimation tasks witness advancements in robustness and accuracy with the proposed methods, they also reveal specific constraints. Despite our model's capability to achieve state-of-the-art performance, GS-Net fails to capture long-range dependencies between body joints as well as different relations between neighboring joints and distant ones, as it only takes the immediate neighbors into consideration. We also thoroughly investigate several instances where our GS-Net model did not perform as expected. We scrutinize these failure cases to gain a deeper understanding of the limitations of our model. Figure 4.1 illustrates failure cases of our model predictions for the "Greeting" and "Sitting Down" actions from Human3.6M. As can be seen, our predictions do not align perfectly with the ground truth poses in situations where self-occlusions occur. It is important to point out that these failure cases are not specific to GS-Net alone but rather a common challenge encountered in previous works as well [30, 34]. This is attributed in part to the diverse actions performed in different ways within the Human3.6M training dataset. In addition, since our model does not incorporate temporal information, the uncertainty inherent in human motion further adds complexity to the prediction process.

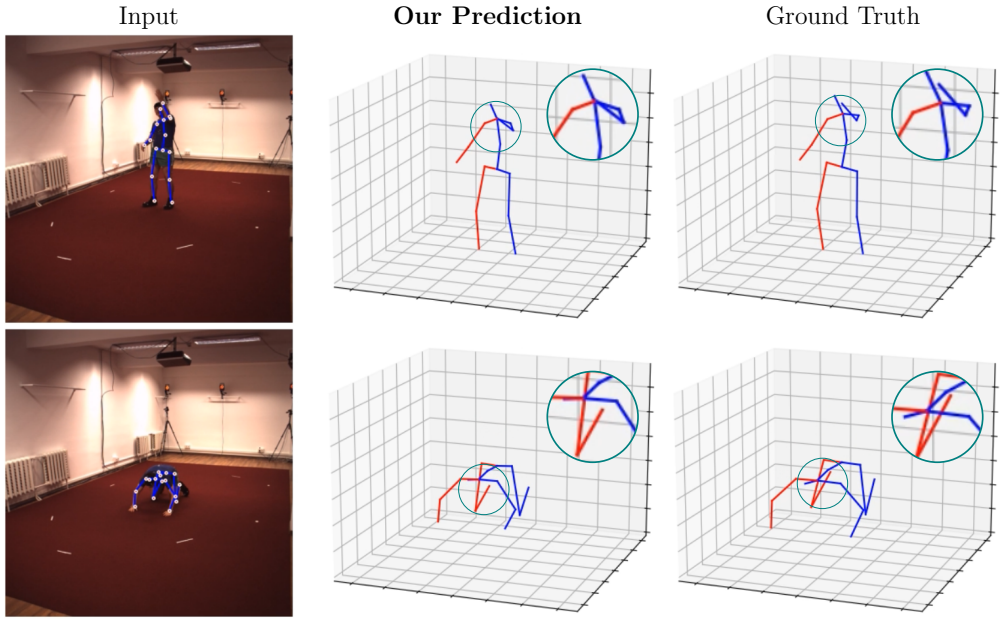


Figure 4.1: Example of the failure cases of our model on the "Greeting" and "Sitting Down" actions from Human3.6M.

Moreover, the efficacy of our Graph Attention block in the MGT-Net model heavily depends on multi-head self-attention for capturing global information. However, this approach can lead to computational overhead, especially when dealing with extensive datasets or deep networks. Conse-

quently, this limitation poses a significant challenge in scaling the model for real-time applications, where low latency plays a vital role. Furthermore, these methods do not provide end-to-end solutions for regressing 3D keypoints from images or videos since they comprise two separate stages that are usually decoupled. To overcome this limitation, a promising approach is to utilize extensively trained, deep models. By doing so, we can address the issue and potentially achieve more accurate and integrated results when regressing 3D keypoints from visual data.

4.3 Future Work

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

4.3.1 GS-Net with Multi-hop Neighbors

In order to further improve the accuracy of 3D pose estimation, we aim to devise a method that takes into account the high-order connectivity between joints. To achieve this, we plan to incorporate multi-hop graph convolutions with the aim of explicitly taking advantage of the graph structure information, allowing our model to better capture the relationships between body joints at various hop distances. We also plan to explore the applicability of our model to other computer vision and graph representation learning tasks, as well as to improve its computational efficiency and interpretability.

4.3.2 MGT-Net Exploiting Frequency Domain

We aim to further enhance our approach through architectural improvements by effectively fusing features both in the time domain and frequency domain. By exploiting the compact representation of input skeleton sequences in the frequency domain, we hope to improve the model's robustness to 2D noisy detection. Besides, we plan to design a dedicated temporal transformer module and incorporate it into MGT-Net for better learning the global dependencies of the entire sequence. In addition, we plan to explore its application in other human motion analysis tasks, such as 3D human motion prediction, multi-person 3D human pose estimation, and human mesh recovery.

References

[1] 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.

[2] 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*, pp. 506–516, 2017.

[3] 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 International Conference on Computer Vision*, 2021.

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

[5] 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. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2272–2281, 2019.

[6] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, and S. Lin, "SRNet: Improving generalization in 3D human pose estimation with a split-and-recombine approach," in *Proc. European Conference on Computer Vision*, pp. 507–523, 2020.

[7] R. Liu, J. Shen, H. Wang, C. Chen, S.-C. Cheung, and V. Asari, "Attention mechanism exploits temporal contexts: Real-time 3D human pose reconstruction," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5064–5073, 2020.

[8] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, and J. Luo, "Anatomy-aware 3D human pose estimation with bone-based pose decomposition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 1, pp. 198–209, 2021.

[9] J. Cai, H. Liu, R. Ding, W. Li, J. Wu, and M. Ban, "HTNet: human topology aware network for 3D human pose estimation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[10] L. Song, G. Yu, J. Yuan, and Z. Liu, "Human pose estimation and its application to action recognition: A survey," *Journal of Visual Communication and Image Representation*, vol. 76, 2021.

[11] Y. Zhao, Z. Yuan, and B. Chen, "Accurate pedestrian detection by human pose regression," *IEEE Transactions on Image Processing*, vol. 29, pp. 1591–1605, 2019.

[12] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3D head tracking to detect falls of elderly people," in *Proc. International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6384–6387, 2006.

[13] W. Liu, Q. Bao, Y. Sun, and T. Mei, "Recent advances of monocular 2D and 3D human pose estimation: A deep learning perspective," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–41, 2022.

[14] X. Zhou, X. Sun, W. Zhang, S. Liang, and Y. Wei, "Deep kinematic pose regression," in *Proc. European Conference on Computer Vision*, pp. 186–201, 2016.

[15] S. Park, J. Hwang, and N. Kwak, "3D human pose estimation using convolutional neural networks with 2D pose information," in *Proc. European Conference on Computer Vision*, pp. 156–169, Springer, 2016.

[16] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *Proc. European Conference on Computer Vision*, pp. 529–545, 2018.

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

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

[19] 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. IEEE conference on Computer Vision and Pattern Recognition*, pp. 5255–5264, 2018.

[20] Z. Chen, Y. Huang, H. Yu, B. Xue, K. Han, Y. Guo, and L. Wang, "Towards part-aware monocular 3D human pose estimation: An architecture search approach," in *Proc. European Conference on Computer Vision*, pp. 715–732, 2020.

[21] K. Lee, I. Lee, and S. Lee, "Propagating LSTM: 3D pose estimation based on joint interdependency," in *Proc. European Conference on Computer Vision*, pp. 119–135, 2018.

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

[23] 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. IEEE International Conference on Computer Vision*, pp. 2272–2281, 2019.

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

[25] H. Wu and B. Xiao, "3D human pose estimation via explicit compositional depth maps," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12378–12385, 2020.

[26] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang, "Deep kinematics analysis for monocular 3D human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognitionn*, pp. 899–908, 2020.

[27] H. Choi, G. Moon, and K. M. Lee, "Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose," in *Proc. European Conference on Computer Vision*, pp. 769–787, 2020.

[28] J. Wang, S. Yan, Y. Xiong, and D. Lin, "Motion guided 3D pose estimation from videos," in *Proc. European Conference on Computer Vision*, pp. 764–780, Springer, 2020.

[29] K. Liu, Z. Zou, and W. Tang, "Learning global pose features in graph convolutional networks for 3D human pose estimation," in *Proc. Asian Conference on Computer Vision*, 2020.

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

[31] 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.

[32] 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, 2020.

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

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

[35] 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, IEEE, 2021.

[36] J. Y. Lee and I. G. Kim, "Multi-hop modulated graph convolutional networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2022.

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

[38] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[39] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[40] 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.

[41] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation+ matching," in *Proc. IEEE conference on Computer Vision and Pattern Recognition*, pp. 7035–7043, 2017.

[42] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3D pose estimation from a single image," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2500–2509, 2017.

[43] B. Tekin, P. Márquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," in *Proc. IEEE International Conference on Computer Vision*, pp. 3941–3950, 2017.

[44] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1653–1660, 2014.

[45] H. Qiu, C. Wang, J. Wang, N. Wang, and W. Zeng, "Cross view fusion for 3D human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4342–4351, 2019.

[46] Y. He, R. Yan, K. Fragkiadaki, and S.-I. Yu, "Epipolar transformers," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7779–7788, 2020.

[47] Z. Liu, H. Chen, R. Feng, S. Wu, S. Ji, B. Yang, and X. Wang, "Deep dual consecutive network for human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 525–534, 2021.

[48] Z. Liu, R. Feng, H. Chen, S. Wu, Y. Gao, Y. Gao, and X. Wang, "Temporal feature alignment and mutual information maximization for video-based human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11006–11016, 2022.

[49] Y. Sun, Y. Ye, W. Liu, W. Gao, Y. Fu, and T. Mei, "Human mesh recovery from monocular images via a skeleton-disentangled representation," in *Proc. IEEE International Conference on Computer Vision*, pp. 5349–5358, 2019.

[50] W. Li, H. Liu, H. Tang, P. Wang, and L. V. Gool, "MHFormer: Multi-hypothesis transformer for 3D human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13147–13156, 2022.

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

[52] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognitionn*, pp. 5693–5703, 2019.

[53] 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.

[54] 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.

[55] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

[56] Z. Islam and A. B. Hamza, "Iterative graph filtering network for 3D human pose estimation," *Journal of Visual Communication and Image Representation*, 2023.

[57] 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.

[58] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. International Conference on Machine Learning*, pp. 5453–5462, 2018.

[59] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized pagerank," in *International Conference on Learning Representations*, 2019.

[60] T. Chen, K. Zhou, K. Duan, W. Zheng, P. Wang, X. Hu, and Z. Wang, "Bag of tricks for training deeper graph neural networks: A comprehensive benchmark study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[61] 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.

[62] 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 International Conference on Computer Vision*, 2021.

[63] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE International Conference on Computer Vision*, pp. 10012–10022, 2021.

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

[65] 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.

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

[67] T. Xu and W. Takano, "Graph stacked hourglass networks for 3D human pose estimation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 16105–16114, 2021.

[68] 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. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10905–10914, 2019.

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

[70] 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.

[71] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for end-to-end object detection," in *International Conference on Learning Representations*, 2021.

[72] 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*, vol. 32, 2018.

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

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

[75] S. Banik, A. M. GarcÍa, and A. Knoll, "3D human pose regression using graph convolutional network," in *Proc. IEEE International Conference on Image Processing*, pp. 924–928, 2021.

[76] 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.

[77] 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. IEEE International Conference on Computer Vision*, pp. 398–407, 2017.

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

[79] J. Wang, S. Huang, X. Wang, and D. Tao, "Not all parts are created equal: 3D pose estimation by modeling bi-directional dependencies of body parts," in *Proc. IEEE International Conference on Computer Vision*, pp. 7771–7780, 2019.

[80] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu, "HEMlets pose: Learning part-centric heatmap triplets for accurate 3D human pose estimation," in *Proc. IEEE International Conference on Computer Vision*, pp. 2344–2353, 2019.

[81] D. C. Luvizon, D. Picardu, and H. Tabia, "Multi-task deep learning for real-time 3D human pose estimation and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, pp. 2752–2764, 2021.

[82] A. Zanfir, M. Zanfir, A. Gorban, J. Ji, Y. Zhou, D. Anguelov, and C. Sminchisescu, "HUM3DIL: Semi-supervised multi-modal 3D human pose estimation for autonomous driving," in *Proc. Conference on Robot Learning*, 2023.

[83] C. K. Ingwersen, C. Mikkelstrup, J. N. Jensen, M. R. Hannemose, and A. B. Dahl, "SportsPose – a dynamic 3D sports pose dataset," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2023.

[84] Y. Gu, S. Pandit, E. Saraee, T. Nordahl, T. Ellis, and M. Betke, "Home-based physical therapy with an interactive computer vision system," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[85] N. Azizi, H. Possegger, E. Rodolà, and H. Bischof, "3D human pose estimation using Möbius graph convolutional networks," in *Proc. European Conference on Computer Vision*, pp. 160–178, 2022.

[86] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and unifying graph convolutions for skeleton-based action recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 143–152, 2020.

[87] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations*, 2016.

[88] J. Y. Lee and I. G. Kim, "Multi-hop modulated graph convolutional networks for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2022.

[89] J. Lin and G. H. Lee, "Trajectory space factorization for deep video-based 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2019.

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

[91] 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. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10905–10914, 2019.

[92] 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.

[93] 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.

[94] M. Hassanin, A. Khamiss, M. Bennamoun, F. Boussaid, and I. Radwan, "Crossformer: Cross spatio-temporal transformer for 3D human pose estimation," *arXiv preprint arXiv:2203.13387*, 2022.