Learning Flexible Graph Representations for 3D Human Pose Estimation

Abu Taib Mohammed Shahjahan

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

July 2025

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Abu Taib Mohammed Shahjahan

Entitled: Learning Flexible Graph Representations 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:

______________________ Chair
Dr. Mohammad Mannan

______________________ Examiner
Dr. Mohammad Mannan

______________________ Examiner
Dr. Manar Amayri

______________________ Thesis Supervisor(s)
Dr. Abdessamad Ben Hamza

______________________ Thesis Supervisor(s)

Approved by ______________________
Dr. Chun Wang Chair of Department or Graduate Program Director

____
Dr. Mourad Debbabi Dean of Gina Cody School

Abstract

Learning Flexible Graph Representations for 3D Human Pose Estimation

Abu Taib Mohammed Shahjahan

Accurate 3D human pose estimation remains a significant challenge in computer vision, especially under occlusions, complex joint articulation, and depth ambiguities. Graph Convolutional Network (GCN)-based methods have proven effective by modeling the human skeleton as a graph of joints and bones. However, standard GCNs are limited by one-hop neighbor aggregation, as well as spectral bias, which emphasizes low-frequency features while overlooking fine-grained motion. This thesis addresses these issues by introducing flexible graph convolutional network (Flex-GCN), a novel architecture that enhances spatial awareness through multi-hop aggregation controlled by a scaling parameter. Flex-GCN integrates residual graph convolutional blocks and a global response normalization layer to improve feature selectivity and contextual understanding. Moreover, adjacency modulation enables dynamic graph restructuring, allowing better representation of distant joint relationships. Building upon these findings, the second part of this thesis introduces the Flexible Graph Kolmogorov-Arnold Network (FG-KAN), a more expressive framework that integrates the Kolmogorov-Arnold Network (KAN) with graph-based learning. FG-KAN replaces the fixed activation functions in standard GCNs with learnable, univariate functions applied directly to graph edges, enhancing both interpretability and adaptability, which not only mitigates spectral bias but also enables the model to capture fine-grained joint dynamics crucial for accurately estimating complex and fast body movements. FG-KAN incorporates residual connections, scalable multi-hop feature aggregation, and symmetric adjacency modulation, ensuring both computational efficiency and improved generalization. Comprehensive experimental evaluations on benchmark datasets such as Human3.6M and MPI-INF-3DHP demonstrate that both Flex-GCN and FG-KAN outperform competing baseline methods in terms of Mean Per Joint Position Error, Procrustes Aligned Mean Per Joint Position Error, and Percentage of Correct Keypoints. Notably, while both models demonstrate strong robustness, interpretability is a distinct advantage of FG-KAN, as evidenced by qualitative visualizations and ablation studies.

Table of Contents

List of Figures

List of Tables

List of Acronyms

GCN	Graph Convolutional Network
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
ReLU	Rectified Linear Unit
GELU	Gaussian Error Linear Unit
Flex-GCN	Flexible graph convolutional network
GRN	Global Response Normalization
FG-KAN	Flexible Graph Kolmogorov-Arnold Network
HPE	Human Pose Estimation
MM-GCN	Multi-hop Modulated GCN
SemGCN	Semantic Graph Convolutional Networks
CompGCN	Compositional GCN
HOIF-Net	Higher-Order Implicit Fairing Network
WSGN	Weakly Supervised Generative Network
NLP	Natural Language Processing

CHAPTER 1

Introduction

In this chapter, we begin by outlining the core motivation behind this research, emphasizing the challenges and opportunities in 3D human pose estimation that inspire our proposed approach. We then provide the necessary technical background, covering key architectural components such as residual connections, layer normalization, and dropout, which play a vital role in stabilizing and optimizing deep neural networks. This is followed by presenting a formulation of the problem and articulating the main objectives of the work. Next, we present a comprehensive literature review that highlights the evolution of the field, focusing in particular on graph convolutional network (GCN)-based models and Transformer-based architectures that have emerged as leading paradigms for 3D human pose estimation. This review situates our work within the broader research landscape, identifying limitations in existing approaches and motivating our proposed contributions. Finally, we summarize the specific research objectives and key contributions of this thesis.

1.1 Framework, Motivation and Background

Human Pose Estimation (HPE) refers to the task of predicting the configuration of human body parts, typically represented as 2D or 3D joint coordinates, from still images or videos. The goal is to reconstruct a coherent skeletal representation of the human body that captures its posture and movement. Traditional HPE systems, primarily based on marker-based motion capture (MoCap), relied on specialized hardware to track body joints and reconstruct skeletons. While accurate, these systems are costly, cumbersome, and impractical for deployment in uncontrolled environments. Over the past decade, significant progress in computer vision and deep learning has led to the development of vision-based HPE methods that no longer require wearable markers. These approaches have substantially improved the detection accuracy of body joints across varied conditions and have made HPE accessible to a wide range of real-world applications. As artificial intelligence continues to expand its reach, HPE has found use in numerous domains including media and entertainment, healthcare and rehabilitation, sports and fitness, surveillance and security, human-computer interaction (HCI), robotics, autonomous systems, retail analytics, education, and cultural heritage preservation. In healthcare and rehabilitation, 3D HPE [8], which aims to recover the 3D spatial coordinates of body joints, plays a vital role in tasks such as physical therapy monitoring, gait analysis, and posture correction. Gait, in particular, serves as a distinctive biometric signature and carries valuable spatial and temporal information about an individual's locomotion. Spatial characteristics include limb coordination, stride length, and torso orientation, while temporal features capture the rhythm and periodicity of movement. Gait analysis is especially useful for early detection of movement disorders such as Parkinson's disease (PD) and cerebral palsy. For example, a recent study [9] demonstrated the feasibility 3D HPE in analyzing kinematic gait parameters in patients with PD, offering a non-invasive tool for clinical assessment.

From a technical standpoint, 3D HPE provides a more realistic and accurate representation of body posture in real-world environments. However, the task presents multiple challenges. First, collecting accurately labeled 3D data is difficult due to the need for expensive and intrusive equipment, limiting the availability of large-scale datasets. Second, monocular 3D HPE suffers from depth ambiguity and occlusion, where certain body parts are hidden behind others, leading to uncertainty in spatial reconstruction. To mitigate these issues, several strategies have been proposed, such as synthesizing occlusion-rich training data [10], incorporating kinematic priors, and inferring hidden joints through contextual relationships with visible parts. Furthermore, the computational cost of HPE models is a critical consideration. Highly accurate models often require substantial computational resources, limiting their scalability and deployment in edge devices or real-time applications. As a result, recent work increasingly emphasizes lightweight architectures that strike a balance between performance and efficiency, reducing model size while maintaining competitive accuracy. Most existing 3D HPE approaches rely on monocular RGB images or video streams, making the problem inherently ill-posed due to the loss of depth information in the 2D-to-3D projection. Broadly, 3D HPE methods can be categorized into one-stage and two-stage pipelines, or alternatively, based on the input modality: single-view single-person, single-view multi-person, and multi-view methods. Each setting introduces unique challenges in terms of occlusion handling, depth inference, scalability, and computational demands.

Residual Connections. Residual Connections are a foundational approach in modern deep learning architectures, developed to address the optimization challenges that arise when training very deep neural networks. As neural networks grow deeper, it becomes increasingly difficult to optimize them effectively, even when strategies like normalized initialization and batch normalization help facilitate convergence. One unexpected phenomenon is the degradation of training accuracy as the depth of a network increases an issue not attributed to overfitting, but to inherent optimization difficulties. The core idea of deep residual learning is to reformulate the learning process by explicitly modeling the residual functions [11]. Instead of expecting a stack of nonlinear layers to directly approximate a desired function , the residual learning approach enables these layers to approximate the residual function effectively shifting the learning task from function approximation to learning a perturbation from the identity mapping. This simple yet powerful reparameterization helps alleviate the degradation problem by making it easier for the optimization algorithm to approximate functions that are close to the identity. If the optimal mapping is indeed close to an identity transformation, driving the residual to zero becomes simpler than learning the identity from scratch through a stack of nonlinear layers. Residual mappings are implemented through shortcut connections, which perform identity mappings and are added directly to the outputs of the stacked layers. These shortcuts introduce no additional parameters or computational cost and allow gradients to propagate more effectively during backpropagation, mitigating the vanishing gradient problem. This structure enables training of significantly deeper networks without the degradation observed in traditional architectures. Residual connections not only simplifies optimization but also offers a scalable architecture capable of handling extremely deep models. The empirical success of residual networks across multiple tasks underscores the broad applicability and robustness of this framework.

Layer Normalization. Layer normalization (LN) [12] was introduced as a solution to several optimization difficulties encountered when training deep neural networks, particularly those related to internal covariate shift, where the distribution of layer inputs changes during training, disrupting gradient-based optimization. In conventional feed-forward networks, each hidden layer applies a linear transformation followed by a non-linear activation function. However, this setup often leads to highly correlated neuron activations, which in turn slows convergence and increases sensitivity to parameter initialization. To mitigate this, batch normalization (BN) was originally proposed to normalize activations across the batch dimension by computing the mean and variance over mini-batches. While BN has proven effective in convolutional architectures and large-scale training setups, it suffers from notable limitations: its dependency on sufficiently large and consistent batch sizes, and its inapplicability to certain settings such as recurrent neural networks (RNNs), variable-length sequences, or online learning where batch computation is not feasible.

Layer normalization offers a more generalizable and architecture-agnostic alternative by normalizing activations across the feature dimension of each individual input sample, rather than across the batch. Specifically, for each training example, LN computes the mean and variance across all hidden units in a layer, ensuring that the output distribution remains stable regardless of the batch size. This property makes LN especially suitable for sequence modeling tasks, including RNNs and Transformers, where consistent normalization across time steps and small batches is often required. Formally, given an input vector $\mathbf{x} \in \mathbb{R}^d$, the output $\mathbf{y}$ of a layer normalization operation is computed as:

$$\mathbf{y} = \frac{\mathbf{x} - \mathrm{E}[\mathbf{x}]}{\sqrt{\mathrm{Var}[\mathbf{x}] + \epsilon}} \cdot \gamma + \beta, \tag{1.1}$$

where $\mathrm{E}[\cdot]$ and $\mathrm{Var}[\cdot]$ denote the mean and variance of the features within the input. The parameters γ and β are learnable scale and shift parameters. The small constant ϵ ensures numerical stability during division.

Dropout. Dropout [13] is a regularization technique developed to reduce overfitting in deep feed-forward neural networks. In a typical architecture, layers of non-linear hidden units learn feature detectors by adjusting the incoming weights to optimize prediction performance. However, when the model is sufficiently expressive and the labeled dataset is limited, numerous weight configurations can fit the training data perfectly. These solutions, though accurate on training data, often generalize poorly due to overfitting arising from complex co-adaptations among hidden units. To counteract this, dropout introduces stochasticity by randomly deactivating (or "dropping out") each hidden unit during training with a fixed probability, typically 0.5. This means a hidden unit cannot depend on the presence of specific other units, forcing the network to learn more robust and independent feature detectors. From another perspective, dropout can be seen as an efficient form of model averaging. Instead of training and averaging predictions from an ensemble of individual network, which is computationally expensive, dropout trains a vast number of subnetworks, each defined by a different random subset of active hidden units, all sharing the same parameters. Training is conducted using stochastic gradient descent on mini-batches. To further control overfitting, rather than applying a global penalty on the L2 norm of all weights, dropout constrains the L2 norm of the incoming weight vector for each hidden unit individually. If an update causes this norm to exceed a predefined threshold, the weights are rescaled to remain within bounds. This approach permits the use of larger initial learning rates, enhancing exploration of the parameter space and improving convergence. At inference time, the full network is used, but with each hidden unit's outgoing weight scaled by the dropout probability (e.g., multiplied by 0.5) to approximate the ensemble behavior. In networks with a single hidden layer of N units and a softmax output, this "mean network" is mathematically equivalent to the geometric mean of the predictions from all

2^N possible subnetworks. Notably, the mean network's output is guaranteed to achieve a higher expected log-probability for the correct class than the average log-probability of the individual dropout networks. A similar improvement is observed for regression tasks with linear output units, where the squared error of the mean network is lower than the average squared error across all subnetworks.Dropout thus provides a practical and theoretically grounded method for reducing overfitting and enhancing generalization in deep learning models.

1.2 Problem Statement

In this section, we briefly describe the main problem addressed in this thesis: learning flexible graph representations for 3D human pose estimation. Graphs serve as powerful tools for representing relationships between entities in various domains, including social networks, e-commerce platforms, citation networks, shape classification and retrieval [14–18], geometry processing [19], and mesh watermarking [20]. Formally, a graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{1, \dots, N\}$ is the set of N nodes or vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges or links connecting pairs of vertices. Given a a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ 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

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} l(f(\mathbf{x}_i), \mathbf{y}_i), \tag{1.2}$$

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

1.3 Objectives

The aim of this thesis is to develop graph representation leaning models for 3D Human Pose Estimation Specifically, the objectives are to:

- Develop a Flexible Graph Convolutional Network (Flex-GCN), which effectively captures higher-order dependencies among body joints for 3D Human Pose Estimation, thereby addressing limitations related to occlusion and depth ambiguity while maintaining computational efficiency.
- Design a Flexible Graph Kolmogorov-Arnold Network (FG-KAN) for 3D Human Pose Estimation, enabling flexible, data-driven feature transformations and multi-hop spatial reason-

ing to overcome the limitations of traditional GCNs in modeling complex joint dependencies and high-frequency pose variations.

1.4 Literature Review

The objective of 3D human pose estimation is to accurately infer the spatial coordinates of a set of biologically plausible joint landmarks, such as the head, torso, shoulders, elbows, hips, knees, and ankles, that collectively describe the full configuration of the human body in three-dimensional space. This task is typically performed from monocular or multi-view RGB images or video sequences, and serves as a critical foundation for downstream applications in motion analysis, activity recognition, animation, healthcare, and human-robot interaction. Unlike 2D pose estimation, which only captures joint positions in the image plane, 3D pose estimation must recover depth information, making the problem inherently ill-posed due to occlusions, projection ambiguity, and the absence of depth cues in monocular input.

In recent years, several lines of research have emerged to tackle the challenges of 3D human pose estimation using deep learning-based approaches. These efforts can broadly be categorized into (i) methods based on Graph Convolutional Networks (GCNs), which exploit the topological structure of the human skeleton and learn spatial dependencies between joints through graph-based message passing; (ii) Transformer-based models, which leverage self-attention mechanisms to model long-range dependencies and capture complex spatial-temporal relationships across joints and frames; and (iii) spatio-temporal methods that explicitly integrate both spatial configuration and temporal continuity, either through recurrent layers, temporal convolutions, or sequence modeling mechanisms. In this section, we present a comprehensive review of representative works in these three categories, highlighting their core methodologies, strengths, and limitations. Our discussion sets the stage for motivating the development of our proposed approaches, which builds on these foundations to address the remaining challenges in robust and efficient 3D human pose estimation.

Graph Convolutional Network based Methods. Given that the human skeletal structure can be naturally represented as a graph with joints as nodes and bones as edges, Graph Convolutional Networks (GCNs) have become increasingly prominent in the task of lifting 2D human poses to 3D[[3], [21], [6]]. Zhao *et al.* [3] introduced SemGCN, which employs semantic-aware graph convolutions alongside non-local blocks to effectively model both local and global joint dependencies for improved 3D pose estimation. Unlike Convolutional Neural Networks (CNNs), which operate on regular grid-like structures with fixed receptive fields, GCNs are inherently well-suited for capturing the irregular topology of the human skeleton. By explicitly encoding joint relationships as graph edges, GCNs facilitate the modeling of both spatially proximal and distal dependencies, thereby enabling more comprehensive and flexible feature aggregation across the entire pose graph. A key limitation of conventional Graph Convolutional Networks (GCNs) lies in their use of shared feature transformation weights across all nodes within a graph convolution layer. While this weight sharing reduces the number of trainable parameters and improves computational efficiency, it constrains the model's capacity to capture the diverse spatial and kinematic variations inherent in human motion. Liu *et al.* [21] presented one of the first comprehensive studies on weight sharing in Graph Convolutional Networks (GCNs) for 3D Human Pose Estimation. Their analysis systematically examined how weight sharing across different layers and network architectures affects model performance. To eliminate shared transformations, they introduced a pre-aggregation strategy in which each node's input features undergo unique transformations prior to aggregation, thereby enhancing the network capacity to capture joint-specific variations. Nonetheless, it notably enlarged the size of the model. Zou *et al.* [6] proposed a modulated GCN that retains a shared weight matrix but learns node-specific modulation vectors, enabling the network to distinguish among various joint connections while maintaining a compact model footprint. Similarly, Ci *et al.* [22] enhanced GCN expressiveness by removing weight sharing altogether and decoupling structural and transformation matrices, resulting in a locally connected graph framework that better captures joint-specific characteristics. Conventional GCNs primarily capture dependencies among immediate (first-order) neighboring joints, often neglecting higher-order relationships between more distant joints. This limitation hampers their ability to model long-range spatial dependencies critical for understanding complex body configurations. Recent advancements have shifted focus toward constructing richer, higher-order joint relationships, enabling more comprehensive and expressive representations of skeletal structures. Zou *et al.* [4] enhanced the modeling of long-range joint dependencies by aggregating features from nodes at varying graph distances, enabling the capture of remote inter-joint relationships. Quan *et al.* [5] employed multi-hop neighborhood aggregation to further strengthen the network capacity to model spatially distant dependencies. More recently, Li *et al.* [7] introduced a hybrid architecture that integrates multilayer perceptron (**MLP!**) with graph convolutional networks using SG-MLP and CG-MLP blocks. This design effectively captures both global contextual information and local structural patterns within human skeletal data.

Transformer-based HPE. The Transformer architecture, originally introduced as a sequence-to-sequence (seq2seq) model utilizing self-attention, comprises an encoder-decoder structure built from Multi-Head Attention mechanisms and Feed-Forward Networks. Unlike Recurrent Neural Networks (RNNs), which suffer from limited parallelization capabilities, and Convolutional Neural Networks (CNNs), which may fail to capture global dependencies, Transformers enable efficient parallel processing while preserving long-range contextual information. Their remarkable success in natural language processing (NLP) has motivated extensive exploration of their applicability in computer vision tasks. Recognizing the inherent ambiguity in projecting 2D human poses into 3D space—where multiple 3D configurations can correspond to the same 2D representation [23] introduced the Multi-Hypothesis Transformer (MHFormer). This model processes sequential 2D pose inputs using a cascaded Transformer architecture to generate multiple plausible 3D pose estimations, effectively addressing depth uncertainty. Additionally, MHFormer leverages temporal dynamics to model dependencies not only across the sequence but also within each individual pose hypothesis, thereby improving the robustness and accuracy of 3D pose predictions. Zheng *et al.* [24] introduced PoseFormer, a transformer-based architecture for 3D Human Pose Estimation that relies exclusively on transformer modules. The method employs two distinct transformer networks to separately capture spatial and temporal characteristics of 2D pose sequences derived from consecutive image frames. Specifically, the Spatial Transformer focuses on encoding high-dimensional spatial representations within individual frames, while the Temporal Transformer models global temporal dependencies across the entire sequence, enabling robust long-range pose reasoning. Building on the integration of convolutional and attention-based approaches, Shan *et al.* [25] proposed a regression-driven framework named MSRT (Multi-Scale Representation Transformer). This method retains convolutional neural networks (CNNs) for low-level feature extraction but introduces a Feature Aggregation Module (FAM) designed to partition and merge deep semantic features with fine-grained local details. The fusion of CNN-based representations with transformer-based global modeling enhances both accuracy and generalization in human pose regression. Wang *et al.* [26] proposed MTNet, a dual-branch architecture that leverages features from the third stage of HRNet and incorporates a multi-scale transformer to model global dependencies across HRNet's multi-resolution feature streams. This design enables more comprehensive spatial reasoning across different scales. Recent studies have introduced various enhancements to transformer-based architectures for 3D Human Pose Estimation. Li *et al.* [27] revised the feed-forward network of the Vanilla Transformer Encoder to more efficiently capture long-range dependencies while reducing computational overhead. Building upon Vitpose, Nicola *et al.* [28] proposed an unsupervised transformer model for pose reconstruction. Hong *et al.* [29] developed a transformer framework capable of adaptively integrating multi-view and temporal information. Cheng *et al.* [30] and Dong *et al.* [31] both employed multi-scale feature representations, combining up/down-sampling and CNNs with transformers to model spatial hierarchies across different resolutions. Yujun *et al.* [32] connected CNN and transformer modules to learn spatial relationships in 3D space, enhancing performance through residual connections.

Siyuan *et al.* [33] separated and recombined spatial and channel features to improve representation prior to transformer encoding. Dong *et al.* [34] extended transformer capabilities by incorporating third-order interactions via convolutional operations, achieving performance gains without additional computational cost. Kai *et al.* [35] introduced an enhancement transformer that processes temporal and spatial data separately, employing a custom similarity function to refine attention weights. Meanwhile, Hong *et al.* [36] fused CNNs and transformers in parallel, combining global and local cues and integrating spatio-temporal modules for robust joint motion aggregation across frames.

Spatial-temporal based Methods. Spatio-temporal modeling plays a pivotal role in 3D Human Pose Estimation (HPE), particularly when handling video data, where understanding both spatial structure and temporal consistency is essential. Temporal information contributes to robustness against occlusions, noise, and abrupt motion, enabling more accurate and temporally coherent pose predictions. Traditional methods relied heavily on Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) architectures to model temporal dependencies. These networks, as demonstrated by Hossain *et al.* [37], incorporated temporal smoothness constraints, enabling the prediction of consistent pose sequences. However, RNNs suffered from gradient issues and limited long-range dependency modeling. To address these, Temporal Convolutional Networks (TCNs) were introduced. Pavllo *et al.* [38] proposed dilated TCNs to capture long-term dependencies efficiently with reduced memory overhead. Overall, spatial-temporal based methods have evolved from sequential modeling with RNNs to advanced attention-based architectures. Transformer-based models now lead the field, offering flexible and powerful tools for integrating spatial structure and temporal dynamics. Future research may explore hybrid architectures that combine diffusion processes, semantic priors, or self-supervised learning to further boost robustness and generalization in unconstrained environments.

3D Human Pose Estimation. 3D Human Pose Estimation (HPE) from monocular images has gained significant attention in computer vision. Broadly, the approaches are categorized into one-stage (end-to-end) methods and two-stage (2D-to-3D lifting) methods. end-to-end estimation methods directly output the corresponding 3D pose information from the input images. In end-to-end estimation methods, deep learning models such as CNNs are typically employed to learn the mapping from images to 3D poses. These models extract image features and output corresponding pose estimations. Zhou *et al.* [39] proposed a weakly supervised transfer learning approach that utilized both 2D and 3D labels in combination. The 2D HPE subnetwork and the 3D depth regression subnetwork shared the same features, enabling 3D labels from indoor environments to be transferred to outdoor environments. In contrast to end-to-end models, two-stage methods decompose the 3D HPE pipeline into two sequential tasks: 2D pose estimation followed by 2D-to-3D lifting. Initially, a 2D keypoint detector extracts joint locations from monocular images. These 2D poses are then lifted to 3D using regression networks or optimization frameworks. Zhao *et al.* [40] utilized intermediate visual representations from pre-trained 2D pose detectors, leveraging the joint-centric spatial context encoded within these representations to significantly enhance the accuracy of 3D pose estimation. In the 2D keypoint detection stage, CNN-based keypoint detectors are commonly used, such as OpenPose [41] and hourglass [42]. In the 3D pose reconstruction stage, optimization methods such as adding geometric constraints to the deep learning model can be utilized to map 2D keypoints to 3D space.

While graph convolutional networks (GCNs) have demonstrated success in 3D Human Pose Estimation by modeling the human skeleton as a structured graph, their reliance on one-hop neighbor aggregation constrains their receptive field. This limitation hampers the ability to model long-range spatial dependencies critical for resolving occlusions and depth ambiguities and contributes to spectral bias, which favors low-frequency information while underrepresenting finer structural details. To address these challenges, we propose two complementary models. First, Flex-GCN, a flexible graph convolutional network, expands the neighborhood aggregation to include both immediate and second-order neighbors, enhancing the capture of global contextual features without increasing computational overhead. Flex-GCN is built with residual graph convolutional blocks and a global response normalization (GRN) module, enabling robust global feature calibration and improved network stability. Second, we introduce FG-KAN (Flexible Graph Kolmogorov-Arnold Network), a novel architecture that extends Kolmogorov-Arnold Networks to graph-based learning for 3D pose estimation. FG-KAN replaces traditional fixed activation functions with learnable transformations on graph edges, allowing the model to adaptively extract complex, high-frequency pose details. This design supports multi-hop aggregation and incorporates residual FG-KAN blocks with GRN layers, significantly improving spatial reasoning and joint localization. Extensive evaluations on standard benchmark datasets validate the effectiveness of both models. Flex-GCN and FG-KAN outperform existing methods in handling complex body configurations and occlusion scenarios, highlighting their ability to generalize better in real-world 3D Human Pose Estimation tasks.

1.5 Overview and Contributions

The rest of the thesis is structured as follows:

- In Chapter 2, we propose a graph convolutional network, Flex-GCN [43] ("Flexible graph convolutional network for 3D human pose estimation," in *Proc. British Machine Vision Conference*, 2024), which aggregates features from both first- and second-order neighbors without increasing computational complexity. The architecture integrates residual blocks and a global response normalization layer to enhance feature aggregation and calibration. Experimental results validate the model's effectiveness, showing competitive performance on standard benchmarks.

- In Chapter 3, we propose a flexible graph Kolmogorov-Arnold network (FG-KAN), which uses learnable edge functions for adaptive feature transformation. FG-KAN incorporates multi-hop aggregation, residual blocks, and global response normalization to enhance spatial awareness and feature refinement. Extensive experiments on benchmark datasets confirm its strong generalization and competitive performance.
- Chapter 4 summarizes the thesis's main contributions, acknowledges its limitations, and outlines potential future work directions.

CHAPTER 2

Flexible GCN for 3D Human Pose Estimation

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

2.1 Introduction

The objective of 3D Human Pose Estimation is to predict the 3D positions of body joints from images or videos. This task is essential for interpreting human movements and actions in various computer vision applications, including sports performance analytics and pedestrian behavior analysis [44]. For instance, accurately identifying skeletal joints is crucial for assessing sports activities, as it enables a meaningful evaluation of athletes' performance.

Existing approaches to 3D Human Pose Estimation can generally be categorized into one- and two-stage methods. One-stage approaches, also known as direct regression techniques, aim to predict 3D joint locations directly from input images or video frames without intermediary steps. However, these methods often face depth ambiguity, where multiple plausible 3D poses can explain the same 2D observations. They also struggle with complex poses and occlusions [45–48]. On the other hand, two-stage approaches, also known as 2D-to-3D lifting methods, typically consist of separate stages for joint detection and pose regression. The first stage detects 2D joint positions in the image, and the second stage uses these 2D detections to estimate the 3D joint positions. By incorporating an intermediate step for 2D joint detection, two-stage methods can mitigate challenges such as occlusions and depth ambiguity, resulting in more robust 3D pose estimations compared to their one-stage counterparts. Moreover, they allow for the use of different 2D pose detectors and lifting networks, providing flexibility in designing and optimizing each component separately, thereby potentially leading to higher accuracy [22,49–51].

Graph convolutional network (GCN)-based methods have recently demonstrated considerable promise in 3D Human Pose Estimation [3,6,21,52,53], leveraging the inherent graph structure of the human body, where joints serve as nodes interconnected by edges representing skeletal connections. By capitalizing on this representation, GCN-based models can effectively capture spatial dependencies crucial for accurate pose estimation. While these methods have demonstrated effectiveness in capturing dependencies between body joints, they are, however, inherently limited in their ability to model interactions beyond immediate neighbors. To overcome this challenge, recent approaches have introduced high-order graph convolutions [4,5,54], which enable information propagation through multiple hops in the graph, allowing the model to gather insights from not only immediate neighbors but also nodes located farther away. This enhances the model's capacity to capture global context and complex relationships between body joints. Another limitation of GCN-based methods is their inherent reliance on the adjacency matrix, which represents the connectivity between body joints in a graph, with non-zero entries indicating the presence of connections between neighboring joints. By modulating the adjacency matrix [6], we can incorporate additional information from nodes that are further apart in the graph, allowing the model to capture more complex dependencies and contextual cues.

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

Our contributions are summarized as follows:

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

2.2 Related Work

3D Human Pose Estimation aims to estimate the 3D coordinates of the joints in the human body from images or videos. One-stage and two-stage approaches are two common strategies employed in this task. One-stage methods directly regress the 3D pose from input images [45–48], while two-stage methods first predict intermediate representations, such as 2D joint locations, before lifting them to 3D space [22,49–51]. Two-stage methods, often combined with robust 2D joint detectors, typically exhibit better performance, particularly in addressing depth ambiguity challenges. Our proposed method falls under the category of two-stage approaches, with a network architecture design inspired by the ConvNeXt V2 framework [55], which leverages a global response normalization layer.

GCN-based methods for 3D Human Pose Estimation offer an intuitive paradigm by representing the human body as a graph structure [3,6,21,52,53], where the joints of the body serve as nodes and the connections between them represent the bones. This approach leverages the inherent spatial relationships between body parts, allowing for the modeling of complex human poses through graph-based representations. Also, by analyzing the connectivity patterns within the skeletal graph, GCN-based methods can infer the positions of individual joints based on information propagated from neighboring nodes, facilitating robust and contextually informed pose predictions. For instance, SemGCN [3] integrates semantic information into the graph convolution, allowing the model to combine structural information from the graph with semantic features derived from the data. In GroupGCN [53], convolutional operations are performed within distinct groups, each of which has its own weight matrix and spatial aggregation kernel. Weight Unsharing [21] analyzes the trade-offs between weight sharing and unsharing in GCNs. Modulated GCN [6] combines weight modulation to learn unique modulation vectors for individual nodes and adjacency modulation to account for additional edges beyond the human skeleton connections. One major limitation of the standard GCN architecture is that it typically operates with one-hop neighbors, which can restrict the ability of GCN-based methods to capture long-range dependencies and complex interactions within the graph. In other words, these methods provide a relatively local perspective of the graph structure, potentially overlooking long-range interactions and intricate dependencies present in human body movements. To mitigate this issue, High-order GCN [4] incorporates high-order dependencies among body joints by considering neighbors located multiple hops away during the update of joint features. Similarly, multi-hop Modulated GCN (MM-GCN) [54] involves modulating and fusing features from multi-hop neighbors. Our proposed model differs from these GCN-based approaches in that we employ a new update rule for graph node feature propagation that seamlessly integrates both first- and second-order neighboring information, combined with an initial residual connection, with the aim of learning graph representations that capture more global information and dependencies, while maintaining the time and memory complexity of the standard GCN. We also leverage adjacency modulation to learn additional connections between body joints.

2.3 Method

2.3.1 Preliminaries and Problem Statement

An attributed graph is a type of graph data structure where each node in the graph is associated with attributes or features. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an attributed graph, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_N)^\intercal$ an $N \times F$ feature matrix of node attributes (i.e., $\mathbf{x}_i$ is an F-dimensional row vector for node i). We denote by $\mathbf{A}$ an $N \times N$ adjacency matrix whose (i, j)-th entry is equal to 1 if i and j are neighboring nodes, and 0 otherwise. We also denote by $\mathbf{L} = \mathbf{I} - \hat{\mathbf{A}}$ the normalized Laplacian matrix, where $\hat{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is the normalized adjacency matrix, $\mathbf{D} = \operatorname{diag}(\mathbf{A}\mathbf{1})$ is the diagonal degree matrix, and $\mathbf{1}$ is an N-dimensional vector of all ones. Since the normalized Laplacian matrix is symmetric positive semi-definite, it admits an eigendecomposition given by $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\intercal$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ is an orthonormal matrix whose columns constitute an orthonormal basis of eigenvectors and $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ is a diagonal matrix comprised of the corresponding eigenvalues such that $0 = \lambda_1 \leq \cdots \leq \lambda_N \leq 2$ in increasing order [56].

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

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} l(f(\mathbf{x}_i), \mathbf{y}_i), \tag{2.1}$$

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

2.3.2 Spectral Modulation Filtering

Spectral graph filtering employs filters defined as functions of the graph normalized Laplacian (or its eigenvalues). The goal of these filters, often referred to as frequency responses or transfer functions, is to reduce or eliminate high-frequency noise in the graph signal. These functions basically describe how a filter affects the input graph signal to produce the output graph signal. We define a spectral modulation filter as follows:

$$h_s(\lambda) = \frac{1}{(1+s)\lambda - s\lambda^2}, \tag{2.2}$$

where $s \in (0, 1)$ is a positive scaling parameter that allows for the modulation or adjustment of the spectral characteristics, indicating its capability to control the filtering effect on different frequency components of the graph signal. The filter h_s is a rational polynomial function of the eigenvalues of the normalized Laplacian matrix. It is a flexible-pass filter in the sense that it exhibits low-pass characteristics, as it allows low-frequency components (corresponding to small eigenvalues) to flexibly pass through with little attenuation, while attenuating high-frequency components (associated with large eigenvalues). As shown in Figure 2.1, the attenuation behavior of the filter is determined by the scalar s, which serves as a tuning or modulation parameter. By adjusting s, we can control the trade-off between preserving low-frequency structural information and reducing high-frequency noise or variations in the graph signal. When s is large, the filter is less selective, allowing a wider range of frequencies to pass through with less attenuation. As s decreases, the filter becomes more selective and significantly attenuates higher frequencies, effectively filtering out more of the high-frequency noise or variations in the graph signal.

Graph Filtering System. Applying the spectral modulation filter on the graph signal $\mathbf{X} \in \mathbb{R}^{N \times F}$ yields a filtered graph signal $\mathbf{H}$ given by

$$\mathbf{H} = h_s(\mathbf{L})\mathbf{X} = ((1+s)\mathbf{L} - s\mathbf{L}^2)^{-1}\mathbf{X}, \tag{2.3}$$

which can be rewritten as

$$((1+s)\mathbf{L} - s\mathbf{L}^2)\mathbf{H} = \mathbf{X}, \tag{2.4}$$

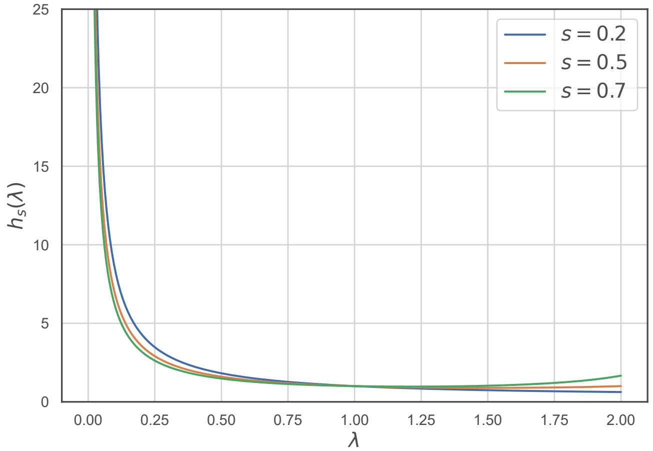


Figure 2.1: Transfer function of the spectral modulation filter. Lower values of the scaling parameter make the filter attenuate high-frequency components more strongly.

or equivalently

$$\begin{aligned}\mathbf{H} &= (\mathbf{I} - (1+s)\mathbf{L} + s\mathbf{L}^2)\mathbf{H} + \mathbf{X} \\ &= (\mathbf{I} - s\mathbf{L})(\mathbf{I} - \mathbf{L})\mathbf{H} + \mathbf{X}.\end{aligned} \tag{2.5}$$

Since $\mathbf{L} = \mathbf{I} - \hat{\mathbf{A}}$, the spectral modulation filter equation becomes

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

which can be solved using, for instance, the fixed point iteration method [57], where $\mathbf{H} = \varphi(\mathbf{H})$ with the function φ defined by the right-hand side term of Eq. (2.6).

Fixed Point Iterative Solution. Fixed-point iteration is an iterative numerical method used to find a fixed point of a given function [57]. The process involves repeatedly applying the function to an initial guess or estimate and updating this estimate in each iteration until it converges to the fixed point. For the spectral modulation filter equation $\mathbf{H} = \varphi(\mathbf{H})$, The fixed point iterative solution is given by

$$\mathbf{H}^{(t+1)} = ((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}}\mathbf{H}^{(t)} + \mathbf{X}, \tag{2.7}$$

with some initial guess $\mathbf{H}^{(0)}$, and $t \in \mathbb{N}$ denotes the iteration number.

2.3.3 Flexible Graph Convolutional Network

At the core of graph neural networks is the concept of feature propagation rule, which determines how information is passed between nodes in a graph. It involves updating the current node features by aggregating information from their immediate and high-order neighboring nodes, followed by a non-linear activation function to produce an updated representation for the node. Inspired by the fixed point iterative solution of the spectral modulation filter equation, we propose a flexible graph convolutional network (Flex-GCN) with the following layer-wise update rule for node feature propagation:

$$\mathbf{H}^{(\ell+1)} = \sigma\Big(((1-s)\mathbf{I} + s\hat{\mathbf{A}})\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)} + \mathbf{X}\widetilde{\mathbf{W}}^{(\ell)}\Big), \tag{2.8}$$

where $\mathbf{W}^{(\ell)}$ and $\widetilde{\mathbf{W}}^{(\ell)}$ are learnable weight matrices, $\sigma(\cdot)$ is an element-wise activation function, $\mathbf{H}^{(\ell)} \in \mathbb{R}^{N\times F_\ell}$ is the input feature matrix of the ℓ-th layer with F_ℓ feature maps for $\ell = 0,\ldots,L-1$. The input of the first layer is the initial feature matrix $\mathbf{H}^{(0)} = \mathbf{X}$. It is worth pointing out that the key difference between Eq. (2.7) and Eq. (3.6) is that the latter defines a representation updating rule for propagating node features layer-wise using trainable weight matrices for learning an efficient representation of the graph, followed by an activation function to introduce non-linearity into the network in a bid to enhance its expressive power.

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

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

For memory complexity, an L-layer Flex-GCN requires $\mathcal{O}(LNF + LF^2)$ in memory, where $\mathcal{O}(LNF)$ is for storing all embeddings and $\mathcal{O}(LF^2)$ is for storing all layer-wise weight matrices. Therefore, our proposed Flex-GCN model has the same time and memory complexity as that of GCN, albeit Flex-GCN takes into account both immediate and distant graph nodes for improved learned node representations. It is important to note that there is no need to explicitly compute the square of the normalized adjacency matrix in the Flex-GCN model. Instead, we perform right-to-left multiplication of the normalized adjacency matrix with the embedding. This process avoids the computational cost associated with matrix exponentiation and simplifies the computation, making our model more efficient while achieving its objectives.

2.3.4 Numerical Stability of Flex-GCN

In order to demonstrate the numerical stability of the proposed Flex-GCN model, we start with a useful result in matrix analysis [58], which states that the spectral radius of the sum of two commuting matrices is bounded by the sum of the spectral radii of the individual matrices.

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

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

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

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

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

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

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

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

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

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

Adjacency Modulation. We modulate the normalized adjacency matrix to capture not just the interactions between adjacent nodes, but also the relationships between distant nodes beyond the natural connections of body joints [6].

$$\breve{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}, \tag{2.9}$$

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., $(\mathbf{Q} + \mathbf{Q}^T)/2$.

Model Architecture. Figure 1 illustrates the architecture of our proposed Flex-GCN model for 3D Human Pose Estimation. The input to the model consists of 2D keypoints, obtained via an off-the-shelf 2D detector [59]. Following the architectural design of the ConvNeXt block [60], our residual block consists of three graph convolutional (Flex-GConv) layers. The first two convolutional layers are followed by layer normalization, while the third 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. The last convolutional layer of the model is preceded a global response normalization (GRN) [55], which aims to increase the contrast and selectivity of channels. The brain has a variety of systems that support neuron diversity. For instance, lateral inhibition [61,62] can enhance the contrast and selectivity of individual neurons to the stimulus, sharpen the response of the activated neuron, and increase the diversity of responses among the neurons in the population. Response normalization is one way that deep learning can apply this type of lateral inhibition [63]. The last Flex-GConv layer of the network generates the 3D pose.

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

$$\hat{\mathbf{Y}} = (\hat{\mathbf{y}}_1, \ldots, \hat{\mathbf{y}}_N)^{\intercal} \in \mathbb{R}^{N \times 3}, \tag{2.10}$$

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 Flex-GCN model for 3D Human Pose Estimation are learned by minimizing the following loss function

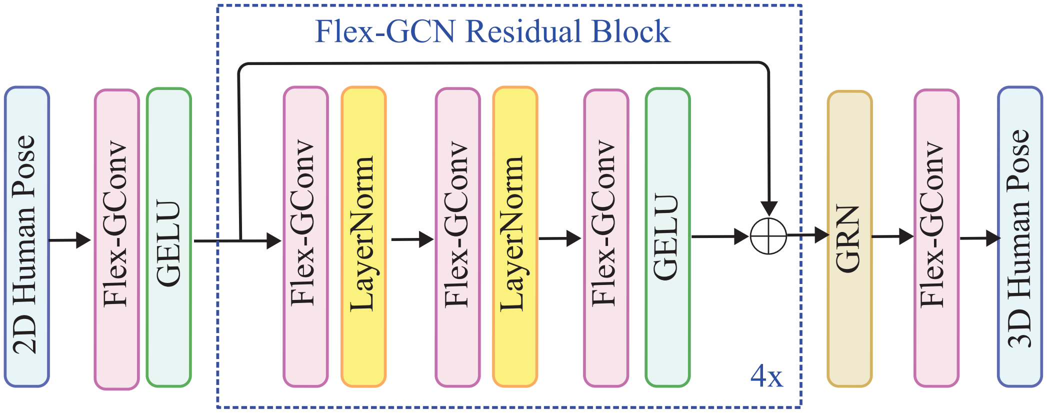


Figure 2.2: Network architecture of the proposed Flex-GCN model for 3D Human Pose Estimation. The model takes 2D pose coordinates (16 or 17 joints) as input and produces 3D pose predictions (16 or 17 joints) as output. It consists of ten Flex-GCN graph convolutional layers with four residual blocks. Within each residual block, the first two convolutional layers are followed by layer normalization, while the third convolutional layer is followed by a GELU activation function. The final convolutional layer is preceded by a Global Response Normalization (GRN).

$$\mathcal{L} = \frac{1}{N}\left[(1-\alpha)\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha\sum_{i=1}^{N}\|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1\right], \tag{2.11}$$

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. 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).

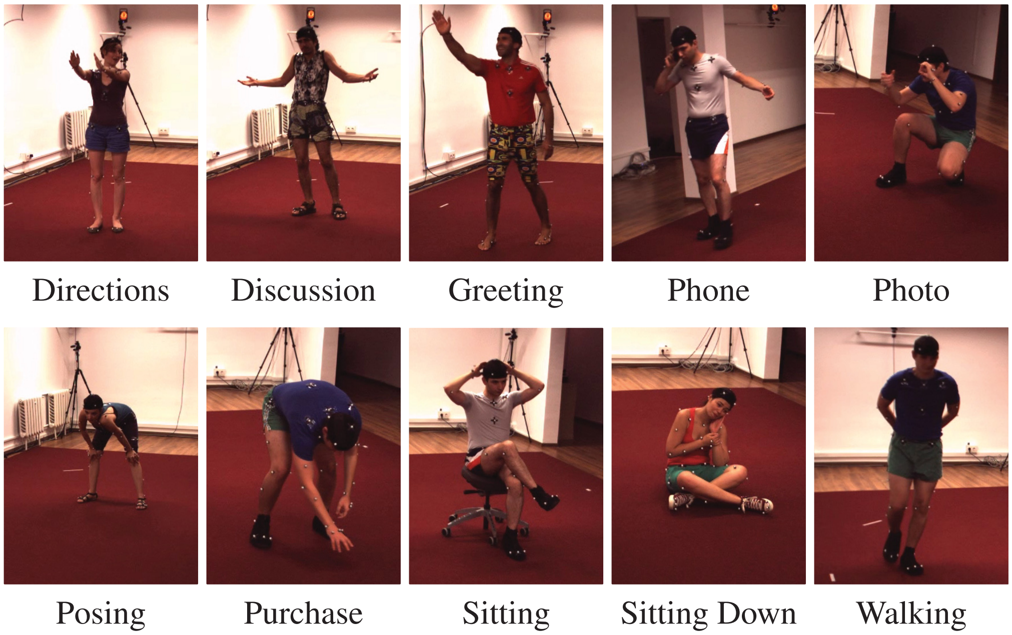
2.4 Experiments

2.4.1 Experimental Setup

Human3.6M The Human3.6M dataset comprises an extensive collection of 3.6 million images recorded at a rate of 50Hz using four synchronized cameras, offering a wide array of viewpoints and configurations. This dataset features 11 skilled actors (6 male, 5 female) performing 15 distinct activities (Directions, Discussion, Eating, Greeting, Phoning, Posing, Purchases, Sitting, Sitting

Down, Smoking, Photo, Waiting, Walk Dog, Walking, and Walk Together) within indoor settings, as depicted in Figure 2.3. Precise 3D coordinates of body joints are annotated using a motion capture system, which are then projected onto 2D poses using known camera parameters. The dataset includes annotations for seven subjects with detailed 3D joint information.The dataset is divided into two distinct sets for training and testing purposes. The training set comprises data from five actors (S1, S5, S6, S7, and S8), while the test set includes data from two additional actors (S9 and S11). Both sets are balanced, ensuring an equal number of samples for each activity and subject. Prior to model input, we preprocess the data by normalizing both 2D and 3D postures [3,4,49,64]. To achieve zero-centering, the hip joint is used as the root joint in 3D postures.

MPI-INF-3DHP MPI-INF-3DHP is a benchmark dataset designed for 3D Human Pose Estimation using monocular RGB images. It encompasses both indoor scenarios with limited space and complex outdoor scenes, as illustrated in Figure 2.4. The dataset includes video recordings of eight actors (4 male and 4 female) from 14 different angles, each performing 8 sets of activities that cover a wide range of pose categories. These activities range from simple movements like walking and sitting to more challenging workouts and dynamic actions compared to the Human3.6 million dataset. Each activity lasts approximately a minute, and the actors wore two different outfits that were alternated between. One clothing set consisted of casual attire suitable for daily use, while the other set was plain-colored to facilitate easy augmentation. Ground-truth annotations for the 3D joint positions are provided in the dataset.


Directions Discussion Greeting Phone Photo

Posing Purchase Sitting Sitting Down Walking

Figure 2.3: Examples of actions performed by different actors in the Human3.6M dataset [1].

Figure 2.4: Examples of activities in the MPI-INF-3DHP dataset [2].

Evaluation Protocols and Metrics. We utilized two standard evaluation protocols for training and testing on the Human 3.6M benchmark, designated as Protocol #1 and Protocol #2 [49]. We calculate the average Euclidean distance between the predicted and ground-truth 3D coordinates of each joint, after aligning the root joint (hip joint) which is recorded as the results under Protocol #1. Another commonly used metric for evaluating 3D Human Pose Estimation algorithms is the Procrustes-aligned mean per-joint position error (PA-MPJPE) used under Protocol #2. PA-MPJPE is computed after aligning the prediction with the ground truth using Procrustes analysis. This alignment involves scaling, rotating, and translating the predicted joint positions to match the ground-truth joint locations in a single coordinate system. The PA-MPJPE is then calculated as the mean of the Euclidean distances between the aligned predicted and ground-truth joint locations for each joint. Both MPJPE and PA-MPJPE are measured in millimeters (mm), with lower error levels indicating higher performance.Two individuals (S9 and S11) are used for testing, while five subjects (S1, S5, S6, S7, and S8) are trained in both Protocols #1 and Protocol #2 . Every action is trained using a single model that includes all camera views. We use Area Under Curve (AUC) for a variety of PCK thresholds and Percentage of Correct Keypoint (PCK) with a 150-mm threshold as assessment metrics for the MPI-INF-3DHP dataset. Within a certain distance threshold, the PCK and AUC metrics provide an indication of how closely the predicted joint positions match the ground-truth joint positions. Superior performance is indicated by higher PCK and AUC scores-For testing, two individuals (S9 and S11) are utilized, while five subjects (S1, S5, S6, S7, and S8) are trained in both Protocols #1 and Protocol #2. Each action is trained using a single model that encompasses all camera views. Assessment metrics for the MPI-INF-3DHP dataset include Area

Under Curve (AUC) for various PCK thresholds and Percentage of Correct Keypoint (PCK) with a 150-mm threshold. Both PCK and AUC metrics indicate how closely the predicted joint positions align with the ground-truth joint positions within a specified distance threshold. Higher PCK and AUC scores indicate superior performance [22, 65–69].

Baseline Methods. We compare the performance of our model with several state-of-the-art GCN-based methods for 3D pose estimation. Semantic Graph Convolutional Networks (SemGCN) [3] is a GCN-based model that simultaneously optimizes 3D joint positions and body joint angles using multi-task learning. Compositional GCN (CompGCN) [52] employs a hierarchical composition technique, dynamically weighting the contributions of different body parts at various hierarchy levels with a multi-level attention mechanism. High-order GCN [4] utilizes high-order GCNs to model complex relationships between body joints. Weight Unsharing [21] analyzes the trade-offs between weight sharing and unsharing in GCNs for different body components. Multi-hop Modulated GCN (MM-GCN) [54] is a multi-hop GCN method that uses modulated attention mechanisms to capture interactions between body joints over multiple hops. GroupGCN [53] is a GCN variant that combines group interaction and group convolution for 3D Human Pose Estimation. Finally, Modulated GCN [6] is a GCN-based architecture that employs adjacency and weight modulation techniques to capture intricate interactions among body parts.

Implementation Details. Our model is implemented in PyTorch and all experiments are conducted on a single NVIDIA GeForce RTX A4500 GPU with 20GB of memory. We employ the AMSGrad optimizer for training, running for 30 epochs on both 2D ground truth and 2D pose detections [59]. The initial learning rate is set to 0.001, with a decay factor of 0.99 every four epochs, a batch size of 512, and 384 channels. The hyperparameter S is determined as 0.2 through grid search and cross-validation on the training set. Additionally, we set the weighting factor α to 0.03. To prevent overfitting, we apply dropout with a factor of 0.2 after each graph convolution layer. We also include a Global Response Normalization (GRN) layer to enhance inter-channel feature competition [55] and a pose refinement module [70]. For improved performance, we add an extra pose refinement network consisting of two fully connected layers. However, in the ablation study, we exclude the pose refinement network to ensure fair comparability.

2.4.2 Results and Analysis

Quantitative Results on Human3.6M. In Tables 3.1 and 3.2, we provide a performance comparison of our Flex-GCN model to other cutting-edge approaches for 3D pose estimation on Human3.6M. In both tables, we present the results of all 15 actions, as well as the average performance. Table 3.1 shows that our technique outperforms all baselines, with an average MPJPE of 46.9mm and PA-MPJPE of 38.6mm. Under Protocol #1, Table 3.1 reveals that our Flex-GCN model performs better than Modulated GCN [6] in 14 out of 15 actions, yielding 2.5mm error reduction on average, improving upon this best performing baseline by a relative improvement of 5.08%, while maintaining a fairly small number of learnable parameters. Our method also consistently performs better in almost all actions and outperforms SemGCN [3] by a significant relative improvement of 18.57% on average. Apart from that, our model achieves better predictions than the best baseline on challenging actions like hard poses involving activities of self-occlusion such as "Eating", "Sitting" and "Smoking", showing relative error reductions of 1.53%, 7.47% and 8.24%, respectively, in terms of MPJPE. The presence of self-occlusions during activities can pose challenges for Human Pose Estimation, as they restrict the model's access to observable information. For instance, some activities like eating or smoking can lead to occlusions where a person's hands and arms obstruct parts of their face and upper body besides, when sitting, a person's legs and arms may obstruct other body parts such as the torso or feet.

Table 2.1: Comparison of our model and baseline methods in terms of Mean Per Joint Position Error (MPJPE) in millimeters, computed between the ground truth and estimated poses on the Human3.6M dataset under Protocol #1. The last column displays the average errors, with boldface numbers denoting the best 3D pose estimation performance and underlined numbers indicating 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.* [49]	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.* [48]	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.* [67]	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.* [71]	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 [72]	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.* [66]	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.* [3]	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.* [4]	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.* [5]	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.* [52]	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.* [21]	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.* [6]	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
Lee *et al.* [54]	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.* [53]	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
Ours	**40.2**	**45.8**	**45.0**	**46.8**	**48.6**	54.0	**42.4**	**42.1**	**53.2**	66.7	**45.6**	**45.4**	48.8	**38.4**	**40.1**	**46.9**

Under Protocol #2, Table 3.2 shows that our model on average reduces the error by 1.28%

compared to Modulated GCN [6], and achieves better results in 11 out of 15 actions, with same performance in the action "Phone". Also, our method outperforms Modulated GCN on the challenging actions of "Greeting", "Sitting" and "Smoking", yielding relative error reductions of 2%, 4.74% and 5.67%, respectively, in terms of PA-MPJPE. Moreover, our model performs better than Modulated GCN on the challenging "Photo" action, yielding a relative error reduction of 2%. In addition, Flex-GCN outperforms High-order GCN [4] by a relative improvement of 11.67% on average, as well as on all actions.

Table 2.2: Comparison of our model and baseline methods in terms of Procrustes-aligned Mean Per Joint Position Error (PA-MPJPE), computed between the ground truth and estimated poses on the Human3.6M dataset under Protocol #2.

	Action															
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Pavlakos *et al.* [47]	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.* [39]	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.* [49]	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.* [48]	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.* [71]	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 [72]	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.* [77]	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.* [4]	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.* [5]	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.* [52]	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.* [21]	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.* [6]	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.* [54]	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.* [53]	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	**34.1**	**38.0**	36.8	**39.7**	**39.2**	**43.6**	**33.4**	**34.5**	**44.2**	57.1	**38.3**	36.0	41.0	**29.9**	**33.1**	**38.6**

Cross-Dataset Results on MPI-INF-3DHP. In Table 3.3, We evaluate the generalization ability of our method by comparing it against strong baselines using different datasets. Our model is trained on the Human3.6M dataset and evaluated on the MPI-INF-3DHP dataset. Results demonstrate that our approach achieves the highest PCK and AUC scores, consistently outperforming the baseline methods across various indoor and outdoor scenes. Compared to the best performing baseline, our model shows relative improvements of 1.05% in PCK and 2.9% percent in terms of AUC metrics. Despite being trained only on indoor scenes from the Human3.6M dataset, our model demonstrates satisfactory performance in outdoor settings, highlighting its strong generalization capabilities to unseen scenarios and datasets.

Table 2.3: Comparison of the performance of our model with baseline methods on the MPI-INF-3DHP dataset, using PCK and AUC as evaluation metrics.

Method	PCK (↑)	AUC (↑)
Martinez *et al.* [49]	42.5	17.0
Mehta *et al.* [2]	64.7	31.7
Li *et al.* [78]	67.9	-
Yang *et al.* [67]	69.0	32.0
Zhou *et al.* [39]	69.2	32.5
Habibie *et al.* [65]	70.4	36.0
Pavlakos *et al.* [66]	71.9	35.3
Wang *et al.* [79]	71.9	35.8
Quan *et al.* [5]	72.8	36.5
Ci *et al.* [22]	74.0	36.7
Zhou *et al.* [80]	75.3	38.0
Zeng *et al.* [81]	77.6	43.8
Liu *et al.* [21]	79.3	47.6
Zhou *et al.* [52]	79.3	45.9
Xu *et al.* [64]	80.1	45.8
Zeng *et al.* [68]	<u>82.1</u>	46.2
Lee *et al.* [54]	81.6	<u>50.3</u>
Zhang *et al.* [53]	81.1	49.9
Ours	**85.2**	**51.8**

Qualitative Results. Figure 3.4 displays the visual results obtained by Flex-GCN on the Human3.6M dataset for various actions are presented. The effectiveness of our proposed approach in addressing the 2D-to-3D pose estimation challenge is demonstrated by the close alignment between the predicted 3D poses and the ground truth, as demonstrated by the accurate outcomes produced by our model from input images. In comparison to Modulated GCN, our model generates posture estimations that closely resemble real poses, particularly in challenging scenarios involving self-occlusion.

Quantitative Results using Ground Truth. We compared our model with GCN-based methods, including SemGCN [3], High-order GCN [4], HOIF-Net [5], Weight Unsharing [21] and, Modulated GCN [6] using ground truth. The results are reported in Table 4, which shows that our model consistently yields better performance than GCN-based approaches under 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, Modulated GCN, and Weight Unsharing by 3.66mm, 1.2mm, 0.19mm, and 0.22mm, with relative error reductions

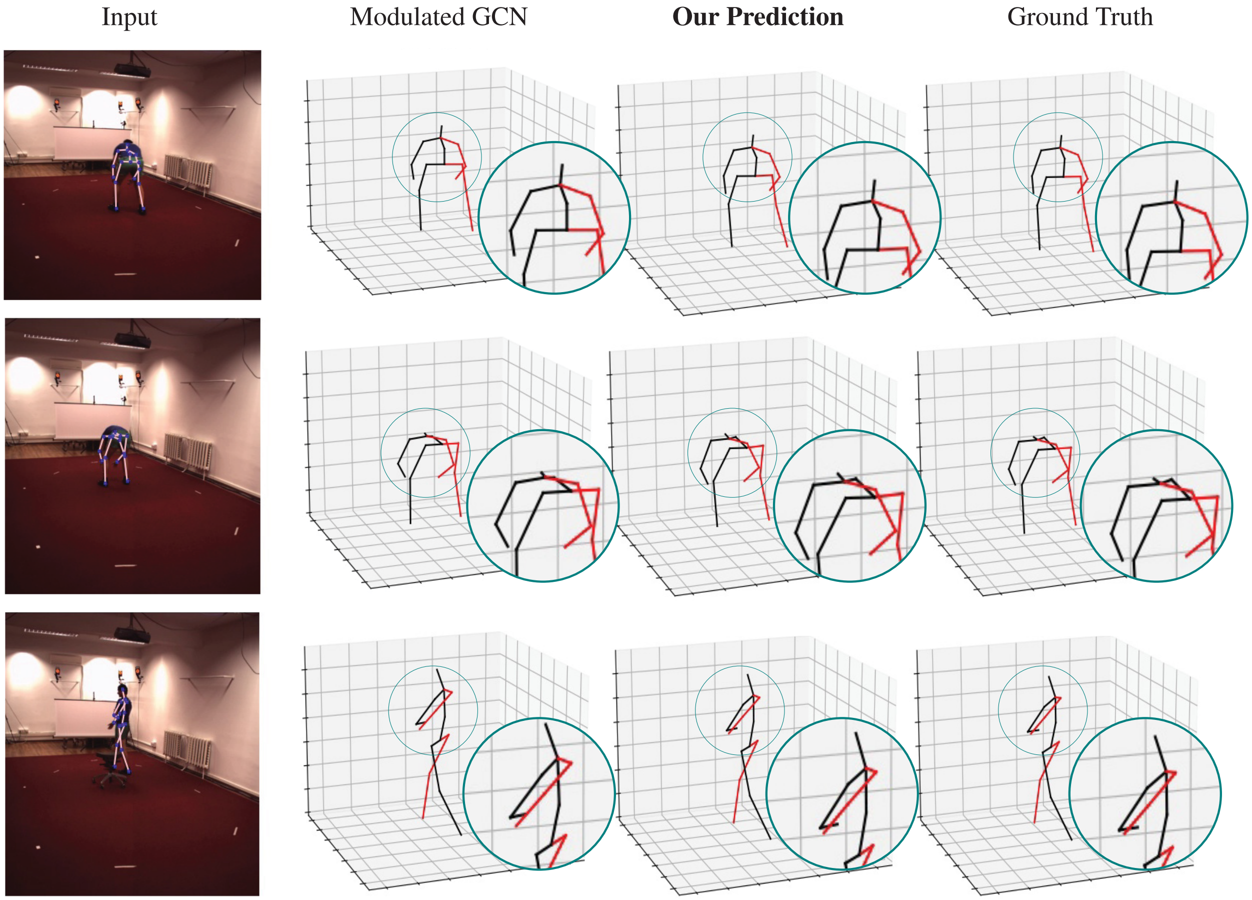


Figure 2.5: Visual comparison between Flex-GCN, Modulated GCN and ground truth on the Human3.6M test set. our model is able to produce better predictions compared to Modulated GCN.

of 10.91%, 3.86%, 0.63%, and 0.73%, respectively.

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

Method	MPJPE (↓)	PA-MPJPE (↓)
SemGCN [3]	42.14	33.53
High-order GCN [4]	39.52	31.07
HOIF-Net [5]	38.12	**29.74**
Modulated GCN [6]	38.25	30.06
Weight Unsharing [21]	37.83	30.09
Ours	**37.41**	29.87

2.4.3 Ablation Study

Impact of Skip Connection. We analyze the impact of the initial residual connection (IRC) in the layer-wise propagation rule on our model performance, and the results are reported in Table 3.5 (left). The inclusion of IRC shows significant improvements, with relative error reductions of 6.58% and 4.41% in terms of MPJPE and PA-MPJPE, respectively. This demonstrates the effectiveness of IRC in enhancing our model's accuracy. By preserving and reinforcing the initial node features throughout the layers, IRC facilitates more stable and effective learning, ensuring that essential positional information is maintained and utilized in subsequent layers.

Table 2.5: Effect of initial residual connection (IRC).

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o IRC	39.76	31.25
w IRC	**37.41**	**29.87**

Impact of Pose Refinement. We also assess the effectiveness of the pose refinement network. The outcomes presented in Table 2.6 indicate that the mean MPJPE and PA-MPJPE errors are reduced by 3.74mm and 1mm, respectively, highlighting the benefit of incorporating pose refinement for improved performance across both protocols. These results are visualized in Figure 2.6, which illustrates the performance contrast with and without the pose refinement model for various challenging actions such as "Eating" and "Photo" under Protocol #1 (top) and Protocol #2 (bottom). For instance, the "Eating" action demonstrates relative reductions in error of 10.62% and 5.98% in terms of MPJPE and PA-MPJPE, respectively.

Table 2.6: Effect of the pose refinement network (PRN). Boldface numbers indicate better performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o PRN	37.41	29.87
w PRN	**33.45**	**28.16**

Impact of Symmetrizing Adjacency Modulation. we assess the effect of symmetrizing the learnable adjacency modulation matrix on model performance. The results, presented in Table 3.6), indicate that introducing symmetry into the adjacency modulation process yields tangible reductions in both MPJPE and PA-MPJPE. Specifically, the MPJPE error decreased by 0.58mm and the PA-MPJPE error decreased by 0.24mm, compared to the configuration without symmetry. This

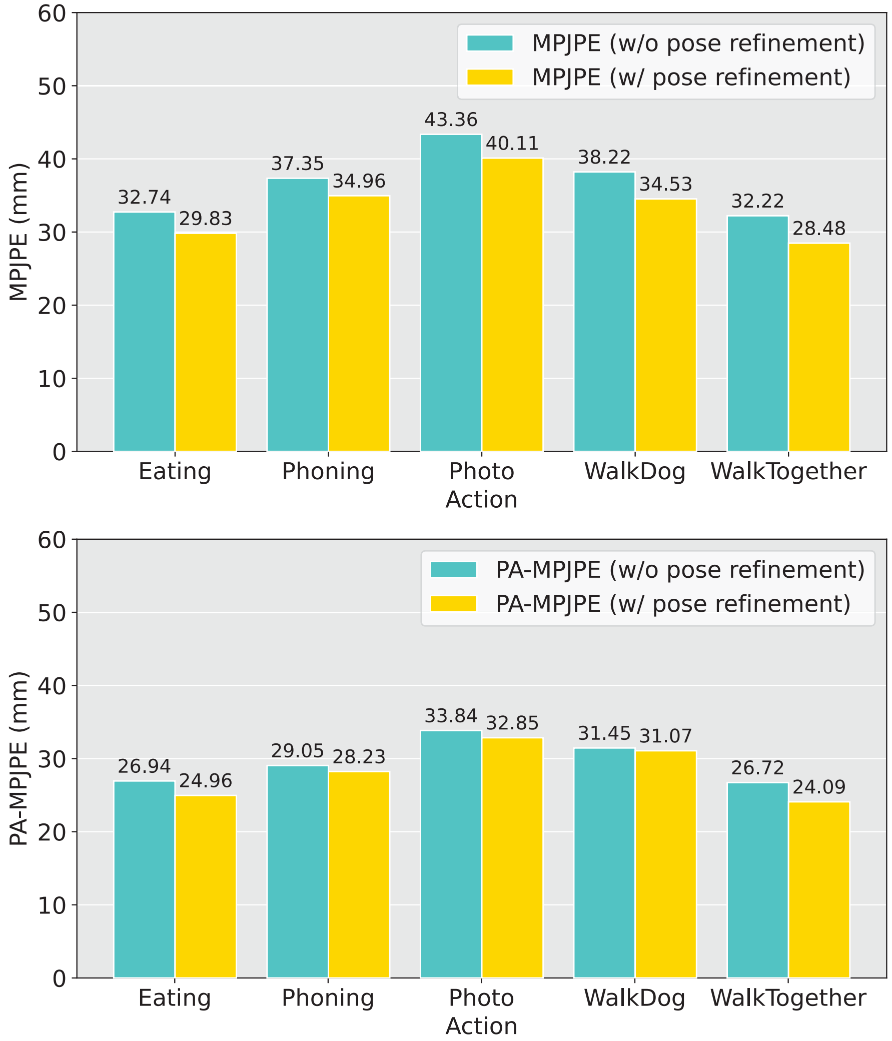


Figure 2.6: Performance evaluation of our model with and without pose refinement using MPJPE (top) and PA-MPJPE (bottom). Integration of a pose refinement network consistently enhances the performance of Flex-GCN, particularly on challenging actions.

improvement highlights the advantage of leveraging skeleton graph symmetry to enhance the precision of our model's estimations. By ensuring that the relational information between joints is consistently balanced, the symmetric adjacency modulation matrix enables more accurate and reliable pose estimations. This approach not only refines the positional accuracy of individual joints but also improves the overall coherence of the estimated poses, demonstrating the critical role of structured graph modulation in 3D Human Pose Estimation.

Impact of Batch and Filter Size. In Figure 3.5, we analyze the impact of varying batch and filter sizes on our model's performance. This analysis is crucial as both parameters play a significant role in the training efficiency and the overall accuracy of the model. Batch size directly influences the stability and speed of the training process. Smaller batch sizes can lead to noisier gradient

Table 2.7: Effect of symmetrizing adjacency modulation.

Method	MPJPE (↓)	PA-MPJPE (↓)
w/o Symmetry	37.99	30.11
w Symmetry	**37.41**	**29.87**

estimates but allow for more frequent updates, potentially improving generalization. Larger batch sizes, on the other hand, provide more stable gradient estimates but require more memory and can lead to slower convergence. Filter size, which determines the number of learnable parameters in each layer of the network, affects the model's capacity to capture complex patterns. A larger filter size increases the model's ability to learn intricate features but also raises the risk of overfitting. Conversely, a smaller filter size may lead to underfitting. Our analysis indicates that a batch size of 512 and a filter size of 384 result in the best performance. This combination yields the lowest MPJPE and PA-MPJPE values, respectively, signifying that the model is accurately estimating 3D human poses. The batch size of 512 strikes a balance between gradient stability and update frequency, while the filter size of 384 provides a sufficient number of parameters to learn detailed features without overfitting.

Hyperparameter Sensitivity Analysis. We examine the effect of the scaling parameter s on model's performance by plotting error metrics against s across a range of values between 0 and 1. Figure 3.6 illustrates that smaller scaling factors often lead to better results, with our model achieving the lowest error values for MPJPE and PA-MPJPE at $s = 0.2$.

2.4.4 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 2.8, which shows that our model significantly outperforms strong baselines.

2.5 Discussion

In this section, we outline the merits of the proposed Fix-GCN model in three key aspects:

- *Flexibility.* The Flex-GCN model introduces a flexible and learnable adjacency modulation mechanism, allowing the model to dynamically capture complex dependencies between joints in 3D Human Pose Estimation. This adaptability enhances the model's ability to gen-

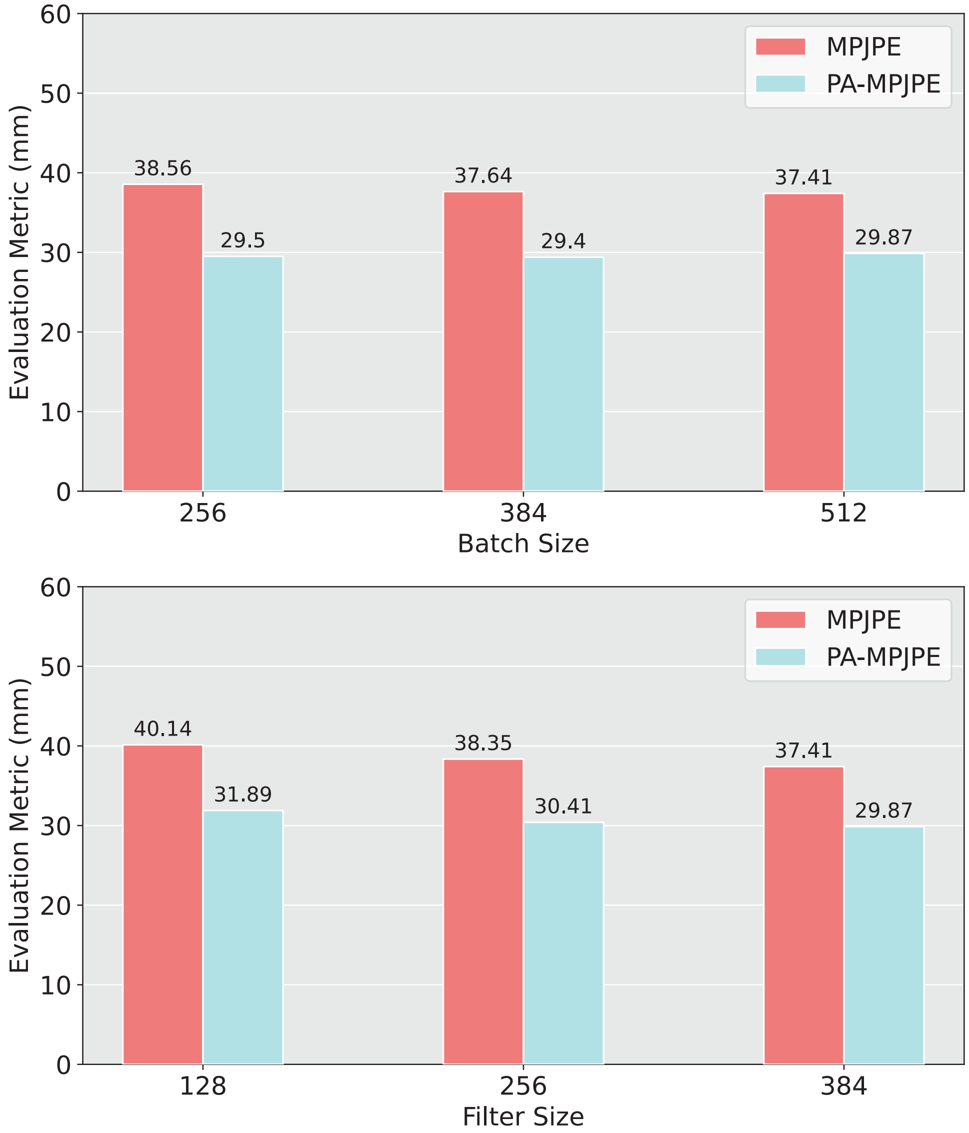


Figure 2.7: Performance of our proposed Flex-GCN model on the Human3.6M dataset using varying batch and filter sizes.

eralize across different poses and datasets. By incorporating initial residual connections in the layer-wise propagation rule, Flex-GCN preserves crucial node feature information, leading to significant improvements in model accuracy. This ensures that the propagation of features is robust and effective, contributing to better overall performance.

- *Performance.* The Flex-GCN model demonstrates substantial reductions in key error metrics such as MPJPE and PA-MPJPE. The inclusion of IRC leads to relative error reductions of 6.58% and 4.41% in terms of MPJPE and PA-MPJPE, respectively. Additionally, the symmetrization of the adjacency modulation matrix further decreases the MPJPE by 0.58mm

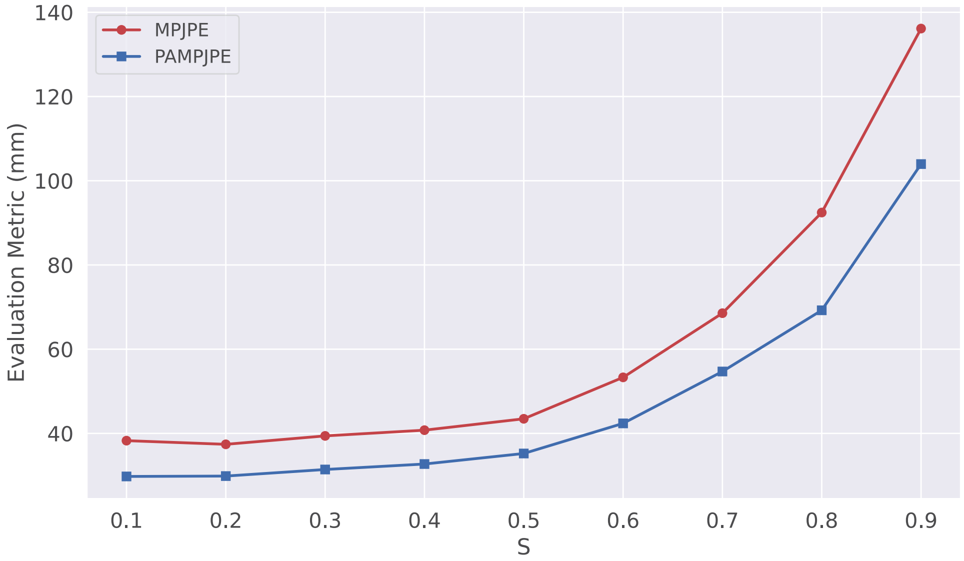


Figure 2.8: Analysis of the model's sensitivity to the selection of the hyperparameter s. Smaller values of s typically lead to reduced MPJPE and PA-MPJPE errors.

Table 2.8: Analysis of the execution time of our model compared to other competing baseline methods.

Method	Inference Time
SemGCN [3]	.012s
High-Order GCN [4]	.013s
HOIF-Net [5]	.016s
Weight Unsharing [21]	.032s
MM-GCN [54]	.009s
Modulated GCN [6]	.010s
Ours	0.06s

and PA-MPJPE by 0.24mm, highlighting the model's precision in 3D Human Pose Estimation. Moreover, the analysis of different batch and filter sizes reveals optimal configurations (batch size of 512 and filter size of 384) that result in the best performance. This optimization ensures that the model operates efficiently while maintaining high accuracy.

- *Efficiency.* The Flex-GCN model is designed to be computationally efficient, significantly outperforming strong baselines in terms of inference time. This efficiency is crucial for practical applications where real-time processing is required. By incorporating symmetry into the adjacency modulation process, the model achieves improved precision in estimations. This approach not only enhances accuracy but also ensures that the model can effectively handle various human poses and movements.

Despite its versatility and enhanced performance, Flex-GCN may still face challenges in accurately estimating extremely complex or rare human poses. Scenarios involving occlusions, interactions with objects, or unusual body movements can present difficulties, potentially impacting the model's robustness in diverse real-world applications.

CHAPTER 3

Graph KAN for 3D Human Pose Estimation

In this chapter, Graph convolutional network (GCN)-based methods have shown strong performance in 3D Human Pose Estimation by leveraging the natural graph structure of the human skeleton. However, their receptive field is inherently constrained by one-hop neighbor aggregation, limiting their ability to capture long-range dependencies essential for handling occlusions and depth ambiguities. They also exhibit spectral bias, which prioritizes low-frequency components while struggling to model high-frequency details. In this chapter, we introduce a flexible graph Kolmogorov-Arnold Network (FG-KAN), a novel framework that extends KANs to graph-based learning for 3D Human Pose Estimation. Unlike GCNs that use fixed activation functions, KANs employ learnable functions on graph edges, allowing data-driven, flexible feature transformations. This enhances the model's adaptability and expressiveness, making it more expressive in learning complex pose variations. Our model employs multi-hop feature aggregation, ensuring the body joints can leverage information from both local and distant neighbors, leading to improved spatial awareness. It also incorporates residual FG-KAN blocks for deeper feature refinement, and a global response normalization for improved feature selectivity and contrast. Extensive experiments on benchmark datasets demonstrate the competitive performance of our model against state-of-the-art methods, highlighting its effectiveness in complex spatial relationships and improving generalization.

3.1 Introduction

Estimating 3D human poses from 2D images or videos is a fundamental task in computer vision, central to applications in sports analytics, human-computer interaction, virtual reality, robotics, and autonomous systems. This task aims to predict the 3D coordinates of body joints, providing crucial geometric and motion information that enables a deeper understanding of human movement dynamics. However, a major challenge in 3D pose estimation lies in recovering depth information from 2D inputs, which often results in depth ambiguities and occlusions.

Mainstream approaches to 3D Human Pose Estimation can be broadly categorized into one- and two-stage methods. One-stage methods, often referred to as direct regression techniques, aim to predict 3D joint coordinates directly from input images or video frames, bypassing any intermediate representation. While this end-to-end approach is computationally appealing and eliminates dependencies on separate components, it faces fundamental challenges. One major issue is depth ambiguity, where multiple plausible 3D poses can project to the same 2D image, making it difficult for the model to disambiguate between different spatial configurations. Moreover, these methods often struggle with complex human poses and occlusions, where self-occlusions hide parts of the body, leading to inaccurate pose predictions [45–48]. In contrast, two-stage methods, also known as 2D-to-3D lifting approaches, decompose the problem into two separate stages: (1) detecting 2D joint locations in the image and (2) lifting these 2D detections into 3D space. This paradigm provides several advantages over direct regression techniques. By first detecting 2D poses, the model can leverage state-of-the-art 2D human pose detectors, which have been extensively optimized on large-scale datasets. The subsequent lifting stage can then focus on recovering depth information, thereby mitigating occlusion-related challenges and depth ambiguities. More importantly, the modular nature of two-stage pipelines helps optimize each component, selecting the best-performing 2D pose detectors and combining them with robust 2D-to-3D lifting models, ultimately leading to better 3D pose estimations and greater flexibility [22,49–51].

GCN-based methods have recently emerged as a powerful approach for 3D Human Pose Estimation, effectively leveraging the inherent skeletal structure of the human body [3,6,21,52,53]. These methods represent human poses as graphs, where joints serve as nodes, and edges represent skeletal connections. By formulating pose estimation as a graph-based learning problem, GCN-based models can effectively capture spatial dependencies between body parts, facilitating the refinement of joint locations based on their structural relationships. One of the primary strengths of GCNs lies in their ability to propagate information across the skeletal structure, ensuring that joint predictions are informed by neighboring joints rather than being processed in isolation. This relational reasoning improves the robustness of pose predictions, particularly when dealing with occlusions, depth ambiguities, and complex body configurations. Moreover, GCN-based architectures allow for flexible graph modeling, where different adjacency matrices can be designed to encode learned connectivity patterns, or even modulated connections [6].

Motivation and Challenges. While GCN-based methods have demonstrated strong performance [3, 6, 21, 52, 53], they still encounter several limitations that hinder their effectiveness in challenging 3D pose estimation scenarios. One of the primary limitations is their inability to model global context effectively due to their reliance on aggregating information mostly from immediate neighbors, thereby struggling to capture long-range dependencies between distant joints. However, many body movements involve coordinated motion between joints that are not directly connected in the skeletal graph (e.g., synchronized hand movements or upper-body coordination during walking). The inability to propagate information beyond local neighborhoods restricts the model's ability to capture global pose structures, making it less effective in handling occlusions and depth ambiguities. Moreover, most GCN-based models incorporate Multi-Layer Perceptrons (MLPs) as their core components for feature transformation, inheriting a fundamental drawback of MLPs, namely spectral bias [82]. As a result, these models tend to prioritize low-frequency components while struggling to capture high-frequency details, which are crucial for accurately modeling rapid movements and fine-grained joint interactions. Since GCNs utilize MLPs to transform node features after aggregation, they inherently exhibit spectral bias, leading to suboptimal performance in tasks that require capturing complex joint coordination patterns. This limitation can significantly impact pose estimation accuracy, particularly in highly articulated actions such as running, dancing, or sports activities, where fine-grained motion details are essential for accurate predictions. Furthermore, another drawback of GCN-based approaches is their lack of interpretability due to their reliance on predefined activation functions. These approaches do not provide insight into how individual joints contribute to the final prediction, making it challenging to analyze learned representations and refine the model for better generalization. Addressing these challenges requires a more flexible and expressive approach that can dynamically adapt feature transformations, model long-range dependencies, and mitigate spectral bias.

Proposed Work and Contributions. To address the aforementioned challenges, we introduce a Flexible Graph Kolmogorov-Arnold Network (FG-KAN), a novel graph-based framework that incorporates learnable function-based transformations. Drawing inspiration from the Kolmogorov-Arnold representation theorem [83, 84], KANs [85] have recently been introduced as a compelling alternative to MLPs, offering significant improvements in both interpretability and adaptability for function approximation tasks while mitigating spectral bias [86]. Instead of applying fixed activation functions at nodes, FG-KAN learns activation functions dynamically on graph edges, providing greater flexibility in feature transformation. We also devise a multi-hop feature aggregation scheme that uses a scaling parameter to balance local and global information propagation, ensuring long-range dependencies are effectively captured. To extend beyond skeletal graph constraints, we employ a learnable adjacency modulation matrix that allows dynamic adjustments to the connectivity between joints, while maintaining graph symmetry for structural consistency. Although our model incurs a slight computational overhead compared to GCN-based methods, it significantly improves performance, as illustrated in Figure 3.1, which compares the performance and model size of FG-KAN with state-of-the-art methods. The main contributions of our work can be summarized as follows:

- We present a flexible graph Kolmogorov-Arnold network, a novel framework that employs learnable function-based transformations, improving adaptability and generalization while maintaining computational efficiency.
- We design a multi-fop feature propagation scheme that extends beyond one-hop aggregation by introducing a scaling parameter that controls the balance between local and global feature aggregation, improving robustness to occlusions and depth ambiguities.
- The effectiveness of our model is rigorously validated through comprehensive experimental evaluation, including a comparative analysis and ablation studies against state-of-the-art methods on two standard 3D Human Pose Estimation benchmarks, demonstrating improved performance.

The rest of this chapter is structured as follows: Section 3.2 reviews related work. Section 3.3 outlines our methodology, covering the problem statement, flexible graph KAN layer, model architecture, training and prediction procedures. Section 3.4 details the experimental setup, and presents quantitative and qualitative results, as well as ablation studies. We conclude the chapter in Section 3.5 with a discussion of our contributions.

3.2 Related Work

Since our FG-KAN model follows the 2D-to-3D lifting pipeline, we primarily focus in this section on methods within this category while also discussing relevant GCN-based approaches and how KANs fit into the broader landscape of graph-based learning for 3D Human Pose Estimation.

3D Human Pose Estimation. The aim of 3D Human Pose Estimation is to predict the 3D coordinates of body joints from 2D images or video frames, playing a crucial role in a wide range

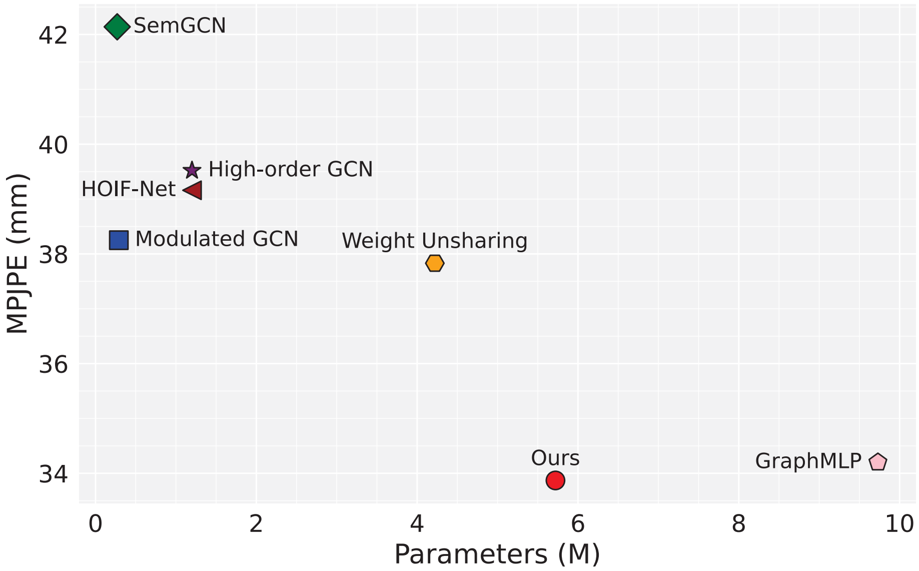


Figure 3.1: Performance and model size comparison between our model and state-of-the-art methods for 3D Human Pose Estimation, including SemGCN [3], High-order GCN [4], HOIF-Net [5], Modulated GCN [6] and GraphMLP [7]. Lower Mean Per Joint Position Error (MPJPE) values indicate better performance. Evaluation is conducted on the Human3.6M dataset with ground truth 2D joints as input.

of applications, including action recognition, motion analysis, and human-computer interaction. The 2D-to-3D lifting pipeline, also known as two-stage approaches, first detects 2D joint positions and then maps them to their 3D coordinates. Compared to one-stage methods, which directly regress 3D joint locations from images [46, 47] without an intermediate representation, two-stage approaches have demonstrated greater robustness, particularly in handling occlusions and depth ambiguities [22, 49–51]. For instance, Martinez *et al.* [49] introduced a fully connected network that predicts 3D joint locations directly from 2D joint inputs. Despite their success, Most existing 2D-to-3D lifting methods utilize fully connected networks, which inherently lack spatial awareness and are prone to overfitting due to their densely connected architecture. Our approach falls under the category of two-stage approaches, focusing on learning flexible graph representations for more accurate 2D-to-3D pose lifting.

Graph-based 3D Human Pose Estimation. Human body joints naturally form a structured graph, where nodes represent joints and edges capture relationships between them. Graph-based learning approaches, particularly GCNs [87], have been widely applied to exploit this structured representation, showing promising results by leveraging the skeletal graph structure to model body joint dependencies. For instance, SemGCN [3] introduced semantic information into graph convolutions, enhancing feature aggregation among body joints. Weight Unsharing [21] explored trade-offs between shared and unshared graph convolutions, improving feature representation. However, most GCN-based methods primarily aggregate information from one-hop neighbors, limiting their ability to model long-range dependencies crucial for understanding complex poses. To address this, High-order GCN [4] extended GCNs to multi-hop neighborhoods, capturing long-range dependencies beyond immediate joint connections. Inspired by the Jacobi's method, HOIF-Net [5] also proposed a higher-order graph convolutional framework with initial residual connections for 2D-to-3D pose lifting using implicit fairing on graphs. Modulated GCN [6] introduced adjacency matrix modulation, enabling the model to learn additional edges beyond the predefined skeletal structure. GroupGCN [53] and Multi-hop Modulated GCN (MM-GCN) [54] further refined graph feature propagation by fusing multi-hop neighborhood information. More recently, GraphMLP [7] combined MLPs and GCNs to leverage both global and local skeletal interactions by learning features with a global receptive field and aggregating local information between neighboring joints, respectively. While graph-based approaches have led to notable improvements in 3D pose estimation, they still face three key limitations. First, standard GCN architectures rely on one-hop neighborhood aggregation, meaning each joint updates its features based only on its immediate neighbors. This restricts the ability to capture long-range dependencies, which are crucial for handling occlusions and depth ambiguities. Second, like MLPs, GCNs exhibit spectral bias, meaning they favor low-frequency components while struggling to capture high-frequency details, thereby hindering their ability to accurately represent complex poses and rapid movement changes. Third, GCN-based methods apply predefined activation functions at the node level, restricting adaptability. By comparison, our FG-KAN framework learns activation functions dynamically on graph edges, providing greater flexibility in feature transformation and improving interpretability, while mitigating the spectral bias thanks in large part to learning function-based transformations that help capture both low- and high-frequency components. Low-frequency components represent smooth, gradual variations in joint positions over time or space, whereas high-frequency components correspond to rapid, abrupt changes in motion. In the Human3.6M dataset [1], for instance, actions such as Walking, Sitting, and Eating predominantly exhibit low-frequency characteristics due to their smooth and predictable patterns. In contrast, high-frequency components are more prominent in actions like Taking Photos, Smoking, and Greeting, where sudden movements and finer pose adjustments are required.

3.3 Method

In this section, we first describe the task at hand, followed by a preliminary background on KANs [85, 86]. Then, we introduce our flexible graph KAN model.

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

Given a training set $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ comprised of 2D joint positions $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^2$ and their associated ground-truth 3D joint positions $\mathbf{y}_i \in \mathcal{Y} \subset \mathbb{R}^3$, the aim of 3D Human Pose Estimation is to learn the parameters $\mathbf{w}$ of a regression model $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ by finding a minimizer of the following objective function

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

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

Kolmogorov-Arnold Networks. KANs are inspired by the Kolmogorov-Arnold representation theorem [83, 84], which states that any continuous multivariate function on a bounded domain can be represented as a finite composition of continuous univariate functions of the input variables and the binary operation of addition. A KAN layer is a fundamental building block of KANs [85], and is given by a matrix $\boldsymbol{\Phi} = (\phi_{q,p})$ of 1D functions, where the trainable activation function ϕ is defined as a weighted combination, with learnable weights, of a sigmoid linear unit (SiLU) function and a spline function. Specifically, each activation function $\phi(x)$ is parameterized by a weighted combination of a basis function $b(x)$ and a spline function with order k and grid size G as follows:

$$\phi(x) = w_b b(x) + w_s \text{spline}(x), \tag{3.2}$$

where $b(x) = \text{silu}(x) = x/(1 + e^{-x})$ and $\text{spline}(x) = \sum_{i=0}^{G+k-1} c_i B_i(x)$ is a weighted sum of B-splines basis functions with trainable coefficients c_i. During training, the parameters w_b and w_s are adjusted to optimize performance, enabling KANs to adapt effectively to different types of data.

Given a input feature vector $\mathbf{x}^{(\ell)} \in \mathbb{R}^{F_\ell}$, the output of the ℓ-th KAN layer is an $F_{\ell+1}$-dimensional feature vector given by

$$\begin{aligned} \mathbf{x}^{(\ell+1)} &= \text{KAN}^{(\ell)}(\mathbf{x}^{(\ell)}) \qquad (3.3) \\ &= \underbrace{\begin{pmatrix} \phi_{1,1}^{(\ell)}(\cdot) & \dots & \phi_{1,F_\ell}^{(\ell)}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{F_{\ell+1},1}^{(\ell)}(\cdot) & \dots & \phi_{F_{\ell+1},F_\ell}^{(\ell)}(\cdot) \end{pmatrix}}_{\boldsymbol{\Phi}^{(\ell)}} \mathbf{x}^{(\ell)} \qquad (3.4) \end{aligned}$$

where $\mathbf{\Phi}^{(\ell)} = (\phi_{q,p}^{(\ell)})$ is an $F_{\ell+1} \times F_\ell$ matrix of functions associated with KAN$^{(\ell)}$. Similar to MLPs, multiple KAN layers can be stacked to form a deep KAN, enabling the model to learn hierarchical representations and capture complex patterns more effectively. A deep KAN consists of L layers, where each layer applies a transformation to the input using a matrix of learnable functions, as illustrated in Figure 3.2.

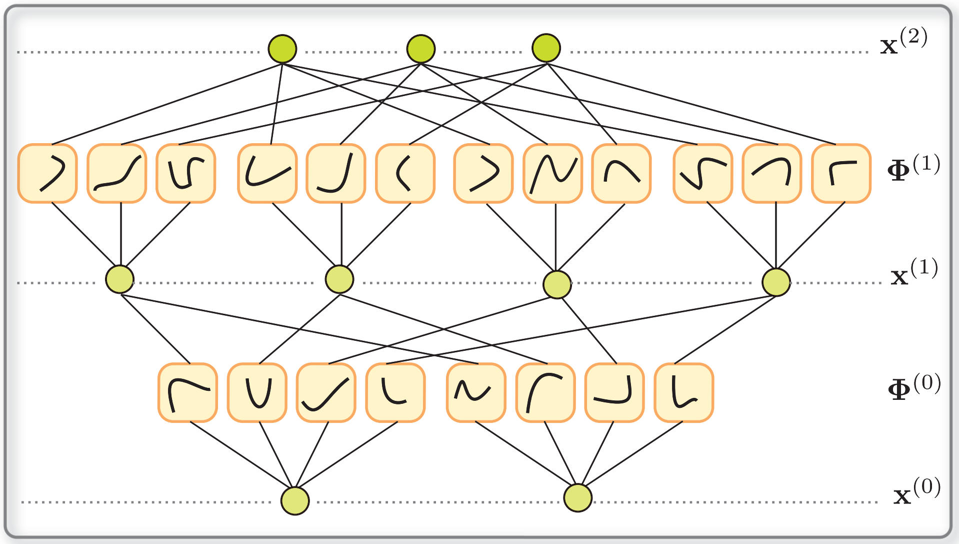


Figure 3.2: Illustration of a two-layer KAN architecture.

3.3.1 Proposed Graph KAN Model

Motivation. Central to GCN-based methods lies the fundamental concept of the feature propagation rule, which determines how information is transmitted among nodes in a graph. This rule entails updating node features by aggregating information from neighboring nodes, performing linear feature transformation via a learnable weight matrix followed by a pre-defined activation function, to generate an updated node representation. For instance, the update rule of an L-layer GCN [87] is given by

$$\mathbf{H}^{(\ell+1)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}), \quad \ell = 0, \dots, L-1 \tag{3.5}$$

where $\mathbf{W}^{(\ell)}$ is a trainable weight matrix, $\sigma(\cdot)$ is an element-wise fixed activation function, and $\mathbf{H}^{(\ell)}$ is the input feature matrix at the ℓ-th layer, with $\mathbf{H}^{(0)} = \mathbf{X}$. Each GCN layer applies a linear transformation followed by a fixed nonlinearity (e.g., ReLU). The transformation is performed using a learnable weight matrix $\mathbf{W}^{(\ell)}$, which is learned during training.

While GCN-based methods have been widely used for 3D Human Pose Estimation due to their ability to model the human skeleton as a structured graph, where joints serve as nodes and bones act as edges, they, however, face several limitations that restrict their effectiveness. First, GCNs rely on fixed activation functions and trainable weights, restricting their ability to dynamically adapt to variations in human poses. This lack of adaptability makes it challenging to generalize across diverse activities and subjects. Second, GCNs primarily aggregate information from one-hop neighbors, limiting their receptive field and making it challenging to capture long-range dependencies that are crucial for handling occlusions and depth ambiguities in 3D pose estimation. Third, GCNs exhibit spectral bias due to their use of MLPs for feature transformation, meaning they tend to prioritize low-frequency information while struggling to capture high-frequency motion details, which are essential for complex pose understanding.

Flexible Graph KAN. To overcome the aforementioned limitations, we introduce a flexible graph KAN (FG-KAN) model, which employs learnable function-based transformations in lieu of fixed activation functions and trainable weight matrices, allowing the model to dynamically adapt its feature learning process. Specifically, the layer-wise update rule of FG-KAN for node feature propagation is given by:

$$\mathbf{H}^{(\ell+1)} = \text{KAN}^{(\ell)}\Big(((1-s)\hat{\mathbf{A}} + s\hat{\mathbf{A}}^2)\mathbf{H}^{(\ell)} + \mathbf{X}\Big),\ \ell = 0, \dots, L-1, \tag{3.6}$$

where $\text{KAN}^{(\ell)}$ is a learnable single KAN layer, $s \in (0,1)$ is a positive scaling parameter that adjusts the contribution of immediate and second-order neighbors, $\mathbf{H}^{(\ell)} \in \mathbb{R}^{J \times F_\ell}$ is the input feature matrix at the ℓ-th layer with embedding dimension F_ℓ. The input of the first layer is $\mathbf{H}^{(0)} = \mathbf{X}$. Note that unlike GCN, which use fixed activation functions, FG-KAN learns its own activation functions dynamically, making it more expressive.

The FG-KAN update rule can be decomposed into two main operations: feature propagation and feature embedding. Feature propagation is given by

$$\mathbf{G}^{(\ell)} = \mathbf{P}\mathbf{H}^{(\ell)} + \mathbf{X}, \tag{3.7}$$

where $\mathbf{P} = (1-s)\hat{\mathbf{A}} + s\hat{\mathbf{A}}^2$ is the propagation matrix defined as a weighted combination of the normalized adjacency matrix and its square, ensuring that multi-hop dependencies are captured. The addition of a residual connection ensures that information from the initial feature matrix is preserved throughout the layers. The initial residual connection allows information from the initial feature matrix to bypass the current layer and be directly added to the output of the current layer. This helps preserve important information that may be lost during the aggregation process, thereby improving the flow of information through the network. The parameter s plays a crucial role in controlling the balance between local and global information. Specifically, it determines the influence of immediate neighbors ($s = 0$) versus nodes that are at most two edges away ($s = 1$).

This tunable mechanism allows the model to adaptively learn graph representations that capture broader contextual dependencies, which is particularly beneficial for handling complex spatial structures in human pose data. After propagation, the feature embedding step refines the node representations by applying a KAN layer as follows:

$$\mathbf{H}^{(\ell+1)} = \mathrm{KAN}^{(\ell)}(\mathbf{G}^{(\ell)}), \tag{3.8}$$

where each edge in the graph is associated with a learnable univariate function via $\mathrm{KAN}^{(\ell)}$, providing greater flexibility in feature adaptation.

The key distinction between the update rule of FG-KAN and its GCN counterpart lies in how feature transformations are performed and how flexibility is introduced into the learning process. While both models use learnable transformations, GCN employs learnable weight matrices for feature transformations, but these transformations are still linear mappings applied at each node. FG-KAN, on the other hand, employs learnable activation functions on edges rather than nodes. This not only allows for greater flexibility and adaptability, but also leads to improved expressiveness and mitigates spectral bias by adapting to both low- and high-frequency components in the data. Moreover, FG-KAN naturally incorporates long-range dependencies using multi-hop aggregation, whereas standard GCNs need deeper architectures to achieve the same effect.

Model Complexity. For simplicity, we assume the embedding dimensions are the same across all layers, i.e., $F_\ell = F$ for all ℓ. The computational cost of multiplying the propagation matrix $\mathbf{P}$ with the embedding $\mathbf{H}^{(\ell)}$ is $\mathcal{O}(\|\hat{\mathbf{A}}\|_0 F)$ in time, where $\|\hat{\mathbf{A}}\|_0$ represents the number of nonzero entries in the sparse normalized adjacency matrix $\hat{\mathbf{A}}$, effectively corresponding to the number of edges in the graph. This term quantifies the complexity of message passing in the graph structure. Applying a KAN layer incurs a computational cost of $\mathcal{O}(GF^2)$, where G is the grid size. Consequently, the overall time complexity of an L-layer FG-KAN is $\mathcal{O}(L\|\hat{\mathbf{A}}\|_0 F + LGF^2)$, where the first term corresponds to feature propagation, while the second term arises from KAN-based feature transformations. Regarding memory complexity, an L-layer FG-KAN requires $\mathcal{O}(LJF + 2^k GLF^2)$ in memory, where $\mathcal{O}(LJF)$ accounts for storing feature embeddings across all layers and $\mathcal{O}(2^k GLF^2)$ represents memory usage in L KAN layers, with 2^k stemming from the recursive computation of order k splines used in function approximation. By comparison, a standard GCN has a lower memory complexity of $\mathcal{O}(LJF + LF^2)$, as it lacks the function-based transformations introduced by KAN layers. However, since k and G are typically small in practical implementations, the additional cost remains manageable. A key computational optimization in FG-KAN is that it avoids explicitly computing the square of the normalized adjacency matrix. Instead, we apply a right-to-left multiplication strategy, where $\hat{\mathbf{A}}$ is first multiplied with the feature embedding before the second multiplication is performed. This optimization eliminates unnecessary matrix exponentiation, significantly reducing computational overhead while preserving the benefits of multi-hop feature propagation. As a result, FG-KAN achieves a strong balance between computational efficiency and enhanced model expressiveness.

Model Architecture. The overall framework of our FG-KAN model architecture is depicted in Figure 3.3. The input to the model consists of 2D keypoints, obtained via an off-the-shelf 2D pose detector [59]. Unlike GCN-based methods that use fixed activation functions, FG-KAN employs learnable function-based transformations, enabling adaptive, expressive, and interpretable feature learning. The FG-KAN architecture consists of start and end FG-KAN Layers, and four residual FG-KAN blocks. The FG-KAN Layer is the core computational unit of the model. The start FG-KAN layer maps input pose representations into latent space for effective feature learning, while the end FG-KAN layer projects refined feature representations back to pose space for final 3D pose predictions. Each residual FG-KAN block is comprised of five FG-KAN layers that learn hierarchical pose features, a layer normalization for stabilized training, an additional FG-KAN layer followed by the Gaussian Error Linear Unit (GELU) nonlinearity, and a residual connection to retain the original information and prevent gradient vanishing. GELU effectively preserves input magnitudes by smoothly blending linear and nonlinear transformations, thereby enhancing adaptability and expressiveness. The end FG-KAN layer generates the 3D pose, and is preceded by global response normalization (GRN) [55] to ensure feature contrast before prediction.

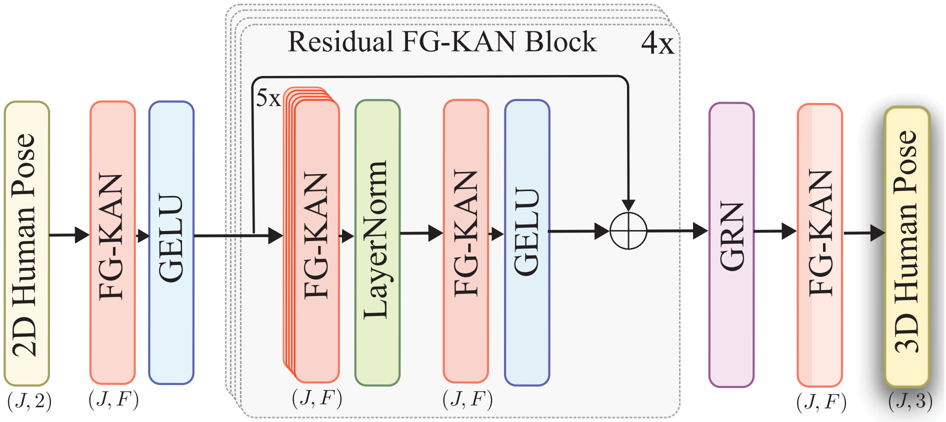


Figure 3.3: Overview of Model Architecture. The model takes 2D pose coordinates as input and produces 3D pose predictions as output, where J is the number of joints and F is the embedding dimension. The architecture consists of a start FG-KAN layer, four residual FG-GCN blocks, and an end FG-KAN layer. Within each residual block, the first five FG-KAN layers are followed by layer normalization, while the last one is followed by a GELU nonlinearity. The end FG-KAN layer is preceded by a global response normalization (GRN) to improve feature contrast and selectivity, ensuring that the most relevant features are emphasized.

Adjacency Modulation. The normalized adjacency matrix is adjusted to account for both local node interactions and long-range dependencies, extending beyond the natural connections in the skeleton graph structure [6]. This modification produces a modulated adjacency matrix, defined as $\check{\mathbf{A}} = \hat{\mathbf{A}} + \mathbf{Q}$, where $\mathbf{Q} \in \mathbb{R}^{J \times J}$ represents a learnable modulation matrix. To respect the inherent symmetry of the skeleton graph, the modulation matrix $\mathbf{Q}$ is symmetrized by averaging it with its transpose, ensuring consistency in the adjacency structure.

Model Prediction. The end FG-KAN layer serves as the output transformation stage, where the learned feature representations are projected back into the 3D pose space to generate the final predicted joint coordinates. The output of this layer consists of the final node embedding, $\hat{\mathbf{y}}_i$, which is the predicted 3D pose coordinates of i-th joint. Global response normalization is applied before the output projection to ensure that the output feature magnitudes are well-calibrated, reducing noise, and improving generalization.

Model Training. The parameters (i.e., univariate activation functions) of the FG-KAN model are learned by minimizing the following loss function:

$$\mathcal{L} = \frac{1}{N}\Big[(1-\alpha)\sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 + \alpha \sum_{i=1}^{N} \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_1\Big], \tag{3.9}$$

which is a weighted combination of the mean squared error (MSE) and mean absolute error (MAE) between the true 3D joint coordinates $\mathbf{y}_i$ and the predicted 3D joint coordinates $\hat{\mathbf{y}}_i$ across N training joints. The MSE term emphasizes large errors by penalizing squared differences, making it particularly effective in reducing substantial deviations between predictions and ground truth. However, MSE can be sensitive to outliers, which may disproportionately influence the optimization process. To mitigate this issue, we incorporate an MAE term, which measures the absolute differences between predictions and ground truth. MAE is more robust to outliers, as it penalizes errors linearly rather than quadratically. The parameter $\alpha \in [0, 1]$ adjusts the relative contributions of each error term, allowing the model to balance stability (by reducing outlier sensitivity through MAE) and precision (by emphasizing squared differences through MSE).

3.4 Experiments

In this section, we present a detailed evaluation of the proposed method, benchmarking it against competing baselines.

3.4.1 Experimental Setup

Datasets. We conduct experimental evaluations on two standard datasets: Human3.6M [1] and MPI-INF-3DHP [88]. We follow standard protocols [3, 49] for data preprocessing and splitting on these public benchmarking datasets.

Evaluation Protocols and Metrics. We employ two standard evaluation protocols for training and testing on Human 3.6M designated as Protocol #1 and Protocol #2 [49], with associated metrics mean per-joint position error (MPJPE) and Procrustes-aligned mean per-joint position error (PA-MPJPE), respectively. For MPI-INF-3DHP, we use the Area Under Curve (AUC) and Percentage of Correct Keypoint (PCK) as assessment metrics.

Baseline Methods. We compare the performance of our model with several state-of-the-art methods for 3D pose estimation, including semantic GCN (SemGCN) [3], MultiPoseNet [73], Weight Unsharing [21], Weakly Supervised Generative Network (WSGN) [74], High-order GCN [4], PoseGraphNet [75], Pose Grammar and and Data Augmentation (PGDA) [76], Pose Grammar and Data Augmentation (PGDA) [76], Compositional GCN (CompGCN) [52], Higher-Order Implicit Fairing Network (HOIF-Net) [5], Multi-hop Modulated GCN (MM-GCN) [54], Group GCN [53], Modulated GCN [6], and GraphMLP [7].

Implementation Details. Our model is implemented in PyTorch and all experiments are conducted on a single NVIDIA GeForce RTX A4500 GPU with 20GB of memory. We employ the AMSGrad optimizer for training, running for 30 epochs on both 2D ground truth and 2D pose detections [59]. The initial learning rate is set to 0.001, with a decay factor of 0.99 every four epochs. The batch size and embedding dimension are set to 64 and $F = 240$, respectively. The scaling parameter $s = 0.2$ and the weighting factor $\alpha = 0.03$ are determined via grid search. To prevent overfitting, we apply dropout with a factor of 0.2 after each FG-KAN layer. We also utilize the GELU nonlinearity as the basis function $b(x)$, which allows adaptive activation scaling, with the spline order set to 3 and the grid size set to 5.

3.4.2 Comparison with State of the Art

Quantitative Results on Human3.6M. In Tables 3.1 and 3.2, we report performance comparison of our FG-KAN model and strong baselines for 3D pose estimation on Human3.6M using the detected 2D pose as input. In both tables, we present the results of all 15 actions, as well as the average performance.

Table 3.1 demonstrates that our proposed FG-KAN model surpasses all baseline methods, achieving an MPJPE of 46.7mm, thereby establishing itself as an effective approach for 3D Human Pose Estimation. Under Protocol #1, FG-KAN consistently outperforms state-of-the-art models across multiple action categories. Compared to GraphMLP, the strongest competing baseline, FG-KAN achieves a relative error reduction of 2.7%, demonstrating its ability to enhance pose prediction accuracy even against the best-performing non-GCN-based approach. Against Modulated GCN, our model achieves a 5.47% reduction in error and performs better in 14 out of 15 action categories, emphasizing the superiority of learnable function-based transformations over pre-defined activation functions in traditional GCNs. Additionally, FG-KAN significantly outperforms High-Order GCN, achieving a relative error reduction of 15.99%, showcasing its ability to capture multi-hop dependencies more effectively without relying on fixed high-order convolutions. Furthermore, FG-KAN provides a substantial 18.92% error reduction over SemGCN, one of the earliest GCN-based models for 3D pose estimation, highlighting its ability to model long-range spatial dependencies and improve feature learning through dynamic adjacency modulation. Our model not only excels in standard poses but also demonstrates superior generalization on challenging actions involving significant self-occlusions, such as Eating, Sitting, and Smoking. Self-occlusion remains a major challenge in monocular 3D pose estimation, as body parts frequently obscure one another, making it difficult for traditional models to infer joint positions accurately. For instance, in activities like eating or smoking, the hands and arms often block facial features, while in sitting postures, the arms and legs can obscure the torso or lower body joints. FG-KAN effectively mitigates these challenges through multi-hop information aggregation, enabling the model to recover occluded joint positions more accurately than prior GCN-based methods.

Under Protocol #2, Table 3.2 demonstrates that FG-KAN achieves the lowest PA-MPJPE of 38.3mm, outperforming all baseline models. When compared to GraphMLP, FG-KAN achieves a relative error reduction of 0.26%, indicating a modest yet consistent improvement over the best competing baseline. Moreover, FG-KAN outperforms GraphMLP in 10 out of 15 action categories, reinforcing its ability to generalize across a diverse range of human activities. Similarly, FG-KAN reduces the error by 2.04% compared to Modulated GCN and achieves better results in 13 out of 15 actions, showcasing the effectiveness of its learnable function-based transformations over the predefined activation functions in GCNs. Notably, FG-KAN demonstrates superior performance in occlusion-prone actions, outperforming Modulated GCN by 1.23% in Greeting, 2.15% in Sitting, and 5.43% in Smoking, highlighting its robustness in handling self-occlusions and complex body poses. Furthermore, FG-KAN surpasses Modulated GCN on the challenging Photo action, achieving a relative error reduction of 3.59%, indicating better feature learning in scenarios with complex backgrounds and fine-grained pose details. Most significantly, FG-KAN outperforms High-Order GCN with a substantial relative error reduction of 12.35%, achieving lower errors across all ac-

Table 3.1: Comparison of our model and baseline methods in terms of Mean Per Joint Position Error (MPJPE) in millimeters, computed between the ground truth and estimated poses on the Human3.6M dataset under Protocol #1. The last column displays the average errors, with boldface numbers denoting the best 3D pose estimation performance and underlined numbers indicating 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.* [49]	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.* [48]	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.* [67]	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.* [71]	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 [37]	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
Ordinal Depth [66]	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
MultiPoseNet [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
SemGCN [3]	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
WSGN [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
PoseGraphNet [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
PGDA [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
High-order GCN [4]	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
HOIF-Net [5]	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
CompGCN [52]	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
Weight Unsharing [21]	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
Modulated GCN [6]	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
MM-GCN [54]	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
Group GCN [53]	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
GraphMLP [7]	43.7	49.3	45.5	47.9	50.5	56.0	46.3	44.1	55.9	59.0	48.4	45.7	51.2	**37.1**	39.1	48.0
FG-KAN (ours)	**40.9**	**45.6**	**44.4**	**47.4**	**48.4**	52.8	**44.1**	**41.6**	**54.6**	63.8	**46.1**	**45.1**	49.0	38.1	**39.0**	**46.7**

tion categories. This significant improvement highlights FG-KAN's ability to capture long-range spatial dependencies more effectively.

Cross-Dataset Results on MPI-INF-3DHP. In Table 3.3, we evaluate the generalization ability of our method by comparing it against strong baselines. Our model is trained on Human3.6M and evaluated on the MPI-INF-3DHP dataset. Results demonstrate that our approach achieves the highest PCK and second best AUC scores, consistently outperforming the baseline methods across various indoor and outdoor scenes. Compared to ICFNet, the best performing baseline, our model shows a relative improvement of 0.5% in terms of the PCK metric. While ICFNet achieves a slightly higher AUC score, FG-KAN remains highly competitive, reinforcing the effectiveness of its learnable function-based transformations and multi-hop feature propagation in capturing complex pose relationships. Despite being trained only on indoor scenes from the Human3.6M dataset, FG-KAN demonstrates strong generalization to outdoor settings, where variations in illumination, camera angles, and occlusions introduce additional challenges. The model's ability to maintain

Table 3.2: Comparison of our model and baseline methods in terms of Procrustes-aligned Mean Per Joint Position Error (PA-MPJPE), computed between the ground truth and estimated poses on the Human3.6M dataset under Protocol #2.

	Action															
Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Pavlakos *et al.* [47]	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.* [39]	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.* [49]	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.* [48]	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.* [71]	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 [72]	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
p-LSTMs [77]	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
WSGN [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
PoseGraphNet [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
PGDA [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
High-order GCN [4]	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 [5]	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
CompGCN [52]	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
Weight Unsharing [21]	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
Modulated GCN [6]	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
MM-GCN [54]	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
Group GCN [53]	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
GraphMLP [7]	35.1	38.2	36.5	**39.8**	39.8	43.5	35.7	34.0	45.6	**47.6**	39.8	**35.1**	41.1	**30.0**	33.4	38.4
FG-KAN (ours)	**34.0**	**37.2**	**36.0**	40.0	**39.0**	**42.9**	**35.1**	**33.3**	**45.4**	54.5	**38.3**	36.0	40.5	30.1	**32.4**	**38.3**

high accuracy in these diverse conditions highlights the effectiveness of its KAN-based architecture and flexible feature aggregation strategies.

Qualitative Results. Figure 3.4 presents qualitative visualizations of FG-KAN's predictions on the Human3.6M dataset across various action categories, illustrating the model's effectiveness in accurately estimating 3D human poses from monocular 2D inputs. The predicted 3D poses exhibit a high degree of alignment with the ground truth, demonstrating the model's ability to capture spatial dependencies and maintain structural consistency even in complex motion scenarios. The precise joint placements and natural articulation of limbs in our predictions further validate FG-KAN's ability to mitigate depth ambiguities and occlusions, which are inherent challenges in 2D-to-3D pose estimation. A comparative analysis with GraphMLP further highlights FG-KAN's advantages in handling challenging cases involving self-occlusion, where certain body parts obscure others, making pose reconstruction significantly harder. Unlike GraphMLP, which occasionally produces unnatural joint positions or misaligned skeletal structures, FG-KAN generates pose estimations that closely resemble the ground truth pose, capturing fine-grained joint interactions more effectively. This improvement can be attributed to FG-KAN's learnable function-based transformations and multi-hop feature aggregation, which enable it to leverage global pose context while refining

Table 3.3: Performance comparison of our model and baseline methods on the MPI-INF-3DHP dataset using PCK and AUC as evaluation metrics.

Method	PCK (↑)	AUC (↑)
Martinez *et al.* [49]	42.5	17.0
Mehta *et al.* [88]	64.7	31.7
Li *et al.* [78]	67.9	-
Yang *et al.* [67]	69.0	32.0
Zhou *et al.* [39]	69.2	32.5
Habibie *et al.* [65]	70.4	36.0
Ordinal Depth [66]	71.9	35.3
Pose Attribute [79]	71.9	35.8
HOIF-Net [5]	72.8	36.5
LCN [22]	74.0	36.7
HEMlets [80]	75.3	38.0
SRNet [81]	77.6	43.8
Weight Unsharing [21]	79.3	47.6
CompGCN [52]	79.3	45.9
GraphSH [64]	80.1	45.8
HCSF [68]	82.1	46.2
MM-GCN [54]	81.6	50.3
Group GCN [53]	81.1	49.9
ICFNet [89]	<u>85.6</u>	**54.3**
FG-KAN (ours)	**86.0**	<u>52.9</u>

local predictions.

Quantitative Results using Ground Truth. In Table 3.4, we report the performance comparison results under Protocol #1 and Protocol #2 for FG-KAN and several state-of-the-art baseline models, including SemGCN, High-Order GCN, HOIF-Net, Modulated GCN, Weight Unsharing, and GraphMLP, using ground truth 2D poses as input. These results highlight FG-KAN's superior performance across both evaluation protocols. Under Protocol #1, FG-KAN achieves an MPJPE of 33.51mm, significantly outperforming all competing baselines. Specifically, compared to SemGCN, High-Order GCN, HOIF-Net, Modulated GCN, Weight Unsharing, and GraphMLP, our model reduces errors by 8.27mm, 5.65mm, 4.25mm, 4.38mm, 3.96mm, and 0.69mm, respectively. These improvements translate into relative error reductions of 19.62%, 14.29%, 11.14%, 11.45%, 10.46%, and 2.01%, demonstrating FG-KAN's ability to learn richer representations from skeletal graphs. Similarly, under Protocol #2, FG-KAN achieves the lowest PA-MPJPE of 28.01mm, outperforming SemGCN, High-Order GCN, HOIF-Net, Modulated GCN, and Weight Unsharing. The relative error reductions amount to 15.41%, 8.72%, 4.64%, 5.65%, and 5.74%, respectively

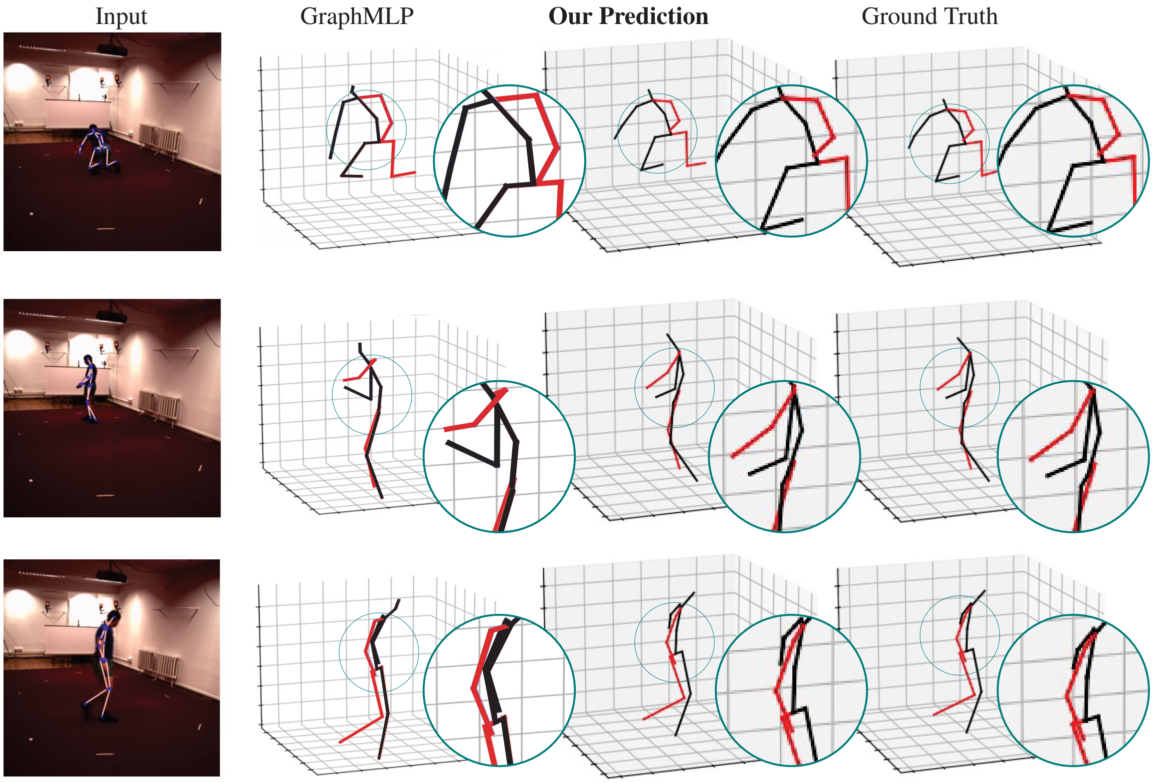


Figure 3.4: Visual comparison between FG-KAN and Modulated GCN on sample actions from the Human3.6M dataset.

Table 3.4: Performance comparison of our model and other state-of-the-art methods using the 2D ground truth pose as input.

Method	MPJPE (↓)	PA-MPJPE (↓)
SemGCN [3]	42.14	33.53
High-order GCN [4]	39.52	31.07
HOIF-Net [5]	38.12	29.74
Modulated GCN [6]	38.25	30.06
Weight Unsharing [21]	37.83	30.09
GraphMLP [7]	34.20	-
FG-KAN (ours)	**33.51**	**28.01**

3.4.3 Ablation Study

We perform an ablation study on the Human3.6M dataset to assess the impact of various design choices within our network architecture. In these experiments, we systematically alter key components of the proposed model to evaluate their impacts to overall performance, providing deeper insights into the effectiveness of each architectural element.

Effect of Residual Connection. We analyze the impact of the initial residual connection (IRC) within the layer-wise propagation rule on our model's performance. The results, summarized in Table 3.5, indicate that incorporating IRC leads to notable performance improvements, reducing the Mean Per Joint Position Error (MPJPE) and Procrustes-Aligned MPJPE (PA-MPJPE) by 1.65% and 1.49%, respectively. These findings highlight the crucial role of IRC in enhancing model accuracy. By preserving and reinforcing the initial node features across layers, IRC ensures that fundamental positional information is effectively retained and leveraged throughout the network. This mechanism not only stabilizes the learning process but also mitigates information loss, allowing for more refined and reliable pose predictions. Furthermore, the consistent flow of initial features helps the model capture both local and global dependencies more effectively, leading to improved generalization and robustness in complex motion scenarios.

Table 3.5: Effect of initial residual connection (IRC) on model performance.

Method	MPJPE (↓)	PA-MPJPE (↓)
Without IRC	34.44	28.79
With IRC	**33.51**	**28.01**

Effect of Symmetrizing Adjacency Modulation. Table 3.6 examines the impact of symmetrizing the adjacency modulation matrix on 3D Human Pose Estimation accuracy. The results demonstrate that enforcing symmetry regularization leads to a significant reduction in both MPJPE and PA-MPJPE errors, improving the model's overall performance. Specifically, introducing symmetry in the adjacency modulation matrix reduces the MPJPE from 36.82mm to 33.51mm, resulting in a relative error reduction of 2.95mm. Similarly, the PA-MPJPE decreases from 28.82mm to 28.01mm, yielding a further 0.46mm improvement. This highlights the benefits of incorporating structural consistency into the adjacency modulation process. This symmetrization step preserves the bidirectional nature of joint interactions, ensuring that the information flow between body joints remains balanced and coherent.

Table 3.6: Effect of symmetrizing adjacency modulation.

Method	MPJPE (↓)	PA-MPJPE (↓)
Without Symmetry	36.82	28.82
With Symmetry	**33.51**	**28.01**

3.4.4 Hyperparameter Sensitivity Analysis

We conduct a sensitivity analysis on three key hyperparameters, namely batch size, embedding dimension, and the scaling factor in the propagation matrix, to assess their impact on the overall performance of our model. By varying these hyperparameters, we aim to understand how they influence the computational efficiency, capacity, and expressiveness of FG-KAN.

Effect of Embedding Dimension. In Figure 3.5, we analyze the impact of a varying embedding dimension on our model's performance. This hyperparameter, which determines the number of learnable parameters in each layer of the network, affects the model's capacity to capture complex patterns. Larger embedding dimensions enable the model to learn richer feature representations, improving its capacity to capture fine-grained motion details. However, increasing the embedding size excessively may lead to overfitting, where the model memorizes training data rather than generalizing to unseen samples. Smaller embedding dimensions reduce the number of parameters, making the model more lightweight and efficient, but may lead to underfitting and reduced prediction accuracy. Our analysis shows that the best performance is achieved using an embedding dimension of 240, providing a sufficient number of parameters to learn detailed features without overfitting.

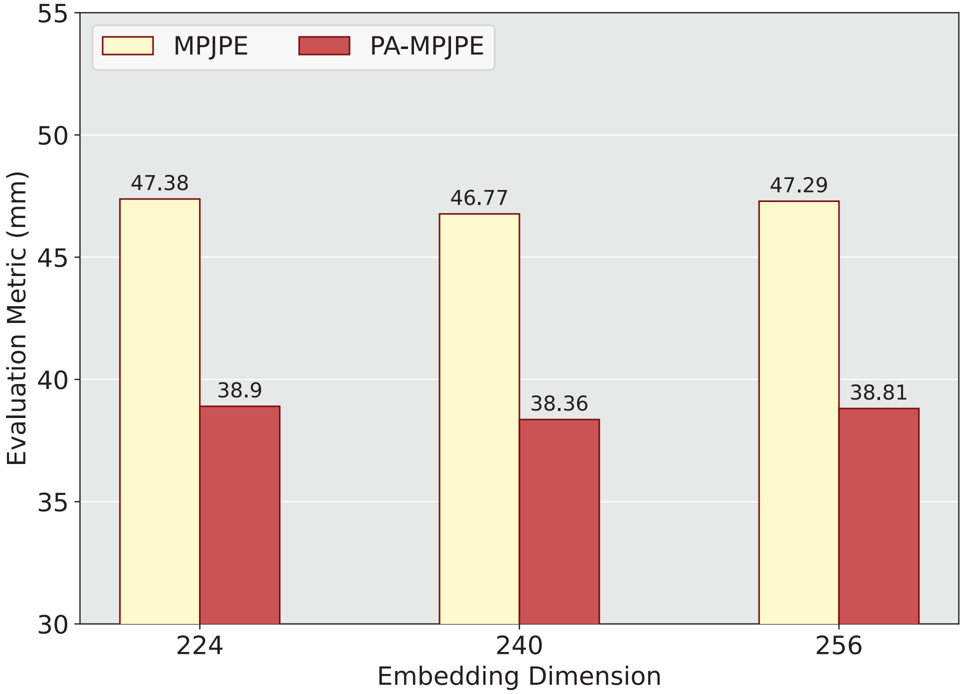


Figure 3.5: Performance of our proposed FG-KAN model on the Human3.6M dataset using various embedding dimensions.

Effect of Scaling Factor. We investigate the impact of the scaling factor s on the model's performance by plotting the error metrics against a range of values for s, from 0 and 1. This hyperparameter helps control the balance between the information from immediate neighbors and the information from nodes that are at most two edges away in the graph. By adjusting this parameter, we control how much weight is given to local versus global dependencies within the graph structure. This ability to adjust the influence of distant nodes is particularly valuable when learning graph representations that need to capture global patterns and long-range dependencies, which are essential for tasks like 3D Human Pose Estimation. In practice, a smaller scaling parameter s emphasizes more localized interactions, effectively allowing the model to focus on the immediate context, while a larger s allows the model to capture broader dependencies across the graph, potentially improving the model's ability to generalize in more complex scenarios. Figure 3.6 illustrates that smaller scaling parameters often lead to better results, with our model achieving the lowest error values for MPJPE and PA-MPJPE at $s = 0.2$. This suggests that a moderate emphasis on local information, while still considering distant nodes, strikes the good balance for our model.

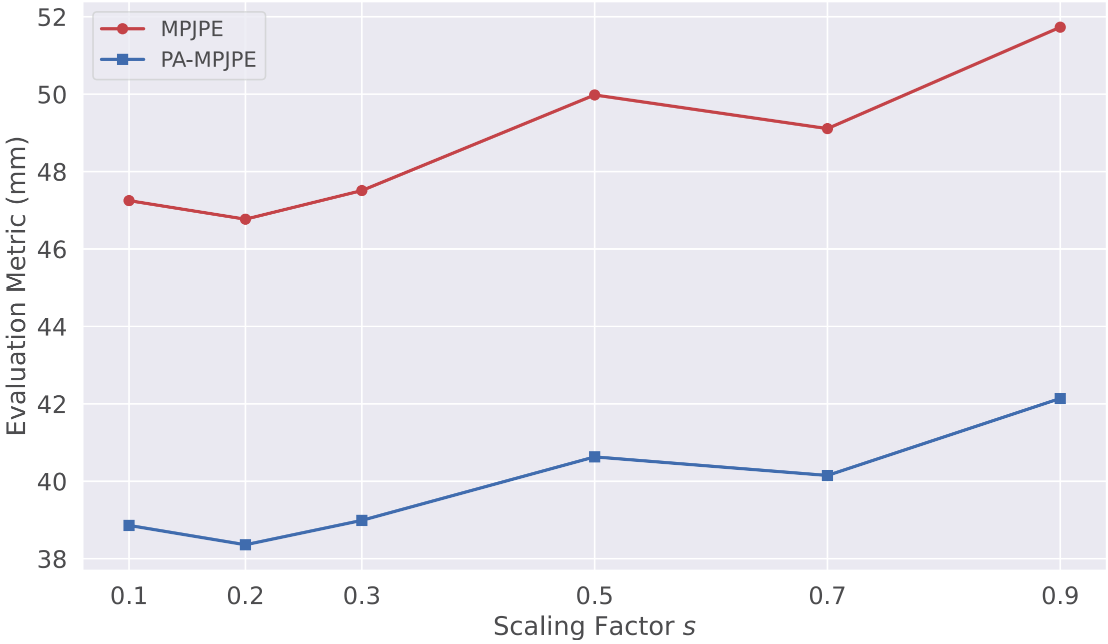


Figure 3.6: Analysis of the model's sensitivity to the selection of the scaling factor s. Smaller values of s typically lead to reduced MPJPE and PA-MPJPE errors.

Effect of Spline Order and Grid Size. To assess the impact of the spline order and grid size on the model's performance, we conduct a sensitivity analysis by evaluating the MPJPE and PA-MPJPE error metrics across different values of these hyperparameters. Figure 3.7 (left) presents the results for varying spline orders, where we observe that the best performance is achieved at a spline order of 3, yielding an MPJPE of 46.77mm and a PA-MPJPE of 38.36mm. Increasing the spline order beyond this point does not lead to further improvements and may introduce unnecessary complexity. Similarly, Figure 3.7 (right) shows the effect of the spline grid size, where the best results are obtained at a grid size of 5, leading to an MPJPE of 46.77mm and a PA-MPJPE of 38.36mm. These results highlight the importance of tuning these hyperparameters to balance model expressiveness and computational efficiency. While a sufficiently high spline order and grid size allow for more flexible function approximations, excessively large values may lead to diminishing returns or increased computational costs.

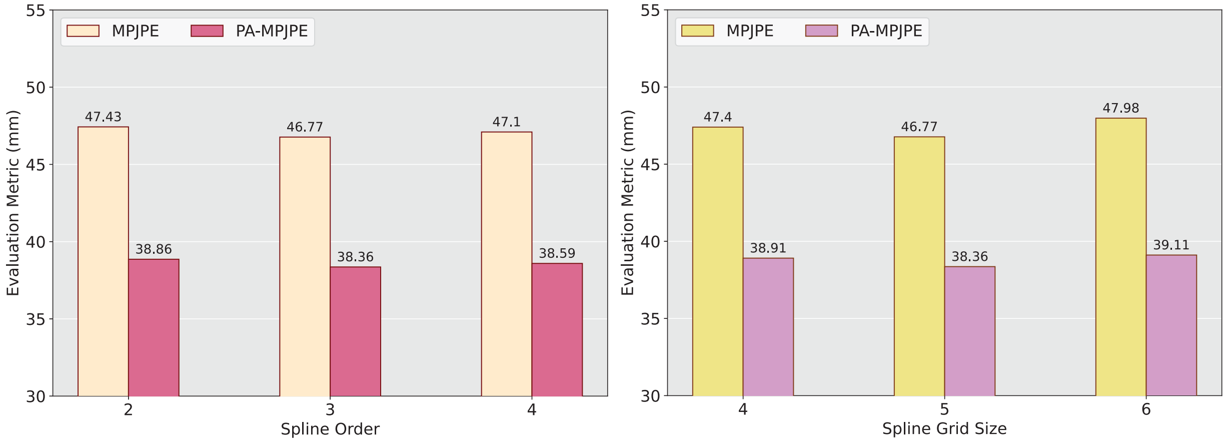


Figure 3.7: Impact of spline order and grid Size on FG-KAN Performance. The left plot shows the effect of varying the spline order, while the right plot illustrates the impact of different grid sizes. The best performance is achieved with a spline order of 3 and a grid size of 5, balancing accuracy and computational efficiency..

3.5 Discussion

The proposed FG-KAN framework presents several advantages over conventional GCN-based approaches for 3D Human Pose Estimation. In this section, we highlight its merits in three key aspects: (1) Flexible and expressive feature learning, (2) Improved long-range dependency modeling, and (3) Reduced spectral bias for enhanced pose estimation.

- *Flexible and Expressive Feature Learning*. Unlike standard GCN-based models that rely on fixed-weight transformations and predefined activation functions, FG-KAN leverages KANs, which introduce learnable activation functions on graph edges. These learnable functions provide greater expressiveness and data-driven adaptability.

- *Improved Long-Range Dependency Modeling*. Standard GCNs are inherently constrained by their reliance on one-hop message passing, which restricts the receptive field of each joint to its immediate neighbors. To overcome this limitation, FG-KAN incorporates a multi-hop propagation mechanism through the flexible propagation matrix with a scaling factor that balances information from immediate neighbors and higher-order dependencies, enabling joints that are not directly connected to exchange information. By modulating the adjacency matrix through a learnable modulation matrix, our model also learns new skeletal connections, further enhancing structure-aware learning.

- *Reduced Spectral Bias.* Since MLPs are core components in GCN-based methods, they exhibit spectral bias, where the model prefers learning low-frequency signals while struggling to capture high-frequency variations. By contrast, FG-KAN mitigates this issue by dynamically learning function-based transformations to capture fine-grained motion variations, allowing our model to learn both low- and high-frequency components more effectively:

While FG-KAN offers greater interpretability compared to GCN-based methods, visualizing how each learnable activation function adapts across different parts of the human skeleton remains a challenge. Moreover, the incorporation of these learnable activations increases the computational cost per layer. The use of spline-based activations further contributes to memory overhead, as storing and evaluating function approximations demand additional resources. For large-scale datasets, this can lead to substantial memory consumption, particularly when training deeper networks. Therefore, further improvements are necessary to enhance computational efficiency, interpretability, and robustness, ensuring broader applicability of FG-KAN.

CHAPTER 4

Conclusions and Future Work

This thesis presented two novel graph-based frameworks, namely Flex-GCN and FG-KAN, for effective 3D human pose estimation. Both models were designed to overcome the inherent limitations of standard Graph Convolutional Networks (GCNs), particularly their constrained receptive fields and susceptibility to spectral bias. The proposed Flex-GCN framework introduced a flexible graph convolution mechanism that aggregates multi-hop neighbor information, enabling the model to capture high-order spatial dependencies critical for mitigating occlusions and depth ambiguities. The architecture further employed adjacency modulation and residual connections to enhance structural awareness and training stability. Empirical evaluations demonstrated that Flex-GCN significantly outperforms baseline models across standard benchmark datasets, supported by detailed ablation studies highlighting the contribution of each architectural component. Building upon these insights, FG-KAN incorporated learnable function-based transformations inspired by the Kolmogorov-Arnold representation theorem. Unlike traditional GCNs that rely on fixed activation functions and MLPs, FG-KAN utilized adaptive, univariate functions on graph edges, effectively mitigating spectral bias and enabling the model to capture both low- and high-frequency motion cues. The model also leveraged multi-hop feature aggregation and a learnable adjacency modulation matrix, allowing dynamic connectivity adjustments to better represent complex skeletal relationships. Extensive experiments on Human3.6M and MPI-INF-3DHP confirmed FG-KAN's superior performance and generalization capability, achieving state-of-the-art results in multiple metrics. Together, these contributions advance the field of graph-based pose estimation by addressing critical limitations in expressiveness and spatial modeling. Future work will explore the extension of these models to multi-person scenarios and their application to broader graph-based vision tasks such as action recognition and motion forecasting. Finally, Section 4.1 summarizes the key findings and contributions of the research presented in the preceding chapters. Section 4.2 outlines the main limitations of the proposed approach, while Section 4.3 offers directions for future research building upon this work.

4.1 Contributions of the Thesis

4.1.1 Flexible GCN for 3D Human Pose Estimation

In Chapter 2, we introduced a simple yet efficient Flex-GCN model, which captures high-order dependencies essential for reducing uncertainty due to occlusion or depth ambiguity in 3D Human Pose Estimation. We also theoretically demonstrated the training stability of Flex-GCN. Experimental results demonstrate that our model outperforms competitive baselines on standard datasets for 3D Human Pose Estimation. Furthermore, our exploration of adjacency modulation enables Flex-GCN to incorporate richer contextual information beyond the natural connections of body joints, leading to enhanced performance in challenging scenarios. Through ablation studies, we have elucidated the contributions of various design choices, such as the initial residual connection and symmetry of modulation adjacency, highlighting their positive impact on model performance.

4.1.2 Graph KAN for 3D Human Pose Estimation

In Chapter 3, we introduced a Flexible Graph Kolmogorov-Arnold Network (FG-KAN), a novel approach for 3D Human Pose Estimation that leverages learnable function-based transformations. Unlike GCN-based models that rely on fixed activation functions, FG-KAN employs learnable univariate functions on graph edges, providing greater flexibility and expressiveness in modeling human skeletal structures. By incorporating multi-hop feature propagation and adjacency modulation, our model effectively captures both local and long-range dependencies, mitigating the challenges posed by occlusions and depth ambiguities in 3D pose estimation. A key advantage of FG-KAN lies in its ability to address spectral bias, a common limitation in MLP-based GCN architectures, where models tend to favor low-frequency information while struggling to capture high-frequency motion details. By leveraging learnable, data-driven activation functions, FG-KAN adapts to both low- and high-frequency components. Our extensive experiments on benchmark datasets, including Human3.6M and MPI-INF-3DHP, demonstrate that FG-KAN outperforms state-of-the-art methods in terms of MPJPE and PA-MPJPE, while also achieving the highest PCK score in cross-dataset evaluations, proving its robust generalization ability to unseen data. Furthermore, we presented visual results of FG-KAN, highlighting the effectiveness of our approach qualitatively.

4.2 Limitations

An important aspect examined in this study is the inference time, which is critical in assessing the practical deployability of the proposed models, especially in resource-constrained scenarios. While our models, including the Flexible Graph Kolmogorov-Arnold Network (FG-KAN), consistently achieve good performance in 3D human pose estimation across multiple benchmarks, we observe that FG-KAN incurs a higher inference time compared to lightweight baselines. This indicates a trade-off between model accuracy and computational efficiency. The slower inference speed is primarily due to the complexity of the graph-based architecture and the use of learnable polynomial functions in FG-KAN, which enhance representational power but increase runtime overhead. Although the accuracy gains justify this trade-off in many use cases, optimizing inference speed remains an area for future improvement to broaden the model's applicability to time-sensitive tasks such as sports analytics, or human-robot interaction.

4.3 Future Work

For future work, we plan to extend the the proposed Flex-GCN and FG-KAN model to a broader range of tasks in computer vision and graph representation learning. In particular, we aim to adapt FG-KAN to multi-person 3D pose estimation, where the challenges of occlusion, interaction, and scalability are more pronounced. We also intend to explore its applicability to other graph-based vision problems, including action recognition and motion forecasting, where modeling temporal dynamics and spatial dependencies jointly is critical. Several promising directions, motivated by this thesis, are outlined as follows.

4.3.1 Kolmogorov-Arnold Transformer for 3D Human Pose Estimation

Building on the success of our current work, we aim to further enhance the accuracy and robustness of 3D human pose estimation by integrating Kolmogorov-Arnold Networks (KANs) into the PoseFormerV2 framework [90]. While KANs offer strong expressive power through learnable polynomial activations, further exploration is needed to fully leverage their potential in spatio-temporal modeling. As part of our future research, we plan to investigate the capabilities of the Kolmogorov-Arnold Transformer (KAT) architecture [91] in this domain. In particular, we will explore the integration of KAT's main components, including rational function bases, group-wise activation sharing, and variance-preserving initialization, into both temporal and spatial encoding modules. This direction is motivated by the need to overcome current computational bottlenecks and initialization instability that may occur when scaling KAN-based modules in Transformer-like pipelines. Moreover, we intend to explore adaptive mechanisms to dynamically balance time- and frequency-domain representations, leveraging KAT's functional decomposition. This could improve model resilience against noisy or missing 2D joint inputs, a common challenge in real-world settings. Ultimately, we envision a fully KAT-enhanced PoseFormerV2 variant that could achieve superior accuracy and computational efficiency.

4.3.2 Bidirectional State Space Model for 3D Human Pose Estimation

To further advance the accuracy and scalability of 3D Human Pose Estimation, our future work will explore integrating Vision Mamba [92], a recently proposed bidirectional state space model, into the PoseFormerV2 framework [90]. Vision Mamba offers a compelling alternative to self-attention by modeling long-range temporal dependencies with linear time complexity, significantly reducing computational overhead while maintaining strong expressive power. Its bidirectional formulation allows it to capture temporal context in both past and future directions, which is particularly valuable in handling complex motion patterns and resolving occlusions in pose estimation tasks. Our plan involves incorporating Mamba's structured state transitions and input-dependent gating mechanisms into both the temporal and spatial encoding layers of PoseFormerV2. This integration will require adapting Mamba's token-mixing operations to capture fine-grained spatio-temporal correlations across human joint trajectories. We anticipate that Mamba's inductive bias toward sequential data will improve the model's robustness to missing or noisy 2D keypoints, conditions that often degrade performance in real-world scenarios. In addition, we intend to conduct a comprehensive comparison between traditional self-attention and Mamba's state space dynamics, aiming to design a hybrid architecture that leverages the strengths of both paradigms. This will involve detailed ablation studies and benchmarking on standard 3D pose estimation datasets to evaluate trade-offs in accuracy, speed, and parameter efficiency.

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] 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.

[4] 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.

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

[6] 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.

[7] W. Li, M. Liu, H. Liu, T. Guo, T. Wang, H. Tang, and N. Sebe, “GraphMLP: A graph MLP-like architecture for 3D human pose estimation,” *Pattern Recognition*, 2025.

[8] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, “Deep learning-based human pose estimation: A survey,” *ACM Computing Surveys*, vol. 56, no. 1, 2024.

[9] J. Kim, R. Kim, K. Byun, N. Kang, and K. Park, “Assessment of temporospatial and kinematic gait parameters using human pose estimation in patients with parkinson’s disease: A comparison between near-frontal and lateral views,” *PLOS One*, vol. 20, no. 1, 2025.

[10] Z. Gao, J. Chen, Y. Liu, Y. Jin, and D. Tian, “A systematic survey on human pose estimation: upstream and downstream tasks, approaches, lightweight models, and prospects,” *Artificial Intelligence Review*, vol. 58, no. 3, 2025.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[12] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016. arXiv preprint arXiv:1607.06450.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” Technical Report, University of Toronto, 2012. arXiv preprint arXiv:1207.0580.

[14] S. Biasotti, A. Cerri, M. Abdelrahman, and et al., “SHREC’14 track: Retrieval and classification on textured 3D models,” in *Proc. Eurographics Workshop on 3D Object Retrieval*, pp. 111–120, 2014.

[15] S. Biasotti, A. Cerri, M. Aono, and et al., “Retrieval and classification methods for textured 3D models: a comparative study,” *The Visual Computer*, vol. 32, pp. 217–241, 2016.

[16] M. Masoumi and A. Ben Hamza, “Spectral shape classification: A deep learning approach,” *Journal of Visual Communication and Image Representation*, vol. 43, pp. 198–211, 2017.

[17] E. Rodola, L. Cosmo, O. Litany, and et al., “SHREC’17: Deformable shape retrieval with missing parts,” in *Proc. Eurographics Workshop on 3D Object Retrieval*, 2017.

[18] M. Masoumi and A. Ben Hamza, “Shape classification using spectral graph wavelets,” *Applied Intelligence*, vol. 47, pp. 1256–1269, 2017.

[19] H. Krim and A. Ben Hamza, *Geometric methods in signal and image analysis*. Cambridge University Press, 2015.

[20] E. E. Abdallah, A. Ben Hamza, and P. Bhattacharya, “Spectral graph-theoretic approach to 3D mesh watermarking,” in *Proceedings of Graphics Interface*, pp. 327–334, 2007.

[21] 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.

[22] 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.

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

[24] 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*, pp. 11656–11665, 2021.

[25] B. Shan, Q. Shi, and F. Yang, "MSRT: Multi-scale representation transformer for regression-based human pose estimation," *Pattern Analysis and Applications*, vol. 26, no. 2, pp. 591–603, 2023.

[26] R. Wang, F. Geng, and X. Wang, "MTPose: Human pose estimation with high-resolution multi-scale transformers," *Neural Processing Letters*, vol. 54, no. 5, pp. 3941–3964, 2022.

[27] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang, "Exploiting temporal contexts with strided transformer for 3D human pose estimation," *IEEE Transactions on Multimedia*, vol. 25, pp. 1282–1293, 2022.

[28] N. Brandizzi, A. Fanti, R. Gallotta, S. Russo, L. Iocchi, D. Nardi, and C. Napoli, "Unsupervised pose estimation by means of an innovative vision transformer," in *Proc. International Conference on Artificial Intelligence and Soft Computing*, pp. 3–20, 2022.

[29] H. Shuai, L. Wu, and Q. Liu, "Adaptive multi-view and temporal fusing transformer for 3D human pose estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4122–4135, 2022.

[30] C. Wu, X. Wei, S. Li, and A. Zhan, "MSTPose: Learning-enriched visual information with multi-scale transformers for human pose estimation," *Electronics*, vol. 12, no. 15, 2023.

[31] D. Wang, W. Xie, Y. Cai, X. Li, and X. Liu, "Transformer-based rapid human pose estimation network," *Computers & Graphics*, vol. 116, pp. 317–326, 2023.

[32] Y. Chen, R. Gu, O. Huang, and G. Jia, "VTP: Volumetric transformer for multi-view multi-person 3D pose estimation," *Applied Intelligence*, vol. 53, no. 22, pp. 26568–26579, 2023.

[33] S. Li, H. Zhang, H. Ma, J. Feng, and M. Jiang, “CSIT: Channel spatial integrated transformer for human pose estimation,” *IET Image Processing*, vol. 17, no. 10, pp. 3002–3011, 2023.

[34] D. Wang, W. Xie, Y. Cai, X. Li, and X. Liu, “Multi-order spatial interaction network for human pose estimation,” *Digital Signal Processing*, vol. 142, 2023.

[35] K. Zhang, X. Luan, T. H. S. Syed, and X. Xiang, “ICRFormer: An improving cos-reweighting transformer for 3D human pose estimation in video,” in *Proc. Chinese Control and Decision Conference*, pp. 436–441, 2023.

[36] H. Zhang, Z. Hu, Z. Sun, M. Zhao, S. Bi, and J. Di, “A fused convolutional spatio-temporal progressive approach for 3D human pose estimation,” *The Visual Computer*, pp. 1–13, 2023.

[37] 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.

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

[39] 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.

[40] Q. Zhao, C. Zheng, M. Liu, and C. Chen, “A single 2D pose with context is worth hundreds for 3D human pose estimation,” in *Advances in Neural Information Processing Systems*, 2023.

[41] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[42] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” in *Proc. European Conference on Computer Vision*, pp. 483–499, 2016.

[43] A. T. M. Shahjahan and A. Ben Hamza, “Flexible graph convolutional network for 3D human pose estimation,” in *Proc. British Machine Vision Conference*, 2024.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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*, 2021.

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

[54] 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.

[55] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie, “ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 16133–16142, 2023.

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

[57] R. L. Burden, J. D. Faires, and A. M. Burden, *Numerical Analysis*. Cengage Learning, 2015.

[58] F. Riesz and B. Sz.-Nagy, *Functional Analysis*. Dover Publications, 1990.

[59] 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.

[60] 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.

[61] F. W. Campbell and J. G. Robson, "Application of Fourier analysis to the visibility of gratings," *The Journal of Physiology*, vol. 197, p. 551—66, 1968.

[62] H. K. Hartline, H. G. Wagner, and F. Ratliff, "Inhibition in the eye of limulus," *Journal of General Physiology*, vol. 39, p. 651—73, 1956.

[63] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 84–90, 2017.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

[72] M. R. I. Hossain and J. J. Little, "Exploiting temporal information for 3d human pose estimation," in *Proc. of the European Conference on Computer Vision*, p. 69–86, 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. Gracia, 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] 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.

[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] 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.

[82] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. A. Hamprecht, Y. Bengio, and A. Courville, "On the spectral bias of neural networks," in *Proc. International Conference on Maching Learning*, 2019.

[83] J. Braun and M. Griebel, "On a constructive proof of Kolmogorov's superposition theorem," *Constructive Approximation*, vol. 30, pp. 653–675, 2009.

[84] J. Schmidt-Hieber, "The Kolmogorov-Arnold representation theorem revisited," *Neural Networks*, vol. 137, pp. 119–126, 2021.

[85] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljacic, T. Y. Hou, and M. Tegmark, "KAN: Kolmogorov-Arnold networks," in *Proc. International Conference on Learning Representations*, 2025.

[86] Y. Wang, J. W. Siegel, Z. Liu, and T. Y. Hou, "On the expressiveness and spectral bias of KANs," in *Proc. International Conference on Learning Representations*, 2025.

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

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

[89] Y. Wang, P. Liu, H. Kang, D. Wu, and D. Miao, "ICFNet: Interactive-complementary fusion network for monocular 3D human pose estimation," *Neurocomputing*, 2025.

[90] Q. Zhao, C. Zheng, M. Liu, P. Wang, and C. Chen, "PoseFormerV2: Exploring frequency domain for efficient and robust 3D human pose estimationg," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8877–8886, 2023.

[91] X. Yang and X. Wang, “Kolmogorov-Arnold Transformer,” *International Conference on Learning Representations*, 2025.

[92] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, “Vision Mamba: Efficient visual representation learning with bidirectional state space model,” *Proc. International Conference on Machine Learning*, 2024.