DEEP LEARNING-BASED 3D HUMAN POSE ESTIMATION FOR FASHION APPLICATIONS

PENG JIHUA

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University
School of Fashion and Textiles

Deep Learning-based 3D Human Pose Estimation for Fashion Applications

PENG Jihua

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

December 2023

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

(Signed)

PENG Jihua (Name of student)

ABSTRACT

3D human pose estimation, a foundational task in computer vision, has received significant attention in recent years due to its crucial applications in robotics, healthcare, and sports science. In particular, it is also a very important research topic in the fashion field due to its ability to yield plausible human body regions for cloth parsing. This study aims to address the issues inherent in exiting state-of-the-art (SOTA) methods of 3D pose estimation by proposing three new and efficient models for 3D pose estimation from various inputs, including video sequence and single image inputs. It is also demonstrated in this study, as an application of these proposed methods, 3D poses predicted from video sequence inputs are being applied and retargeted to game and fashion avatars.

Pose estimation covers both 2D and 3D pose estimation, and the latter are technically more challenging. For 3D pose estimation, most existing methods have converted this challenging task into a local pose estimation problem by partitioning the human body joints into different groups based on the relevant anatomical relationships. Subsequently, the body joint features from various groups are then fused to predict the overall pose of the whole body, which requires a joint feature fusion module. Nevertheless, the joint feature fusion schemes adopted in existing methods involve the learning of extensive parameters and hence are computationally very expensive. Thus, in this study, a novel grouped 3D pose estimation network is first proposed, which involves an optimized feature fusion (OFF) module that not only requires fewer parameters and calculations than existing methods but also is more accurate. Furthermore, this network introduces

a motion amplitude information (MAI) method and a camera intrinsic embedding (CIE) module which are designed to provide better global information and 2D-to-3D conversion knowledge thereby improving the overall robustness and accuracy of the method. In contrast to previous methods, the proposed new network can be trained end-to-end in one single stage, and experiment results have demonstrated that this new method outperforms previous state-of-the-art methods on two benchmarks.

The above first new method for 3D pose estimation is based on convolution neural network (CNN) for grouped feature fusion. In view of the rapid advancement and outstanding performance for transformer-based deep learning models, another novel method, called Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), is also proposed for 3D pose estimation with video inputs. This network contains two novel prior attention modules called Kinematic Prior Attention (KPA) and Trajectory Prior Attention (TPA). KPA models kinematic relationships in the human body by constructing a topology of kinematics. On the other hand, TPA builds a temporal topology to learn the priori knowledge of joint motion trajectory across frames. In this way, the two prior attention mechanisms can yield Q, K, V vectors with prior knowledge for the vanilla self-attention mechanisms, which helps them to model global dependencies and features more effectively. With a lightweight plug-and-play design, KPA and TPA can be easily integrated with various state-of-the-art models to further improve the performance in a significant margin with only a small increase in the computational overhead.

For handling single image inputs, the third new network is designed in this study for

3D pose estimation, which effectively combines the graph and attention mechanism. This method can effectively model the topological information of the human body and learns global correlations among different body joints more efficiently.

Being a demonstration for potential application for these proposed methods, motion retargeting technique is used to transfer the predicted 3D human poses from fashion images/videos to other people, so that different people can perform the same motion, e.g. catwalk, realizing multiplayer motion animation.

Keywords: Deep Learning, 3D Human Pose Estimation, Motion Amplitude, Feature Fusion, Transformer, Self-Attention Mechanisms, Graph Convolutional Network.

PUBLICATIONS

1. **Peng, J.**, Zhou, Y., & Mok, P. Y. (2022, July). 3D POSE ESTIMATION BY GROUPED FEATURE FUSION AND MOTION AMPLITUDE ENCODING. In 16th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, CGVCVIP 2022, 8th International Conference on Connected Smart Cities, CSC 2022, 7th International Conference on Big Data Analytics, Data Mining and Computational Intelligence, BigDaCI 2022, and 11th International Conference on Theory and Practice in Modern Computing, TPMC 2022-Held at the 16th Multi Conference on Computer Science and Information Systems, MCCSIS 2022 (pp. 27-34).
2. **Peng, J.**, Zhou, Y., & Mok, P. Y. (2022). Balanced Feature Fusion for Grouped 3D Pose Estimation. International Conference on Computer Graphics, Visualization and Computer Vision 2022.
3. **Peng, J.**, Zhou, Y., & Mok, P. Y. (2024). KTPFormer: Kinematics and Trajectory Prior Knowledge-Enhanced Transformer for 3D Human Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1123-1132).
4. **Peng, J.**, Zhou, Y., & Mok, P. Y. A cross-feature interaction network for 3D human pose estimation. Pattern Recognition Letters (PRL). under review, 2024.
5. **Peng, J.**, Zhou, Y., & Mok, P. Y. EHFusion: An efficient heterogeneous fusion model for group-based 3D human pose estimation. The Visual Computer (TVC). under review, 2024.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincerest gratitude to my supervisor, Dr Tracy Mok who offers me an invaluable opportunity to pursue my doctorate. I learned a lot from her during my PhD studies, which included critical thinking, research tastes, and academic writing skills. Also, I want to thank her for the careful revisions she made to each of my papers. Without her patient guidance and help, I would not have been able to complete my doctoral studies, nor would I have been able to publish the paper in a top-tier conference. I am deeply honored to be able to engage in academic discussions and collaborate with the members of Dr Tracy Mok's research team.

Then I want to thank my senior, Dr. Zhou Yanghong, who has provided me with great help, whether in research or in life. When I first started conducting research during my doctoral stage, she patiently guided me step-by-step on the research approach, and engaged in discussions with me. I was able to get off to a smooth start in my research, and even to this day, conduct research independently, all of which would not have been possible without her guidance.

Next, I want to express my gratitude to my two best senior fellow students, Zhang Feng and Wang Chen. You have given me a lot of valuable research ideas for my experiments and provided valuable revisions for my papers. From both of you, I've learned many useful research techniques.

I also would like to thank Dr. Hong Zicong, my best friend at PolyU. I really enjoyed the time we spent working out together. Thank you for your encouragement and support, which have enabled me to persevere on the research path. And my other two best friends,

Cheng Yu and Tang Zhenhua. Thank you for taking the time to discuss ideas and paper writing with me. I've learned a lot of research techniques from both of you.

During my long doctoral studies, some friends have made my life more enriching and interesting. I would like to thank Yang Fan, Rao Kaidi, Li Danning, Jiang Zijing, Teng Jingcheng, Liang Guoping and other friends in PolyU. I really enjoyed the times we spent playing basketball together.

Finally, I would like to thank my family for their support and help in keeping me going through my PhD career. I would like to thank all the professors for their valuable revisions of my doctoral thesis.

TABLES OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1 Research Backgrounds

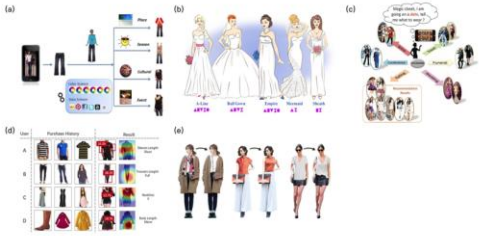
Fashion can reflect the lifestyle and cultural background of its period. It not only has innovative concepts and designs, but also requires certain control over quality. In contemporary society, fashion has a significant impact on all aspects of social life, including social economy, politics, and culture. Since fashion has a significant influence on society and economy, researchers in many disciplines have conducted research and analysis on fashion from different perspectives, making fashion a new type of multidisciplinary research. For example, fashion designers design products by studying fashion trends. Marketing experts use changing consumer habits to maximise profits. Psychologists and sociologists focus on individual and group clothing style.

In recent years, many computing scholars have actively participated in fashion-related research studies because of the readily applicability of machine-learning and computer vision techniques based on the widely available digital resources of online fashion images and videos. Many promising work has been published in top conferences, focusing on semantic segmentation of fashion images (Dong *et al.*, 2015; Liang *et al.*, 2015; Martinsson & Mogren, 2019), semantic recognition of fashion images (Bossard *et al.*, 2012; Di *et al.*, 2013; Verma *et al.*, 2018), fashion analysis (Gu *et al.*, 2020; He & McAuley, 2016; Vittayakorn *et al.*, 2015) and fashion recommendation (Ding *et al.*, 2021; Hu *et al.*, 2015; McAuley *et al.*, 2015; Veit *et al.*, 2015). As shown in Figure 1-1,

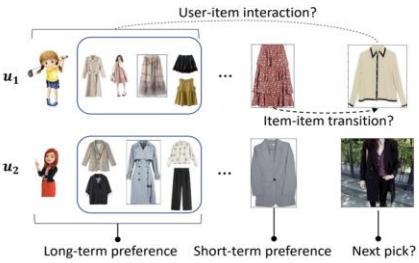
semantic segmentation divides fashion images into multiple regions with semantic labels. Semantic recognition of fashion images focuses on identifying the categories and attributes of clothing from images. The goal of semantic recognition of fashion images is to identify the categories and attributes of clothing from images. Both are very useful for many applications like fashion trend analysis. Fashion recommendation aims to recommend corresponding products according to personal fashion preferences of individual consumer. Fashion analysis studies some valuable fashion cases based on specific data sets and techniques such as clothing recognition.

(a) Semantic Segmentation of Fashion Images (Martinsson & Mogren, 2019)

(b) Semantic Recognition of Fashion Images (Verma, Anand, Arora, & Rai, 2018)



(c) Fashion Analysis (Gu, Gao, Tan, & Peng, 2020)



(d) Fashion Recommendation (Ding, Ma, Wong, & Chua, 2021)

Figure 1-1 Research in the field of fashion.

The fashion industry has an important role in the global economy, the industry is very interested in research applications that improve consumer experience, in particular online shopping experience. For example, some online shopping platforms allow users to take photos of favourite items, and they search for the item in the photo or similar products accordingly. Alibaba iDST video analysis team proposes an online clothing

retrieval system which adopts the most advanced clothing detection and tracking technology to help customers to look for similar style of celebrities or actors/actresses while watching movies and TV. Other examples include applying artificial intelligence to fashion design work, such as Google's Muze and Amazon's Runway projects.

With e-commerce being a major way now people shopping clothing products online, new technology is always in demand to address the need of trying on clothing. Only viewing fashion images at online shopping platform may not satisfy consumers nowadays, and they want to wear their selected clothes virtually to visualise the wearing effects. Revealing how specific clothing will look on people can be achieved by combining clothes parsing technology and motion retargeting technique on personalised avatars, while human pose estimation is the foundation technology for both, which is the key focus of this study.

Human pose estimation, a crucial task in computer vision, is primarily categorized into 2D human pose estimation and 3D human pose estimation. The 2D human pose estimation predicts the pose of a human body, in terms of pixel locations, on input images, while the 3D task involves predicting human pose coordinates in 3D space based on inputs of either single human images or videos. Human pose estimation can be applied for behaviour analysis, human-computer interaction, auxiliary pedestrian detection and virtual reality. The main difficulties of human pose estimation are as follows: First of all, the intricate nature of human body images necessitates the model to acquire a deep understanding of highly nonlinear mapping relations, posing a

significant challenge for threefold reasons. Firstly, human body images were taken in different scenarios, with different shooting angles and lighting conditions; secondly, the interaction between people and objects, as well as the interaction between people will cause random occlusion; thirdly, different wearing and body shapes also increase the complexity of the mapping between joints (also called keypoints) with pixels. Although traditional methods based on handcrafted features can achieve accurate positioning of unobstructed joints under fixed scenes, viewing angles and stable lighting conditions, such ideal situation is very rare in real situation. In this regard, extracting robust features and learning complex mapping relationships is an important research direction in the human pose estimation. On the other hand, the highly non-linear mapping relationship needs to be learned with a higher complexity model which requires significant computational overhead. Hence, speeding up the convergence rate of the model while ensuring the accuracy of the model is a key issue for the practicality of the human pose estimation.

In order to extract more robust features and learn complex mapping relationships, LeCun *et al.* (2015) introduced deep learning techniques. Deep learning denotes a set of machine learning techniques grounded in artificial neural networks, alternatively termed deep structured learning, or deep hierarchical learning. Early deep learning studies mainly focused on the research related to Restricted Boltzmann Machine (RBM) and Auto encoder (AE). In 2012, the outstanding performance of Convolutional Neural Network (CNN) in the ImageNet competition sparked an upsurge in CNN research.

Toshev and Szegedy (2014) introduced the CNN to the field of human pose estimation, sparking a surge in research dedicated to human pose estimation leveraging this neural network architecture. The human body pose estimation method based upon deep learning, primarily on CNN and later on Transformer-based, becomes the current mainstream method for the following reasons: (1) The manual features are designed by the researcher based on experience, and the extracted features are not optimal for the human pose estimation. CNN possess the capability to autonomously acquire image representations from data, thereby circumventing the limitations associated with manual feature engineering. (2) The method based on manual features cannot achieve the joint optimization of feature extraction and human body model, while the end-to-end optimization of network unifies the representation learning and human body modelling. After the researchers have defined the problem, they only need to design a reasonable network architecture and loss function to achieve model learning.

In 2017, Vaswani *et al.* (2017) proposed the transformer architecture for natural language processing tasks. It utilized self-attention mechanisms to capture relationships between different elements in a sequence. Subsequently, transformers have been extended to various domains. In human pose estimation, transformers have demonstrated enhanced modeling capabilities for human pose sequences, achieving the superior performance compared to CNNs. However, the increased consumption of computational resources poses a challenge for transformers in human pose estimation. Given that the human body can be represented as graph-structured data, Graph

Convolutional Networks (GCNs) have been extensively applied in human pose estimation, yielding promising results. Nevertheless, GCNs excel at capturing local information but demonstrate limited capability to model global correlations. This study proposes new CNN-based and Transformer-based networks, addressing the current issues in related work for more effective and efficient 3D pose estimation.

1.2 Statements of the Problem

Here outlines the unique characteristics and issues of existing methods for 3D pose estimation, which this study is endeavour to address.

In the existing CNN-based methods for 3D human pose estimation, the SOTA approach mainly take advantage of grouping strategy, which partitions the human body joints into different parts (arms, legs, and torso). After the joints are divided into different groups, each group's joint features are independently encoded and then a joint feature fusion scheme is usually used to fuse these features from various groups together to predict the overall pose of the full body. Nevertheless, the joint feature fusion schemes adopted in existing methods involves the learning of extensive parameters and hence are computationally very expensive. Moreover, to prevent interference among features from different groups, the grouped method (Shan *et al.*, 2021a) often employ a multi-stage training strategy rather than an end-to-end approach, leading to an increase in training time.

For existing transformer-based methods for 3D human pose estimation (Li *et al.*, 2022b; Li *et al.*, 2022c; Shan *et al.*, 2022; Tang *et al.*, 2023b; Zhang *et al.*, 2022b; Zhao *et al.*,

2023; Zheng *et al.*, 2021a), the main focus was often on developing novel transformer encoders. They model either the spatial correlation between joints within each frame and the pose-to-pose or joint-to-joint temporal correlation across frames. Regardless of spatial or temporal multi-head self-attention (MHSA) calculation, the present transformer-based methods all use linear embedding where 2D pose sequence are tokenized into high dimensional features and treated uniformly to compute the spatial correlation between joints and the temporal correlation across frames in the spatial and temporal MHSA, respectively. This may lead to the problem of 'attention collapse', a phenomenon denoting a circumstance wherein the self-attention becomes too focused on a limited subset of input tokens while disregarding other segments of the sequence. There are some work (Zhao *et al.*, 2022; Zhu *et al.*, 2021) that combines GCN and transformer to learn both local and global dependencies for 3D pose estimation based upon single frame. Nevertheless, these studies merely employ the GCN and the self-attention mechanism in a straightforward manner for feature extraction, without effectively integrating the extracted features. This leads to a situation where local features and global features interfere with each other, rather than being complementary, resulting in a decrease in the performance.

1.3 Research Aims and Objectives

This study aims to develop novel deep-learning based methods for effective and efficient 3D human pose estimation, addressing those issues of existing methods. A total of three new methods are developed in this study. Among the three, the first two

methods take 3D poses from input of video sequences, while the last one is based on single frames.

The specific research objectives are listed as follows:

i. To comprehensively review and understand deep learning techniques about human pose estimation.

ii. To compare and analyse machine learning technologies about 3D human pose estimation.

iii. To design and develop a new CNN-based method for 3D pose estimation with less computational overhead and improved performance, allowing end-to-end training.

iv. To design and develop a novel transformer-based network for 3D pose estimation with video sequence inputs.

v. To present an effective network design that skilfully combines the graph and attention mechanisms for 3D pose estimation from single frames.

vi. To comprehensively evaluate the effectiveness of the proposed methods in comparison with relevant state-of-the-art methods.

vii. To demonstrate the potential fashion application of the prssoposed 3D human pose methods.

1.4 Methodology Overview

As mentioned, this study investigates the cutting-edge deep learning-based methods for

3D human pose estimation and suggests relevant potential fashion applications. Figure 1-2 shows the overall research framework of this study, which include the developments of three DL-based networks, including (1) CNN-based model for group-based 3D pose estimation with an efficient heterogeneous fusion, (2) Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), a transformed based method, and (3) a Cross-Feature Interaction Network, again a transformer-based method. The predicted 3D poses are demonstrated through a fashion application for motion retargeting to several avatars.

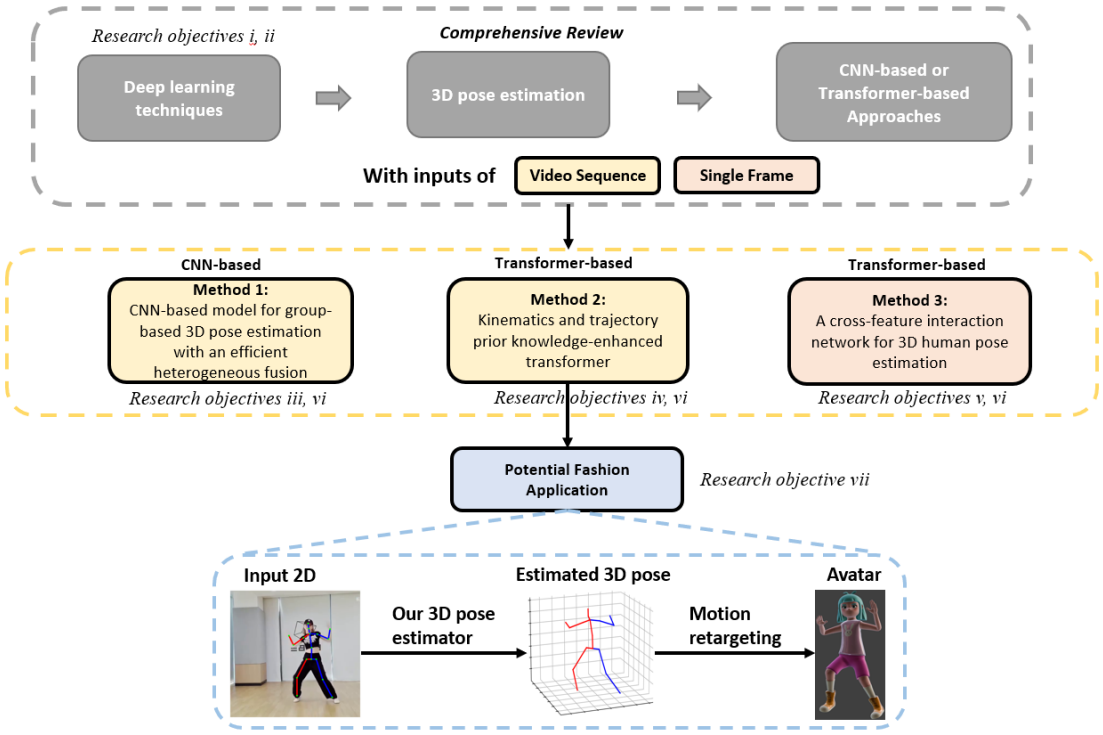


Figure 1-2 Illustration of the overall research framework.

The first model of CNN-based model for group-based 3D pose estimation with an efficient heterogeneous fusion improves the performance and requires fewer parameters and calculations than other existing state-of-the-art CNN-based methods. In this model, a heterogeneous feature fusion (HFF) module is developed to fuse different groups of

joint features more efficiently. Furthermore, this new model introduces a motion amplitude information (MAI) and a camera intrinsic embedding (CIE) sswhich are designed to provide better global information and 2D-to-3D conversion knowledge, thereby improving the overall robustness and accuracy. In contrast to previous SOTA models, the proposed new network can be trained end-to-end in one single stage. Experiments were conducted on two public datasets (Human3.6M (Ionescu *et al.*, 2013) and HumanEva (Sigal *et al.*, 2010)) to validate the effectiveness of this model.

Next, a novel transformer-based model, called Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), is proposed as the second method for 3D pose estimation with video inputs. This network contains two novel prior attention modules − Kinematic Prior Attention (KPA) and Trajectory Prior Attention (TPA). KPA models kinematic relationships in the human body by constructing a topology of kinematics. On the other hand, TPA builds a temporal topology to learn the priori knowledge of joint motion trajectory across frames. In this way, the two prior attention mechanisms can yield Q, K, V vectors with prior knowledge for the vanilla self-attention mechanisms, which helps model global dependencies and features more effectively. With a lightweight plug-and-play design, KPA and TPA can be easily integrated with various state-of-the-art models to further improve the performance significantly with only a small increase in the computational overhead. Extensive experiments were conducted on three benchmarks, including Human3.6M (Ionescu *et al.*, 2013), MPI-INF-3DHP (Mehta *et al.*, 2017) and HumanEva (Sigal *et al.*, 2010), to

evaluate this model in comparison with other state-of-the-art, transformer-based or non-transformer-based methods.

Both the first and second models are designed for 3D pose estimation based on video sequence inputs, while a Cross-Feature Interaction Network, is designed to leverage GCN and the multi-head self-attention (MHSA) to capture the local features and global features, respectively, retaining the initial 2D pose joint features in the third branch of the network. Moreover, we design a specific multi-head cross-attention (MHCA) to facilitate cross-feature communications among three different features (local features, global features and initial 2D pose features) and aggregate them to form the enhanced spatial representations of single pose. Besides, a parallel GCN and multi-layer perceptron (GraMLP) module is introduced to inject the skeletal knowledge of human body into the final 3D pose representation. Again, experiments were conducted on Human3.6M (Ionescu *et al.*, 2013) and MPI-INF-3DHP (Mehta *et al.*, 2017) datasets to validate the effectiveness of this third model.

Being a demonstration for potential application for these proposed methods, motion retargeting technique is used to transfer the predicted 3D human poses from fashion images/videos to other people, so that different people can perform the same motion, e.g. catwalk, realizing multiplayer motion animation.

1.5 Organization of the Thesis

In 0, recent deep learning techniques and human pose estimation methods, including traditional methods, convolutional neural networks and transformers are reviewed.

In 0, the first model of CNN-based model for group-based 3D pose estimation with an efficient heterogeneous fusion is presented, giving detail designs of the heterogeneous feature fusion (HFF) module, motion amplitude information (MAI) and camera intrinsic embedding (CIE). Experimental results on public datasets (Human3.6M and HumanEva) and discussion are presented in the same chapter.

In 0, the second method of this study, namely Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), is introduced and evaluated on public benchmarks (Human3.6M, MPI-INF-3DHP and HumanEva). This method is evaluated and compared with other state-of-the-art methods on these datasets.

In 0, the third network, a Cross-Feature Interaction Network, is introduced for 3D pose estimation with single frame inputs. The model is again evaluated comprehensively by carefully planned experiment on two public datasets (Human3.6M and MPI-INF-3DHP).

0 applies the estimated 3D human pose from the proposed 3D pose estimation methods to various avatars for potential fashion and game applications. Qualitative analysis was conducted through visualising the motion retargeting results on various avatars.

Lastly, the findings of this study are summarised in Chapter 7, with discussions on its limitations and possible future work.

CHAPTER 2. LITERATURE REVIEW

This chapter reviews the related work and important techniques that serve the foundation of the current study on human pose estimation. Since this study is based mainly on deep learning techniques, its development history and major network structures of deep learning are first reviewed in Section 2.1. In Section 2.2, the traditional methods for human pose estimation are summarised, followed by the deep learning based methods of CNN and Transformer related work in recent years in Sections 2.2.2 and 2.2.3, respectively.

2.1 Deep Learning

The exploration of artificial neural networks give rise to the inception of deep learning. Mathematical models inspired by biology and neurology are what constitute artificial neural networks, which constructs neurons like the human brain and connects them according to a certain structure to simulate the biological nervous system. Neural network is a machine learning model that needs to connect individual neurons to achieve complex functions. When it consists of many layers of neurons, it is called the deep neural network. Deep learning was proposed by (LeCun *et al.*, 2015), which employed deep neural networks. We will introduce several commonly used deep learning networks in the following sections.

2.1.1 Feedforward Neural Network

The feedforward neural network, widely utilized, stands out as one of the most fundamental structures in neural networks. Employing a unidirectional multilayer

structure, the feedforward neural network features numerous neurons in each layer. Within this neural network, every neuron can receive signals from the neurons in the preceding layer and produce outputs directed to the subsequent layer. The initial layer, which accepts signal inputs, is termed the input layer, while the concluding layer is known as the output layer. Intermediate layers between them are referred to hidden layers, which may consist of a single layer or multiple layers. In these hidden layers, each node connects to the nodes in the subsequent layer via a weight vector represented by W and a bias denoted by b. Operating without feedback, the network facilitates the unidirectional propagation of signals from the input layer to the output layer. Figure 2-1 illustrates a typical multilayer feedforward neural network. Each small circle represents a perceptron model. Each neuron in the first layer of the network receives the input signal, and then outputs it to the next layer of neurons after weighted summation by its own neural body. The neurons of the second layer get their inputs from the outputs of the previous layer. Finally, after the computation of the intermediate neural network, the Zand this network structure is generally used for regression tasks. The neural network used for classification will output the number of results corresponding to the label at the end.

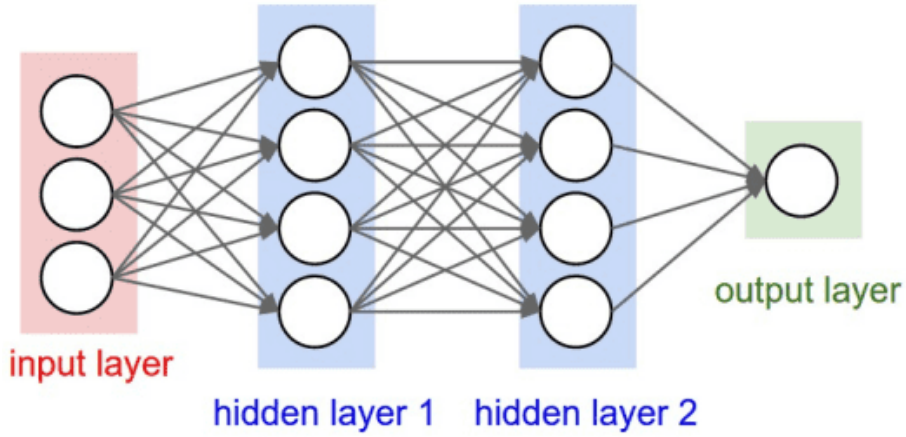


Figure 2-1 A typical multilayer feedforward neural network.

In the feedforward neural network, there is a very important function called the activation function that is generally located in the hidden layer. The activation function operates on the neurons within an artificial neural network, mapping the input of each neuron to its respective output. In the above perceptron model, the neurons treat the input parameters in a weighted summation manner. Therefore, the output is the result of a linear superposition of the input parameters. However, the distribution of the data is mostly non-linear. To adapt to a nonlinear space, most networks introduce activation functions that reinforce the learning ability of the network. Different activation functions have different applications depending on their characteristics. To enrich the representation and learning capability of the network, the activation function generally needs to satisfy these requirements: 1) the activation function should be a continuously derivable nonlinear function; 2) the derivation process of the activation function should be as simple as possible in order to improve the efficiency of the network; 3) The derivative value should be in a certain interval. Too large value will affect the stability of network training, while too small value will affect the efficiency of network training.

In the following section we will introduce several common activation functions.

Sigmoid is a commonly used nonlinear activation function, which has the following mathematical formula:

$$f(z) = \frac{1}{1+e^{-z}} \quad (2\text{-}1)$$

It can convert a continuous real value into an output ranging from 0 to 1. In particular, when confronted with a significantly large negative input, the output is driven to 0, while a correspondingly large positive input results in an output of 1. Sigmoid is a continuous derivative function and the derivative is simple. The output of Sigmoid function can be seen as a gating mechanism to control the amount of information in the output. However, since the value of the Sigmoid function is constantly greater than 0, this non-zero centrality causes bias shifts in the inputs between layers, making convergence slower. When the output value of the neuron is much larger than 0 or much smaller than 0, using the sigmoid function during the training of the neural network causes the gradient to disappear during training. Currently, the network is updated with a return gradient close to 0 and the network stops optimizing.

Another common type of activation function is the Tanh function. Tanh is a hyperbolic tangent function that is like sigmoid. They both belong to the saturation activation function, and the difference is that the output ranges from (0, 1) to (-1, 1). The Tanh activation function is written as follows:

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (2\text{-}2)$$

Tanh is also a zero-centered symmetric function that does not cause bias shifts between

different layers. However, the gradient of the function is still close to 0 when the input value is much larger than 0 and much smaller than 0. The problem of gradient disappearance is still not solved. The most widely utilized activation function in neural networks is the modified activation function ReLU. The formula can be written as follows:

$$ReLU(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2\text{-}3)$$

The ReLU function is a linear function when x is greater than 0, and equal to 0 when x is less than 0. The ReLU function is computationally simple, and the derivative is easy to get, and the neurons using ReLU as the activation function are computationally efficient. The ReLU function has the property of one-sided inhibition and wide excitation band, so it is active when the input is greater than 0. Meanwhile, the ReLU function alleviates the gradient vanishing problem possessed by Sigmoid and Tanh. However, if all training data do not activate a ReLU neuron, then the gradient of the network gradient is always 0. The use of batching operation can solve this problem, so ReLU is widely used in current neural network architectures.

In general, the feedforward neural network is the simplest network because it has no feedback, which is used for learning and correction of parameters.

2.1.2 Convolutional Neural Network

The Convolutional Neural Network (CNN) stands as a representative algorithm within the domain of deep learning. It embodies a feedforward neural network with a deep structure, incorporating convolutional computation. CNN can achieve translational

invariant classification of input data based on its hierarchical structure. It takes advantage of the local similarity of images, and the semantic information in the images is not affected when the images are scaled and panned.

2.1.2.1 Development History

The exploration of CNN commenced during the 1980s and 1990s. The Japanese scholar Fukushima and Miyake (1982) first proposed the neocognitron model in 1982. He conceived the neural network called "neocognitron" with the goal of emulating the visual cortex of living beings. Neocognitron is characterized by its deep structure and stands as one of the earliest deep learning algorithms, featuring alternating S-layers (Simple-layer) and C-layers (Complex-layer) as implicit components. The integration of S and C layers in neocognitron facilitates feature extraction and filtering, partially fulfilling the roles of the convolution layer and the pooling layer in the CNNs. This groundbreaking research serves as a seminal inspiration for convolutional neural networks.

The inaugural convolutional neural network, known as the Time Delay Neural Network (TDNN), was introduced by (Waibel *et al.*, 1989). TDNN applied the CNN to address the speech recognition challenge, utilizing an FFT pre-processed speech signal as its input. The network featured an implicit layer incorporating two 1D convolutional kernels aimed at extracting translational invariant features in the frequency domain. Notably, TDNN benefited from the progress in Backpropagation (BP) algorithms within the field of artificial intelligence that preceded its emergence, allowing it to

leverage the BP framework for learning.

In 1988, Zhang *et al.* (1996) introduced the initial 2D convolutional neural network, termed the translation-invariant artificial neural network (SIANN). They successfully applied SIANN to the detection of medical images. LeCun *et al.* (1989b) developed a convolutional neural network designed for computer vision challenges, and this marked the inception of the original version of LeNet. LeNet comprised two convolutional layers, two fully connected layers, and a total of 60,000 learning parameters. Notably, the network's architecture was substantially larger than those of TDNN and SIANN. Structurally, LeNet bore a close resemblance to contemporary convolutional neural networks. LeCun *et al.* (1989b) employed Stochastic Gradient Descent (SGD) for learning following the random initialization of weights. This approach was subsequently maintained in subsequent deep learning research. In 1998, LeCun *et al.* (1998b) developed a more complete neural network called LeNet-5, and succeeded in the problem of handwritten digit recognition. LeNet-5 adheres to the learning strategy of (LeCun *et al.*, 1989b) and extends the original design by incorporating a pooling layer to filter input features. This addition of pooling layers is instrumental in shaping the basic structure of modern convolutional neural networks. LeNet-5 and its subsequent variations establish a foundational framework where alternating convolutional-pooling layers effectively extract translation-invariant features from input images.

Although convolutional neural networks were already established in 1998, they did not

show significant advantages over the mainstream methods of combining support vector machines with manual features due to the limitations of computer performance and the lack of training datasets. Therefore, they did not receive much attention. Following the proposal of deep learning theory by (Hinton & Salakhutdinov, 2006), there was a notable surge in interest and development of the representational learning capability of convolutional neural networks. This progress was further facilitated by advancements in numerical computing devices. In 2008, NVIDIA introduced the concept of General-purpose Graphics Processing Units (GPGPU) and created the Compute Unified Device Architecture (CUDA) computing library to enable acceleration of scientific computing. In 2008, NVIDIA introduced the concept of General-purpose Graphics Processing Units (GPUs) and created the Compute Unified Device Architecture (CUDA), which enables acceleration of scientific computing. In 2012, Krizhevsky *et al.* (2012a) introduced the AlexNet and achieved efficient training of the network based on CUDA, and won the ImageNet classification competition, which quickly attracted widespread attention to convolutional neural networks. Since then, research on convolutional neural networks has been flourishing. Researchers propose many improvements to the network architecture and apply convolutional neural networks to various application scenarios (e.g., target detection, face recognition, object classification, semantic segmentation, pedestrian re-identification, pose estimation).

2.1.2.2 Network Structure

Modern convolutional neural networks mainly consist of convolutional layer, pooling

layer, fully connected layer. Earlier classification tasks or regression tasks used a fully connected layer after the feature extractor, and the dimensionality of the features was reduced by the fully connected layer. However, too many parameters in the fully connected layer can increase the computational overhead of the network and cause overfitting. In order to reduce the number of parameters and computation of the network and avoid overfitting, some researchers have tried to use the global pooling layer instead of the fully connected layer and obtained the same results as the fully connected layer. In some dense prediction assignments like semantic segmentation and pose estimation, convolutional layers are typically positioned close to the output layer within the network. This placement is essential because these tasks necessitate the implementation of a fully convolutional network to retain spatial details, enabling the creation of accurate segmentation maps or heat maps. The subsequent section provides an elaborate breakdown of each component within the convolutional neural network.

The convolutional layer is the core component of a convolutional neural network and consists of convolutional kernels, which aim to extract the local features. Convolution kernels share parameters when sliding over images or feature maps. This type of parameter sharing can significantly reduce the number of parameters. Figure 2-2 illustrates the 2D convolution operation. While each convolutional kernel possesses a small receptive field, the cumulative effect of stacking multiple convolutional layers allows the entire network to encompass a large receptive field. When a convolutional neural network is forwarded, each convolutional kernel slides over the input image or

feature map and calculates the dot product with the current local receptive field, which is used as the activation value for a location in the feature map. When the sliding is over, the convolution layer outputs a new feature map.

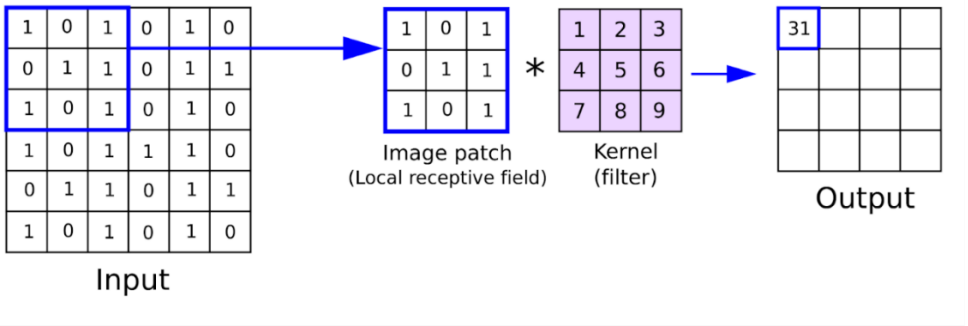


Figure 2-2 2D convolution operation.

Besides, researchers also add a zero-padding operation and introduce different strides to increase the expressiveness of the convolution, making feature extraction more flexible.

Figure 2-3 shows the zero padding in the convolution. Convoluting the input image with a convolution kernel can result in the loss of information at the image boundaries. This occurs because pixels at the edges of the image are never positioned at the centre of the convolution kernel, and the kernel cannot extend beyond the edge region. Introducing zero padding enables the convolution kernel to extend beyond the edges, incorporating pseudo-pixels when scanning the input data. This ensures that the output and input maintain the same size.

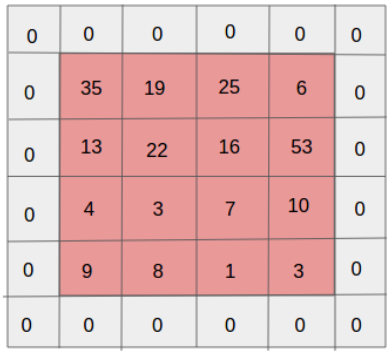


Figure 2-3 Zero padding.

Figure 2-4 shows the case where the stride in the convolution is equal to 2. The convolution kernel starts with the top-left corner of the input and slides one column to the left or one row down to calculate the output one by one. The number of rows and columns in each slide is called Stride. The purpose of the Stride parameter is to exponentially decrease the size, with the specific reduction factor determined by its numerical value. For instance, if the stride is set to 2, the output size becomes half of the input; similarly, a stride of 3 results in an output size one-third of the input.

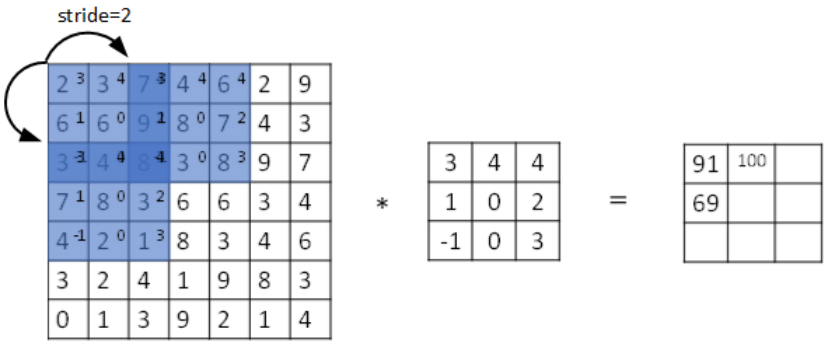


Figure 2-4 Stride=2.

Changing the strides of the convolution and zero-padding methods can improve the feature extraction ability of traditional convolution, but these methods still cannot solve the following drawbacks of traditional convolution: a) The local operation of the

convolution kernel prevents convolution from directly obtaining the global features of the image. The only way to get the global features is to stack the convolution layers. The constant stacking of convolutional layers will make the number of parameters larger and the computational overhead too high. b) Convolution does not enable recovery from smaller feature maps to larger ones. In order to solve these problems, some new convolutional structures have been proposed, such as dilated convolutions (Yu & Koltun, 2015), transposed convolution (Dumoulin & Visin, 2016).

Dilated convolution is executed by introducing gaps or "holes" into the standard convolutional map, thereby expanding the receptive field of the network. The original convolution gets local information from adjacent positions, while the dilated convolution gets local information from partially adjacent positions. In addition to determining the length and width of the convolution kernel, the dilated convolution also requires determining the dilation rate that refers to the interval distance between each weight in the convolution kernel. The dilation rate of the original convolution is 1. Figure 2-5 shows the dilated convolution. The kernel size in the figure is 3×3. As the dilation rate increases, the receptive field of the convolution also increases. Dilated convolution does not need to increase the convolution kernel or stack convolution layers to increase the reception field, which saves computational resources and does not cause overfitting problems. In Figure 2-5, the dilated convolution with a dilation rate of 2 has a reception field of 5×5 with the same parameters.

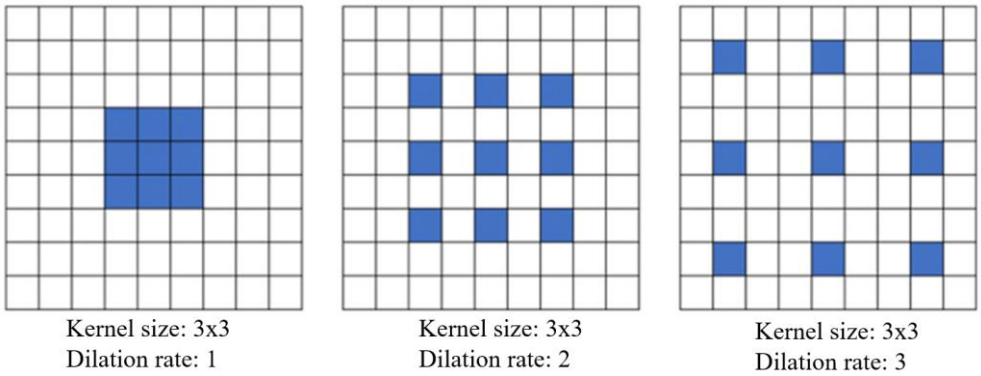


Figure 2-5 Dilated convolution.

As an input image undergoes feature extraction via a convolutional neural network, the output size often diminishes. Occasionally, it becomes necessary to restore the image to its original size for subsequent computations, such as semantic segmentation. This process, aimed at mapping the image from a smaller to a larger resolution, is referred to as upsampling. There are 3 common methods for upsampling: bilinear interpolation, transposed convolution, and nearest neighbour interpolation. Transposed convolution solves the upsampling problem of feature map. As shown in Figure 2-6, transposed convolution expands the original low-resolution feature map by the zero-padding operation, and then generate the feature map of the next layer, which is used to achieve upsampling.

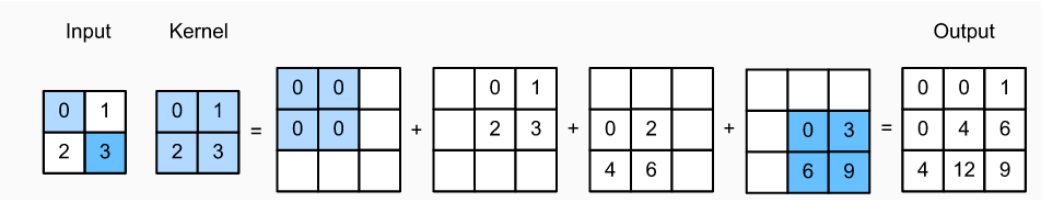


Figure 2-6 Transposed convolution.

The pooling layer constitutes another crucial element of the convolutional neural network, serving as a form of nonlinear downsampling. The primary objective of incorporating pooling layers is to achieve translation invariance, enabling the model to

prioritize the presence or absence of a feature rather than its precise location. This layer effectively reduces the resolution of the feature map, mitigating computational overhead in the network while helping to prevent overfitting. Common types of pooling layers include the maximum pooling layer, average pooling layer, and global maximum pooling layer. Among these, the maximum pooling layer is the most commonly utilized, dividing the input into non-overlapping sub-regions and extracting the maximum value from each sub-region to represent its characteristics. As shown in Figure 2-7, the pooling process is similar to the convolution process. It uses a 2×2 filter with a stride 2 to scan the values in the neighbourhood of a feature map and selects the maximum value to output to the next layer. The pooling operation does not affect the dimensionality of the output and the feature channels remain unchanged.

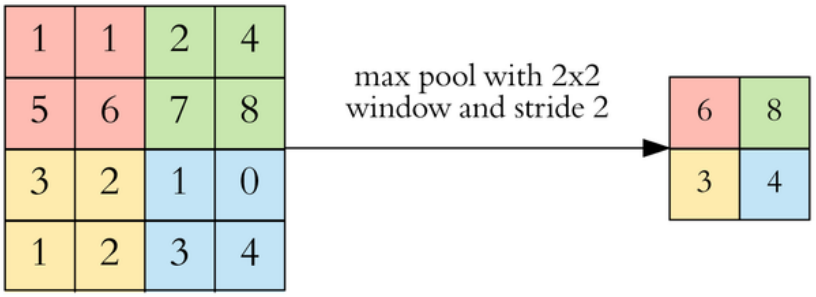


Figure 2-7 Max pooling operation.

The fully connected layer in a convolutional neural network corresponds to the hidden layer in a traditional feedforward neural network. Positioned at the end of the hidden layers in the convolutional neural network, the fully connected layer exclusively transmits signals to other fully connected layers. As shown in Figure 2-8, each neuron within the fully connected layer forms connections with neurons in the preceding layer. Serving as a "classifier" in the convolutional neural network, the fully connected layer

plays a pivotal role. The convolutional layer, pooling layer, and activation function collaboratively map the initial data into a hidden feature space, undertaking the processes of feature extraction and selection. Meanwhile, the fully connected layer further maps the learned feature representation to the labeled space of the samples. Essentially, it integrates these features and channels them towards the final classifier or regression. It's worth noting that the fully connected layer discards location information present in the feature map, thereby reducing the sensitivity of parameters during model learning. However, it is susceptible to parameter redundancy. The parameters associated with the fully connected layer can constitute a significant portion, around 80%, of the overall network parameters. This not only slows down the training speed but also increases the risk of overfitting.

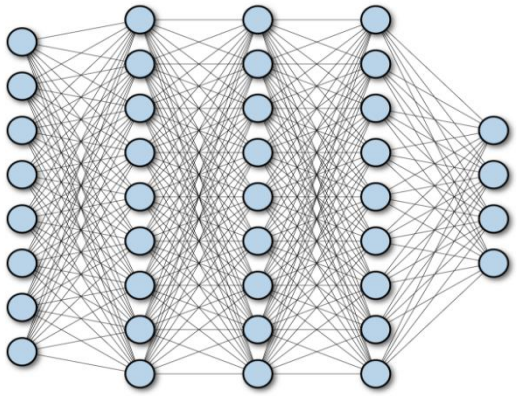


Figure 2-8 Fully connected layer.

2.1.3 Transformer

In 2017, Vaswani *et al.* (2017) proposed the transformer architecture and showed remarkable performance in natural language processing (NLP), as the self-attention can model long-range dependencies and also capture global features. Figure 2-9 shows the

vanilla transformer architecture. Within the self-attention layer, the input vector undergoes an initial transformation into three distinct vectors: the query vector Q, the key vector K, and the value vector V. Then, the attention between different input vectors is calculated as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \quad (2-4)$$

To boost the performance of the vanilla self-attention mechanism, the multi-head attention mechanism is proposed. When considering a particular reference word within a sentence, some key words are often emphasized. The constraint imposed by a single-head self-attention layer impedes the capacity to selectively concentrate on one or more specific positions without concurrently affecting attention toward other positions of equal significance. To address this limitation, divergence in representation subspace is introduced across attention heads. Specifically, distinct query, key, and value matrices are employed for different heads. In this way, these matrices can project input vectors into different feature subspaces. The equation of multi-head attention can be written as follows:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W \quad (2-5)$$

$$head_i = Attention(Q, K, V) \quad (2-6)$$

Where h is the number of heads, W is the projection matrix. Following the multi-head attention layers in each encoder and decoder, a feed-forward network (FFN) is employed. This network comprises two linear transformation layers with a nonlinear activation function embedded between them. In addition, a residual connection is

introduced to every sub-layer within both the encoder and decoder, enhancing the information flow to achieve better performance.

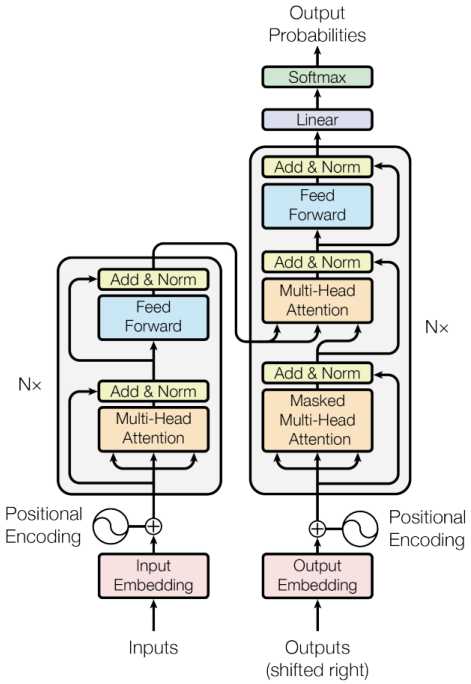


Figure 2-9. The vanilla transformer architecture (Vaswani et al., 2017).

Benefiting from the powerful modeling capability of the multi-head attention mechanism, scholars have recently endeavoured to employ transformers in addressing the computer vision tasks. Chen *et al.* (2020) trained a sequence transformer with the objective of auto-regressively predicting pixels. Dosovitskiy *et al.* (2020) applied a vanilla transformer directly to sequences of image patches for the purpose of classifying the entire image, achieving state-of-the-art performance across various image recognition benchmarks. In addition to image classification, transformer has been employed to tackle other computer vision tasks. Carion *et al.* (2020) presented a transformer encoder-decoder architecture DETR for object detection, eliminating the necessity for numerous manually crafted components including a non-maximum

suppression procedure or anchor generation. Zhu *et al.* (2020) proposed Deformable DETR, exploiting the attention modules to selectively attend to a concise set of key sampling points surrounding a specified reference. Zheng *et al.* (2021b) adopted a pure transformer to encode an image as a sequence of patches, and the encoder can be seamlessly integrated with a transformer decoder. This was a semantic segmentation method entirely based on the transformer. Chen *et al.* (2021a) developed an image processing transformer (IPT) model which is optimized on ImageNet (Deng *et al.*, 2009) benchmark with multi-heads and multi-tails. Zhou *et al.* (2018) introduced an end-to-end transformer network for dense video captioning. In particular, they utilized a self-attention mechanism to incorporate an efficient non-recurrent structure during the encoding process, thereby improving the performance. Due to the outstanding performance of transformers, an increasing number of researchers are proposing transformer-based models to improve a diverse array of visual tasks.

2.2 Human Pose Estimation

2.2.1 Traditional Methods

Before the success of deep learning, there are some traditional human pose estimation methods in the early research. These methods can be divided into feature engineering, pictorial structure and poselets.

2.2.1.1 Feature engineering

Research on human pose estimation first appeared in 1980. Early methods used feature

engineering and assumptions for human pose estimation. Forsyth and Fleck (1997) proposed the Body Plan method which refers to a series of human features learned from image data under the constraints of colour, texture, and geometric attributes. It can be used to segment and recognize the human body in a complex environment. Mori and Malik (2002) obtained the pose of the human body by matching the shape. This method could not only obtain the position of the joints but also realize the tracking of the joints in motion. Ren *et al.* (2005) used the segmentation method to obtain the characteristics of each part of the human body, and then used the relative position between the joints and the consistency of the scale to constrain the model to predict the human pose. Hua *et al.* (2005) used Markov model to infer the human pose from the shape, edge, colour and other information in the image.

2.2.1.2 Pictorial structure

Compared with the earlier methods, the method of pictorial structure has the advantages of low computational complexity and fast prediction. This method is used in some fields such as human tracking, human pose estimation, and automatic discovery of object regions in videos. The research based on the pictorial structure method mainly focuses on three aspects: realizing the rapid calculation of the model, improving the modelling ability of the Appearance Model and the performance of the human pose estimation. Eichner and Ferrar (2009) proposed to use the relationship between the appearance of different parts of the body to improve the modelling performance of the appearance model. Johnson and Everingham (2009) proposed to use Histogram of Oriented

Gradients (HOG) to improve the performance of human component detectors. Sapp *et al.* (2011) proposed to model joints instead of human limbs and used several tree-like sub-models to track joints between video frames. Based on the pictorial structure framework, Felzenszwalb *et al.* (2008) and Felzenszwalb *et al.* (2010) proposed Deformable Part-based Models to improve the modelling capabilities of appearance models. Yang and Ramanan (2011) and Yang and Ramanan (2012) introduced hybrid deformable components into the tree structure model to improve the modelling capabilities of the appearance model. This method uses component-based models (Felzenszwalb *et al.*, 2008) and structured Support Vector Machine (SVM) for learning.

2.2.1.3 Poselets

Poselets (Bourdev & Malik, 2009) is another mainstream method in traditional human pose estimation. This method first needs to construct a data set containing 3D human pose information, and then use a clustering method to divide samples with the same pose in the data set into the same sub-data set. The sub-data sets formed in this way have the same pose but different shapes. Bourdev and Malik (2009) used sub-data sets to train several linear SVM classifiers which are Poselets. Poselets could be used to scan the image at multiple scales after obtained. During the scanning process, the output of the Poselets was fused to determine whether the current image block contains joints and the types of joints. In order to estimate the pose of the upper body, Bourdev and Malik (2013) proposed the Armlet method based on the Poselets method, which divided the data set according to the pose of the arms, and then used the divided data set to train

Armlet. In order to add higher-order information between body parts to the graph structure model, Pishchulin *et al.* (2013) integrated Posetlets into the graph structure model, using the information extracted by the Poselets detector as the middle layer to directly predict the body joints in the image. In order to obtain a stronger local multi-modal shape model, the author uses a rotation-independent part detector. By taking the local shape features obtained by the part detector and the mid-level features obtained by the Poselets detector as the input of the graph structure model, it improves the performance of graph structure model. Since the deformable part model cannot use the annotation information of the joints in the data set like Poselet, Gkioxari *et al.* (2014) proposed to use Poselet for the deformable part model to enhance its performance.

2.2.1.4 Problems with traditional methods

The manual features used in traditional human pose estimation methods include the directional gradient histogram using local gradient contours (Dalal & Triggs, 2005), the directional gradient histogram of gPb contours (Arbeláez *et al.*, 2010), scale-invariant feature transform (Lowe, 2004), color characteristics (Wren *et al.*, 1997) and contextual features (Gkioxari *et al.*, 2013). Traditional methods usually use SVM classifier to classify (Finley & Joachims, 2008) or use deformable component model to model the human body structure (Felzenszwalb *et al.*, 2008). Although traditional methods have achieved good results on simple data sets, they still have many intractable problems. These problems can be explained from the two aspects of feature extractor and human body model respectively. The problems faced by feature extractors: a) The hand-

designed feature extractors can only extract low-level features and cannot capture high-level semantic information; b) The hand-designed feature extractors do not work well with the human body model. The extractor cannot interact with the model during training; c) The hand-designed feature extractor is designed by the researcher based on experience. It is not the optimal feature extraction method and is not necessarily suitable for the task of human pose estimation.

The problems faced by the human body model: a) The graph structure model is only suitable for data with a small number of joints and cannot be extended to the situation with more parameter parts; b) The graph structure model and the deformable component model need to be used manually for the designed features, while the model itself does not have the ability to extract features and cannot interact with feature extraction methods; c) Poselets are more computationally intensive and the training process is cumbersome and complex. When combined with other methods, the model increases the complexity. The above-mentioned problems have prompted researchers to turn their attention to find methods that can perform characterization learning without manual modelling. With the rise of deep learning, especially the continuous development of convolutional neural networks, LeCun *et al.* (1998c) have brought new dawn to solve these problems. This is mainly because the convolutional neural network has the following characteristics: Firstly, the convolutional neural network can directly use the data marked in the training set, and the convolutional neural network is a data-driven method. The more high-quality annotated data, the better the effect of the model.

However, traditional methods cannot learn from a large number of Benefit from data; Secondly, the convolutional neural network can learn very complex nonlinear mapping relationships and deal with problems that plague traditional methods such as random occlusion, complex pose, and changeable appearance; Thirdly, the convolutional neural network can realize representation learning, avoiding the cumbersome manual design of features; Fourthly, the feature extraction inside the end-to-end trained convolutional neural network is integrated with the human body model. The convolutional neural network can automatically learn the feature representation and the human body model from the labelled data set according to the defined loss function.

2.2.2 CNN-based Methods

Convolutional neural networks (CNN) have experienced several years of development before they have matured. Fukushima and Miyake (1982) and Fukushima (1975) proposed the prototype of the early convolutional neural network based on the study of the visual system of cats by Hubel and Wiesel (Hubel & Wiesel, 1962; Hubel & Wiesel, 1965; Hubel & Wiesel, 1977). Subsequently, LeCun (1989) and LeCun *et al.* (1989a) applied the Back Propagation algorithm (Rumelhart *et al.*, 1986) to the convolutional neural network after studying the convolutional neural network to realize the effective training of the network. LeCun *et al.* (1998a) then improved the architecture of the network by proposing the multilayer cascaded convolutional neural network architecture LeNet-5. From an architectural perspective, the early LeNet-5 network lacks some of the key methods in the modern network architecture: rectified linear unit

(ReLU) (Glorot *et al.*, 2011) is used instead of Sigmoid as the activation function to improve network training, and dropout (Srivastava *et al.*, 2014) is used to deal with over-fitting problems. Due to the limitations of the machine performance at the time, the running speed of the convolutional neural network was very slow, which hindered its further development. When convolutional networks were created, the Internet had just emerged without much data accumulation, which was one of the reasons why convolutional neural networks were not popular at that time. It wasn't until 2012 that AlexNet proposed by (Krizhevsky *et al.*, 2012b) won the championship of the ImageNet classification competition at the time that made the world realize the importance of convolutional neural networks. Later, researchers gradually applied convolutional neural networks to their respective research fields, such as 3D human pose estimation. Presently, convolutional neural network-based approaches for 3D human pose estimation can be broadly categorized into two groups: i) directly predicting the 3D coordinates of each joint from 2D images (one-step method); ii) initially predicting 2D joint positions in image space followed by a subsequent lifting to 3D (two-step method).

The one-step method can be subdivided into two categories: regression-based methods and detection-based methods. Regression-based approaches are to directly predict each joint position relative to the root joint position. Li and Chan (2014) used a shallow network to predict 3D joint coordinates directly and realize the task of body part detection simultaneously. Park *et al.* (2016) employed an end-to-end network with

synchronous training of both 2D joint classification and 3D joint estimation. Li *et al.* (2015) applied an embedding sub-network to learn potential human posture structure information and realized the matching of 3D coordinates. The sub-network can use the maximum margin cost function to allocate matching scores to the input image-pose pairs. Tekin *et al.* (2016a) learned a high-dimensional potential pose representation for adding some constraints about the human body with an unsupervised auto-encoder and then introduced a shallow network to predict the 3D coordinates of poses. Sun *et al.* (2017) believed that regression-based approaches did not make good use of the structural information of the human body, so he designed a skeleton-based network by using human body structure information. Furthermore, he also proposed a compositional loss function to solve the problem of no association of bones in the L2 loss. Zhou *et al.* (2016) proposed a deep kinematic neutral network to learn motion parameters and joint locations. Motion parameters include the fixed bones length and angles of bones rotation around combined joints. However, the fixed bones length does not improve the generalization ability of the model well. Nibali *et al.* (2018) believed that the method of predicting by heatmap is completely non-differentiable, while that of regressing the coordinates with the fully connected layer lacks spatial generalization. Therefore, he proposed a module named differentiable spatial to numerical transform (DSNT) to solve these two problems. Luvizon *et al.* (2019) proposed an end-to-end differentiable network, converting feature heatmaps to joint coordinates by a soft-argmax function. Detection-based methods mainly converts the image containing

people into a heatmap for each joint first, and then takes the maximum value of the heat map as the joint coordinates. Pavlakos *et al.* (2017) created a convolutional neural network derived from the stacked hourglass architecture (Newell *et al.*, 2016), predicting the possibility of each voxel of each joint through the fine discretization of 3D space. Liu *et al.* (2019) designed a feature learning neural network to predict 3D hand pose and human pose from an image, which used a new long short-term dependence-aware network to generate 2D heatmaps of joints. Sun *et al.* (2018) connected and unified the heatmap representation and joint regression with an integral operation, thus correcting some non-differentiable error. Luvizon *et al.* (2018) designed a multitask network to enable estimation of 2D and 3D poses jointly and action recognition. It is worth noting that 2D and 3D pose are uniformly predicted by using volumetric heatmaps.

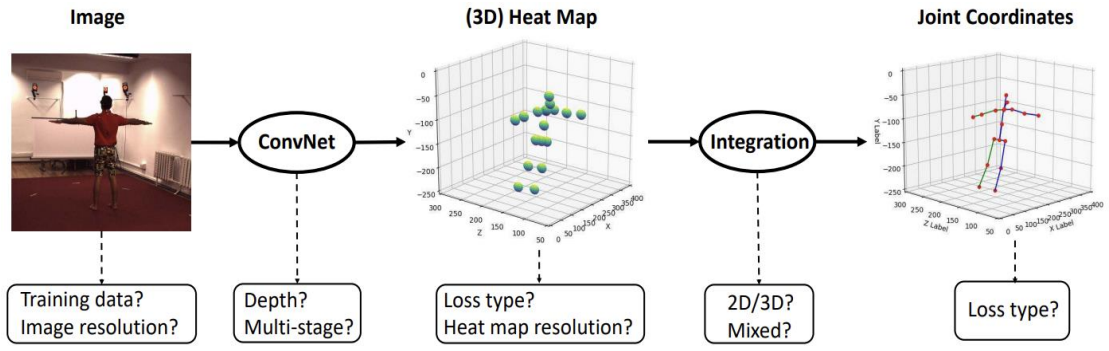


Figure 2-10 Integral human pose regression with 3D heatmaps (Sun et al., 2018).

Benefiting from the high accuracy and generalization capabilities of 2D human pose estimation, many researchers use off-the-shelf 2D human pose estimation networks as an intermediate supervision step, lifting 2D poses to 3D space. This two-step method is generally superior to the directly regressing method because of the excellent

performance of 2D pose detectors.

2.2.2.1 Methods based on Single Frame

Martinez *et al.* (2017b) introduced a classical method wherein 2D keypoint coordinates serve as input, and the model maps the 2D pose directly to 3D space using a fully connected layer with residual connections. Although the model was relatively simple, it achieved state-of-the-art results. The experiments conducted suggested that the errors in many contemporary 3D human pose estimation algorithms primarily stem from challenges in understanding 2D human pose estimation rather than issues with the 2D-to-3D lifting process. Chen and Ramanan (2017) added a K-nearest neighbour search algorithms into the network to search similar 3d pose among 2d pose dataset and then output the correct 3D pose. However, the predictions of this method can be wrong when 3D pose and 2D pose are not conditionally independent. Fang *et al.* (2018) used the grammar information to encode the anatomy relations and dependencies in the network because previous works rarely applied domain-specific knowledge and the generalization ability is poor when performing cross-view pose estimation. Zhou *et al.* (2017) jointly trained a network capable of estimating 3D human pose in the wild, using the heatmap and features of the predicted 2D joints as input for the regression depth. Brau and Jiang (2016) incorporated prior knowledge regarding bone length and projection consistency to perform regression of 3D joint coordinates. Tekin *et al.* (2017) introduced a two-branch network to estimate 2D heatmaps and extract features from images, which are then fused with the 2D heatmaps through a fusion layer. Jahangiri

and Yuille (2017), Sharma *et al.* (2019) and Li and Lee (2019) generated multiple feasible hypotheses of 3D poses from 2D poses and chose the best one with 2D reprojections. Moreno-Noguer (2017) trained a neural network to learn the mapping of the two matrices from encoded pairwise Euclidean distances of 2D and 3D body joints. Euclidean Distance Matrices (EDMs) are invariant to rotations and translations in the plane, as well as scaling invariance when applying normalization operations. Wang *et al.* (2018) introduced a Pairwise Ranking Convolutional Neural Network (PRCNN) to predict the depth information of human joints and ranked the information that is used as a cue to infer coordinates of 3D joints. Yang *et al.* (2018) introduced a multi-source architecture including image, geometric descriptor, joint location information, heatmaps and depth maps.

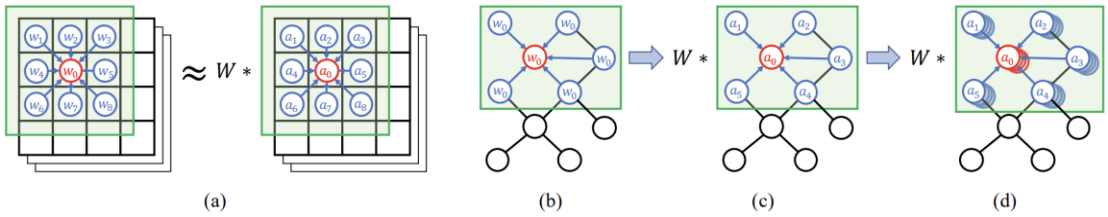


Figure 2-11 Semantic Graph Convolutions (Zhao et al., 2019b).

Graph Neural Network. Since a human pose can be represented as a graph where joints are the nodes and skeletons are the edges, many researchers have used Graph Convolution Networks (GCNs) to estimate 3D poses from 2D poses, achieving promising results. Zhao *et al.* (2019b) introduced Semantic Graph Convolutional Networks in Figure 2-11, a novel neural network architecture designed specifically for regression tasks involving graph-structured data. This network can learn semantic information, including local and global node relationships, that may not be explicitly

represented in the graph. Besides, it also addressed the limitation associated with the GCNs, which is confined to a small receptive field of the convolution filter and the shared transformation matrix for each node. Ci *et al.* (2019a) proposed a generic network called Locally Connected Network (LCN) that is composed of the GCN and the fully connected network to model the relationship of adjacent joints. It can mitigate the issues of GCNs that weight sharing strategy affects the representational power of the network. In addition, the LCN can greatly improve network characterization and generalization, and enable end-to-end deployment and application to different scenarios. Liu *et al.* (2020a) carried out a thorough and systematic investigation into the challenge of weight sharing in GCNs for the purpose of 3D human pose estimation. They concluded that the way of weight sharing in GCNs has a significant impact on the performance of 3D human pose estimation and more parameters do not necessarily lead to better performance. Decoupled self-connection is beneficial for reaching good performance and pre-aggregation is the best weight sharing method in terms of GCN. Zeng *et al.* (2021) aimed to improve the performance on challenging poses characterized by depth ambiguity, self-occlusion, and complexity or rarity. Therefore, they proposed a hop-aware hierarchical channel-squeezing fusion layer to suppresses the learning of noise by the adjacency matrix of graph neural networks (GNN). Also, they build temporal-aware dynamic skeletal graphs to dynamically change the weights of the adjacency matrix based on temporal action changes. However, the above GCNs-based methods all represent the human skeleton as an undirected graph for processing,

ignoring the hierarchical orders among human joints and failing to reflect the articulated characteristic of human skeletons. Therefore, Hu *et al.* (2021) depicted the human skeleton as a directed graph, with joints as nodes and bones as edges directed from parent joints to child joints. Based on this representation, they introduced a U-shaped Conditional Directed Graph Convolutional Network to exploit different non-local dependencies for different poses.

2.2.2.2 Methods based on Video Sequence

Compared with estimating 3D human pose from monocular images, inferring 3D joints from video could exploit temporal information, achieving more stable and jitter-free prediction results. Tekin *et al.* (2016b) inferred 3D poses with the information of histograms of oriented gradients and demonstrated that motion information in the volumes can improve the accuracy of some challenging poses with mirroring and self-occlusion. Hossain and Little (2018b) proposed a sequence-to-sequence architecture which used Long Short-Term Memory units (LSTM) to predict the 3D human pose from given 2D pose sequence. This network encoded a 2D pose sequence into a fixed feature vector. Then, it decoded the 2D pose sequence into a 3D pose sequence using residual connections. However, encoding the 2D pose sequences into a 1D vector ignored the expression of the spatial configuration of 2D poses and this model needed fixed length when inputting temporal data. To solve these problems, Pavllo *et al.* (2019a) proposed the temporal convolutional network (TCN) that leveraged dilated temporal convolutional to extract continuous frame information of human in the video.

Compared with RNN and LSTM, TCN can process multiple frames in parallel and flexibly capture varying sequences. The dilated convolutions were employed to capture long-term dependencies while requiring fewer training parameters and achieving superior computational speed compared to sequence-to-sequence models. Besides, Pavllo *et al.* (2019a) used back-projection to add non-labelled data for semi-supervised training. However, the vector encoding of joint sequences lacks the capacity to adequately express spatial relationships, which is essential for addressing challenges associated with depth ambiguities and self-occlusions. Dabral *et al.* (2018), Cai *et al.* (2019a) and Li *et al.* (2019) made additional use of spatial information on top of the temporal information and added some constraints to the loss function, such as fixed bone length and symmetrical relationship between the left and right of the human body. In addition, there are other works that deform the TCN (Pavllo *et al.*, 2019a) to improve the prediction accuracy. Cheng *et al.* (2019) and Cheng *et al.* (2020) introduced occlusion labels to the temporal convolutional network (TCN) to improve estimation accuracy on some images with occluded human. Cheng *et al.* (2019) proposed an occlusion-aware network with a "cylinder man model" producing occlusion labels, which enabled the network to perform statistics on occlusion labels and thus design regularization penalties. The key stage in this method is that the occlusion model uses incomplete 2D keypoints with ignoring self-occluded points, allowing the network to be less affected by the error-prone estimations of occluded keypoints. However, when the bounding box of the detected human body deviates significantly from the ground

truth, the estimation becomes highly inaccurate. If two or more people are very close, this method may not be able to distinguish the keypoints of different people. The "cylinder man model" cannot produce occlusion from other objects. Cheng *et al.* (2020) developed an end-to-end trainable model that leveraged multi-scale features in space and time to process target human at different distances and different speeds. They used a multi-scale convolutional network (HRNet) proposed by (Sun *et al.*, 2019) which fused these spatial features. Therefore, 3D joint coordinates are predicted with multi-scale features embedding obtained from those heatmaps based on TCN (Pavllo *et al.*, 2019a). Besides, Cheng *et al.* (2020) designed a discriminant model based on spatio-temporal kinematic chains enforcing limbs angular and length constraints for validation of pose sequences. Liu *et al.* (2020c) applied attention mechanism to TCN, which determined key frames and output tensor in every layer. Different from (Pavllo *et al.*, 2019a) who used a voting mechanism to select important frames, Liu *et al.* (2020c) systematically assigned a weight distribution to frames, all of which might contribute to the inference. At the same time, this attention mechanism also modelled long-range dependencies to increase temporal receptive fields. Wang *et al.* (2020c) proposed a loss function called motion loss that used the model to reconstruct the keypoints motion trajectories, considering the similarity of temporal structure between the estimated pose sequence and the ground truth. Meanwhile, they designed a U-shaped GCN based on (Cai *et al.*, 2019a) to combine long-range information through temporal pooling operations. However, the local-to-global network architecture is constrained by its

ability to embed fixed-length spatial-temporal sequences. Chen *et al.* (2021b) decomposed the task of predicting the 3D human pose into two components: predicting bone direction and predicting bone length. By doing so, the 3D joint coordinates can be entirely derived since the bone lengths of a human skeleton remain constant over time. This approach involved predicting the bone directions for the target frame using consecutive local frames and determining bone lengths by considering randomly sampled frames from the entire video. Zeng *et al.* (2020) introduced the split-and-recombine scheme to enhance the generalization of rare and unseen poses. This innovative approach involved segmenting human joints into distinct groups and applying temporal convolution within each group. Subsequently, the joints from different groups were recombined to reconstruct a comprehensive human pose. Similarly, Shan *et al.* (2021b) classified joints into five distinct groups: torso, left and right arms, and left and right legs. They devised a feature fusion module to merge five different features, performing the TCN (Pavllo *et al.*, 2019b) within each group prior to fusion.

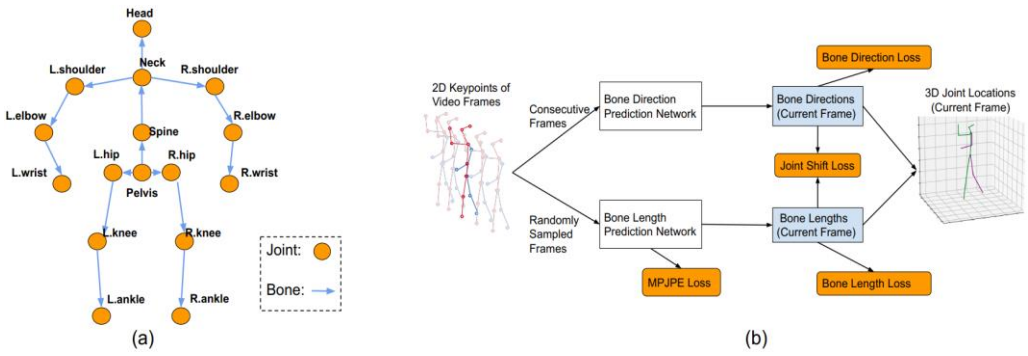


Figure 2-12 Anatomy-aware network for predicting bone directions and bone lengths (Chen et al., 2021c).

2.2.3 Transformer-based Methods

Transformer was first proposed by (Vaswani *et al.*, 2017) and showed remarkable performance in natural language processing (NLP), as the self-attention can model long-range dependencies and also capture global features. Recently, several studies on transformer-based methods for 3D human pose estimation have been reported, with PoseFormer (Zheng *et al.*, 2021a) being the first that predicts the 3D pose of the central frame by modeling spatial and temporal information. However, the computational burden is huge when the frame number increases. PoseformerV2 (Zhao *et al.*, 2023) introduces a time-frequency feature to the transformer structure, efficiently extends the input sequence length, and achieves a good trade-off between speed and accuracy. MHFormer (Li *et al.*, 2022c) a transformer-based network, generates multiple hypotheses at the pose level and calculates the target 3D pose by averaging. MixSTE (Zhang *et al.*, 2022b) stacks spatial and temporal transformer blocks to capture spatial-temporal features alternatively and models the trajectory of joints over frame sequence. STCFormer (Tang *et al.*, 2023b) slices the input joint features into two partitions and uses MHSA to encapsulate the spatial and temporal context in parallel. D3DP (Shan *et al.*, 2023), a diffusion-based method, recovers the noisy 3D poses by assembling joint-by-joint multiple hypotheses. By introducing new encoders for better modeling the spatial and temporal relations, these methods all have unavoidably changed the internal structure or altered the MHSA of the transformer, resulting in largely increased network complexity.

Recently, some studies combined graph and transformer, introducing graph-transformer methods (Gong *et al.*, 2023; Li *et al.*, 2023; Zhao *et al.*, 2022; Zhu *et al.*, 2021). PoseGTAC (Zhu *et al.*, 2021) uses graph atrous convolution to learn the multi-scale information among 1-to-3 top neighbours and utilizes the graph transformer layer to capture long-range features. GraFormer (Zhao *et al.*, 2022) replaces the MLP in the transformer with learnable GCN layers to form the GraAttention block, which also contains MHSA. Li *et al.* (2023) introduces a graph POT, where each element is the relative distance between a pair of joints, which are being encoded as the attention bias in the MHSA module. DiffPose (Gong *et al.*, 2023) interlaces GCN layers with self-attention layers as a diffusion model, which can capture spatial features between joints based on the human skeleton. Nevertheless, these graph-transformer methods (Gong *et al.*, 2023; Li *et al.*, 2023; Zhao *et al.*, 2022; Zhu *et al.*, 2021) learn merely the spatial information of individual pose, without considering temporal correlation across frames. Moreover, they (Gong *et al.*, 2023; Li *et al.*, 2023; Zhao *et al.*, 2022; Zhu *et al.*, 2021) modify the structure of the transformer by introducing the graph convolution, resulting in much larger and more complex networks.

CHAPTER 3. GROUP-BASED 3D POSE ESTIMATION WITH AN EFFICIENT HETEROGENEOUS FUSION

3.1 Introduction

With reference to the overall method as outlined in Figure 1-2 (page 9), a total of three novel network models are developed in this study for 3D pose estimation. Here we explain the first model, a CNN-based model with an efficient heterogeneous fusion.

As reviewed in Section 2.2.2 on page 35, of the various models following the two-step method, those involving temporal information and anatomical grouping strategies are the two which are the most frequently investigated. Some researchers (Cai *et al.*, 2019b; Chen *et al.*, 2021c; Liu *et al.*, 2020d; Pavllo *et al.*, 2019b) exploited temporal information of the input videos to achieve a more accurate and jitter-free result, in which the temporal information of a few adjacent frames is aggregated by means of the network. Other researchers (Park & Kwak, 2018; Shan *et al.*, 2021a; Zeng *et al.*, 2020), however, grouped the various human joints into parts, such as arms, legs and torso, based on human anatomy, so as to improve the prediction accuracy. By integrating features from different groups to enhance the interdependence among different body parts, these group-based methods achieve remarkable estimation performance. Nevertheless, they treat each group of features equally without considering the importance of the torso as the interconnected section of limb groups. Consequently, the torso joints were not given adequate attention, resulting in inaccurate predictions for the limb joints. In addition, these group-based methods impose a high computational

workload when integrating features and require a multi-stage training strategy to avoid interference between the feature fusion module and the encoding module (Shan *et al.*, 2021a). In other words, existing group-based methods cannot be trained end-to-end, and the training is both time consuming and computationally expensive.

Therefore, we propose an efficient heterogeneous group-based method called 'EHFusion' for 3D human pose estimation, as illustrated in Figure 3-1. Inspired by interesting feature fusion work in other domain (Jiang *et al.*, 2022; Nazir *et al.*, 2020; Xie *et al.*, 2021), we design a heterogeneous feature fusion (HFF) module to integrate the relative information of different groups to effectively facilitate kinematic interaction among various body parts. Rather than utilizing identical modules to integrate features from different groups, the HFF module emphasizes the importance of the torso as the core component of the human body and leverages a heterogeneous network structure. By combining convolutional and fully connected operations, the HFF module can reduce model parameters and computational costs while simultaneously improving performance.

Moreover, to further enhance the accuracy of 3D pose estimation, motion amplitude information (MAI) and a camera intrinsic embedding (CIE) module are introduced in EHFusion, as illustrated in Figure 3-1. MAI aims to incorporate a global body motion context without resorting to feature fusion, improving the estimation accuracy of actions with large motion amplitudes (e.g., sitting). CIE mitigates the gap in coordinate system transformation during the process of 2D-to-3D lifting. The main contributions

of this section are summarized as follows:

- The development of a heterogeneous and efficient feature fusion module (HFF) lowers the computational burden for feature fusion while improving both the prediction accuracy and efficiency of the overall 3D pose estimation network.
- MAI has been introduced to enable the network to learn the body motion amplitude information between the target pose and the others. Unlike the relative information in (Shan *et al.*, 2021a), our MAI aims to incorporate a global body motion context without resorting to feature fusion.
- A camera embedding, by means of a Multi-Layer Perceptron (MLP), has been developed to learn the transformation from the image coordinate system (ICS) to the camera coordinate system (CCS), thereby boosting the performance of 2D-to-3D pose estimation.
- A multi-tasking network has been designed enabling end-to-end training. The encoders of different body sections/parts and the fusion module grouping features of different body sections can be trained at the same time without interfering with each other, thus saving significant computational time and training resources.
- Related analysis, involving extensive datasets, have demonstrated both the effectiveness and efficiency of the proposed network in comparison to various existing state-of-the-art methods (i.e., transformer-based).

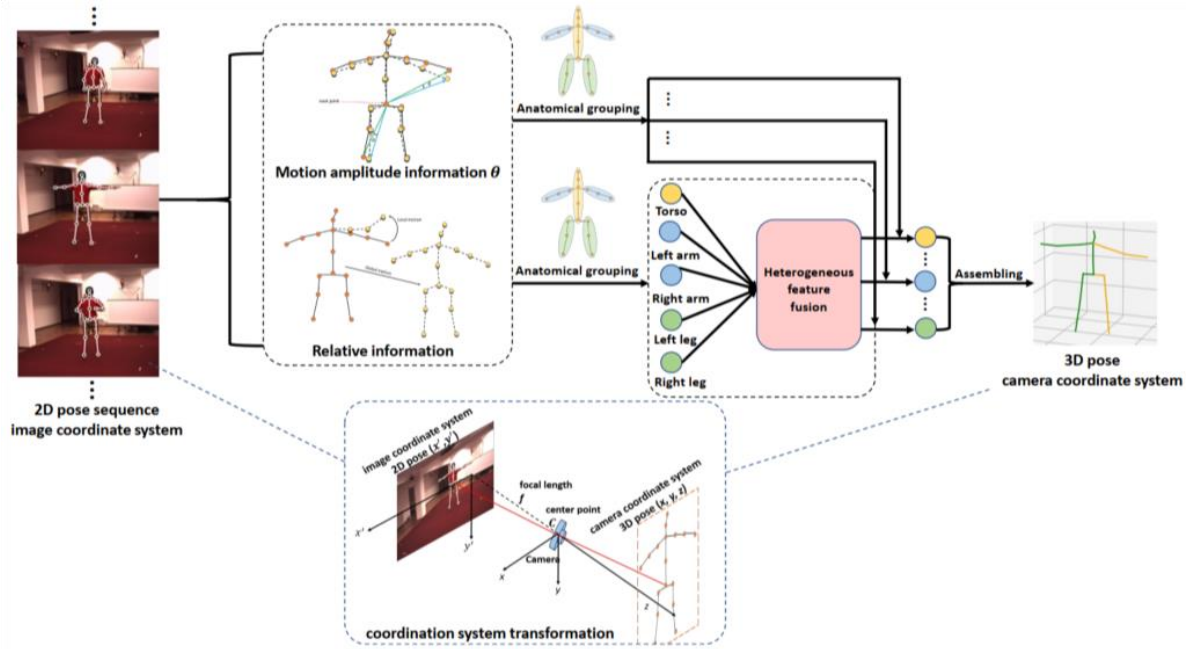


Figure 3-1 Architecture of the proposed EHFusion model.

3.2 Method

3.2.1 Problem Formulation

Let an input 2D pose sequence be denoted as $K = \{k_t^j\}$, $k_t^j = (x_t^j, y_t^j) \in \mathbb{R}^2$, where (x_t^j, y_t^j) denotes a keypoint defined in image coordinate system; j represents the joint index of a human pose and $j = 0, 1, 2, \dots, J$; and $j = 0$ is called the root joint. The total number of joints is set as 17 in current study, i.e., $J = 16$; while t is the frame index and $t = 1, 2, \dots, T$, representing each input 2D pose sequence is compose of T number of frames. In this study, the relative information from (Shan *et al.*, 2021a) is used for 3D pose estimation.

Relative information. The relative information contains positional information and temporal information. First, the positional information refers to the relative joint coordinates, in which the joint coordinates in each pose are calculated as relative

coordinates to the root joint and can be expressed as follows:

$$K_P = \left\{k_t^j - k_t^0\right\}_{j=1}^{J} \quad (3-1)$$

For each joint $j > 0$, the relative coordinates are defined as:

$$\overline{k_t^j} = \left(x_t^j - x_t^0, y_t^j - y_t^0\right) \quad (3-2)$$

The temporal information denotes the pose differences between frames, in which the coordinates of the target frame are subtracted from the pose coordinates of the other frames and can be expressed as:

$$K_T = \left\{k_t - k_{\frac{T}{2}}\right\}_{t=1}^{T} = \left\{(x_t, y_t) - \left(x_{\frac{T}{2}}, y_{\frac{T}{2}}\right)\right\}_{t=1}^{T} \quad (3-3)$$

j is omitted for the sake of simplicity.

Topology-based grouping aims to divide the human body into smaller anatomical parts or groups (e.g., limbs, torso) and estimate the pose of each group independently before integrating them to obtain the pose of the whole body. This method is robust in terms of occlusions since the human body is divided into smaller anatomical groups to learn the unique features (e.g., positional and temporal information) of each group. By focusing on smaller parts, it is less likely that the entire part will be occluded, allowing for more accurate pose estimation, even under highly challenging situations.

For each 2D pose, all 17 body joints are divided into 5 nonoverlapping groups, $K^i = \left\{k_t^j\right\}_{j=1}^{J_i}$, where J_i is the number of joints in group i and $i = 1, \ldots, 5$, corresponding to torso, left arm, right arm, left leg, and right leg, as shown in Figure 3-3(b). The limbs are not inter-connected to each other but treated as four independent parts. Each joint group represents the local commonality of related joints and provides better local

features. With reference to equation (3-1) and (3-2), the positional and temporal information of the i^{th} joint group can be expressed as K_P^i and K_P^i, respectively.

TCN encoding: For each joint group i, by concatenating the input pose K^i, the positional information K_P^i and the temporal information K_T^i together, an enhanced representation K_E^i can be obtained, which is further processed by TCN encoding:

$$F_E^i = E_T^i\left(K_E^i\right) \qquad (3-4)$$

Where $E_T^i(\cdot)$ stands for the TCN encoder (Pavllo *et al.*, 2019b). F_E^i indicates the encoded relative information.

Target pose encoding. The target pose is defined as the centre frame of the 2D pose sequence, $K_G = \left\{k_{T\over 2}^i\right\}_{j=1}^{J}$, which is encoded by an independent MLP network to give global feature:

$$F_G = E_m(K_G) \qquad (3-5)$$

where $E_m(\cdot)$ stands for the MLP encoder. Instead, we incorporate the encoded information F_G into the encoded relative information.

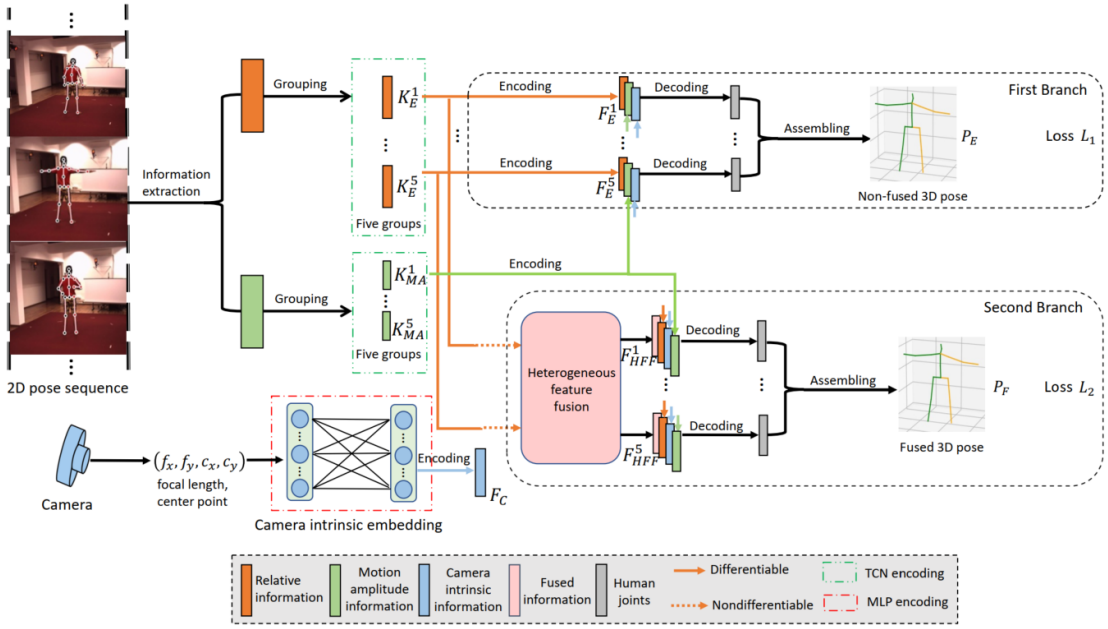


Figure 3-2 Our multi-task end-to-end EHFusion network.

3.2.2 *Heterogeneous Feature Fusion (HFF)*

By topology-based grouping, the spatial relationships of the connecting joints have been preserved within each joint group, representing strong local features, while the connection between groups is not included, making the joint positions of other groups unknown to the current group. In order to obtain a complete 3D pose prediction, it is important to fuse features of different joint groups together. Shan *et al.* (2021a) used a feature fusion module, based on fully connected layers (FCN), to fuse grouped features. Nevertheless, for the prediction of each group, all the grouped features are fused with the uniform FCN feature fusion block, which ignores the different relationships between groups. Additionally, the use of FCNs also results in over 90% of the network parameters, generating a large volume of redundant information in the network design (Cheng *et al.*, 2015). A new module called heterogeneous feature fusion (HFF) module is proposed here to address the above issues.

The proposed HFF module, as illustrated in Figure 3-4, combines one FCN feature fusion block with four conv feature fusion blocks, where the former block is employed for the prediction of torso joints while the latter block is employed for that of other groups. With this heterogeneous fusion module, the FCNs can capture comprehensive relationships and patterns within the limb group features to assist in the prediction of torso joints, while the convolutional layers utilize localized features for predicting joints within groups. This is not only beneficial to make the network focus more on the torso joint prediction which has close relationship with all other group joints and plays a more important role in the prediction stability of the model, but also helps to alleviate the over-fitting problem and reduce model parameters and computation costs.

More specifically, for a specific group i, we concatenate grouped features of the other four groups together according to the channel dimension, $\left[F_E^{i+1}, \dots, F_E^{i+4}\right]$, as input to be further processed in the FCN or Conv feature fusion block. The FCN feature fusion block consists of the fully connected layer, 1D batch normalization (BN), rectified linear unit (ReLU) and dropout, raising the feature dimensions to obtain the fused features, see the top right corner of Figure 3-4. In the design of conv feature fusion block, we used the discriminative dimensional reduction method (Su *et al.*, 2017) to find a lower-dimensional representation of the feature that maximizes the separability between different classes, namely the other four body parts. By so doing, we can preserve the essential features while improve computational efficiency and reduce the risk of over-fitting. Specifically, each grouped feature is processed by a 1D convolution

with 1 stride and 1 kernel size, followed by batch normalization (BN) and rectified linear unit (ReLU) for a discriminative feature:

$$F^{i'} = ReLU\left(BN\left(Conv1D\left(F_E^i\right)\right)\right) \qquad (3-6)$$

The final fused feature of conv block is obtained by concatenating the four resulting features $\left[F^{(i+1)'}, F^{(i+2)'}, F^{(i+3)'}, F^{(i+4)'}\right]$.

In the proposed HFF module, the FCN feature fusion block is used for the torso by fusing the other four grouped features of limbs (left/right arm and left/right leg). In contrast, for each of the four limbs, the conv feature fusion block is used to fuse the grouped features of torso and other limb parts. There is weight sharing in the convolution operation, and this enable learning the common features of the connected parts. Compared with the feature fusion module of (Shan *et al.*, 2021a), the proposed HFF module not only avoids over-fitting and reduces the number of parameters, but also improves the performance (see experimental comparison).

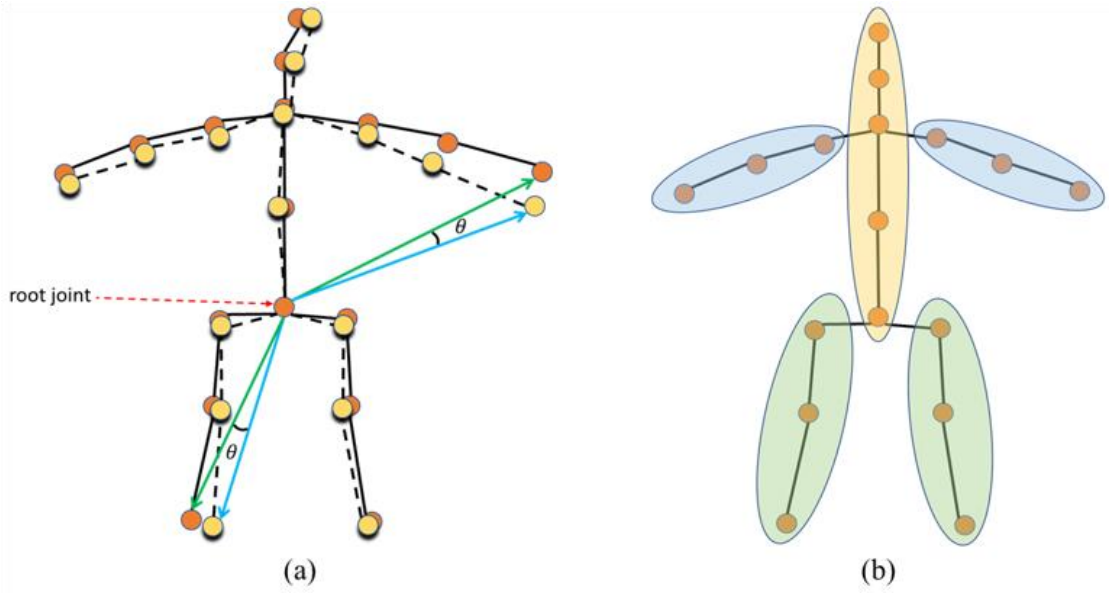


Figure 3-3 Illustrations of (a) motion amplitude θ and (b) the group configuration.

3.2.3 *Motion Amplitude Information (MAI)*

In this study, **motion amplitude information (MAI)** is introduced to enhance the description of body motion, which can be viewed as another kind of global information. As shown in Figure 3-3(a), the orange dots connected with the black solid lines indicate the target pose, while the yellow dots and the dashed lines indicate another pose at time frame t. These two poses share the same root joint. The motion amplitude is defined as the angle θ between the green vector and the blue vector, and is derived as follows:

$$\phi\left(\overline{k_G^J}, \overline{k_t^J}\right) = exp[\theta - \pi] \tag{3-7}$$

Where $\overline{k_G^J}$ denotes the target pose and $\overline{k_t^J}$ denotes the other pose at frame t. The angle θ is calculated as follows:

$$\theta = cos^{-1} \frac{\left(\overline{k_G^J} * \overline{k_t^J}\right)}{\left\|\overline{k_G^J}\right\| \left\|\overline{k_t^J}\right\|} \tag{3-8}$$

This encoding method normalizes the joint motion amplitude θ between 0 to 1 to avoid the gradient explosion. Finally, the motion amplitude (MA) is calculated as:

$$K_{MA} = \phi(\overline{K}, \overline{K_G}) \tag{3-9}$$

Where $\overline{K_G}$ is the target pose sequence and $\overline{K}$ is the other pose sequence. We divide the pose motion amplitude K_{MA} into the same five joint groups (torso, left arm, right arm, left leg, right leg) by anatomical grouping, and each grouped motion amplitude is denoted as K_{MA}^i and processed by TCN encoding:

$$F_{MA}^i = E_T^i\left(K_{MA}^i\right) \tag{3-10}$$

where E_T^i stands for the TCN encoder (Pavllo *et al.*, 2019b).

3.2.4 *Camera Intrinsic Embedding (CIE)*

The process of lifting 2D pose to 3D pose implies the transformation of coordinate systems from the image coordinate system (ICS) to the camera coordinate system (CCS). Specifically, $p_i = [u_i, v_i]$ represents the i-th joint coordinates in the ICS of a 2D pose. $P_i = [X_i, Y_i, Z_i]$ stands for the corresponding joint coordinates in the CCS of the 3D pose. If the depth Z in the CCS is known, the 2D pose coordinates in the ICS can be converted to 3D pose in the CCS as follows:

$$X_i = Z_i \frac{u_i - c_x}{f_x} \tag{3-11}$$

$$Y_i = Z_i \frac{v_i - c_y}{f_y} \tag{3-12}$$

Where f_x and f_y denote the camera focal length; c_x and c_y represent the coordinates of the camera centre point.

According to equation (3-11) and (3-12), the prediction of 3D pose involves the prediction of X and Y coordinates and the depth Z, while the former can be derived from Z using camera intrinsic parameters (focal length and camera centre point). The focal length determines the scale factor between the 2D image plane and the 3D space, which help to estimate the relative depth. The camera center point is where the optical axis of the camera intersects with the image plane, providing an offset of the coordinate system origin that maps 2D image coordinates onto 3D coordinates.

Thus, we propose a **camera intrinsic embedding (CIE)** network to exploit the focal length (f_x, f_y) and the camera centre point (c_x, c_y) as a priori information for more accurate predictions of 3D pose in CCS. More specifically, the focal length (f_x, f_y) and the camera centre point (c_x, c_y) are first concatenated together to form a tensor

(f_x, f_y, c_x, c_y), which is then fed into the CIE network to obtain a high-dimensional camera intrinsic information F_C:

$$F_C = E_m(f_x, f_y, c_x, c_y) \qquad (3-13)$$

where $E_m(\cdot)$ stands for the MLP encoder. The CIE network consists of two fully connected layers, 1D batch normalization, rectified linear unit and dropout.

3.2.5 Model Optimization

A multitasking end-to-end network has been developed to incorporate four kinds of information including positional, temporal, motion amplitude and camera intrinsic information for 3D pose estimation. The network framework has two branches and can be trained end-to-end in one single stage or by multiple stages. In contrast to the RIE network (Shan *et al.*, 2021a), which can only be trained in multiple stages, our network executes rapidly with fewer parameters and lower computational cost. As illustrated in Figure 3-4, the input to the network is 2D pose sequences; either ground-truth sequence or poses predicted by 2D pose detectors can be used as inputs. To predict 3D poses, the input 2D poses are transformed into positional and temporal information, which are concatenated to theoriginal input 2D pose. The concatenated information is then divided into five groups (torso, left arm, right arm, left leg, right leg) and sent for TCN encoding. Furthermore, the motion amplitude is encoded independently by means of TCN encoder, while the target pose and the camera intrinsic parameters are encoded by two different MLP networks. All the encoded information is concatenated together and fed into the decoder of the first branch for decoding. The second branch includes an

additional HFF module for fused features before decoding, and the gradients are not back propagated during training in the HFF module. By doing so, the training of the fused information and the other encoded information do not interfere with each other, leading to the final fused 3D pose.

Loss functions. Our multitasking network uses two loss functions to govern the learning of the two branches. The first branch, which the HFF module is not involved, only learns the parameters of the encoders and decoders, generating a non-fused 3D pose. Hence, the first loss function L_1 only constrains the non-fused 3D pose and facilitates the parameter learning of the encoders and decoders. The second branch, with the HFF module incorporated, aims to yield a fused 3D pose which is optimized by the second loss function L_2. Since the second branch network directly uses the encoders trained by the first branch network, the loss L_2 drops along with the dropping of loss L_1 in the training. After training, the prediction of the second branch is chosen as the final result.

In the first branch, we concatenate four pieces of features per group including enhanced representation F_E^i, global feature of target pose F_G, motion amplitude F_{MA}^i, and the camera intrinsic embedding F_C and input them into the decoder as follows:

$$F_D^i = D\big(Concat\big[F_E^i, F_{MA}^i, F_g, F_C\big]\big) \qquad (3-14)$$

Next, the five groups of decoded features are concatenated together as a non-fused 3D pose P_E:

$$P_E = Concat[F_D^1, F_D^2, \dots, F_D^5] \qquad (3-15)$$

The non-fused pose P_E and the ground-truth 3D pose P_G are compared to calculate the loss function as follows:

$$L_1 = \|P_E - P_G\|_2^2 \quad (3-16)$$

In the second branch, the fused features from the HFF module F_{OFF}^i is concatenated with the four other features F_E^i, F_{MA}^i, F_g, F_C and input to the decoder as follows:

$$F_{D\_fuse}^i = D\left(\left[F_E^i, F_{MA}^i, F_g, F_C, F_{OFF}^i\right]\right) \quad (3-17)$$

Similarly, the decoded features $F_{D\_fuse}^i$ are concatenated together to give a fused 3D pose:

$$P_F = Concat\left[F_{D\_fuse}^1, F_{D\_fuse}^2, \dots, F_{D\_fuse}^5\right] \quad (3-18)$$

The loss function of the second branch is calculated as follows:

$$L_2 = \|P_F - P_G\|_2^2 \quad (3-19)$$

Finally, the overall loss function of our network is obtained as follows:

$$L = L_1 + L_2 \quad (3-20)$$

One-stage optimization: In our multitask network (Figure 3-2), the first branch learns the parameters for information encoding (relative, motion amplitude, and camera intrinsic embedding), while the second branch shares the encoded information with the first branch and additionally introduces the HFF module to fuse the encoded positional and temporal information. The HFF module in the second branch is designed without gradient back propagation, making it non-differentiable. By doing so, the encoded information (positional and temporal) can be trained independently without interfering with the feature fusion between the different groups. During training, the parameters of

the information encoding module are gradually optimized in the first branch. The encoded information (positional and temporal) is fed into the HFF module in the second branch for feature fusion. In this way, the information encoding and feature fusion in both branches are optimized simultaneously without interfering with each other. The accuracy of the predicted 3D pose in the second branch is higher than that of the first branch because of feature fusion. Hence, the 3D pose from the second branch is selected as the final result.

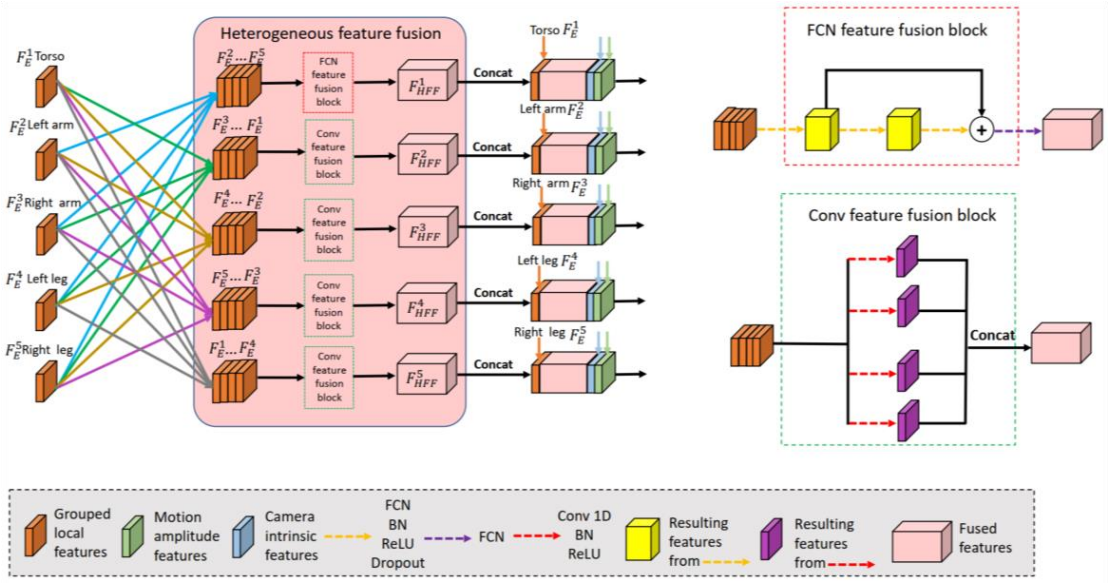


Figure 3-4 Illustration of the proposed heterogeneous feature fusion (HFF) module. FCN –Fully Connected Layer; BN – 1D Batch Normalization; Conv 1D – 1D convolution..

Three-stage training: For comparative purpose, we also investigated the three-stage training strategy, as illustrated in Figure 3-5. In stage 1 of the three-stage training, the encoding modules for the four types of information, including enhanced representation F_E^i, motion amplitude F_{MA}^i, target pose F_G and camera intrinsic embedding F_C are trained. The enhanced representation F_E^i per group is first encoded by TCN, while the

motion amplitude F_{MA}^{i} is independently encoded per group by another encoding module of TCN. The target pose F_G and the camera intrinsic embedding F_C are separately encoded with two MLP networks. After the training of stage 1 is finished, the parameters of the decoders are discarded and only the parameters for encoder are fixed in stage 2 training. In stage 2, the HFF module and decoders are trained. The parameters of the encoders and the encoded features of the positional and temporal information are first input into the HFF module for feature fusion. The fused features are then concatenated with the other encoded features, namely the motion amplitude, target pose and camera intrinsic embedding and sent to the decoders. In stage 3, the entire network including encoders, HFF module and decoders are fine-tuned simultaneously. The three-stage training strategy ensures that encoders and the HFF module are trained without interfering with each other, but this strategy requires plenty of training time and computational costs.

Our multitask network can be trained end-to-end, saving considerable computational time and costs, while the RIE (Shan *et al.*, 2021a) can only be trained in three stages. A comparative study will be given later.

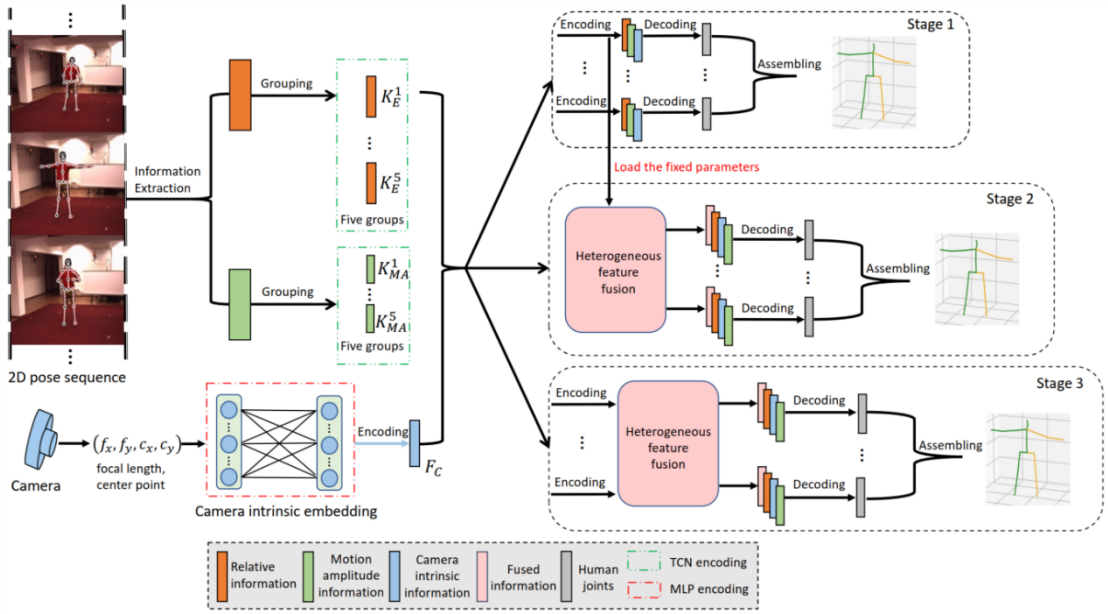


Figure 3-5 Three-stage training network.

3.3 Experimental Results and Discussion

3.3.1 Datasets and Evaluation Protocol

Datasets: We evaluated our model on two public datasets Human3.6M (Ionescu *et al.*, 2013) and HumanEva-I (Sigal *et al.*, 2010). **Human3.6M** is an indoor scene dataset collected by motion capture systems with a total of 3.6 million video frames. It includes 7 professional actors wearing markers which record the coordinates of each body joint. These actors perform 15 typical daily actions, such as walking dogs, taking photos, sitting, greeting, eating, and so forth, captured in 4 synchronized camera angles. In line with the previous research (Liu *et al.*, 2020d; Shan *et al.*, 2021a; Zeng *et al.*, 2020), we used five actors (S1, S5, S6, S7, S8) for training and two (S9 and S11) for testing. **HumanEva-I** is a smaller dataset, covering only three subjects performing six actions, captured in a controlled indoor environment by three cameras.

Protocols: Protocol#1 is denoted as Mean Per Joint Position Error (MPJPE), which is

the average Euclidean distance, in millimeters, between the predicted joint coordinates J_i and the ground-truth joint coordinates J_i^*, and is expressed by:

$$MPJPE = \frac{1}{N}\sum_{i=1}^{N}\|J_i - J_i^*\|_2 \qquad (3-21)$$

Protocol#2, denoted by P-MPJPE, refers to the error after the predicted pose being aligned with the ground truth via rigid transformation of translation, rotation and scale using Procrustes analysis. Compared to Protocol#1, Protocal#2 is more robust to individual joint prediction failure, thus is also referred as post-processing protocol and is expressed by:

$$P - MPJPE = \frac{1}{N}\sum_{i=1}^{N}\|J_i' - J_i^*\|_2 \qquad (3-22)$$

Where J_i' represents the predicted joint coordinates after they are aligned to the ground truth joint coordinates J_i^* by means of Procrustes analysis.

3.3.2 Ablation Studies

To verify the effectiveness of our new model, ablation experiments were conducted by training the network model on the Human3.6M dataset based on ground-truth 2D poses as inputs. Table 3-1 gives the evaluation results by means of Protocol#1. Our baseline network, without MAI or HFF modules but with fully connect feature fusion module of (Shan *et al.*, 2021a), trained in one stage with gradient back-propagation removed, has the prediction error of 32.3mm in MPJPE. After the introduction of the MAI module, the prediction error drops by 1.2mm without increasing the number of floating-point operations per second (FLOPs) and model parameters significantly. By only replacing

fusion module of the baseline with the new HFF module, MPJPE drops by 1.4mm while reducing substantially the number of FLOPs by 13.64M (49.20M→35.56M) and the model parameters by 13.40M (54.98M→41.58M). This demonstrates the effectiveness of our proposed HFF module for efficient feature fusion. With the simultaneous introduction of the MAI and HFF modules, the Protocol#1 (MPJPE) result is reduced by 2.3mm. The introduction of the CIE module, the performance is further reduced by 0.4mm, reaching 29.6mm in MPJPE, while the number of FLOPs and model parameters are significantly lower than that of the baseline. It demonstrates that the combination of the MAI, HFF and CIE modules is very effective in reducing prediction errors and the number of FLOPs and parameters.

Table 3-1 Ablation study results based on human3.6m dataset. GT-ground-truth 2D poses.

Method (GT)	MPJPE (mm)	FLOPs (M)	Parameters (M)
Baseline	32.3	49.20	54.98
+MAI	31.1	49.52	56.66
+HFF	30.9	35.56	41.58
+HFF+MAI	30.0	36.21	47.59
+HFF+MAI+CIE	**29.6**	36.87	48.25

3.3.3 Comparison with State-of-the-art Methods

Results on Human3.6M dataset. *We compared our results with recent state-of-the-art (SOTA) methods using the public dataset Human3.6M. First, we used 2D poses detected by means of CPN (Chen et al., 2018) as inputs and trained under the receptive field of T =243 frames using one-stage training strategy. In addition to the one-stage training strategy, we also investigated a three-stage strategy to train the model. At stage 1, the encoders, MAI and CIE modules were trained without the HFF module.*

At stage 2, the parameters of the MAI and CIE modules were loaded, and the HFF module was then trained independently. At stage 3, the whole network was fine-tuned, see Figure 3-5. As shown in Table 3-3, our method obtained 44.1mm and 43.8mm in MPJPE under one-stage and three-stage training strategies, respectively, surpassing the recent methods (Tang et al., 2023a; Yu et al., 2023). For a fair comparison, we also employed the refining module in (Cai et al., 2019b) to refine the initial estimated 3D poses, following (Li et al., 2022b; Shan et al., 2022). Our refined results (three-stage) slightly underperform than (Shan et al., 2022) by 0.3mm, becoming the runner-up result in all methods. However, it is noteworthy that our approach demonstrates significantly lower FLOPs (as shown in

Table 3-2) in comparison to (Shan *et al.*, 2022). This has demonstrated that our model

is efficient and can yield robust predictions of 3D poses.

Table 3-5 compares the results obtained under Protocol#2 with those of existing SOTA methods. Our method obtained 35.2mm and 34.8mm in P-MPJPE with one-stage and three-stage training strategies, respectively, with the P-MPJPE of 34.8mm using three-stage training strategy outperforming the refined results (Li *et al.*, 2022b). After using the refining module in (Cai *et al.*, 2019b), our P-MPJPE of 34.5mm under one-stage training strategy achieves the runner-up result of all the methods.

Table 3-5 compares our results with those of SOTA models using ground-truth 2D poses as inputs on Human3.6M dataset. Our method obtained a superior or comparable result of 29.6mm in MPJPE when using the one-stage training strategy. We also trained our model using the three-stage training strategy (Shan *et al.*, 2021a) as illustrated in Figure 3-5, and also obtained 29.6mm under MPJPE. It is worth noting that our model achieves the same result (29.6mm) using both training strategies. This indicates that our one-stage network succeeds in online end-to-end training without any loss in performance, while the RIE (Shan *et al.*, 2021a) can only be trained offline stage-by-stage. After using the refining module in (Cai *et al.*, 2019b), our results obtained from the one-stage and three-stage training strategies show improvements of 1.3mm and 2.0mm, respectively, compared to (Li *et al.*, 2022b), which also utilizes the refining module (Cai *et al.*, 2019b).

Table 3-2. Comparison of computational complexity and MPJPE with 2D ground truth poses as inputs on Human3.6M. The lowest prediction error is in bold. † indicates the transformer-based methods. ∗ uses the refining module propose in (Cai et al., 2019b).

Method (GT)	Parameters ↓	FLOPs ↓	MPJPE (mm) ↓
Shan *et al.* (2021a) (stage 1)	23.39M	17M	33.0
Shan *et al.* (2021a) (stage 2)	41.78M	36M	30.9
Shan *et al.* (2021a) (stage 3)	41.78M	36M	30.3
Zheng *et al.* (2021a)†	9.60M	815M	31.3
Li *et al.* (2022c)†	24.76M	4826M	30.9
Shan *et al.* (2022)†	6.70M	1737M	29.3
Li *et al.* (2022b)†∗	**4.34M**	2193M	28.5
Tang *et al.* (2023a)	4.49M	1037M	29.2
Yu *et al.* (2023)	37.81M	43821M	28.5
Ours (one-stage)	48.25M	**36M**	29.6
Ours (one-stage)∗	48.39M	152M	27.2
Ours (three-stage) (stage 1)	29.40M	18M	31.4

Ours (three-stage) (stage 2)	34.39M	23M	29.7
Ours (three-stage) (stage 3)	34.39M	23M	29.6
Ours (three-stage) (stage 3)∗	34.53M	138M	**26.5**

Table 3-3 Results of MPJPE (mm) on Human3.6m Dataset using Protocol#1 with 2D poses detected by CPN (Chen et al., 2018) as inputs. The lowest prediction error is in bold. † indicates the transformer-based methods. ∗ uses the refining module propose in (Cai et al., 2019b).

MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Wang *et al.* (2020b)	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
Liu *et al.* (2020d)	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Zeng *et al.* (2020)	46.6	47.1	43.9	41.6	45.8	49.6	46.5	40.0	53.4	61.1	46.1	42.6	43.1	31.5	32.6	44.8
Shan *et al.* (2021a)	40.8	44.5	41.4	42.7	46.3	55.6	41.8	41.9	53.7	60.8	45.0	41.5	44.8	30.8	31.9	44.3
Zheng *et al.* (2021a)†	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Chen *et al.* (2021c)	41.4	43.5	40.1	42.9	46.6	51.9	41.7	42.3	53.9	60.2	45.4	41.7	46.0	31.5	32.7	44.1
Li *et al.* (2022b)†∗	40.3	43.3	40.2	42.3	45.6	52.3	41.8	40.5	55.9	60.6	44.2	43.0	44.2	30.0	30.2	43.7
Li *et al.* (2022c)†	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	**43.7**	41.1	43.8	29.8	30.6	43.0
Shan *et al.* (2022)†∗	38.4	**42.1**	39.8	40.2	45.2	**48.9**	**40.4**	**38.3**	53.8	**57.3**	43.9	41.6	42.2	**29.3**	**29.3**	**42.1**
Tang *et al.* (2023a)	41.3	44.7	42.2	42.9	47.9	55.2	43.3	40.9	58.0	66.4	46.2	44.2	45.2	30.7	31.5	45.4
Yu *et al.* (2023)	41.3	44.3	40.8	41.8	45.9	54.1	42.1	41.5	57.8	62.9	45.0	42.8	45.9	29.4	29.9	44.4
Ours (one-stage)	40.0	44.2	40.8	42.2	45.8	55.9	42.1	40.7	55.1	60.3	45.4	42.2	44.1	31.0	31.4	44.1
Ours (three-stage)	39.9	44.0	40.9	41.8	46.0	55.4	41.4	40.8	53.8	60.6	44.8	41.3	44.7	30.1	30.8	43.8
Ours (one-stage)∗	38.6	43.5	39.7	40.7	**44.3**	53.6	41.1	39.7	**52.5**	57.6	43.9	41.2	42.3	29.8	30.1	42.6
Ours (three-stage)∗	**37.7**	42.6	**39.0**	40.0	44.6	53.1	41.1	39.0	53.4	59.6	43.8	**40.7**	**42.0**	**29.3**	29.8	42.4

Table 3-4 Results of P-MPJPE (mm) on Human3.6m Dataset using Protocol#2 with 2D poses detected by CPN (Chen et al., 2018) as inputs. The lowest prediction error is in bold. † indicates the transformer-based methods. ∗ uses the refining module propose in (Cai et al., 2019b).

P-MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Liu *et al.* (2020d)	32.3	35.2	33.3	35.8	35.9	41.5	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	35.6
Wang *et al.* (2020b)	32.9	35.2	35.6	34.4	36.4	42.7	**31.2**	32.5	45.6	50.2	37.3	32.8	36.3	26.0	**23.9**	35.5
Zheng *et al.* (2021a)†	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Shan *et al.* (2021a)	32.5	36.2	33.2	35.3	35.6	42.1	32.6	31.9	**42.6**	**47.9**	36.6	32.1	34.8	24.2	25.8	35.0
Chen *et al.* (2021c)	32.6	35.1	32.8	35.4	36.3	40.4	32.4	32.3	42.7	49.0	36.8	32.4	36.0	24.9	26.5	35.0

Li *et al.* (2022b)†∗	32.7	35.5	32.5	35.4	35.9	41.6	33.0	31.9	45.1	50.1	36.3	33.5	35.1	23.9	25.0	35.2
Li *et al.* (2022c)†	**31.5**	**34.9**	32.8	**33.6**	35.3	**39.6**	32.0	32.2	43.5	48.7	36.4	32.6	34.3	23.9	25.1	**34.4**
Tang *et al.* (2023a)	31.6	35.5	34.4	34.9	36.6	42.5	33.1	**30.9**	46.5	52.2	37.0	33.4	35.3	24.4	24.9	35.6
Yu *et al.* (2023)	32.4	35.3	32.6	34.2	**35.0**	42.1	32.1	31.9	45.5	49.5	**36.1**	32.4	35.6	**23.5**	24.7	34.8
Ours (one-stage)	31.9	36.0	33.3	34.5	36.0	42.5	32.7	31.5	44.4	49.2	37.3	32.5	35.1	24.5	25.4	35.2
Ours (three-stage)	31.9	35.7	32.9	34.9	35.6	42.3	32.6	31.5	42.9	48.3	36.6	32.1	35.0	23.9	25.4	34.8
Ours (one-stage)∗	32.0	35.5	**32.3**	33.9	35.1	41.8	32.6	31.0	42.7	48.0	36.7	**32.0**	34.3	24.2	25.3	34.5
Ours (three-stage)∗	32.1	35.2	32.6	34.0	35.1	41.5	32.6	31.2	42.9	48.9	36.6	**32.0**	**34.0**	24.1	25.3	34.6

Table 3-5 Results on Human3.6M under Protocol#1 with MPJPE (mm). The ground truth of 2D poses is used as inputs. The lowest prediction error is in bold. † indicates the transformer-based methods. ∗ uses the refining module propose in (Cai et al., 2019b).

MPJPE (GT)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Liu *et al.* (2020d)	34.5	37.1	33.6	34.2	32.9	37.1	39.6	35.8	40.7	41.4	33.0	33.8	33.0	26.6	26.9	34.7
Zeng *et al.* (2020)	34.8	32.1	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
Zheng *et al.* (2021a)†	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
Shan *et al.* (2021a)	29.5	30.8	28.8	29.1	30.7	35.2	31.7	27.8	34.5	36.0	30.3	29.4	28.9	24.1	24.7	30.1
Li *et al.* (2022c)†	27.7	32.1	29.1	28.9	30.0	33.9	33.0	31.2	37.0	39.3	30.0	31.0	29.4	22.2	23.0	30.5
Shan *et al.* (2022)†	28.5	30.1	28.6	27.9	29.8	33.2	31.3	27.8	36.0	37.4	29.7	29.5	28.1	21.0	21.0	29.3
Li *et al.* (2022b)†∗	27.1	29.4	26.5	27.1	28.6	33.0	30.7	26.8	38.2	34.7	29.1	29.8	26.8	19.1	**19.8**	28.5
Tang *et al.* (2023a)	29.3	31.0	25.3	27.4	31.3	35.0	30.5	27.1	33.9	38.1	29.2	28.1	28.6	20.9	22.1	29.2
Yu *et al.* (2023)	26.5	27.2	29.2	25.4	**28.2**	**31.7**	29.5	26.9	37.8	39.9	29.9	27.0	27.3	20.5	20.8	28.5
Ours (one-stage)	27.7	29.4	27.6	27.4	30.9	37.2	31.1	27.7	36.8	36.3	30.6	29.0	28.0	20.9	22.0	29.6
Ours (three-stage)	28.3	28.4	27.6	28.5	31.3	35.2	31.1	27.5	35.4	36.5	30.6	28.9	27.8	22.8	23.6	29.6
Ours (one-stage)∗	25.8	27.7	**24.6**	25.7	28.6	35.0	28.4	24.3	33.5	34.3	27.8	27.2	25.5	**19.0**	**19.8**	27.2
Ours (three-stage)∗	**25.0**	**25.8**	25.3	**25.3**	28.6	33.6	**28.3**	**23.9**	**32.3**	**31.7**	**27.3**	**25.5**	**24.2**	20.0	20.5	**26.5**

As shown in

Table 3-2, we compared the cost-effectiveness between our method and other methods. Our MPJPE result (both one-stage and three-stage) is 0.7mm lower than that of RIE (Shan *et al.*, 2021a). Moreover, we only needed to train 80 epochs for one stage, while the RIE (Shan *et al.*, 2021a) needed three stages of training with a total of 240 epochs. Our method saves a great deal of training time and enables end-to-end training. Furthermore, for comparison purpose, we also trained our network with three-stage training strategy (Shan *et al.*, 2021a). At the first stage, the MPJPE result of our method is 31.4mm, which is 1.6mm lower than that of RIE (Shan *et al.*, 2021a), while the number of parameters and FLOPs in our model increases by 6.01M and 0.65M,

respectively, compared to those of RIE (Shan *et al.*, 2021a) in stage 1. At the second stage, our model obtained 29.7mm in MPJPE, which is 1.2mm and 0.6mm lower than those of RIE (Shan *et al.*, 2021a) in stages 2 and 3, respectively. At the same time, the number of parameters and FLOPs in our method in stage 2 is 7.39M and 12.98M, respectively, lower than the corresponding ones of RIE (Shan *et al.*, 2021a). At stage 3, the MPJPE result in our model is 0.7mm lower than that of RIE (Shan *et al.*, 2021a), and the number of parameters and FLOPs in our model are also much lower than those of RIE (Shan *et al.*, 2021a). Furthermore, our method exhibits a significantly lower number of FLOPs (36M) in comparison to other SOTA methods (Li *et al.*, 2022c; Shan *et al.*, 2022; Tang *et al.*, 2023a; Zheng *et al.*, 2021a) (i.e., transformer-based), while simultaneously maintaining a competitive estimation accuracy (29.6mm). After using the refining module (Cai *et al.*, 2019b), our method achieves the best results (27.2mm and 26.5mm) in both the one-stage and three-stage training settings, compared with other SOTA methods including the refined result of (Li *et al.*, 2022b). More importantly, our method demonstrates substantially lower FLOPs compared to other methods, even only 0.03% of that of (Yu *et al.*, 2023). These results demonstrates the superiority of our model in terms of computational complexity, as well as its lightweight design with promising estimation performance.

Table 3-6 Results based on HumanEva-I dataset using Protocol#1 of MPJPE (mm).

Protocol #1	Walk			Jog			Avg
	S1	S2	S3	S1	S2	S3	
Shan *et al.* (2021a)	17.9	11.9	38.0	27.6	18.1	19.2	22.1
Zheng *et al.* (2021a)	**16.3**	**11.0**	47.1	**25.0**	**15.2**	**15.1**	21.6
Zhang *et al.* (2022b)	20.3	22.4	**34.8**	27.3	32.1	34.3	28.5
Ours (T=27) (one-stage)	19.0	13.1	38.4	28.7	18.2	20.4	22.9
Ours (T=27) (three-stage)	17.2	11.9	36.8	26.8	17.0	18.5	**21.3**

In Figure 3-6, we compared the MPJPE performance of our method (one-stage) with that of RIE (Shan *et al.*, 2021a) (stage 3) based on Human3.6m, using ground-truth 2D poses as inputs. Both methods used 80 epochs to train, but our method requires only one stage of training and achieved the best results (29.6mm) at the 37th epoch. In contrast to this, RIE (Shan *et al.*, 2021a) required three stages of training with 80 epochs each, and the best results (30.3mm) were achieved at the 63rd epoch of stage 3. Compared to the RIE (Shan *et al.*, 2021a), our method converged quicker, achieved the best result within one stage of training. In other words, our method required fewer than 80 epochs to train because the two losses are optimized simultaneously. The second loss (fused 3D pose) does not need to wait for the convergence of the first one (non-fused 3D pose). We used one stage and fewer epochs to train and end up with a better result than did the RIE (Shan *et al.*, 2021a).

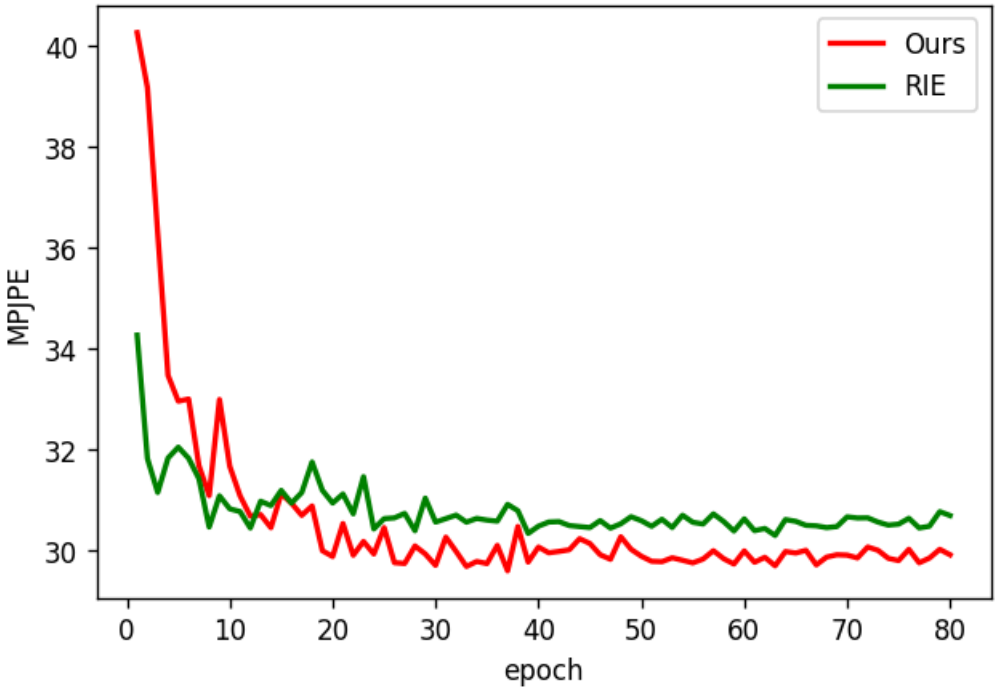


Figure 3-6 Comparison of MPJPE performance of our method and that of RIE (Shan et al., 2021a).

Results on HumanEva-I dataset. We evaluated our model in terms of Protocol#1 on the HumanEva-I dataset in

Table 3-7. For comparative purpose, the input 2D poses were detected using Mask R-CNN (He *et al.*, 2017), being in line with other SOTA results. In addition, we trained our model with a receptive field of T=27 frames. As shown, our method achieves 21.3mm in MPJPE under three-stage training, which is the best result compared to other methods. These results highlight the superior performance of our method on small datasets in comparison to transformer-based models.

Table 3-7 Analysis of hyperparameters setting for the MAI module based on Human3.6M dataset using Protocol#1.

Input Channel	Output Channel	with HFF?	MPJPE (mm) ↓	FLOPs ↓	Parameters ↓
128	32	No	31.1	49.52M	56.66M
256	64	No	31.1	49.85M	61.00M
128	32	Yes	30.3	35.88M	43.25M
256	64	Yes	**30.0**	36.21M	47.59M

3.3.4 Discussion

We investigated in this section the different settings of the MAI, HFF and CIE modules, analyzing the impact of different hyperparameters and design options in each module.

Different design options for the MAI. MAI has been introduced to encode joint motion amplitude information, in addition to positional and temporal information, to enable the network to better predict 3D poses. As a kind of global information, motion amplitude was encoded by TCN and introduced after the feature fusion, as illustrated in Figure 3-2. This section analyses the impacts of different input and output channel sizes for MAI and different ways of motion amplitude encoding. Table 3-8 compares the effects for different input and output channels, in which the results were obtained by one-stage training using ground-truth 2D poses as inputs. Table 3-8 shows that the performance of MPJPE cannot be improved when the input and output channels of MAI increase. When the HFF module is applied, however, MAI with larger input and output channels (256 and 64) would result in better MPJPE (30.0mm).

Table 3-8 Ablation study on whether to encode the MAI module separately on Human3.6M under Protocol#1.

Method (GT)	MPJPE (mm)	FLOPs (M)	Parameters (M)
Baseline	32.3	49.20	54.98
+MAI (together)	32.3	49.20	55.00
+MAI (together)+HFF	30.5	35.56	41.59
+MAI (separately)	31.1	49.85	61.00
+MAI (separately)+HFF	**30.0**	36.21	47.59

The motion amplitude information F_{MA}^{i} in equation (3-7) is separately encoded, instead of like input pose, positional and temporal information being encoded together as enhanced representation F_{E}^{i} (with reference to equation (3-4)). It is interesting to investigate how motion amplitude information should be encoded, whether together with other types of information or separately. The results of experimental analysis is shown in Table 3-9. When the four types of information (including input pose, position, temporal and motion amplitude information) were encoded together, the performance (MPJPE) did not improve. When we encoded the motion amplitude information separately (256 for the input channel and 64 for the output channel) as shown in Figure 3-5, the performance of MPJPE improved by 1.2mm over the baseline. By applying HFF module further, a MPJPE result of 30.0mm was obtained, representing an improvement of 0.5mm comparing to the method of encoding all three information together and applying the HFF module afterwards. There is, however, not much of an increase in FLOPs and number of parameters when encoding motion amplitude information separately compared to encoding them together. This has validated that the motion amplitude information should be encoded separately in general.

Table 3-9 Ablation study involving different settings of feature fusion module.

Feature Fusion	MPJPE (mm)	FLOPs (M)	Parameters (M)
Module design 1	32.3	49.20	54.98
Module design 2	31.1	32.15	39.21
Module design 3	31.1	45.79	51.83
Module design 4	**30.9**	35.56	41.58

Table 3-10 Ablation study on the hyperparameters of CIE module on Human3.6M under Protocol#1.

Embedding Channel	MPJPE (mm)	FLOPs (M)	Parameters (M)
32	29.8	36.54	47.92
64	29.6	36.87	48.25
128	29.7	37.53	48.91

Different design options for the HFF module. We conducted ablation experiments involving the HFF module based on the Human3.6M dataset using Protocol#1. We compared four different design settings for feature fusion module, as illustrated in Figure 3-7. Design 1 used the FCN feature fusion block for all five body parts to form the feature fusion module, while module design 2 adopted the Conv feature fusion block for all five body parts. Module design 3 used the Conv feature fusion block for the torso and the FCN feature fusion block for the four limb parts, while module design 4 utilized the FCN feature fusion block for the torso and the Conv feature fusion block for the four limb parts. As shown in Table 3-10, module design 1 produced a MPJPE result of 32.3mm. By replacing all the FCN feature fusion blocks with the Conv feature fusion blocks, the MPJPE result of module design 2 was improved by 1.2mm compared to module design 1, and the number of FLOPs and parameters were reduced substantially to 32.15M and 39.21M, respectively. This demonstrated the effectiveness of the Conv feature fusion block. In contrast to the design 2, module design 3 replaces the Conv feature fusion blocks with the FCN feature fusion modules for the limbs and achieved

the same MPJPE result of 31.1mm as the design 2. Nevertheless, the FLOPs and parameters of module design 3 were higher than those of module design 2. Module design 4 achieved a MPJPE of 30.9mm with the reduced number of FLOPs and parameters when comparing to design 3. This is because the torso has more joints and requires the FCN to learn with more parameters, while the limbs can be effectively learned by the Conv feature fusion block and prevent overfitting. Therefore, module design 4 is selected as our HFF module setting.

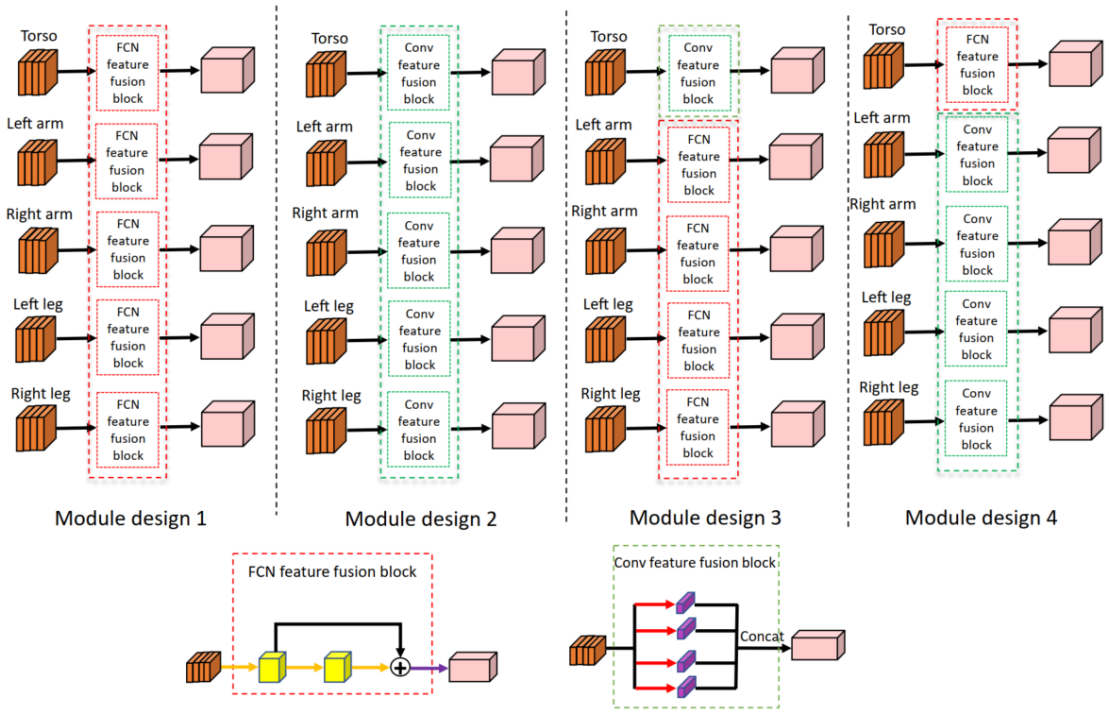


Figure 3-7 Comparison of different feature fusion modules.

Different design options for the CIE module. We analyzed the effect of embedding channel dimensions of the CIE module on the performance in terms of MPJPE (mm) in **Error! Reference source not found.**. Ablation experiments were again conducted for o ne-stage network based on Human3.6M dataset using Protocol#1. It became apparent from **Error! Reference source not found.** that simply increasing the channel d

imensions of the CIE module in an uninformed manner would not improve the performance. The CIE module, with 64 channels, gave the best MPJPE result (29.6 mm), indicating that this setting performed slightly better in terms of accuracy compared to the other two settings. The number of FLOPs and parameters increased with the number of embedding channels. This is because larger models require more computational resources and have higher complexity. In conclusion, the CIE module, with 64 embedding channels, seemed to be the best trade-off in terms of accuracy and computational complexity.

3.3.5 Qualitative Results

Figure 3-8 compares the qualitative results of our method obtained by one-stage training with those of RIE (Shan *et al.*, 2021a) obtained by three-stage training. Compared to RIE (Shan *et al.*, 2021a), our method produced more accurate predictions in 3D poses when the range of motion of the limbs is large or in actions that are heavily occluded.

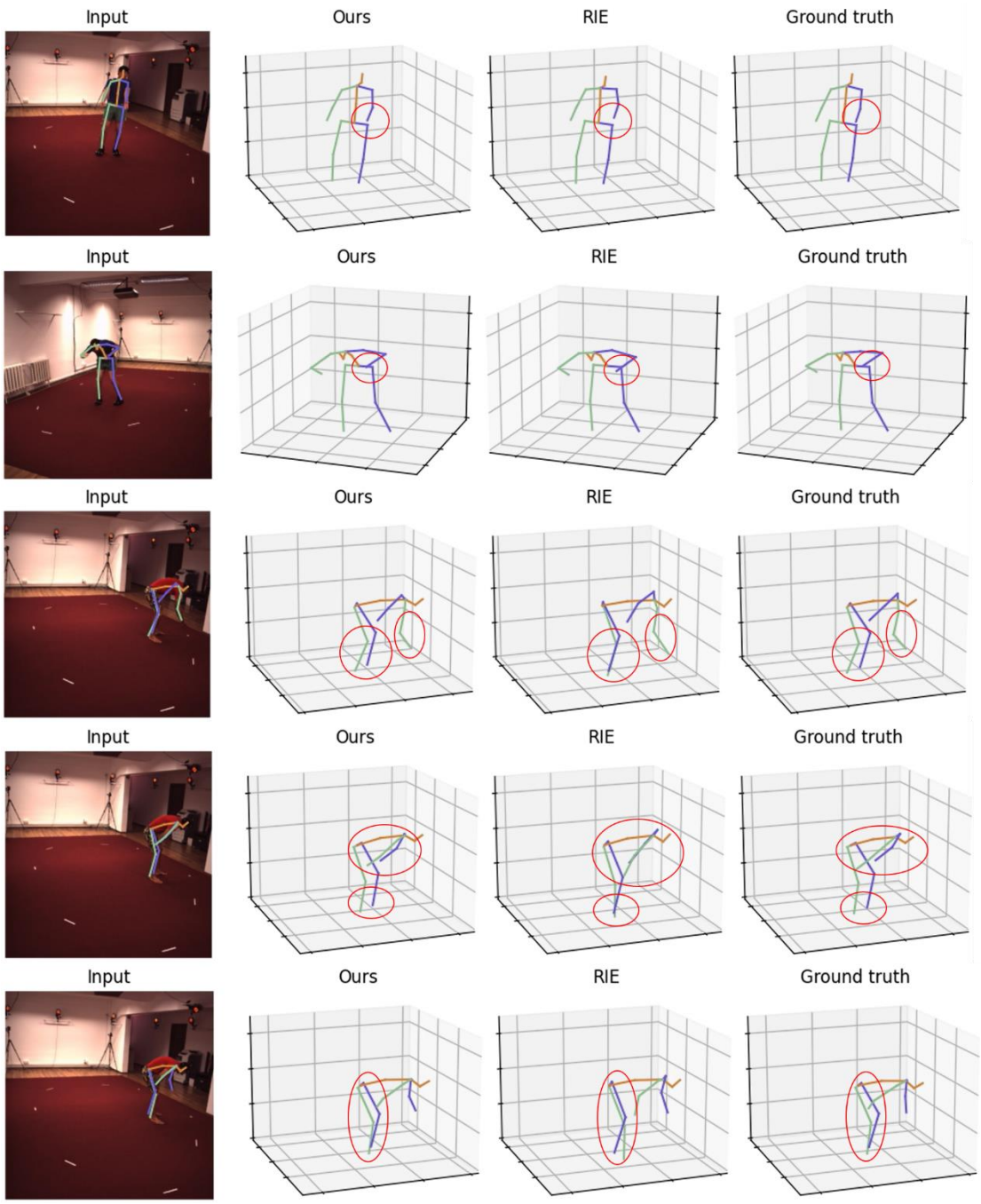


Figure 3-8 Qualitative results output by our method and those of RIE (Shan et al., 2021a).

3.4 Chapter Summary

In this chapter, a new network with three new encoding modules, including MAI, HFF and CIE, has been developed for grouped 3D pose estimation. It can be concluded that the motion amplitude encoding (MAI) and camera intrinsic embedding (CIE) modules

could provide global information to the network and improve the accuracy of 3D pose estimation. Furthermore, the optimized feature fusion (HFF) module could significantly reduce model complexity while ensuring the accuracy of the model. Compared to a previous approach (Shan *et al.*, 2021a), our method has used fewer parameters to fuse different groups of human pose features and also improved the performance. Moreover, a one-stage training scheme based on gradient detaching has been proposed to train, in an end-to-end manner, the new CNN-based network for 3D human pose estimation with grouped feature fusion, and this could greatly reduce the number of training epochs, saving training time with only a slight drop in accuracy in comparison to the multi-stage offline training strategy. Recently, the transformer exhibited stronger modeling capabilities than CNN, but it requires more computational resources. Thus, this current approach is designed to accommodate situations with limited computational resources. In the next chapter, a transformer-based method will be introduced to yield more accurate 3D pose in situations where there are ample computational resources.

CHAPTER 4. KINEMATICS AND TRAJECTORY PRIOR KNOWLEDGE-ENHANCED TRANSFORMER

4.1 Introduction

With reference to the overall research framework of the current study defined in Chapter 1, a total of three novel network models are developed. Chapter 3 explained the first model, a CNN-based network, while the rest two are transformer-based networks. This chapter explains the detail of the second model developed in this study, a transformer-based network for 3D pose estimation with video sequence inputs.

Transformer, a deep learning architecture, has revolutionized first in natural language processing (NLP) and later in other areas such as computer vision since its introduction in 2017 (Vaswani *et al.*, 2017). The name 'transformer' comes from the fact that these architectures use a self-attention mechanism to transform layers of inputs into layers of outputs in a way that allows the model to focus on (attend to) certain inputs. In terms of 3D pose estimation, the transformer first processes an input video into a sequence of tokens, the basic units of processing namely 2D poses, and then models the spatial-temporal relationship between tokens using multi-head self-attention (MHSA) mechanism.

The existing works of transformer-based methods for 3D human pose estimation (Li *et al.*, 2022b; Li *et al.*, 2022c; Shan *et al.*, 2022; Tang *et al.*, 2023b; Zhang *et al.*, 2022b; Zhao *et al.*, 2023; Zheng *et al.*, 2021a) mainly focus on developing novel transformer encoders. They model either the spatial correlation between joints within each frame

and the pose-to-pose or joint-to-joint temporal correlation across frames. Regardless of spatial or temporal MHSA calculation, the present transformer-based methods all use linear embedding where 2D pose sequence are tokenized into high dimensional features and treated uniformly to compute the spatial correlation between joints and the temporal correlation across frames in the spatial and temporal MHSA, respectively. This may lead to the problem of 'attention collapse', a phenomenon denoting a circumstance wherein the self-attention becomes too focused on a limited subset of input tokens while disregarding other segments of the sequence. In contrast to previous works, with the known anatomical structure of the human body as well as joint motion trajectory across frames as a priori knowledge, we propose a graph-based method to formulate such prior knowledge-attention for better learning the spatial and temporal correlations. Our graph-based prior attention mechanism is different from other existing graph-transformer methods (Gong *et al.*, 2023; Li *et al.*, 2023; Zhao *et al.*, 2022; Zhu *et al.*, 2021); without modifying the transformer structure or introducing complex network, instead, we design plug-and-play modules to be placed in front of MHSA modules of a vanilla transformer. Our method is simple yet effective, highly flexible and adaptable, allowing it to be integrated into different transformer-based methods.

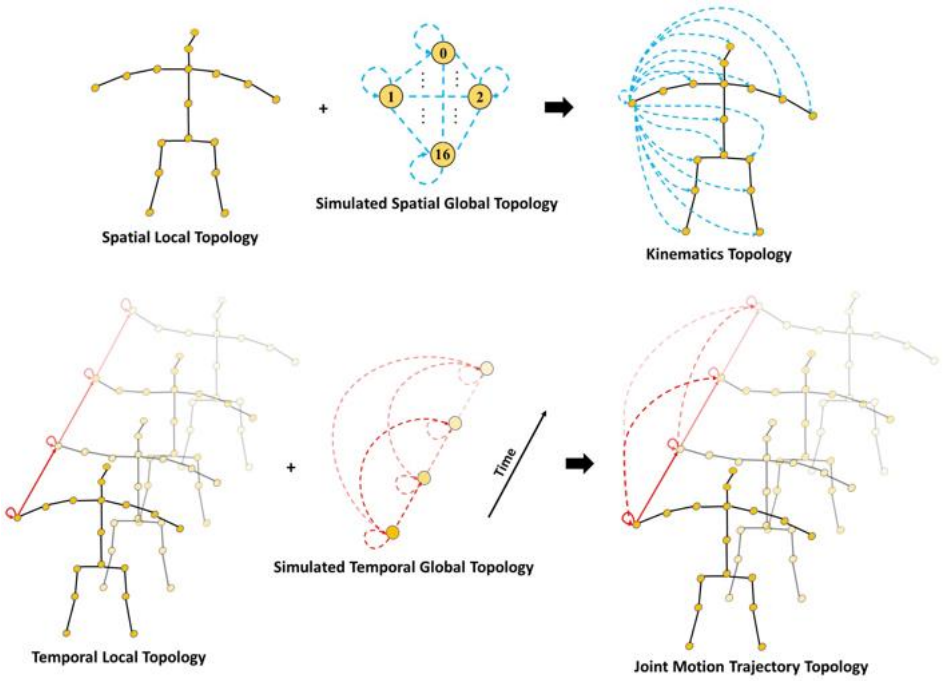


Figure 4-1 *Top: the spatial local topology (fixed) plus the simulated spatial global topology (learnable) to form the kinematics topology (learnable). Bottom: the temporal local topology (fixed plus the simulated temporal global topology (learnable) to form the joint motion trajectory topology (learnable).*

To be specific, we introduce two novel prior attention modules, namely Kinematics Prior Attention (KPA) and Trajectory Prior Attention (TPA), and the key concepts are illustrated in Figure 4-1. KPA first constructs a spatial local topology based on the anatomy of the human body, as shown at the top of Figure 4-1. The way these joints are physically connected to each other is fixed and is represented by solid lines. To introduce the kinematic relations among non-connected joints, we use a fully connected spatial topology to calculate the joint-to-joint attention weights, called simulated spatial global topology. In this topology, the strength of the connectivity relationship between each joint (including itself) is learnable, and thus we denote it with a dotted line in

Figure 4-1. We combine the spatial local topology and the simulated spatial global topology to obtain a kinematics topology, where each joint has a learnable kinematic relationship with each other. This kinematic topological information aims to provide a priori knowledge to the spatial MHSA, enabling it to assign weights to joints based on the kinematic relationships in different actions. Similarly, as shown in the bottom of Figure 4-1, TPA connects the same joint across consecutive frames to build the temporal local topology. Next, we construct a temporal global topology by exploiting learnable vectors (dotted line) to connect the joints among all neighbouring and non-neighbouring frames, which is equivalent to the computation of attention weights among all frames by self-attention, called simulated temporal global topology. Then, we combine the two topologies to obtain a new topology called joint motion trajectory topology, which allows the network to learn both the temporal sequentiality and periodicity (joints in non-neighbouring frames have similar motions to each other) for the joint motion. The temporal tokens embedded with the trajectory information will be more effectively activated in the temporal MHSA, which enhances the temporal modeling ability for MHSA. The KPA and TPA modules are combined with vanilla MHSA and MLP to form the Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer) for 3D pose estimation, as shown in Figure 4-2. The main contributions of this section are summarized as follows:

- We propose two novel prior attention modules, KPA and TPA, which can be combined with MHSA and MLP in a simple yet effective way, forming the

KTPFormer for 3D pose estimation.

- Our KTPFormer outperforms the state-of-the-art methods on Human3.6M, MPI-INF-3DHP and HumanEva benchmarks, respectively.
- KPA and TPA are designed as lightweight plug-and-play modules, which can be integrated into various transformer-based methods (including diffusion-based) for 3D pose estimation. Extensive experiments show that our method can significantly improve the performance without largely increasing computational resources.

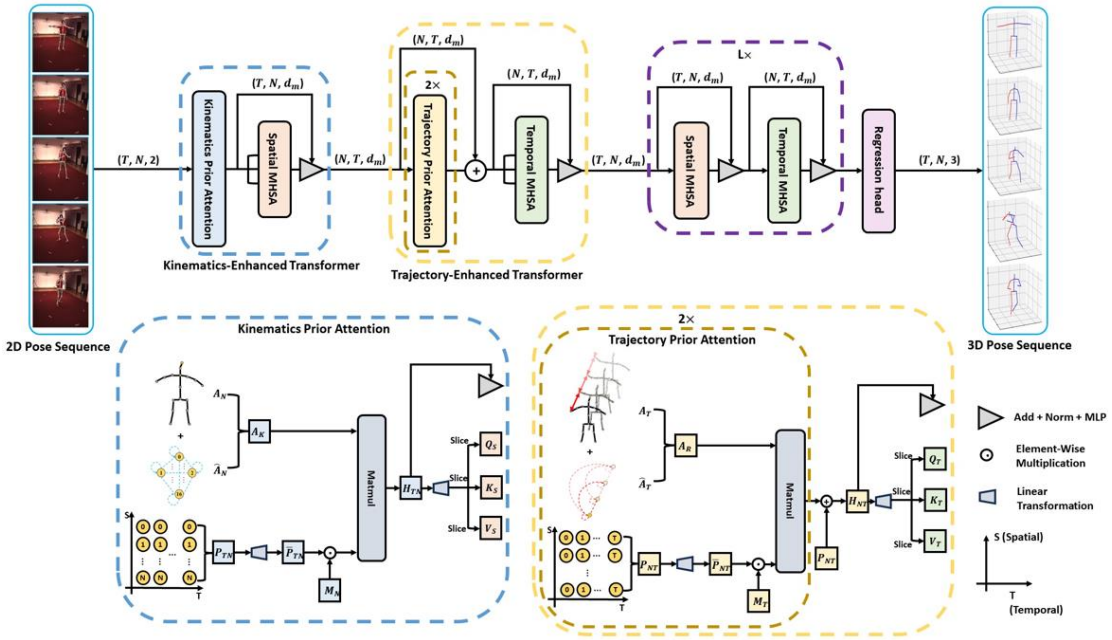


Figure 4-2 Overview of Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer). The input 2D pose sequence $P_{TN} \in \mathbb{R}^{T\times N\times 2}$ with T frames and N joints is first fed into the Kinematics-Enhanced Transformer.

4.2 Method

In this thesis, we propose a novel Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer), which combines kinematics and trajectory prior attentions and MHSA in a direct but effective way. Our KTPFormer can model both

spatial and temporal information simultaneously. Moreover, our method preserves the inherent structure of the transformer and is more flexible.

Our KTPFormer utilizes the seq2seq pipeline for 3D human pose estimation, which can simultaneously predict 3D pose sequence corresponding to the input 2D keypoint sequence. As shown in Figure 4-2, an input 2D pose sequence $P_{TN} \in \mathbb{R}^{T\times N\times 2}$ is first fed into the Kinematics-Enhanced Transformer, with T denotes the number of frames, N denotes the number of joints, and 2 is the channel size. KPA injects the kinematic topological information into the 2D pose $P_N \in \mathbb{R}^{N\times 2}$ in each frame, aiming to obtain high-dimensional spatial tokens $H_{TN} \in \mathbb{R}^{T\times N\times d_m}$. Next, the spatial MHSA transforms H_{TN} into matrics Q_S, K_S, V_S for learning the global correlation between joints. The Trajectory-Enhanced Transformer takes a sequence of reshaped tokens $P_{NT} \in \mathbb{R}^{N\times T\times d_m}$ as inputs. We stack two TPA blocks with the residual connection to generate the temporal tokens $H_{NT} \in \mathbb{R}^{N\times T\times d_m}$ with incorporated prior information on joint motion trajectories. Next, the temporal MHSA transforms H_{NT} into Q_T, K_T, V_T for modeling global coherence among frames. The output features from Temporal MHSA are reshaped and fed into stacked spatio-temporal transformers for encoding. Finally, the regression head predicts the coordinates of the 3D pose sequence based on the learned features.

4.2.1 Kinematics-Enhanced Transformer

Kinematics-Enhanced Transformer receives the input 2D keypoint sequence and

transforms them into high-dimensional spatial tokens. The 2D keypoint sequence first goes through the KPA for embedding the prior knowledge of kinematics, which is then fed into the spatial MHSA for global correlation learning between joints.

To be specific, given a 2D pose sequence as $P_{TN} \in \mathbb{R}^{T\times N\times 2}$, we regard each joint $p_i \in \mathbb{R}^2$ of a 2D pose $P_N \in \mathbb{R}^{N\times 2}$ as a keypoint patch. Next, we define a learnable transformation matrix $W \in \mathbb{R}^{2\times d_m}$ to map all keypoint patches P_{TN} into high-dimensional tokens $\bar{P}_{TN} \in \mathbb{R}^{T\times N\times d_m}$. In order to inject the prior information of kinematics into $\bar{P}_{TN}$, KPA first constructs a symmetric affinity matrix $A_N \in \mathbb{R}^{N\times N}$ based on the skeletal structure of the human body, namely spatial local topology, as shown in Figure 4-1. If two joints are physically connected in the human body structure, the corresponding element in the affinity matrix $A_N \in \mathbb{R}^{N\times N}$ is non-zero and 0 otherwise. The affinity matrix A_N can allow each 2D keypoint to learn anatomical structure information of the human body. Besides, KPA also considers the implicit kinematic relationships among adjacent and non-adjacent keypoints. Similar to the self-attention in MHSA, we establish a fully connected spatial topology, called simulated spatial global topology, as shown in Figure 4-1. In this topology, all the joints are interconnected by dotted lines, indicating that the connectivity relationship between each joint is learnable. The simulated spatial global topology is denoted as an affinity matrix $\hat{A}_N \in \mathbb{R}^{N\times N}$, where each element is learnable. Lastly, we combine the spatial local topology A_N with the simulated spatial global topology $\hat{A}_N$ to derive a kinematics topology A_K, which is shown as:

$$A_K = \frac{(A_N+\hat{A}_N)+(A_N+\hat{A}_N)'}{2} \quad (4-1)$$

where $'$ denotes the matrix transpose, $A_K \in \mathbb{R}^{N\times N}$ is a learnable affinity matrix. The reason why we construct the A_K with the above formula is that the original spatial local topology matrix A_N is also symmetric. In order to ensure that different keypoints can learn different kinematic knowledge, we introduce a learnable weight matrix $M_N \in \mathbb{R}^{N\times d_m}$ and multiply it with tokens $\bar{P}_{TN} \in \mathbb{R}^{T\times N\times d_m}$ by element-wise multiplication, which is an economic and effective way. Thus, we can obtain the tokens $H_{TN} \in \mathbb{R}^{T\times N\times d_m}$ including the prior knowledge of kinematics. The formula is represented as:

$$H_{TN} = (M_N \odot \bar{P}_{TN})A_K \quad (4-2)$$

Where $\odot$ represents element-wise multiplication. Moreover, we add the learnable spatial positional embedding to H_{TN}. After that, $H_{TN} \in \mathbb{R}^{T\times N\times d_m}$ is transformed into queries $Q_S \in \mathbb{R}^{T\times N\times d_m}$, keys $K_S \in \mathbb{R}^{T\times N\times d_m}$ and values $V_S \in \mathbb{R}^{T\times N\times d_m}$ by a linear transformation matrix. Then, we design a spatial MHSA ($MHSA_S$) to model global spatial correlation between keypoints within an identical frame. Each attention head ($i = 1, \dots, h$) can be represented as:

$$head_i = Softmax\left(\frac{Q_S^i\left(K_S^i\right)'}{\sqrt{d}}\right)V_S^i \quad (4-3)$$

where $'$ denotes the matrix transpose. All the attention heads are concatenated together to form the $MHSA_S$:

$$MHSA_S(Q_S, K_S, V_S) = \boldsymbol{cat}(head_1, \dots, head_h)W_S \quad (4-4)$$

Where $W_S \in \mathbb{R}^{d_m\times d_m}$ is the linear transformation matrix. Concurrently, H_{TN} as a

residual adds the output of $MHSA_S$ to form the new output $H_S \in \mathbb{R}^{T\times N\times d_m}$, which is then fed into the layer normalization (LN) and MLP followed by a residual connection and LN. The formula can be represented as:

$$H_S = MHSA_S(Q_S, K_S, V_S) + H_{TN} \quad (4-5)$$

$$P_{NT} = MLP\big(LN(H_S)\big) + H_S \quad (4-6)$$

Where $P_{NT} \in \mathbb{R}^{N\times T\times d_m}$ is the output of the Kinematics-Enhanced Transformer after being reshaped.

4.2.2 Trajectory-Enhanced Transformer

Trajectory-Enhanced Transformer aims to integrate the prior trajectory information of joint motion across frames into a sequence of tokens $P_{NT} \in \mathbb{R}^{N\times T\times d_m}$, in which each joint is regarded as an individual token in the time dimension. TPA first connects the identical keypoints (including itself) across neighboring frames to construct the temporal local topology, as shown in Figure 4-1, which is denoted as the symmetric affinity matrix $A_T \in \mathbb{R}^{T\times T}$. In order to enhance the global attention of temporal coherence in the MHSA, we simulate a temporal global topology that considers the implicit temporal correlation among neighboring and non-neighboring frames. These keypoints belonging to the identical trajectory among neighboring and non-neighboring frames are connected by the learnable vector (dotted line) to form the simulated temporal global topology, as shown in Figure 4-1. This topology can be expressed in

the form of a learnable matrix $\hat{A}_T \in \mathbb{R}^{T\times T}$. Thus, the equation of joint motion trajectory topology can be represented as:

$$A_R = \frac{\left(A_T + \hat{A}_T\right) + \left(A_T + \hat{A}_T\right)'}{2} \quad (4-7)$$

where $'$ denotes the matrix transpose, $A_R \in \mathbb{R}^{T\times T}$ is a learnable affinity matrix. Then, we transform P_{NT} to embeddings $\bar{P}_{NT} \in \mathbb{R}^{N\times T\times d_m}$ by the linear transformation. Also, we utilize a learnable weight matrix $M_T \in \mathbb{R}^{T\times d_m}$ to allow different keypoints for different prior knowledge learning. The formula of one TPA is represented as:

$$TPA(P_{NT}) \coloneqq (M_T \odot \bar{P}_{NT})A_R \quad (4-8)$$

We stack two TPA blocks with a residual connection to obtain the temporal tokens $H_{NT} \in \mathbb{R}^{N\times T\times d_m}$ as follows:

$$H_{NT} = TPA\big(TPA(P_{NT})\big) + P_{NT} \quad (4-9)$$

The learnable temporal positional embedding is then added to H_{NT}. After that, the $H_{NT} \in \mathbb{R}^{N\times T\times d_m}$ is converted into queries $Q_T \in \mathbb{R}^{N\times T\times d_m}$, keys $K_T \in \mathbb{R}^{N\times T\times d_m}$ and values $V_T \in \mathbb{R}^{N\times T\times d_m}$ by the linear transformation. We use a temporal MHSA ($MHSA_T$) to model the global temporal correlation between joints across all frames as follows:

$$head_i = Softmax\left(\frac{Q_T^i\left(K_T^i\right)'}{\sqrt{d}}\right)V_T^i \quad (4-10)$$

$$MHSA_T(Q_T, K_T, V_T) = \boldsymbol{cat}(head_1, \dots, head_h)W_T \quad (4-11)$$

where $W_T \in \mathbb{R}^{d_m\times d_m}$ is the linear transformation matrix. Similar to $MHSA_S$, we can obtain the final output H_{ST}:

$$H_T = MHSA_T(Q_T, K_T, V_T) + H_{NT} \quad (4-12)$$

$$H_{ST} = MLP\big(LN(H_T)\big) + H_T \quad (4-13)$$

where $H_{ST} \in \mathbb{R}^{N \times T \times d_m}$ is the output of Trajectory-Enhanced Transformer.

4.2.3 Stacked Spatio-Temporal Encoders

After being reshaped, H_{ST} is fed into the stacked spatio-temporal encoders which consist of alternating spatial and temporal transformers. The number of stacks is L. The sequential features are reshaped according to the type of the MHSA before fed into the encoder (spatial or temporal).

4.2.4 Regression Head

We utilize the linear layer as a regression head to predict the 3D pose sequence $\hat{P}_{3D} \in \mathbb{R}^{T \times N \times 3}$. The overall loss function for our network is given as:

$$\mathcal{L} = \mathcal{L}_W + \lambda_T \mathcal{L}_T + \lambda_M \mathcal{L}_M \qquad (4-14)$$

where $\mathcal{L}_W$ denotes the weighted mean per-joint position error (WMPJPE) loss (Zhang *et al.*, 2022b), $\mathcal{L}_T$ is the temporal consistency loss (Hossain & Little, 2018a), and $\mathcal{L}_M$ indicates the mean per joint velocity error (MPJVE) loss (Pavllo *et al.*, 2019b). Here λ_T and λ_M are hyper-parameters.

4.3 Experiments

4.3.1 Datasets and Protocols

Datasets. We evaluated our model on three public datasets, namely Human3.6M (Ionescu *et al.*, 2013), MPI-INF-3DHP (Mehta *et al.*, 2017) and HumanEva (Sigal *et al.*, 2010). Human3.6M is an indoor scenes dataset with 3.6 million video frames. It has

11 professional actors, performing 15 actions under 4 synchronized camera views. Following previous work (Tang *et al.*, 2023b; Zhang *et al.*, 2022b), we used subjects 1, 5, 6, 7 and 8 for training, and subjects 9 and 11 for testing. MPI-INF-3DHP is also a public large-scale dataset. Following the setting of (Tang *et al.*, 2023b; Zhang *et al.*, 2022b), we use the area under the curve (AUC), percentage of correct keypoints (PCK) and MPJPE as evaluation metrics. HumanEva is a smaller dataset in the indoor environment. To have a fair comparison with (Zhang *et al.*, 2022b; Zheng *et al.*, 2021a), we valuated our method for actions (Walk and Jog) of subjects S1, S2, S3.

Protocol. Protocol#1 is denoted as the mean per-joint position error (MPJPE), which is the average Euclidean distance in millimetres (mm) between the predicted and the ground-truth 3D joint coordinates. Protocol#2 refers to the reconstruction error after the predicted 3D pose is aligned to the ground-truth 3D pose using procrustes analysis (Gower, 1975), denoted as P-MPJPE (mm).

4.3.2 Implementation Details

We implemented our method in the Pytorch framework on one GeForce RTX 3090 GPU. The input 2D keypoints were detected by 2D pose detector (Chen *et al.*, 2018) or 2D ground truth. The W in WMPJPE follows the setting (1.0, 1.5, 2.5, and 4.0) of MixSTE (Zhang *et al.*, 2022b). We set the number of stacked spatio-temporal encoders L to 7. Thus, the encoders contain 14 spatial and temporal transformer layers with number of heads $h = 8$, feature size $C = 512$. During the training stage, we use the

Adam (Kingma & Ba, 2014) optimizer to train our model. The batch size, dropout rate, and activation function are set to 7, 0.1, and GELU. The learning rate is initialized to 0.00007 and decayed by 0.99 per epoch. Recently, diffusion models have been introduced in 3D pose estimation (Gong *et al.*, 2023; Shan *et al.*, 2023) and have achieved significant improvements in performance, as the diffusion process can be viewed as an augmentation method for pose data. To demonstrate the adaptability of our method, we introduced a diffusion process to our network, following the setting of D3DP (Shan *et al.*, 2023), which also uses the transformer-based network as the backbone. We used our KTPFormer as the denoiser in the D3DP (Shan *et al.*, 2023). For the design of the remaining diffusion process, our experimental parameters were set to be the same as D3DP (Shan *et al.*, 2023).

4.3.3 Comparison with State-of-the-art Methods

Results based on Human3.6M. We compared our results with those of recent state-of-the-art methods based on the dataset Human3.6M. As shown in Table 4-1, our method (diffusion-based) achieves the state-of-the-art (SOTA) result 33.0mm in MPJPE and 26.2mm in P-MPJPE using the 2D poses detected by CPN (Chen *et al.*, 2018) as inputs. Our method (diffusion-based) outperforms D3DP (Shan *et al.*, 2023) by 2.4mm under MPJPE and 2.5mm under P-MPJPE with the same settings (the number of frames, hypotheses, and iterations) as D3DP (Shan *et al.*, 2023). This demonstrates that our network can serve as an excellent backbone for diffusion-based methods, effectively

improving the performance for 3D pose estimation. Besides, we obtain the best results 40.1mm under T =243 setting and 41.8mm under T =81 setting in MPJPE among all methods that are not diffusion-based.

Table 4-1 *Quantitative comparison results with the state-of-the-art methods on Human3.6M. The 2D poses obtained by CPN (Chen et al., 2018) are used as inputs. Top table: evaluation results of MPJPE (mm); Bottom table: evaluation results of P-MPJPE (mm); T is the number of input frames. (†) denotes using temporal information, and (*) indicates the diffusion-based methods. Red: Best results. Blue: Runner-up results.*

MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Wang *et al.* (2020b) (T=96)†	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
Zheng *et al.* (2021a) (T=81)†	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
Li *et al.* (2022b) (T=351)†	40.3	43.3	40.2	42.3	45.6	52.3	41.8	40.5	55.9	60.6	44.2	43.0	44.2	30.0	30.2	43.7
Zhao *et al.* (2022)	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
Li *et al.* (2022c) (T=351)†	39.2	43.1	40.1	40.9	44.9	51.2	40.6	41.3	53.5	60.3	43.7	41.1	43.8	29.8	30.6	43.0
Shan *et al.* (2022) (T=243)†	38.9	42.7	40.4	41.1	45.6	49.7	40.9	39.9	55.5	59.4	44.9	42.2	42.7	29.4	29.4	42.8
Zhang *et al.* (2022b) (T=81)†	39.8	43.0	38.6	40.1	43.4	50.6	40.6	41.4	52.2	56.7	43.8	40.8	43.9	29.4	30.3	42.4
Zhang *et al.* (2022b) (T=243)†	37.6	40.9	37.3	39.7	42.3	49.9	40.1	39.8	51.7	55.0	42.1	39.8	41.0	27.9	27.9	40.9
Zhang *et al.* (2022a) (T=300)†	37.9	41.9	36.8	39.5	40.8	49.2	40.1	40.7	47.9	53.3	40.2	41.1	40.3	30.8	28.6	40.6
Yu *et al.* (2023) (T=243)†	41.3	44.3	40.8	41.8	45.9	54.1	42.1	41.5	57.8	62.9	45.0	42.8	45.9	29.4	29.9	44.4
Li *et al.* (2023)	47.9	50.0	47.1	51.3	51.2	59.5	48.7	46.9	56.0	61.9	51.1	48.9	54.3	40.0	42.9	50.5
Tang *et al.* (2023b) (T=81)†	40.6	43.0	38.3	40.2	43.5	52.6	40.3	40.1	51.8	57.7	42.8	39.8	42.3	28.0	29.5	42.0
Tang *et al.* (2023b) (T=243)†	38.4	41.2	36.8	38.0	42.7	50.5	38.7	38.2	52.5	56.8	41.8	38.4	40.2	26.2	27.7	40.5
Gong *et al.* (2023) (T=243)†*	33.2	36.6	33.0	35.6	37.6	45.1	35.7	35.5	46.4	49.9	37.3	35.6	36.5	24.4	24.1	36.9
Shan *et al.* (2023) (T=243)†*	33.0	34.8	31.7	33.1	37.5	43.7	34.8	33.6	45.7	47.8	37.0	35.0	35.0	24.3	24.1	35.4
Ours (T=81)†	39.1	41.9	37.3	40.1	44.0	51.3	39.8	41.0	51.4	56.0	43.0	41.0	42.6	28.8	29.5	41.8
Ours (T=243)†	37.3	39.2	35.9	37.6	42.5	48.2	38.6	39.0	51.4	55.9	41.6	39.0	40.0	27.0	27.4	40.1
Ours (T=243)†*	30.1	32.1	29.1	30.6	35.4	39.3	32.8	30.9	43.1	45.5	34.7	33.2	32.7	22.1	23.0	33.0
P-MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Wang *et al.* (2020b) (T=96)†	31.8	34.3	35.4	33.5	35.4	41.7	31.1	31.6	44.4	49.0	36.4	32.2	35.0	24.9	23.0	34.5
Zheng *et al.* (2021a) (T=81)†	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Li *et al.* (2022b) (T=351)†	32.7	35.5	32.5	35.4	35.9	41.6	33.0	31.9	45.1	50.1	36.3	33.5	35.1	23.9	25.0	35.2
Shan *et al.* (2022) (T=243)†	31.3	35.2	32.9	33.9	35.4	39.3	32.5	31.5	44.6	48.2	36.3	32.9	34.4	23.8	23.9	34.4
Zhang *et al.* (2022b) (T=81)†	32.0	34.2	31.7	33.7	34.4	39.2	32.0	31.8	42.9	46.9	35.5	32.0	34.4	23.6	25.2	33.9
Zhang *et al.* (2022b) (T=243)†	30.8	33.1	30.3	31.8	33.1	39.1	31.1	30.5	42.5	44.5	34.0	30.8	32.7	22.1	22.9	32.6
Zhang *et al.* (2022a) (T=300)†	30.3	34.6	29.6	31.7	31.6	38.9	31.8	31.9	39.2	42.8	32.1	32.6	31.4	25.1	23.8	32.5

Yu *et al.* (2023) (T=243)†	32.4	35.3	32.6	34.2	35.0	42.1	32.1	31.9	45.5	49.5	36.1	32.4	35.6	23.5	24.7	34.8
Tang *et al.* (2023b) (T=81)†	30.4	33.8	31.1	31.7	33.5	39.5	30.8	30.0	41.8	45.8	34.3	30.1	32.8	21.9	23.4	32.7
Tang *et al.* (2023b) (T=243)†	29.3	33.0	30.7	30.6	32.7	38.2	29.7	28.8	42.2	45.0	33.3	29.4	31.5	20.9	22.3	31.8
Shan *et al.* (2023) (T=243)†*	27.5	29.4	26.6	27.7	29.2	34.3	27.5	26.2	37.3	39.0	30.3	27.7	28.2	19.6	20.3	28.7
Ours (T=81)†	30.6	33.4	30.1	31.9	33.7	38.2	30.6	30.7	40.9	44.8	34.4	30.5	32.7	22.3	24.0	32.6
Ours (T=243)†	30.1	32.3	29.6	30.8	32.3	37.3	30.0	30.2	41.0	45.3	33.6	29.9	31.4	21.5	22.6	31.9
Ours (T=243)†*	24.1	26.7	24.2	24.9	27.3	30.6	25.2	23.4	34.1	35.9	28.1	25.3	25.9	17.8	18.8	26.2

Table 4-2 *Quantitative comparison results of MPJPE (mm) with the state-of-the-art methods on Human3.6M using ground-truth (GT) 2D poses as inputs. T is the number of input frames. (†) denotes using temporal information, and (*) indicates the diffusion-based methods. Red: Best results. Blue: Runner-up results.*

MPJPE (GT)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Wang *et al.* (2020b) (T=96)†	23.0	25.7	22.8	22.6	24.1	30.6	24.9	24.5	31.1	35.0	25.6	24.3	25.1	19.8	18.4	25.6
Zhu *et al.* (2021)	37.2	42.2	32.6	38.6	38.0	44.0	40.7	35.2	41.0	45.5	38.2	39.5	38.2	29.8	33.0	38.2
Zheng *et al.* (2021a) (T=81)†	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
Li *et al.* (2022b) (T=351)†	27.1	29.4	26.5	27.1	28.6	33.0	30.7	26.8	38.2	34.7	29.1	29.8	26.8	19.1	19.8	28.5
Zhao *et al.* (2022)	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
Li *et al.* (2022c) (T=351)†	27.7	32.1	29.1	28.9	30.0	33.9	33.0	31.2	37.0	39.3	30.0	31.0	29.4	22.2	23.0	30.5
Shan *et al.* (2022) (T=243)†	28.5	30.1	28.6	27.9	29.8	33.2	31.3	27.8	36.0	37.4	29.7	29.5	28.1	21.0	21.0	29.3
Zhang *et al.* (2022a) (T=300)†	22.1	23.1	20.1	22.7	21.3	24.1	23.6	21.6	26.3	24.8	21.7	21.4	21.8	16.7	18.7	22.0
Zhang *et al.* (2022b) (T=81)†	25.6	27.8	24.5	25.7	24.9	29.9	28.6	27.4	29.9	29.0	26.1	25.0	25.2	18.7	19.9	25.9
Zhang *et al.* (2022b) (T=243)†	21.6	22.0	20.4	21.0	20.8	24.3	24.7	21.9	26.9	24.9	21.2	21.5	20.8	14.7	15.7	21.6
Li *et al.* (2023)	32.9	38.3	28.3	33.8	34.9	38.7	37.2	30.7	34.5	39.7	33.9	34.7	34.3	26.1	28.9	33.8
Tang *et al.* (2023b) (T=81)†	26.2	26.5	23.4	24.6	25.0	28.6	28.3	24.6	30.9	33.7	25.7	25.3	24.6	18.6	19.7	25.7
Tang *et al.* (2023b) (T=243)†	21.4	22.6	21.0	21.3	23.8	26.0	24.2	20.0	28.9	28.0	22.3	21.4	20.1	14.2	15.0	22.0
Yu *et al.* (2023) (T=243)†	20.1	21.2	20.0	19.6	21.5	26.7	23.3	19.8	27.0	29.4	20.8	20.1	19.2	12.8	13.8	21.0
Gong *et al.* (2023) (T=243)†*	18.6	19.3	18.0	18.4	18.3	21.5	21.5	19.1	23.6	22.3	18.6	18.8	18.3	12.8	13.9	18.9
Shan *et al.* (2023) (T=243)†*	18.7	18.2	18.4	17.8	18.6	20.9	20.2	17.7	23.8	21.8	18.5	17.4	17.4	13.1	13.6	18.4
Ours (T=81)†	22.5	22.4	21.3	21.4	21.2	25.5	24.2	22.4	24.4	27.5	22.7	21.4	21.7	16.3	17.3	22.2
Ours (T=243)†	19.6	18.6	18.5	18.1	18.7	22.1	20.8	18.3	22.8	22.4	18.8	18.1	18.4	13.9	15.2	19.0
Ours (T=243)†*	18.8	17.4	18.1	17.7	18.3	20.6	19.6	17.7	23.3	22.0	18.7	17.0	16.8	12.4	13.5	18.1

As shown in the results of Table 4-2, ground-truth 2D poses were used as input for experiments. Among all the methods, our method (diffusion-based) achieves the SOTA result 18.1mm, with the same settings (the number of frames, hypotheses, and iterations) as D3DP (Shan *et al.*, 2023). On the other hand, we also obtain the best result 19.0mm under T=243 setting and 22.2mm under T=81 setting in MPJPE without diffusion process. Compared to GLA-GCN (Yu *et al.*, 2023), there is a noticeable improvement (21.0→19.0mm) with T=243. Under T=81 setting, our method significantly outperforms the second-best result by 3.5mm (25.7→22.2mm).

Results based on MPI-INF-3DHP. We evaluate the performance on MPI-INF-3DHP dataset to verify the generalization capability of our method. Following previous work (Tang *et al.*, 2023b; Zhao *et al.*, 2023), we input the ground-truth 2D poses to train our model. Table 4-3 reports the comparison results on the MPI-INF-3DHP test set. Our method with T=81 achieves the SOTA result with PCK of 98.9%, AUC of 85.9% and MPJPE of 16.7mm, outperforming the existing SOTA models by 0.2% in PCK, 2.0% in AUC and 6.4mm in MPJPE. Moreover, our method with T=27 also surpasses all other methods in terms of three metrics. These results demonstrate the strong generalization capability of our method on complicated datasets.

Table 4-3 Performance comparisons on MPI-INF-3DHP with PCK, AUC and MPJPE. The ↑ denotes the higher, the better, the ↓ denotes the lower, the better.

Method	PCK↑	AUC↑	MPJPE↓
Wang *et al.* (2020b) (T=96)	86.9	62.1	68.1
Zheng *et al.* (2021a) (T=9)	88.6	56.4	77.1
Li *et al.* (2022c) (T=9)	93.8	63.3	58.0

Zhang *et al.* (2022b) (T=27)	94.4	66.5	54.9
Shan *et al.* (2022) (T=81)	97.9	75.8	32.2
Gong *et al.* (2023) (T=81)	98.0	75.9	29.1
Shan *et al.* (2023) (T=243)	98.0	79.1	28.1
Zhao *et al.* (2023) (T=81)	97.9	78.8	27.8
Yu *et al.* (2023) (T=81)	98.5	79.1	27.7
Tang *et al.* (2023b) (T=27)	98.4	83.4	24.2
Tang *et al.* (2023b) (T=81)	98.7	83.9	23.1
Ours (T=27)	98.9	84.4	19.2
Ours (T=81)	98.9	85.9	16.7

Results based on HumanEva. Table 4-4 shows the performance in comparison to other methods on HumanEva dataset. Our method yields the best MPJPE result of 15.3mm under T=27. Also, our method is superior to other algorithms under T=81. Compared with MixSTE (Zhang *et al.*, 2022b), we achieve 36.8% improvement (28.5→18.0mm) under T=81. Due to the short video length in HumanEva, our method gives better results under T=27 than T=81. These results highlight the effectiveness of our method on small datasets.

Table 4-4 The MPJPE evaluation results on HumanEva testset.

Method	Walk			Jog			**Avg**
	S1	S2	S3	S1	S2	S3	
Pavllo *et al.* (2019b) (T=81)	13.1	10.1	39.8	20.7	13.9	15.6	18.9
Zheng *et al.* (2021a) (T=43)	16.3	11.0	47.1	25.0	15.2	15.1	21.6
Zhang *et al.* (2022b) (T=43)	20.3	22.4	34.8	27.3	32.1	34.3	28.5
Ours (T=43)	16.5	13.9	19.9	25.3	15.9	16.5	18.0
Ours (T=27)	12.3	11.5	19.5	20.9	13.1	14.5	15.3

4.3.4 Ablation Study

Effect of each module. To verify the effectiveness of the proposed modules, we conducted ablation experiments under T=243 on Human3.6M using ground-truth 2D poses as inputs. Table 4-5 presents the results of ablation study of each module. Our

baseline network first utilizes a linear layer to lift the 2D pose sequence to the high-dimensional space and then exploits the stacked spatio-temporal encoders (L=8) to predict the 3D pose sequence, reaching 22.1mm of MPJPE. The introduction KPA and TPA brings 2.1mm and 2.4mm of MPJPE drops, respectively. With both KPA and TPA modules, the performance has improved 3.1mm. More remarkably, the number of parameters and FLOPs merely increase by 0.0016M and 21M, respectively, showing that our method is both effective and efficient.

Table 4-5 Results of ablation study of each module in our KPTFormer on Human3.6M dataset.

Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
Baseline	22.1	33.6506	139038
+KPA	20.0	33.6501	139042
+TPA	19.7	33.6527	139055
+KPA+TPA	19.0	33.6522	139059

Effect on different combinations of KPA and TPA. We analyzed the impacts to performance for four different combinations of KPA and TPA, including the United Mode (UMD), the Separate Mode (SMD), the Separate Mode-S and the Parallel Mode (PMD). UMD indicates that the output of the KPA is fed into the two TPA blocks with a residual connection, followed by stacked spatio-temporal encoders. SMD represents that KPA is followed by spatial MHSA and two TPA blocks with a residual connection are followed by temporal MHSA. The SMD-S differs from the SMD in that only one TPA block is followed by temporal MHSA. For PMD, the input is fed into TPA and Kinematics-Enhanced Transformer simultaneously, and the outputs of them are then added together and fed into temporal MHSA. We evaluated the four modes on

Human3.6M with input T=243 frames. Table 4-6 shows the comparison results between the four modes. The MPJPE results of UMD are worse than those of SMD because these features from KPA are fed directly into TPA, which leads to the confusion of spatial and temporal information. The comparisons between the PMD and SMD illustrate that TPA is more suitable to inject the trajectory information into high-dimensional tokens, rather than the initial 2D pose sequence. Besides, KPA and TPA should be independently followed by spatial MHSA and temporal MSHA, without introducing other information to cause disruption. The comparison between the SMD-S and SMD indicates that the stacked TPA blocks can inject the prior information into high-dimensional tokens more effectively.

Table 4-6 Results of ablation study involving different combinations of KPA and TPA in the network.

Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
Baseline	22.1	33.6506	139038
United Mode (UMD)	20.0	33.6522	139059
Parallel Mode (PMD)	19.8	33.6512	139051
Separate Mode-S (SMD-S)	20.4	33.6512	139051
Separate Mode (SMD)	19.0	33.6522	139059

Different Numbers of Modules. We validate the impact of different numbers of KPA and TPA blocks in the KTPFormer. Table 4-7 reports the MPJPE and P-MPJPE comparisons on Human3.6M dataset. We take the estimated 2D poses by CPN as input and train these models under 81 frames. The baseline network utilizes the stacked spatio-temporal encoders (L=8) with number of heads H=8 and feature size C=512 to predict the 3D pose sequence. In our KTPFormer (first block), we combine KPA and TPA respectively with vanilla spatial transformer and temporal transformer, forming

Kinematics-Enhanced Transformer and Trajectory-Enhanced Transformer, which are placed at the beginning of the network. Subsequently, we employ the stacked spatio-temporal encoders (L=7) to encode features. In the KTPFormer (all blocks), we stack the Kinematics-Enhanced Transformer and Trajectory-Enhanced Transformer for L=8 loops. As indicated by the results, our KTPFormer (first block) obtains the lowest errors of MPJPE and P-MPJPE, indicating that KPA and TPA are better suited for processing the initial 2D pose sequence. Also, the KTPFormer (first block) can improve the performance more efficiently and has only a smaller increase in the computational overhead compared to the KTPFormer (all blocks). The design of KTPFormer (first block) is more effectively applicable to different 3D pose estimators.

Table 4-7 *The MPJPE and P-MPJPE comparisons with different numbers of KPA and TPA blocks in the KTPFormer. The evaluation is performed on Human3.6M with 81 input frames. The best result in each column is marked in red.*

Method	Parameters (M)	FLOPs (M)	MPJPE (mm)	P-MPJPE (mm)
Baseline	33.650	46346	43.1	34.1
KTPFormer (all blocks)	33.673	46412	42.3	33.4
KTPFormer (first block)	33.652	46353	41.8	32.6

Different Combination Ways of Topologies. We compare two different ways of combining the local topology and the simulated global topology. The first combination has been illustrated in the main text. We apply the first combination way to our KTPFormer, namely KTPFormer (average). The second combination is to directly add the local topology and the simulated global topology to obtain the kinematics topology

or the joint motion trajectory topology. The second combination way is also applied to the KTPFormer, called KTPFormer (add). We train the two networks using the estimated 2D poses by CPN with 81 frames as input. As shown in Table 4-8, the KTPFormer (average) achieves the best results of MPJPE and P-MPJPE. It suggests that the KTPFormer (average) which ensures the symmetry of the final topology allows the nodes to learn the spatial or temporal prior knowledge between them without being influenced by the direction of node connections.

Table 4-8 *The MPJPE and P-MPJPE comparisons with different combination ways of topologies in the KPA and TPA. The evaluation is performed on Human3.6M with 81 input frames. The best result in each column is marked in red.*

Method	Parameters (M)	FLOPs (M)	MPJPE (mm)	P-MPJPE (mm)
KTPFormer (add)	33.652	46353	42.1	33.3
KTPFormer (average)	33.652	46353	41.8	32.6

Free Parameters. We conduct experiments on the KTPFormer under three free parameters, including the number of spatio-temporal encoders L, the feature size of transformer layers C and the number of heads H, to examine different architectures of KTPFormer. During the experiment, we alter each free parameter while maintaining a constant value for the remaining two parameters. Table 4-9 reports the comparisons on Human3.6M using the CPN's 2D pose detection with 81 frames as input. The KTPFormer with L=7, C=512 and H=8 achieves the runner-up result of MPJPE and the best result of P-MPJPE, and strikes a balance between regression capacity and

computational cost. Thus, we choose this configuration as the standard version of KTPFormer.

Table 4-9 *The MPJPE and P-MPJPE of KTPFormer with different number of spatio-temporal encoders L, feature size of transformer layers C, and the number of heads H in self-attention on Human3.6M dataset. Red: Best results. Blue: Runner-up results.*

L	C	H	Parameters (M)	FLOPs (M)	MPJPE (mm)	P-MPJPE (mm)
6	512	4	29.446	40560	42.2	33.4
7	512	4	33.652	46353	41.7	33.1
8	512	4	37.857	52145	43.0	33.6
7	256	4	8.437	11625	43.0	33.7
7	512	4	33.652	46353	41.7	33.1
7	1024	4	134.413	185115	42.5	33.7
7	512	1	33.652	46353	43.0	34.2
7	512	2	33.652	46353	42.8	33.8
7	512	4	33.652	46353	41.7	33.1
7	512	8	33.652	46353	41.8	32.6
7	512	16	33.652	46353	42.5	33.4

4.3.5 Qualitative Analysis

We visualize the 3D pose estimation results and attention maps to validate the efficacy of our method in comparison to MixSTE (Zhang *et al.*, 2022b). As shown in Figure 4-3, the spatial and temporal attention outputs from different heads are both averaged to show the distribution of attention weights of joints and frames. Figure 4-3(a) illustrates the phenomenon of unreasonable attention weight allocation to the right arm, right leg and torso in the spatial attention map of MixSTE (Zhang *et al.*, 2022b), leading to poor predictions of the 3D pose (top of Figure 4-3). In contrast, the spatial attention weights (Figure 4-3(b)) are activated by KPA in regions of right arm, right leg and torso. In

particular, the three joints of the right arm exhibit stronger attention weights in the thorax column, owing to the anatomical connection between the right hand and the torso. The attention allocation is more reasonable (Figure 4-3(b)), contributing to an enhanced performance of 3D pose predicted by our method. Moreover, Figure 4-3(c) depicts the averaged temporal attention outputs of the three joints of right arm. In contrast, TPA (Figure 4-3(d)) yields stronger correlations across adjacent frames due to the continuity of human movements. The enhanced temporal attention also contributes to the performance improvement of the right arm. Also, we present more qualitative results of KTPFormer. Figure 4-4 and Figure 4-5 show some visualized examples of spatial attention maps and temporal attention maps for all layers in KTPFormer. The attention weights of different heads are averaged to observe the overall correlations of joints and frames, and the attention weights are normalized from 0 to 1. Additionally, Figure 4-6 presents visual comparisons of 3D pose estimation results between our KTPFormer and MixSTE (Zhang *et al.*, 2022b). The green circle highlights locations where we can achieve more accurate 3D pose estimations compared to MixSTE (Zhang *et al.*, 2022b). Furthermore, we collect several in-the-wild videos as an additional real-world test to validate the generalization ability of our method. As shown in Figure 4-7, our method demonstrates remarkable robustness and accuracy across the majority of frames in the wild videos, especially in challenging scenarios with severe occlusion and extremely fast movements.

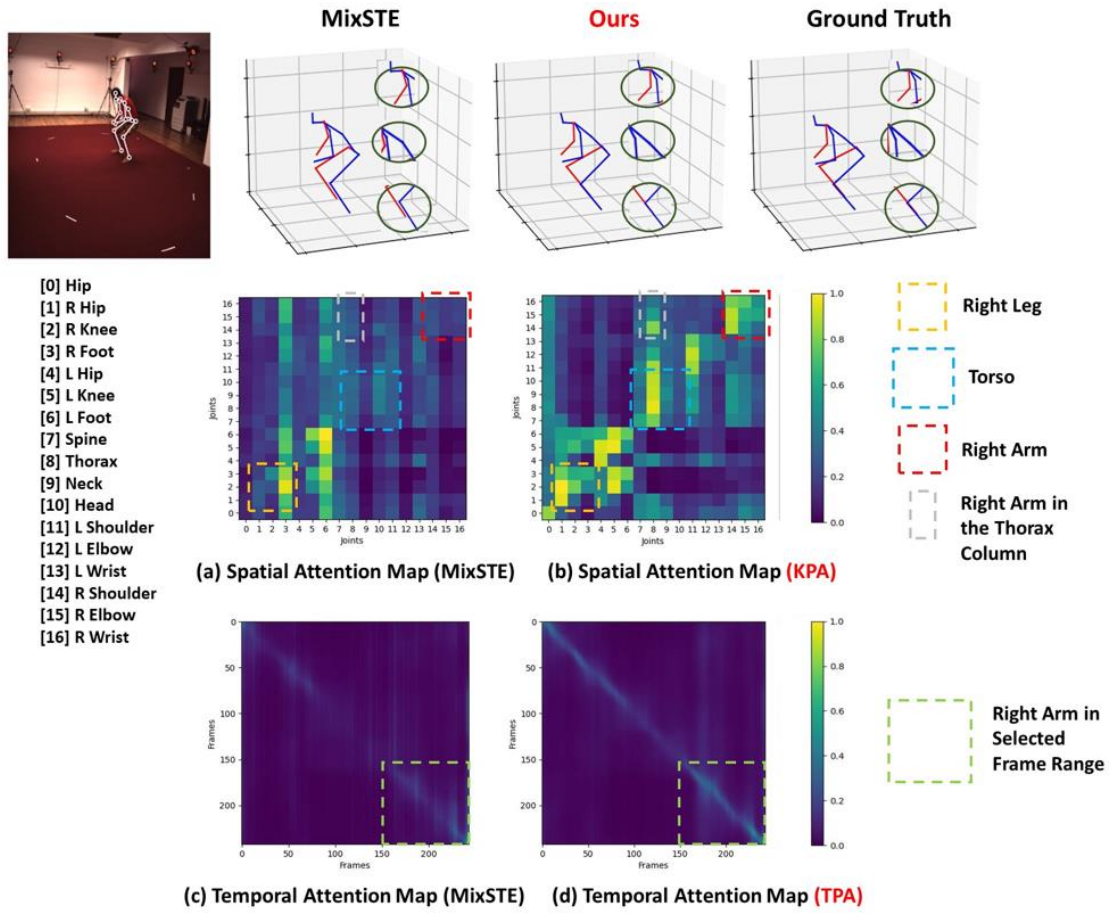


Figure 4-3 *Comparison of visualization results and attention maps between ours and MixSTE (Zhang et al., 2022b). The x-axis and y-axis correspond to the queries and the predicted outputs, respectively.*

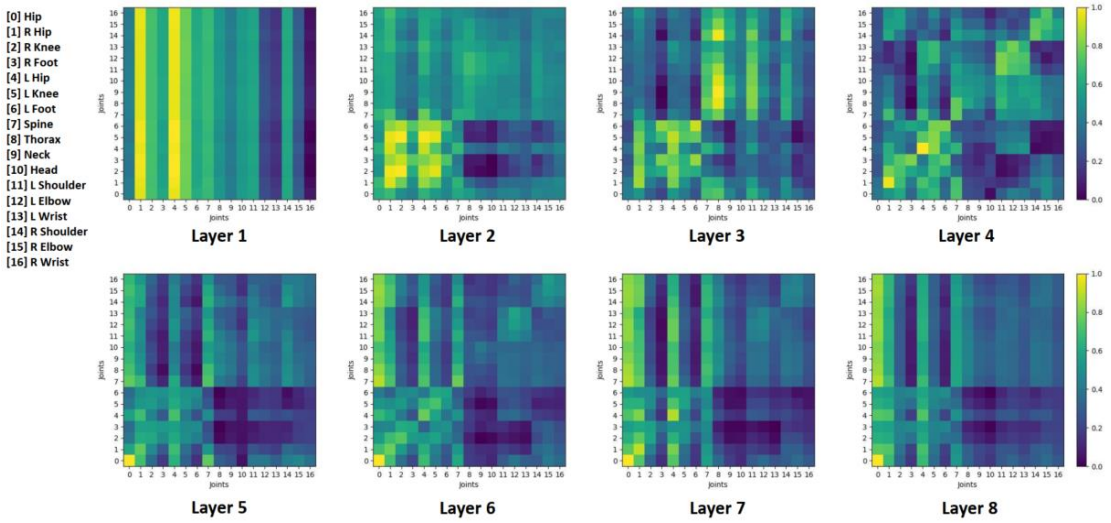


Figure 4-4 *Visualizations of attention maps from the spatial self-attention in KTPFormer. The x-axis and y-axis correspond to the joints queries and*

the predicted outputs, respectively. The attention weights are normalized from 0 to 1, and the lighter color indicates stronger attention.

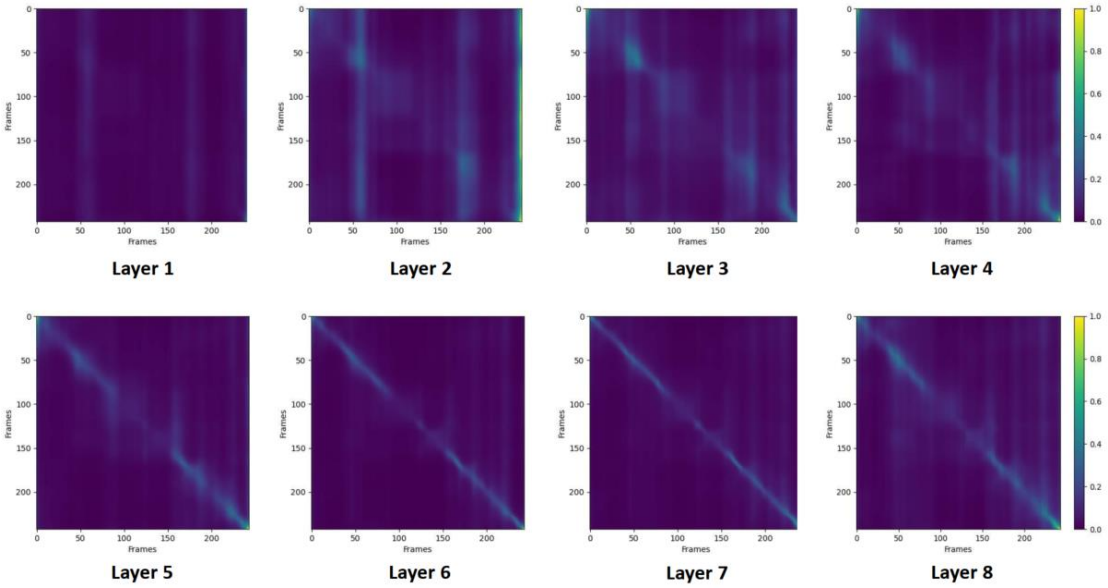


Figure 4-5 Visualizations of attention maps from the temporal self-attention in KTPFormer. The x-axis and y-axis correspond to the frames queries and the predicted outputs, respectively. The attention weights are normalized from 0 to 1, and the lighter color indicates stronger attention.

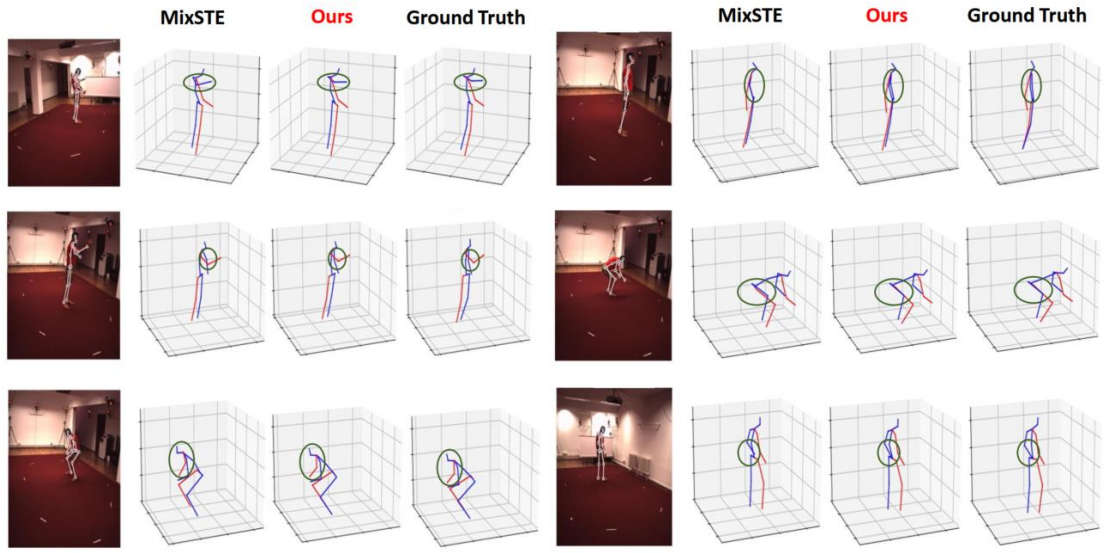


Figure 4-6 Visual comparisons of 3D pose estimation between MixSTE (Zhang et al., 2022b) and our KTPFormer on Human3.6M dataset. The green circle highlights locations where our KTPFormer yields better results.

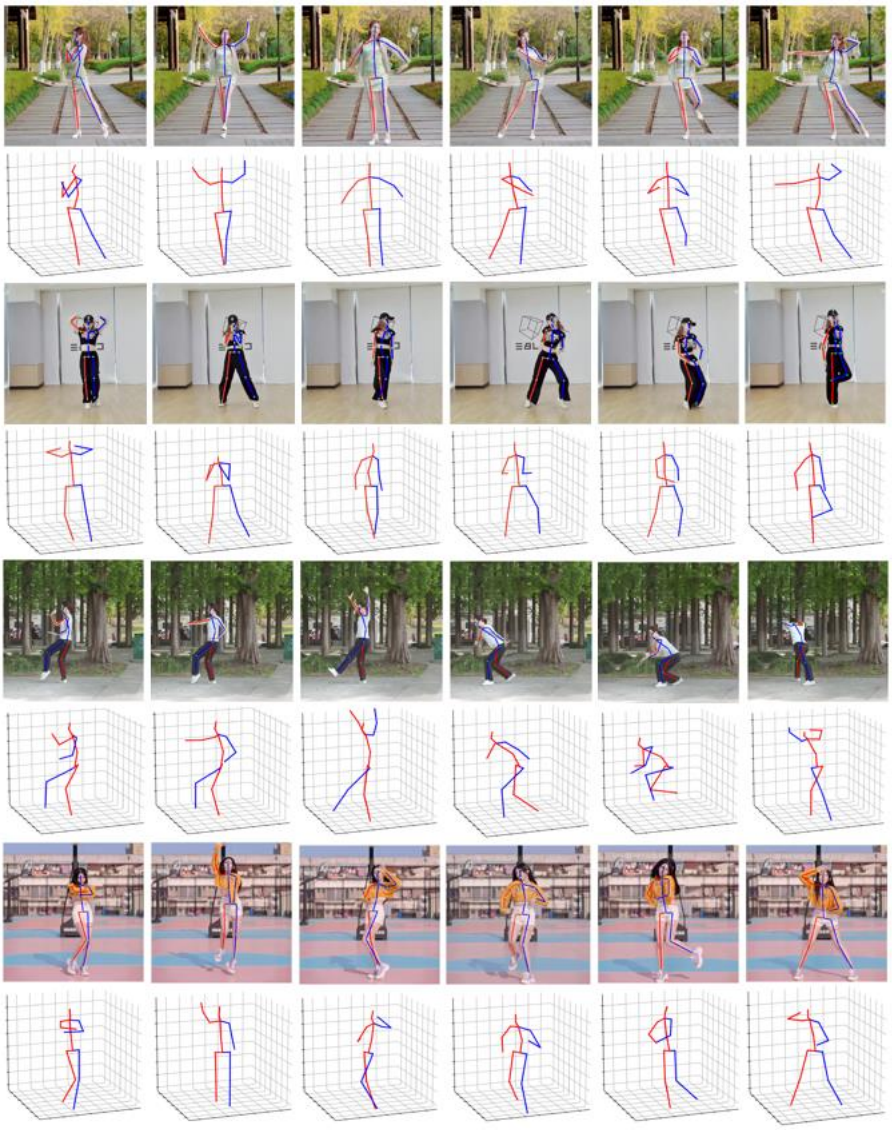


Figure 4-7 Some visualisation results of 3D pose estimation by our KTPFormer on in-the-wild videos.

4.3.6 Adaptable to Different 3D Pose Estimators

Ablation Study. Our KPA and TPA are generic and can be applied in various transformer-based 3D pose estimators. To verify the adaptability, we selected five transformer-based 3D pose estimators as backbones. We removed the linear embedding before the first spatial encoder and put the KPA in front of the first MHSA in these

models. We used the TPA to encode the features of different poses across frames on (Li *et al.*, 2022b; Li *et al.*, 2022c; Zheng *et al.*, 2021a) and different joints across frames on (Tang *et al.*, 2023b; Zhang *et al.*, 2022b). We trained these models on the Human3.6M dataset using 2D ground-truth poses as inputs. As shown in Table 4-10, our method brings about noticeable improvements in all the models in terms of MPJPE (mm), with very slight increases in the number of parameters and FLOPs, indicating that our KPA and TPA modules are lightweight and plug-and-play to different models for 3D pose estimation.

Table 4-10 Comparative results obtained with different 3D pose estimators trained with and without KPA and TPA modules on Human3.6M dataset.

Method	MPJPE (mm)	Parameters (M)	FLOPs (M)
Zheng *et al.* (2021a) (T=81)	31.3	9.558	815.522
+KPA+TPA	28.8$^{(-2.5)}$	9.560$^{(+0.02)}$	815.885$^{(+0.363)}$
Li *et al.* (2022b) (T=351)	28.5	3.979	801.093
+KPA+TPA	27.4$^{(-1.1)}$	3.980$^{(+0.01)}$	801.859$^{(+0.766)}$
Li *et al.* (2022c) (T=243)	30.9	24.767	4826.854
+KPA+TPA	28.8$^{(-2.1)}$	24.773$^{(+0.06)}$	4829.873$^{(+3.019)}$
Zhang *et al.* (2022b) (T=243)	21.6	33.650	139038.488
+KPA+TPA	19.0$^{(-2.6)}$	33.652$^{(+0.02)}$	139059.638$^{(+21.15)}$
Tang *et al.* (2023b) (T=81)	25.7	4.747	6535.219
+KPA+TPA	25.1$^{(-0.6)}$	4.748$^{(+0.01)}$	6541.565$^{(+6.346)}$

Implementation Details. We illustrate in detail on how our Kinematics Prior Attention (KPA) and Trajectory Prior Attention (TPA) are applied to different 3D pose estimators. Our TPA possesses the capability to not only model joint-to-joint motion trajectory across frames but also to model pose-to-pose motion trajectory across frames. Figure 4-8 shows the joint-to-joint and pose-to-pose motion trajectory topology. In Figure

4-8(b), TPA connects the different poses across consecutive adjacent frames to build the temporal local topology (pose-to-pose), including self-connection. Next, we exploit learnable vectors (dotted line) to connect the poses among all neighbouring and non-neighbouring frames to construct the simulated temporal global topology (pose-to-pose), which is equivalent to the computation of attention weights among all frames by the self-attention. Then, the two topologies are integrated together through the combination method identical to joint motion trajectory topology (Figure 4-8(a)), resulting in the pose motion trajectory topology. The pose motion trajectory topology (Figure 4-8(b)) is incorporated into the stacked TPA (pose) to encode the pose-to-pose features across frames for these works (Li *et al.*, 2022b; Li *et al.*, 2022c; Zheng *et al.*, 2021a). On the other hand, we introduce joint motion trajectory topology (Figure 4-8(a)) into the stacked TPA (joint) to learn joint-to-joint temporal information for other works (Tang *et al.*, 2023b; Zhang *et al.*, 2022b). Figure 4-9 depicts the framework overview of our KPA and TPA applied to different 3D pose estimators. For PoseFormer (Zheng *et al.*, 2021a), the KPA and the stacked TPA (pose) are placed ahead of the stacked spatial transformers and stacked temporal transformers, respectively. The model architecture of StridedTransformer (Li *et al.*, 2022b) with our method is similar to PoseFormer (Zheng *et al.*, 2021a). Hence, we have not depicted it. For MHFormer (Li *et al.*, 2022c), we employ the KPA to process the initial 2D pose sequence, generating Q, K and V vectors for the first spatial transformer. Then, we utilize three parallel stacked TPA (pose) blocks to encode the pose-to-pose temporal features for multiple

hypotheses, respectively. The three outputs from three stacked TPA (pose) blocks are fed into the next layer. In terms of STCFormer (Tang *et al.*, 2023b), the KPA and the stacked TPA (joint) blocks are positioned ahead of the spatial attention and temporal attention in parallel. They yield spatial and temporal Q, K and V vectors with priori knowledge for the spatial attention and temporal attention, respectively. For D3DP (Shan *et al.*, 2023), we employ two KPA blocks to concurrently process the 2D pose sequence and noisy 3D pose sequence, subsequently concatenating the output fea tures and feeding them into the spatial transformer. Then, the stacked TPA (joint) blocks are placed between the spatial transformer and temporal transformer. D3DP (Shan *et al.*, 2023) adopts the MixSTE (Zhang *et al.*, 2022b) as the denoiser, so the model architecture of MixSTE (Zhang *et al.*, 2022b) with our method is similar to D3DP (Shan *et al.*, 2023).

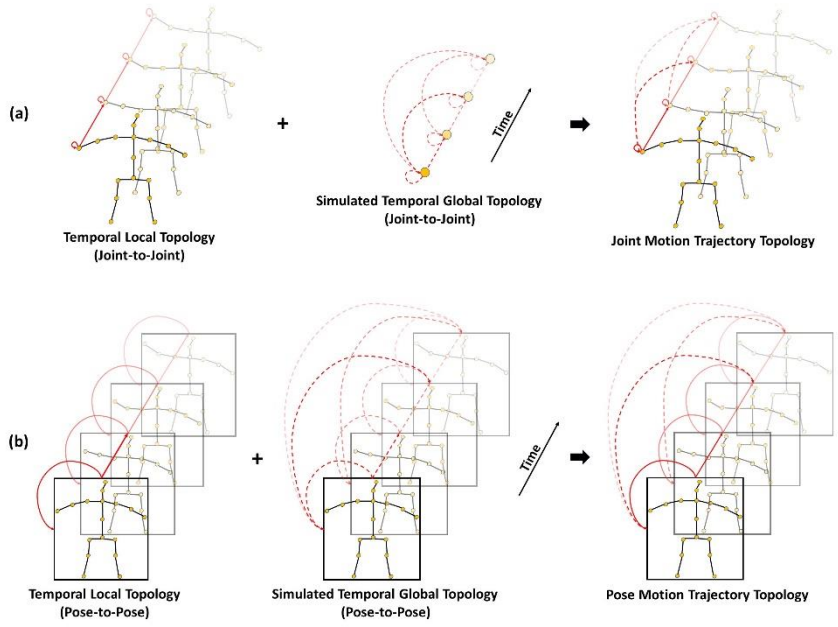


Figure 4-8 Overview of different motion trajectory topology. (a) The temporal local topology (joint-to-joint) plus the simulated temporal global topology

(joint-to-joint) to form the joint motion trajectory topology. (b) The temporal local topology (pose-to-pose) plus the simulated temporal global topology (pose-to-pose) to form the pose motion trajectory topology.

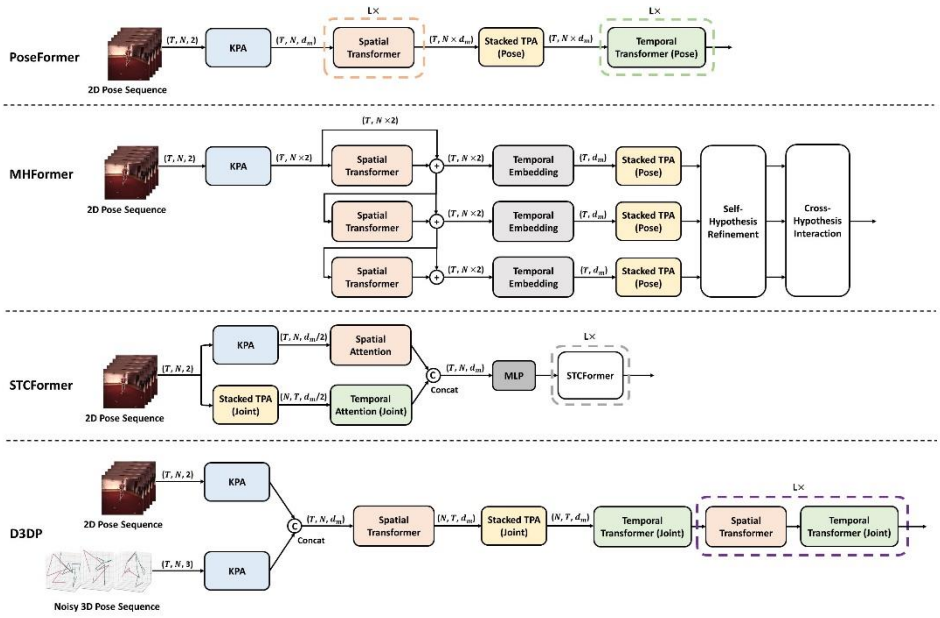


Figure 4-9 The framework overview of our KPA and TPA applied to different 3D pose estimators. The stacked TPA indicates that two TPA blocks are stacked with a residual connection. In terms of PoseFormer (Zheng et al., 2021a) and MHFormer (Li et al., 2022c), we use the stacked TPA (pose) to model temporal correlations between poses across frames. In contrast, the stacked TPA (joint) is utilized to encode the temporal features between joints across frames for STCFormer (Tang et al., 2023b) and D3DP (Shan et al., 2023).

Enhanced Attention Maps. We visualize the enhanced attention maps of (Li *et al.*, 2022c; Tang *et al.*, 2023b; Zheng *et al.*, 2021a) after applying our KPA and TPA on Human3.6M, to validate the effectiveness of our method. Figure 4-10 illustrates enhanced spatial and temporal attention maps from PoseFormer (Zheng *et al.*, 2021a), MHFormer (Li *et al.*, 2022c) and STCFormer (Tang *et al.*, 2023b), by integrating our

KPA and TPA into their networks. In terms of spatial attention maps, our KPA enhances attention weights between certain joints based on human anatomical structures and kinematic relationships, facilitating the explicit representation of human body topological relationships in the attention maps. On the other hand, our TPA enhances the temporal correlations between adjacent frames based on the motion trajectories of poses or joints in MHFormer (Li *et al.*, 2022c) and STCFormer (Tang *et al.*, 2023b). In particular, our TPA enhances attention weights between the frames of central region and other frames in PoseFormer (Zheng *et al.*, 2021a), recognizing the periodic nature of human motion in videos.

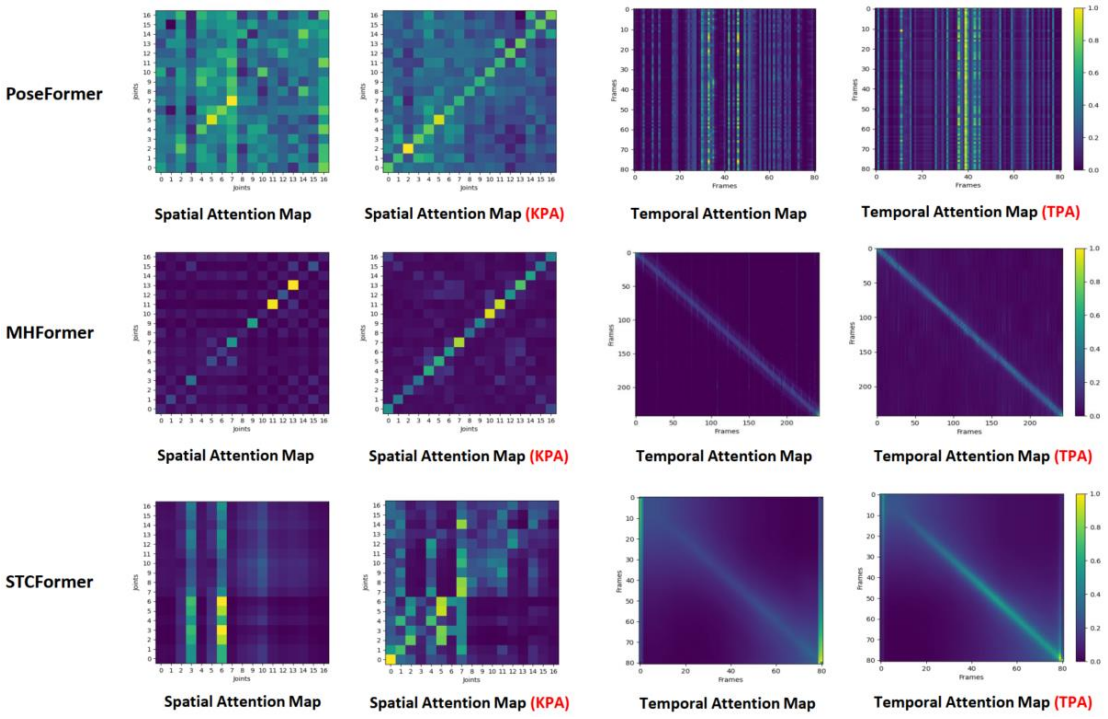


Figure 4-10 Visualizations of enhanced spatial and temporal attention maps by our KPA and TPA. The x-axis and y-axis correspond to the queries and the predicted outputs, respectively. The attention weights are normalized from 0 to 1, and the lighter color indicates stronger attention.

4.4 Chapter Summary

In this chapter, a Kinematics and Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer) has been proposed and developed, which introduces two novel prior attention mechanisms, KPA and TPA, for 3D pose estimation. Specifically, KPA constructs a kinematics topology to inject the kinematics prior knowledge into spatial tokens. TPA incorporate the prior information of joint motion trajectory into temporal tokens. The two prior attention mechanisms can enhance the capabilities of modeling global correlations in the self-attention mechanisms effectively. Experimental results on three benchmarks demonstrate that our method is effective in improving the performance with only a very small increase in the number of parameters and computation. Furthermore, the KPA and TPA can be integrated with various transformer-based 3D human pose estimators as lightweight plug-and-play modules. While the preceding two methods taking videos as input can achieve excellent results, there are occasions when only data of single images are available. To broaden the scope of application, another novel network that takes single images as input to predict 3D poses will be presented in next Chapter.

CHAPTER 5. A CROSS-FEATURE INTERACTION NETWORK

5.1 Introduction

Recently, the graph convolutional networks (GCNs) have been widely used for single-frame based 3D human pose estimation with the outstanding performance. These GCN-based methods (Liu *et al.*, 2020b; Zhao *et al.*, 2019a; Zou & Tang, 2021) utilize the topological information of the human skeleton by aggregating feature representations of the neighbouring body joints. However, these methods (Liu *et al.*, 2020b; Zhao *et al.*, 2019a; Zou & Tang, 2021) focus only on modeling the motion characteristics of adjacent or connecting joints, namely the local information. There are additional implicit kinematic information between joints that are not physically connected. For example, in the action of 'walking a dog', the joints of two hands and two feet move in the same direction along the dog's motion. In order to better capture the global information of human skeleton representations, some transformer-based methods (Li *et al.*, 2023; Li *et al.*, 2022c; Zhang *et al.*, 2022b; Zheng *et al.*, 2021a) are proposed. By exploiting the self-attention mechanism, these methods model the spatial dependencies among all body joints. In addition, some studies (Peng *et al.*, 2024; Zhao *et al.*, 2022; Zhu *et al.*, 2021) combine GCNs and transformer architectures to facilitate the learning of spatial correlations in human skeleton. However, all of them utilize GCNs and transformer blocks in a sequential manner, either by using the output of GCNs as the input for a transformer block, or vice versa. The resulting features from GCNs and

transformers lack direct interaction, which may limit the model's capability and performance, preventing it from fully leveraging the strengths of both components.

In order to address the aforementioned issue, we propose a novel Cross-Feature Interaction (CFI) Network to effectively enhance the learning of spatial representations of human skeleton. Figure 5-1 shows the schematic architecture of our method. As shown, we capture the local and global features by GCNs and self-attention mechanisms, respectively. We also obtain the initial 2D pose features by patch embedding (linear layer). The initial features, often neglected by other methods, can serve as an residual connection, to effectively compensate for the information loss that occurs during the layer-to-layer propagation of the other two types of features. Then, we design a specific multi-head cross-attention (MHCA) to facilitate cross-feature interaction among three different features, namely the local features, global features, and the initial 2D pose features. This specially designed MHCA, named as cross-feature interaction (CFI) module, can effectively model dependencies between multiple features and enable the other two features to complement the features of the current branch. Next, these three types of features derived from individual CFI modules are aggregated to form the enhanced spatial features. Finally, we develop a graph-enhanced module (GraMLP) with parallel structure of GCN and multi-layer perceptron (MLP) to incorporate the human skeletal knowledge as an inductive bias into the final representation of 3D pose. The key contributions of this paper are summarized as follows:

- We develop a novel Cross-Feature Interaction Network to effectively enhance the learning of spatial representations for 3D poses.
- A cross-feature interaction (CFI) module is designed to effectively model dependencies among local features, global features, and the initial features, which are further aggregated as enhanced spatial features.
- A graph-enhanced module 'GraMLP' is introduced to integrate vanilla MLP with graph convolutional network (GCN), improving the accuracy of 3D pose estimation.
- Extensive experiments on two benchmarks (Human3.6M and MPI-INF-3DHP) show that our method outperforms other SOTA models.

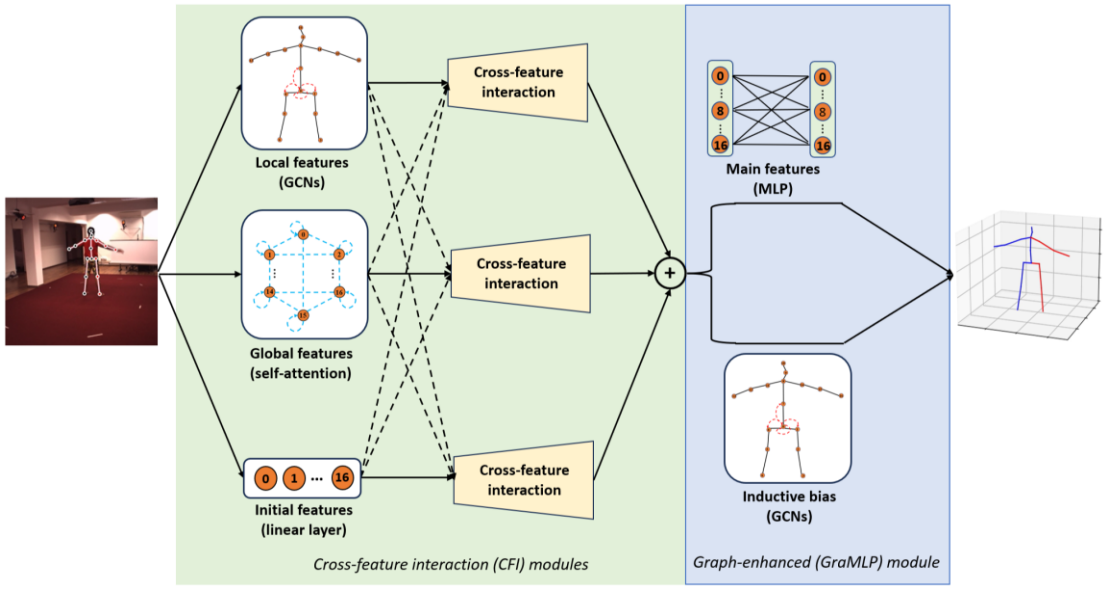


Figure 5-1 Schematic architecture of the proposed method.

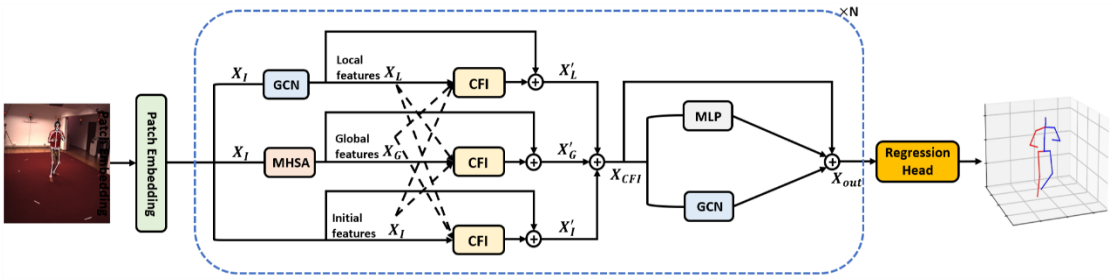


Figure 5-2 An overview of Cross-Feature Interaction Network.

5.2 Method

5.2.1 Preliminary

We first provide a brief overview of Graph Convolutional Networks (GCN) and multi-head self-attention (MHSA).

Graph Convolutional Network (GCN). A graph can be defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a collection of nodes and $\mathcal{E}$ is a set of edges. The representation of edges can be realized through an affinity matrix $A \in \{0,1\}^{N\times N}$, while the set of features of all nodes in the l-th layer can be expressed as a matrix $H_l \in \mathbb{R}^{D\times N}$. N is the number of nodes, and D represents the dimensionality of the features. The graph convolution operation aggregates features from neighboring nodes in the l-th layer following the equation below:

$$H_l = \sigma\left(W_l H_{l-1}\tilde{A}\right) \qquad (5-1)$$

Where $W_l \in \mathbb{R}^{D\times D}$ is the learnable weight matrix, $\tilde{A} = A + I_N$ refers to the adjacency matrix of the graph with the inclusion of self-connections, and I_N is the identity matrix.

Multi-head Self-attention (MHSA). The MHSA computes multiple attention heads via self-attention in parallel. Each attention head ($i = 1, \ldots, h$) is computed as:

$$head_i = Softmax\left(\frac{\left(ZW_i^Q\right)(ZW_i^K)^T}{\sqrt{d_m}}\right)(ZW_i^V) \qquad (5-2)$$

where $Z \in \mathbb{R}^{N\times D}$ is the input token, W_i^Q, W_i^K and $W_i^V \in \mathbb{R}^{D\times D}$ are learnable parameters. All h attention heads are then concatenated together, followed by a linear transformation, to form the output as follows:

$$Z_{MHSA} = Concat(head_1, \dots, head_i, \dots, head_h) \quad (5-3)$$

5.2.2 Cross-Feature Interaction

Figure 5-2 illustrates the proposed Cross-Feature Interaction Network, which consists of two main components of Cross-Feature Interaction module (CFI) and graph-enhanced module (GraMLP). The input 2D pose joints are initially embedded into high-dimensional tokens, denoted as the initial features $X_I \in \mathbb{R}^{N \times D}$. N is the number of joints, and D is the dimensionality of the features. The initial features X_I is then fed into the GCN, yielding the local features $X_L \in \mathbb{R}^{N \times D}$:

$$X_L = \sigma\left(W X_I \tilde{A}\right) \quad (5-4)$$

where $\tilde{A}$ denotes the adjacency matrix of anatomical relationships in the human body. We obtain the global features $X_G \in \mathbb{R}^{N \times D}$ by eq. (5-3) and each head resulted from feeding initial features X_I to the MHSA:

$$head_i^G = Softmax\left(\frac{\left(X_I W_i^Q\right)(X_I W_i^K)^T}{\sqrt{d_m}}\right)(X_I W_i^V) \quad (5-5)$$

To facilitate communication and achieve mutual complementarity among the three types of features, we introduce a **cross-feature interaction** module, a specific multi-head cross attention (see Figure 5-3). The initial features X_I, local features X_L and global features X_G are regarded as queries, keys, and values, respectively, and fed into the CFI unit as follows:

$$head_i = Softmax\left(\frac{\left(X_I W_i^Q\right)(X_L W_i^K)^T}{\sqrt{d_m}}\right)(X_G W_i^V) \quad (5-6)$$

The enhanced global features $X_G' \in \mathbb{R}^{N \times D}$ can be obtained by:

$$X'_G = Concat(head_1, \dots, head_i, \dots, head_h) + X_G \qquad (5-7)$$

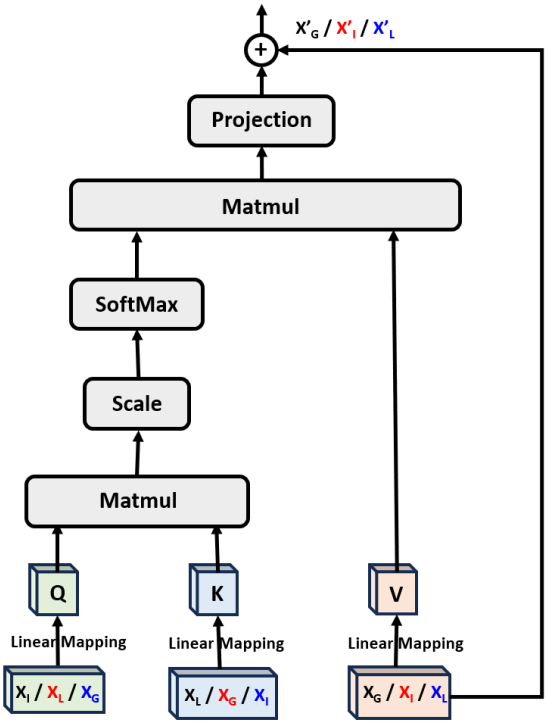


Figure 5-3 Cross-feature interaction module (CFI).

By eq. (5-6), the three types of features engage in interactions and exchange information with each other. The global features can compensate for the limited receptive field of GCN, providing additional implicit kinematic knowledge to the local features. Also, the initial features can offer valuable information that may be lost during the process of feature aggregation by GCN from neighbouring joints. Moreover, the residual term in eq. (5-7) ensures that primary focus of the current branch, namely, the global features. Similarly, we employ the CFI module to obtain the enhanced local features $X'_L \in \mathbb{R}^{N\times D}$ and initial features $X'_I \in \mathbb{R}^{N\times D}$.

Next, the enhanced features X'_G, X'_L and X'_I are sum up to form as the output sequence from CFI module, also the input for the GraMLP module:

$$X_{CFI} = X'_G + X'_L + X'_I \qquad (5-8)$$

5.2.3 GraMLP

The MLP structure in a vanilla transformer is densely connected, which has limited ability to model topological structure information of human skeleton. To inject the human skeleton information into the final 3D pose, we introduce a parallel design of MLP and GCN, namely GraMLP. Considering that the MLP can introduce non-linearities to the input features, by adding GCN in parallel can retain anatomical knowledge of the human body, serving as an inductive bias to enhance the representation of 3D pose. In general, the GraMLP processes the features from the CFI module as follows:

$$X_{out} = X_{CFI} + MLP(X_{CFI}) + GCN(X_{CFI}) \qquad (5-9)$$

where $MLP(\cdot)$ is composed of the linear layer and the GELU activation function. $GCN(\cdot)$ refers to the equation (5-4).

5.2.4 Regression Head

The linear layer is used as a regression head to predict the 3D joint coordinates of the single pose. The loss function for our network is given as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|\tilde{J}_i - J_i\right\|_2^2\right) \qquad (5-10)$$

where $\tilde{J}_i \in \mathbb{R}^{N\times 3}$ and $J_i \in \mathbb{R}^{N\times 3}$ denotes the predicted and ground truth 3D joint coordinates, respectively.

5.3 Experiments

5.3.1 Datasets and Evaluation Metrics

Human3.6M. Human3.6M (Ionescu *et al.*, 2013) is an indoor scenes dataset with 3.6 million video frames. It has 11 professional actors, performing 15 actions under 4 synchronized camera views. Following previous work (Tang *et al.*, 2023b; Zhang *et al.*, 2022b), we used subjects 1, 5, 6, 7 and 8 for training, and subjects 9 and 11 for testing. We use the mean per-joint position error (MPJPE) as the evaluation metric, which is the average Euclidean distance in millimetres (mm) between the predicted and the ground-truth 3D joint coordinates.

MPI-INF-3DHP. MPI-INF-3DHP (Mehta *et al.*, 2017) is also a public large-scale dataset. Following the setting of (Tang *et al.*, 2023b; Zhang *et al.*, 2022b), we use the area under the curve (AUC), percentage of correct keypoints (PCK) as evaluation metrics.

5.3.2 Implementation Details

We implemented our method in the Pytorch framework on one GeForce RTX 3090 GPU. The Graph-Attention Cross-Feature Interaction Network loops $N = 3$ times. Following (Zhao *et al.*, 2019a; Zou & Tang, 2021), the input 2D keypoints are detected by 2D pose detector (Chen *et al.*, 2018) or 2D ground truth. During the training stage, we use the Adam (Kingma & Ba, 2014) optimizer to train our model for 20 epoch. The learning rate is initialized to 0.001 and decayed by 0.95 per epoch. The channel size is set to 512 and the number of heads is set to 8 in the network.

5.3.3 Comparison with State-of-the-Art Methods

Human3.6M. Table 5-1 compares the single-image estimation accuracy of our method with existing SOTA methods using 2D poses detected by CPN (Chen *et al.*, 2018) as inputs. As shown, our method outperforms other SOTA models and achieve the same performance of 49.4mm of MPJPE as MGCN (Zou & Tang, 2021) which adopts the refinement module (Cai *et al.*, 2019b). After applying the refinement module (Cai *et al.*, 2019b) to our model, the performance is improved from 49.4mm to 48.6mm, surpassing MGCN (Zou & Tang, 2021) by 0.8mm error reduction. Moreover, our method obtains the best results of 38.8mm and 38.7mm in terms of P-MPJPE. As shown in Table 5-2, we compare our results with those SOTA methods using 2D ground-truth poses as inputs. Our method attains SOTA performance, validating the effectiveness of our method for different types of input.

Table 5-1 *Quantitative comparisons with SOTA methods based on Human3.6M under MPJPE (mm) and P-MPJPE (mm) with 2D poses detected by CPN (Chen et al., 2018) as inputs. ∗ denotes using the refinement module (Cai et al., 2019b). † indicates the transformer-based methods. Best results are shown in* ***bold****.*

MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Martinez *et al.* (2017a)	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
Zhao *et al.* (2019a)	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
Ci *et al.* (2019b)	46.8	52.3	**44.7**	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Xu and Takano (2021)	45.2	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
Zhao *et al.* (2022) †	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
Cai *et al.* (2019b) ∗	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	53.5	39.2	41.2	50.6
Li *et al.* (2023) †	47.9	50.0	47.1	51.3	51.2	59.5	48.7	46.9	**56.0**	61.9	51.1	48.9	54.3	40.0	42.9	50.5
Zeng *et al.* (2020)	**44.5**	**48.2**	47.1	**47.8**	51.2	56.8	50.1	45.6	59.9	66.4	52.1	**45.3**	54.2	39.1	40.3	49.9
Zou and Tang (2021) ∗	45.4	49.2	45.7	49.4	50.4	58.2	47.9	46.0	57.5	63.0	**49.7**	46.6	52.2	38.9	40.8	49.4
Ours†	45.4	49.5	46.1	49.3	51.7	56.7	**47.3**	**44.6**	58.6	63.0	50.4	47.2	51.8	38.2	41.3	49.4
Ours†∗	45.0	50.3	45.8	48.4	**49.7**	**55.8**	**47.3**	45.4	56.4	59.4	49.9	46.5	50.9	**38.0**	**39.6**	**48.6**

P-MPJPE (CPN)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Martinez *et al.* (2017a)	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
Ci *et al.* (2019b)	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
Liu *et al.* (2020b)	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
Cai *et al.* (2019b) ∗	36.8	38.7	38.2	41.7	40.7	46.8	37.9	35.6	47.6	51.7	41.3	36.8	42.7	31.0	34.7	40.2
Zeng *et al.* (2020)	35.8	39.2	36.6	**36.9**	39.8	45.1	38.4	36.9	47.7	54.4	38.6	36.3	39.4	30.3	35.4	39.4
Zou and Tang (2021) ∗	35.7	38.6	36.3	40.5	**39.2**	44.5	37.0	35.4	46.4	51.2	40.5	35.6	41.7	30.7	33.9	39.1
Ours†	**35.3**	**37.8**	36.8	40.1	40.1	**43.6**	36.2	**34.3**	46.4	50.2	40.8	35.6	41.1	30.0	34.0	38.8
Ours†∗	35.5	38.1	**35.9**	40.4	39.9	43.7	**36.0**	34.7	**46.1**	**48.4**	40.5	35.7	41.3	30.2	33.7	**38.7**

Table 5-2 *Quantitative comparisons on Human3.6M under MPJPE. The input is the ground-truth 2D pose. ∗ denotes using the refinement*

*module (Cai et al., 2019b). † indicates the transformer-based methods. Best results are shown in **bold**.*

Method (GT)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Pur.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg
Martinez *et al.* (2017a)	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Zhao *et al.* (2019a)	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	42.2	53.5	44.3	40.5	47.3	39.0	43.8
Cai *et al.* (2019b) ∗	33.4	39.0	33.8	37.0	38.1	47.3	39.5	37.3	43.2	46.2	37.7	38.0	38.6	30.4	32.1	38.1
Zhu *et al.* (2021) †	37.2	42.2	32.6	38.6	38.0	44.0	40.7	35.2	41.0	45.5	38.2	39.5	38.2	29.8	33.0	38.2
Liu *et al.* (2020b)	36.8	40.3	33.0	36.3	37.5	45.0	39.7	34.9	40.3	47.7	37.4	38.5	38.6	29.6	32.0	37.8
Zou and Tang (2021) ∗	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37.4
Zeng *et al.* (2020)	35.9	**36.7**	29.3	34.5	36.0	42.8	37.7	31.7	40.1	44.3	35.8	37.2	36.2	33.7	34.0	36.4
Xu and Takano (2021)	35.8	38.1	31.0	35.3	35.8	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	30.7	35.8
Zhao *et al.* (2022) †	32.0	38.0	30.4	34.4	34.7	43.3	**35.2**	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
Li *et al.* (2023) †	32.9	38.3	**28.3**	33.8	34.9	38.7	37.2	30.7	**34.5**	39.7	33.9	**34.7**	**34.3**	**26.1**	28.9	33.8
Ours†	35.4	38.7	29.8	34.8	33.6	**36.8**	39.8	30.9	36.6	**36.3**	34.9	37.6	34.4	28.3	30.4	34.6
Ours†∗	**29.1**	37.1	29.5	**31.8**	**33.2**	41.1	36.0	**29.8**	38.2	39.3	**33.3**	36.2	35.8	27.3	**28.6**	**33.7**

MPI-INF-3DHP. Table 5-3 reports the quantitative comparisons with state-of-the-art methods on cross-dataset scenarios. Our model was trained on the Human3.6M dataset and subsequently evaluated on the test set of the MPI-INF-3DHP dataset. The results show that our method achieves the best performance in all metrics, demonstrating the robustness of our method being applied to previously unseen scenarios.

Table 5-3 Quantitative comparisons with state-of-the-art methods on MPI-INF-3DHP test set.

Methods	PCK↑				AUC↑
	GS	noGS	Outdoor	All	
Martinez *et al.* (2017a)	49.8	42.5	31.2	42.5	17.0
Ci *et al.* (2019b)	74.8	70.8	77.3	74.0	36.7
Zeng *et al.* (2020)	-	-	80.3	77.6	43.8
Zhao *et al.* (2022)	80.1	77.9	74.1	79.0	43.8
Liu *et al.* (2020b)	77.6	80.5	80.1	79.3	47.6
Xu and Takano (2021)	81.5	81.7	75.2	80.1	45.8
Li *et al.* (2023)	**86.2**	84.7	81.9	84.1	53.7
Ours	85.0	**86.1**	**85.7**	**85.6**	**54.0**

5.3.4 Ablation Study

To verify the effectiveness of the proposed modules, we conducted ablation experiments on Human3.6M using 2D poses detected by CPN (Chen *et al.*, 2018) as inputs. Table 5-4 shows the results of the ablation study of each module in our method. The vanilla transformer network, composed of the MHSA and MLP, is utilized as our baseline. For consistency, the transformer network is stacked for 3 loops, resulting in an overall accuracy of 51.9mm MPJPE. The notation CFI(·) indicates the application of CFI module to feature representations of the said branch. For example, CFI(local) denotes the application of CFI module to the local features, i.e. eq. (5-8) only has one

component of X_L'. The results show that the application of three CFI modules, i.e., CFI(global), CFI(local) and CFI(initial), contribute 0.5mm, 0.7mm and 1.3mm, respectively, of error reduction. The incorporation of three CFI modules can result in 4.0% improvement of accuracy, decreasing the MPJPE from 51.9mm to 49.8mm. Table 5-4 also shows that the initial features play a crucial role in the interaction of local and global features, which brings the largest contribution of accuracy improvement. This is because the initial features processed by our CFI module can serve as an residual connection to effectively compensate for the information loss that occurs during the layer-to-layer propagation of the other two types of features. Lastly, by the introduction of the GraMLP module on top of three CFI modules, the estimation errors further drop 0.4mm, achieving 49.4mm of MPJPE. The ablation experiments demonstrate the effectiveness of each proposed module in our method.

Table 5-4 Results of ablation study of each module in our method on Human3.6M dataset.

CFI(initial) X_I'	**CFI(local) X_L'**	**CFI(global) X_G'**	**GraMLP**	MPJPE (mm)
				51.9
✓				50.6
	✓			51.2
		✓		51.4
✓	✓			50.2
✓		✓		50.5
	✓	✓		50.6
✓	✓	✓		49.8
✓	✓	✓	✓	**49.4**

5.3.5 Qualitative Results

We visualize the 3D pose estimation results to validate the efficacy of our method in comparison to MGCN (Zou & Tang, 2021). As shown in Figure 5-4, the green circle

highlights locations where we can achieve more accurate 3D pose estimations compared to MGCN (Zou & Tang, 2021). The predicted 3D pose of our method are closer to the ground truth 3D pose under different actions.

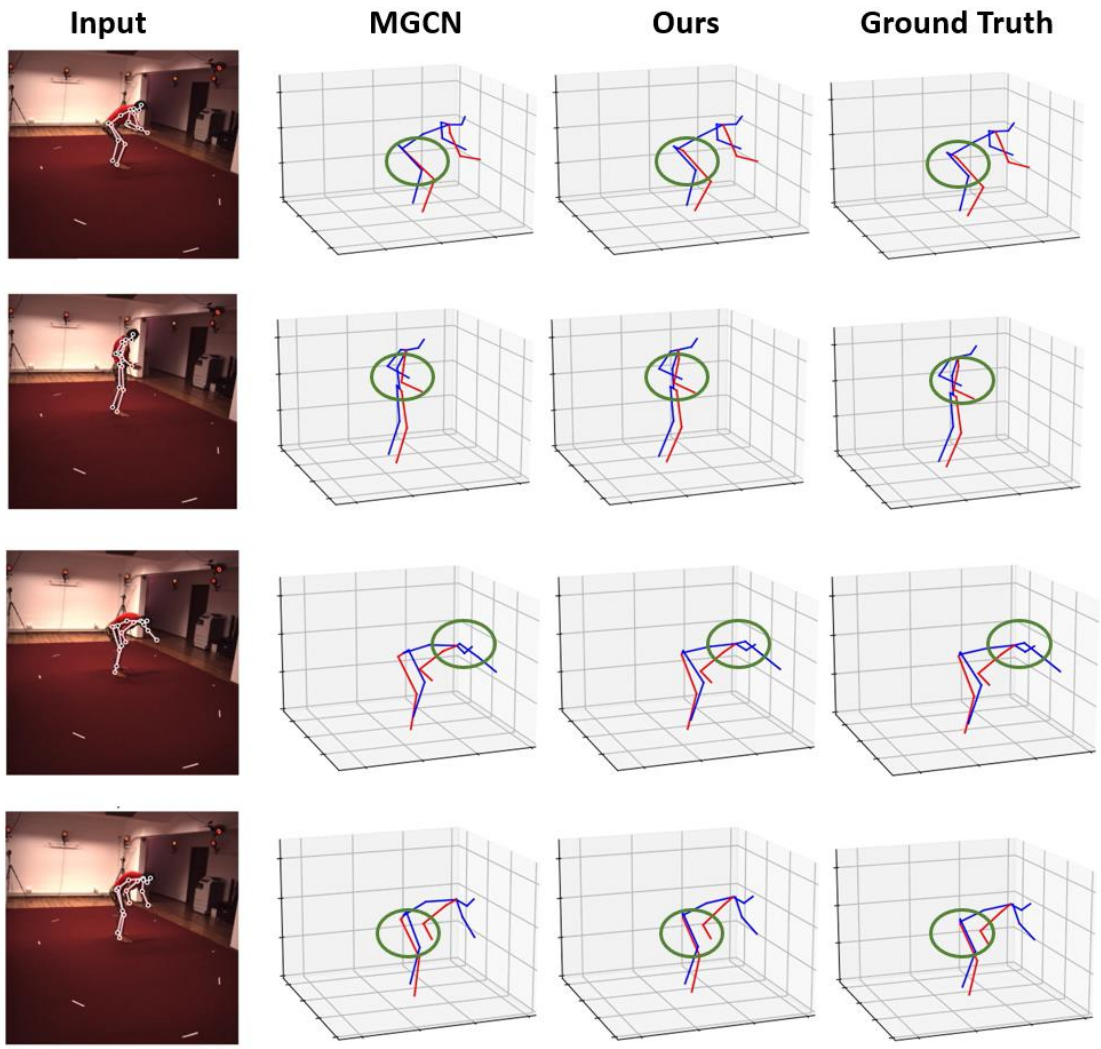


Figure 5-4 Qualitative comparisons with the MGCN (Zou & Tang, 2021) on Human3.6M dataset.

5.4 Chapter Summary

In this chapter, a Cross-Feature Interaction Network has been proposed and developed, which contains two core modules, the Cross-Feature Interaction (CFI) module and the parallel GCN and MLP (GraMLP) module. First of all, local features and global

features are extracted using GCN and MHSA, respectively. Then, the CFI module can facilitate communication and mutual complementation among three types of features (initial, local and global features). The GraMLP aggregates the preceding local features, global features, and initial features in a single layer, generating the final 3D pose. Experimental results on two benchmarks have demonstrated the effectiveness of this transformer-based method for 3D pose estimation based on single frames.

CHAPTER 6. FASHION APPLICATION

In this chapter, the 3D pose estimation method proposed in Chapter 4 is used to predict the 3D poses from video inputs, and the predicted 3 poses are then transferred to a target avatars with motion retargeting technique, animating personalized avatars for potential fashion application.

Motion retargeting involves the transfer of motion or movement from one source to another, commonly between characters or objects in computer graphics, animation, or virtual environments. It aims to implement motion data obtained from a source onto a distinct target, ensuring the preservation of a natural and physically plausible appearance in the results. Motion retargeting finds widespread application in diverse fields such as computer graphics, animation, video games, and virtual reality. It serves to efficiently repurpose existing motion data, enabling the creation of realistic animations for distinct characters or objects. This not only conserves time and resources but also elevates the overall quality of animations. Early optimization-based motion retargeting methods applied various additional constraints to ensure that the retargeted motion did not cause unnatural deformations or collisions with the environment, such as trajectory constraints (Feng *et al.*, 2012), kinematics constraints (Lee & Shin, 1999), dynamics constraints (Tak & Ko, 2005), joint angle constraints (Choi & Ko, 2000) and Euclidean distance (Bernardin *et al.*, 2017). Recently, some deep-learning-based motion retargeting methods have been proposed. Jang *et al.* (2018) proposed a motion retargeting system that integrated the Deep Convolution Inverse Graphics Network

(Kulkarni *et al.*, 2015) and U-Net (Ronneberger *et al.*, 2015) architectures to generate human motions. Villegas *et al.* (2018) designed a Recurrent Kinematics Network for motion retargeting in an unsupervised manner. Lim *et al.* (2019) introduced an unsupervised motion retargeting network to retarget the frame-by-frame pose and learn the movements of a character. Aberman *et al.* (2020) proposed a skeleton-aware motion retargeting framework to learn the skeleton's hierarchical structure and joint adjacency. Li *et al.* (2022a) utilized an iterative motion autoencoder to yield retargeted motions by an unsupervised method. Villegas *et al.* (2021) identified self-contacts and ground contacts from the skinned motion of the source character and preserved these contacts to the target motion by a latent-space optimization method. Zhang *et al.* (2023) presented a Residual RETargeting network with a skeleton-aware module and a shape-aware module to preserve inherent semantics of the source motion and comprehend the geometries of target characters.

Figure 6-1 illustrates the whole process from inputting a video to generating an animated avatar. Specifically, the YOLOv3 (Redmon & Farhadi, 2018) was first adopted to detect a single person in the video. Then, HRNet (Wang *et al.*, 2020a) was exploited to estimate the 2D pose from the detected person. Next, by applying any of the two methods described in Chapters 3 and 4, the 2D pose sequence inputs are lifted to the 3D pose. After obtaining the 3D coordinates of each joint for the 3D pose, these 3D coordinates are transformed into joint angles and the motion data are converted into bvh files. Finally, a motion retargeting method is then applied to bind the skeleton of

the bvh file to the avatar's skeleton, allowing the avatar to perform a series of motions that we predict from the video. Figure 6-2, Figure 6-3, Figure 6-4 and Figure 6-5 showcase some examples of application on animating personalized avatars. The accurate 3D poses predicted by the KTPFormer ensure that these avatars can perform the same coherent motions in situations of heavy occlusion and high-speed movement.

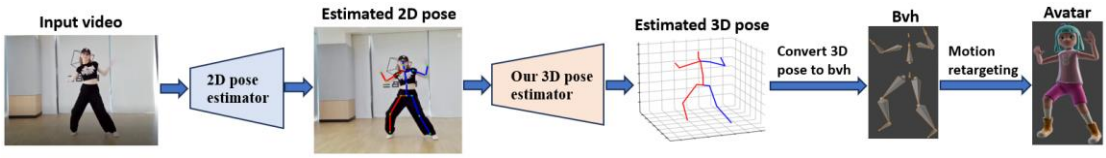


Figure 6-1 The whole process from inputting a video to generating an avatar.

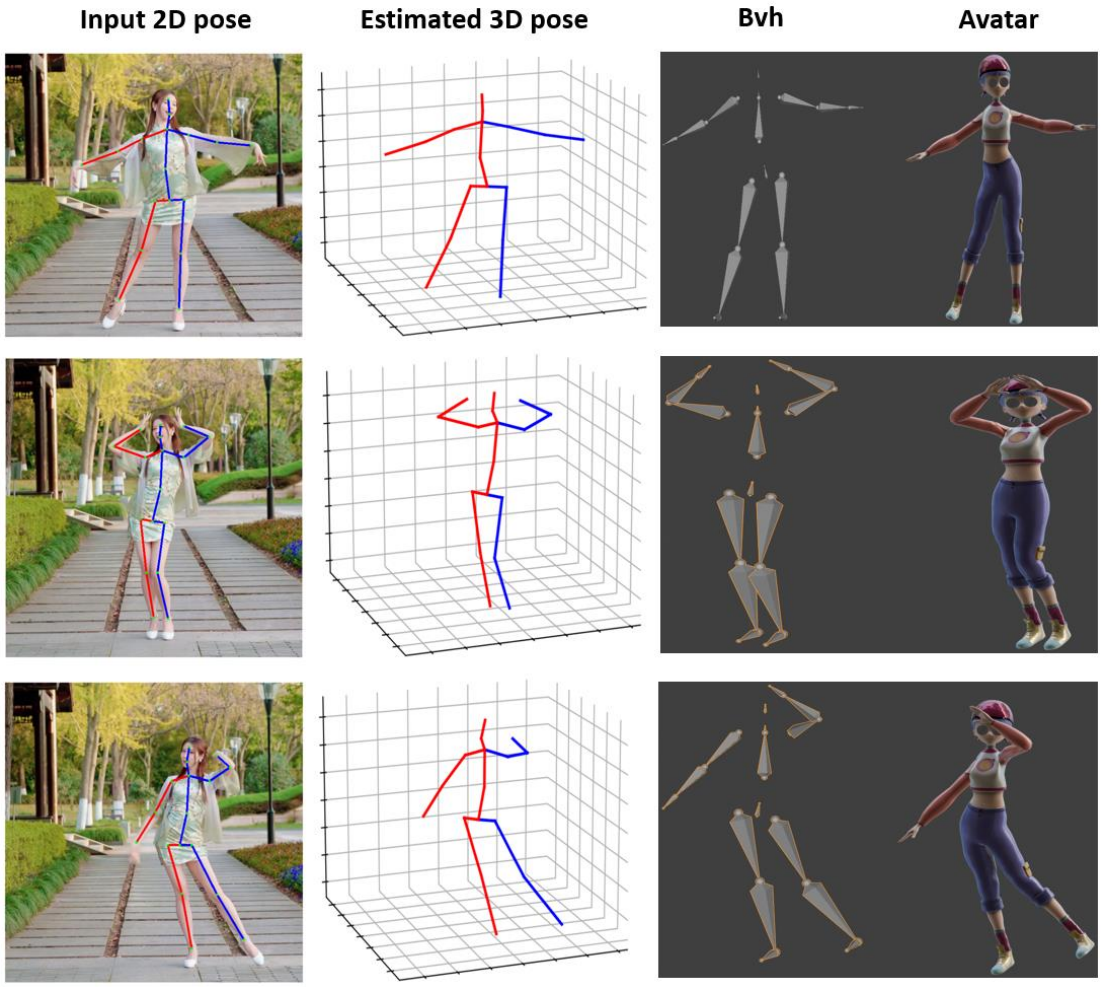


Figure 6-2 Examples of application on animating personalized avatars (a).

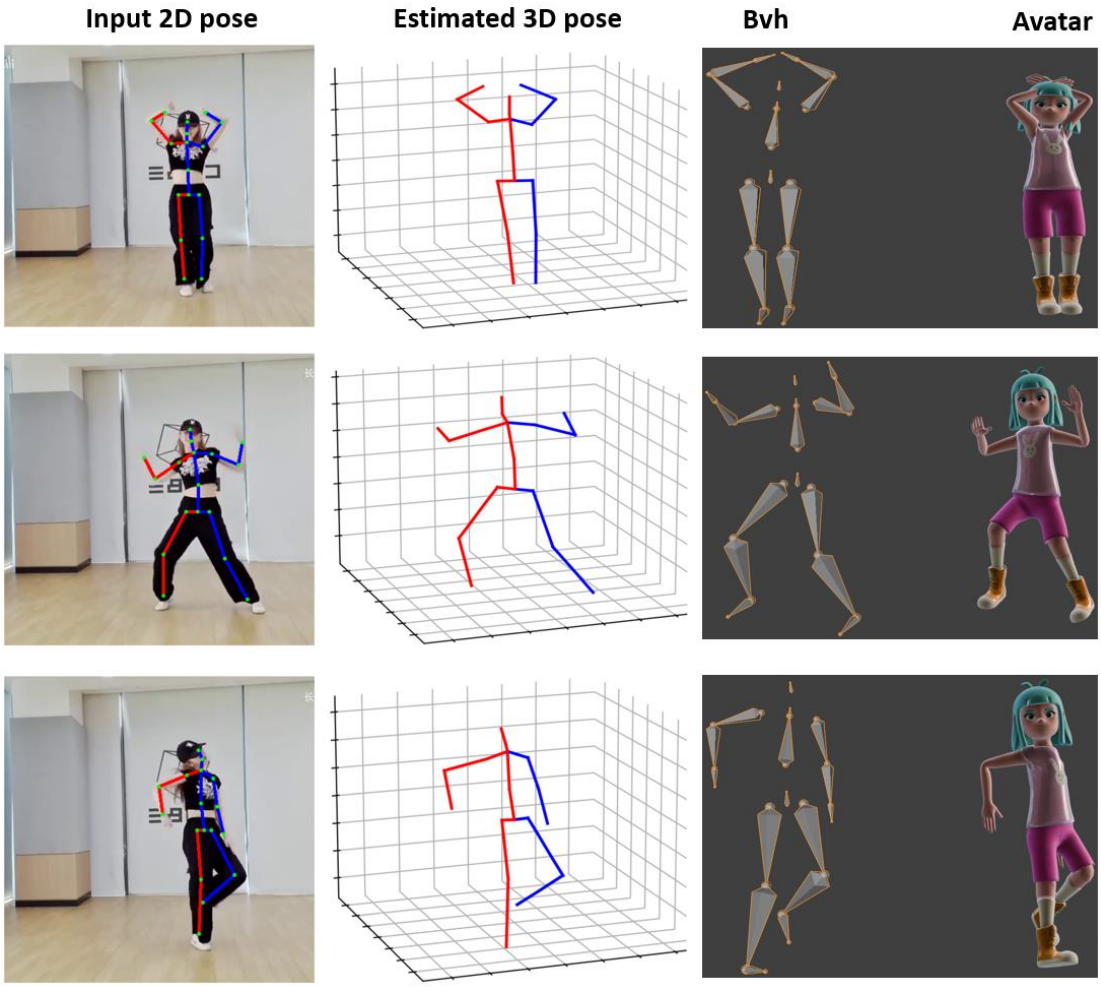


Figure 6-3 Examples of application on animating personalized avatars (b).

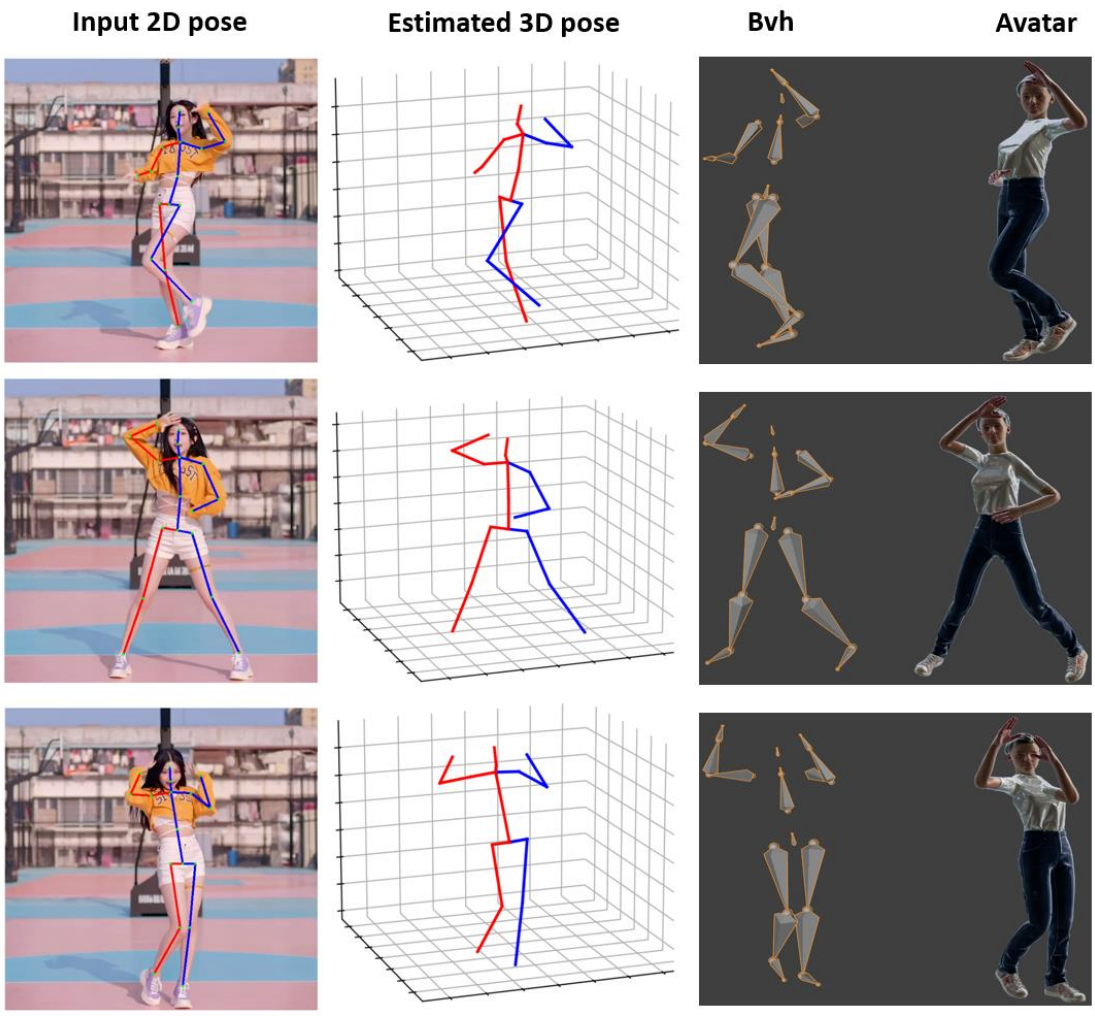


Figure 6-4 Examples of application on animating personalized avatars (c).

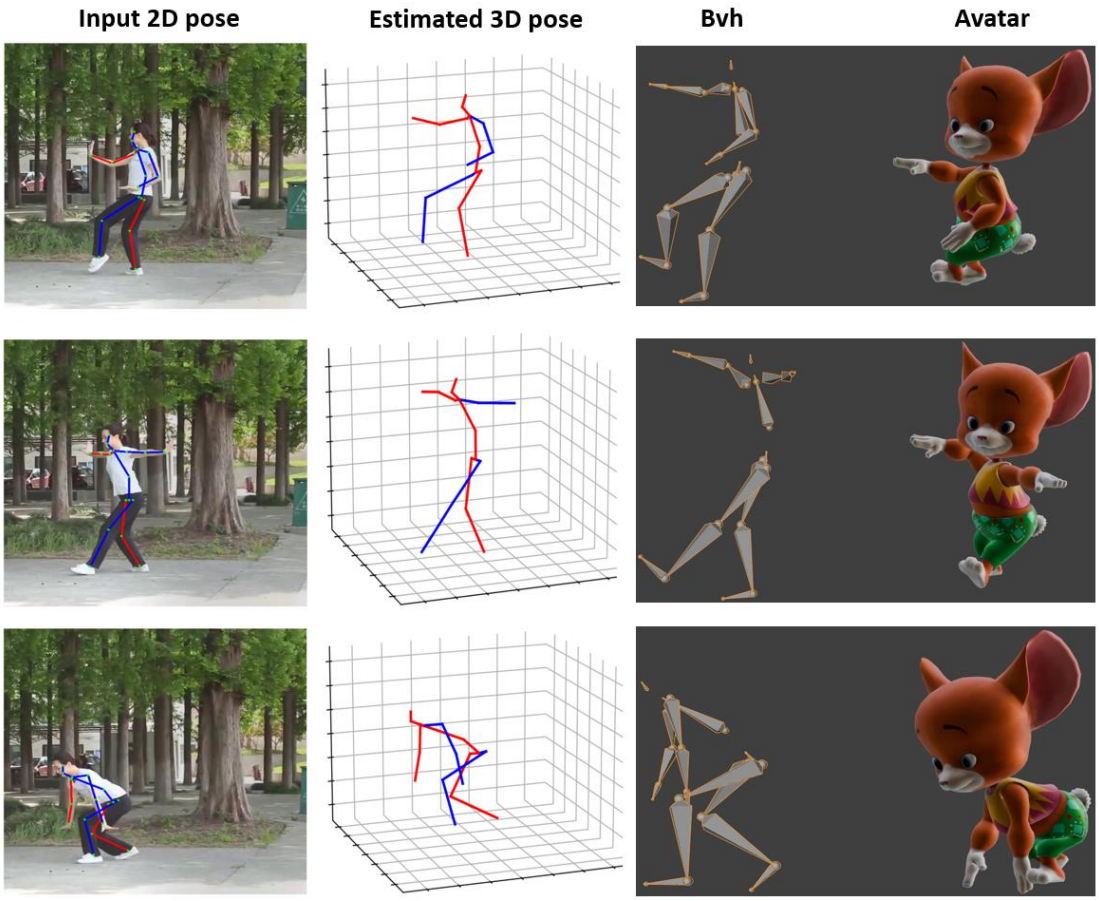


Figure 6-5 Examples of application on animating personalized avatars (d).

As a demonstration of potential fashion applications, a few motion videos and the results of predicted 3D poses and the retargeted avatars are shown at this link https://www.cafilab.com/?page_id=8147 (or QR code below).

CHAPTER 7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Conclusions

This thesis presents three novel 3D pose estimation methods and a potential fashion application of 3D pose estimation for virtual try-on catwalk animation. Among the three network developments, the first one was a CNN-based network with three encoding modules, including MAI, HFF and CIE, for grouped 3D pose estimation. It can be concluded that the motion amplitude information (MAI) and camera intrinsic embedding (CIE) modules can provide global information to the network and improve the accuracy of 3D pose estimation. Furthermore, the optimized feature fusion (HFF) module can significantly reduce model complexity while ensuring the accuracy of the model. Compared to a previous method (Shan *et al.*, 2021a), our method has used fewer parameters to fuse different groups of human pose features and also improves the performance. Moreover, a one-stage training scheme based on gradient detaching has been developed to train the grouped 3D human pose estimation network in an end-to-end manner, which can greatly reduce the number of training epochs and save training time with only a slight drop in accuracy in comparison to the multi-stage offline training strategy.

The second network is a transformer-based method, in which a Kinematics and

Trajectory Prior Knowledge-Enhanced Transformer (KTPFormer) is proposed for 3D pose estimation, which is integrated with two novel prior attention mechanisms. Specifically, KPA constructs a kinematics topology to inject the kinematics prior knowledge into spatial tokens. TPA incorporate the prior information of joint motion trajectory into temporal tokens. The two prior attention mechanisms can enhance the capabilities of modeling global correlations in the self-attention mechanisms effectively. Experimental results on three benchmarks demonstrate that our method is effective in improving the performance with only a very small increase in the number of parameters and computation. Furthermore, the KPA and TPA can be integrated with various transformer-based 3D human pose estimators as lightweight plug-and-play modules.

Lastly, a Graph-Attention Cross-Feature Interaction Network has been developed for 3D human pose estimation based on single frames. This method utilizes the Cross-Feature Interaction (CFI) module to facilitate the exchange of information among three distinct features (initial, local and global features), thereby mutually enhancing each individual feature. Then, the parallel design of GCN and MLP (GraMLP) is proposed to fuse these three features more effectively. The additionally introduced GCN retains anatomical knowledge of the human body, serving as an inductive bias to enhance the learning of local features. Experimental results on two public datasets have demonstrated that the proposed method significantly outperforms other 3D pose estimation methods based upon single frames.

7.2 Recommendations for Future Work

Although the proposed three methods demonstrate high accuracy on different datasets of 3D human pose estimation, further research is suggested to explore advanced 3D human pose estimation methods.

The grouped 3D pose estimation algorithm primarily employs 1D convolutional operations. Recently, many research works has demonstrated that transformers, in terms of performance improvement, surpass traditional convolutional operations. Thus, we will investigate the encoding of joints within distinct body groups using a self-attention mechanism. Furthermore, encoding camera intrinsics into the network can enhance performance in the camera coordinate system, in practical applications, the representation of 3D poses is commonly presented in the world coordinate system. In the future, we will explore effective utilization of camera extrinsics to enhance the accuracy of 3D human pose estimation within the world coordinate system.

Recently, multimodal approaches have been extensively researched in the field of computer vision. We will endeavour to encode the names of actions as textual information into the network to effectively boost the performance of 3D human pose estimation. Our future research will focus on leveraging the attention mechanism in transformers to encode textual information related to actions, exploring how transformed attention can effectively capture and incorporate action-specific details.

We will concurrently apply this multimodal information encoding approach to both video-based and single-frame 3D human pose estimation.

In terms of motion retargeting, we will explore novel methods to achieve effective motion retargeting with limited training data, such as through transfer learning, meta-learning, or other techniques that make the retargeting models more robust and generalizable. Also, we will design a network to jointly train 3D human pose estimation and motion retargeting, enabling the mutual enhancement of performance between the two tasks. In this way, our avatar will be capable of performing more lifelike movements in the field of fashion.

REFERENCES

Aberman, K., Li, P., Lischinski, D., Sorkine-Hornung, O., Cohen-Or, D., & Chen, B. (2020). Skeleton-aware networks for deep motion retargeting. *ACM Transactions on Graphics (TOG)*, *39*(4), 62: 61-62: 14.

Arbeláez, P., Maire, M., Fowlkes, C., & Malik, J. (2010). Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*.

Bernardin, A., Hoyet, L., Mucherino, A., Gonçalves, D., & Multon, F. (2017). Normalized Euclidean distance matrices for human motion retargeting. Proceedings of the 10th International Conference on Motion in Games,

Bossard, L., Dantone, M., Leistner, C., Wengert, C., Quack, T., & Gool, L. V. (2012). Apparel classification with style. Asian conference on computer vision,

Bourdev, L., & Malik, J. (2009). Poselets: Body part detectors trained using 3D human pose annotations. In 2009 IEEE 12th International Conference on Computer Vision,

Bourdev, L., & Malik, J. (2013). Articulated pose estimation using discriminative armlet classifiers. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

Brau, E., & Jiang, H. (2016). 3D Human Pose Estimation via Deep Learning from 2D Annotations. In 2016 Fourth International Conference on 3D Vision (3DV),

Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.-J., Yuan, J., & Thalmann, N. M. (2019a). Exploiting Spatial-temporal Relationships for 3D Pose Estimation via Graph Convolutional Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision,

Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.-J., Yuan, J., & Thalmann, N. M. (2019b). Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. European conference on computer vision,

Chen, C.-H., & Ramanan, D. (2017). 3D Human Pose Estimation = 2D Pose Estimation + Matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021a). Pre-trained image processing transformer. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., & Sutskever, I. (2020). Generative pretraining from pixels. International conference on machine learning,

Chen, T., Fang, C., Shen, X., Zhu, Y., Chen, Z., & Luo, J. (2021b). Anatomy-aware 3d

human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*.

Chen, T., Fang, C., Shen, X., Zhu, Y., Chen, Z., & Luo, J. (2021c). Anatomy-aware 3d human pose estimation with bone-based pose decomposition. *IEEE Transactions on Circuits and Systems for Video Technology*, *32*(1), 198-209.

Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., & Sun, J. (2018). Cascaded pyramid network for multi-person pose estimation. Proceedings of the IEEE conference on computer vision and pattern recognition,

Cheng, Y., Yu, F. X., Feris, R. S., Kumar, S., Choudhary, A., & Chang, S.-F. (2015). An exploration of parameter redundancy in deep networks with circulant projections. Proceedings of the IEEE international conference on computer vision,

Cheng, Y., Yang, B., Wang, B., Wending, Y., & Tan, R. (2019). Occlusion-Aware Networks for 3D Human Pose Estimation in Video. 2019 IEEE/CVF International Conference on Computer Vision (ICCV),

Cheng, Y., Yang, B., Wang, B., & Tan, R. T. (2020). 3D Human Pose Estimation Using Spatio-Temporal Networks with Explicit Occlusion Training. In Proceedings of the AAAI Conference on Artificial Intelligence,

Choi, K. J., & Ko, H. S. (2000). Online motion retargetting. *The Journal of Visualization and Computer Animation*, *11*(5), 223-235.

Ci, H., Wang, C., Ma, X., & Wang, Y. (2019a). Optimizing Network Structure for 3D Human Pose Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision,

Ci, H., Wang, C., Ma, X., & Wang, Y. (2019b). Optimizing network structure for 3d human pose estimation. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Dabral, R., Mundhada, A., Kusupati, U., Afaque, S., Sharma, A., & Jain, A. (2018). Learning 3D Human Pose from Structure and Motion. In Proceedings of the European Conference on Computer Vision (ECCV),

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition,

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition,

Di, W., Wah, C., Bhardwaj, A., Piramuthu, R., & Sundaresan, N. (2013). Style finder: Fine-grained clothing style detection and retrieval. Proceedings of the IEEE Conference on computer vision and pattern recognition workshops,

Ding, Y., Ma, Y., Wong, W. K., & Chua, T.-S. (2021). Leveraging Two Types of Global Graph for Sequential Fashion Recommendation. Proceedings of the 2021 International Conference on Multimedia Retrieval,

Dong, J., Chen, Q., Huang, Z., Yang, J., & Yan, S. (2015). Parsing based on parselets:

A unified deformable mixture model for human parsing. *IEEE transactions on pattern analysis and machine intelligence*, *38*(1), 88-101.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dumoulin, V., & Visin, F. (2016). *A guide to convolution arithmetic for deep learning*.

Eichner, M., & Ferrar, V. (2009). Better appearance models for pictorial structures. In Bmvc,

Fang, H.-S., Xu, Y., Wang, W., Liu, X., & Zhu, S.-C. (2018). Learning pose grammar to encode human body configuration for 3d pose estimation. In Proceedings of the AAAI Conference on Artificial Intelligence,

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In 2008 IEEE conference on computer vision and pattern recognition,

Felzenszwalb, P., Girshick, R., & McAllester, D. (2010). Cascade object detection with deformable part models. In 2010 IEEE Computer society conference on computer vision and pattern recognition,

Feng, A., Huang, Y., Xu, Y., & Shapiro, A. (2012). Automating the transfer of a generic set of behaviors onto a virtual character. Motion in Games: 5th International Conference, MIG 2012, Rennes, France, November 15-17, 2012. Proceedings 5,

Finley, T., & Joachims, T. (2008). Training structural svms when exact inference is intractable. In Proceedings of the 25th international conference on Machine learning,

Forsyth, D., & Fleck, M. M. (1997). Body plans. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition,

Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267-285). Springer.

Fukushima, K., & Miyake , S. (1982). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition. *In Competition and cooperation in neural nets*.

Gkioxari, G., Arbelaez, P., Bourdev, L., & Malik, J. (2013). Articulated pose estimation using discriminative armlet classifiers. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Gkioxari, G., Hariharan, B., Girshick, R., & Malik, J. (2014). Using k-poselets for detecting people and localizing their keypoints. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks.

In Proceedings of the fourteenth international conference on artificial intelligence and statistics,

Gong, J., Foo, L. G., Fan, Z., Ke, Q., Rahmani, H., & Liu, J. (2023). Diffpose: Toward more reliable 3d pose estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*, 33-51.

Gu, X., Gao, F., Tan, M., & Peng, P. (2020). Fashion analysis and understanding with artificial intelligence. *Information Processing & Management*, *57*(5), 102276.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision,

He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. proceedings of the 25th international conference on world wide web,

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *science*, *313*(5786), 504-507.

Hossain, M. R. I., & Little, J. J. (2018a). Exploiting temporal information for 3d human pose estimation. Proceedings of the European conference on computer vision (ECCV),

Hossain, M. R. I., & Little, J. J. (2018b). Exploiting temporal information for 3D human pose estimation. In Proceedings of the European Conference on Computer Vision (ECCV),

Hu, W., Zhang, C., Zhan, F., Zhang, L., & Wong, T.-T. (2021). Conditional Directed Graph Convolution for 3D Human Pose Estimation. In Proceedings of the 29th ACM International Conference on Multimedia,

Hu, Y., Yi, X., & Davis, L. S. (2015). Collaborative fashion recommendation: A functional tensor factorization approach. Proceedings of the 23rd ACM international conference on Multimedia,

Hua, G., Yang, M.-H., & Wu, Y. (2005). Learning to estimate human pose with data driven belief propagation. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,

Hubel, D. H., & Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*.

Hubel, D. H., & Wiesel, T. N. (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of neurophysiology*.

Hubel, D. H., & Wiesel, T. N. (1977). Ferrier lecture-Functional architecture of macaque monkey visual cortex. Proceedings of the Royal Society of London. Series B. Biological Sciences,

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2013). Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, *36*(7), 1325-1339.

Jahangiri, E., & Yuille, A. L. (2017). Generating Multiple Diverse Hypotheses for

Human 3D Pose Consistent with 2D Joint Detections. In Proceedings of the IEEE International Conference on Computer Vision Workshops,

Jang, H., Kwon, B., Yu, M., Kim, S. U., & Kim, J. (2018). A variational u-net for motion retargeting. In *SIGGRAPH Asia 2018 Posters* (pp. 1-2).

Jiang, N., Sheng, B., Li, P., & Lee, T.-Y. (2022). Photohelper: portrait photographing guidance via deep feature retrieval and fusion. *IEEE Transactions on Multimedia*, *25*, 2226-2238.

Johnson, S., & Everingham, M. (2009). Combining discriminative appearance and segmentation cues for articulated human pose estimation. IEEE 12th International Conference on Computer Vision Workshops,

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012a). ImageNet Classification with Deep Convolutional Neural Networks. Advances in neural information processing systems,

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kulkarni, T. D., Whitney, W. F., Kohli, P., & Tenenbaum, J. (2015). Deep convolutional inverse graphics network. *Advances in neural information processing systems*, *28*.

LeCun, Y. (1989). Generalization and network design strategies. *Connectionism in perspective*, *19*(143-155), 18.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989a). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1989b). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998a). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998b). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998c). Gradient-based learning applied to document recognition. Proceedings of the IEEE,

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Lee, J., & Shin, S. Y. (1999). A hierarchical approach to interactive motion editing for human-like figures. Proceedings of the 26th annual conference on Computer graphics and interactive techniques,

Li, C., & Lee, G. H. (2019). Generating multiple hypotheses for 3d human pose estimation with mixture density network. Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition (CVPR),
Li, H., Shi, B., Dai, W., Zheng, H., Wang, B., Sun, Y., Guo, M., Li, C., Zou, J., & Xiong, H. (2023). Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. Proceedings of the AAAI Conference on Artificial Intelligence,
Li, S., & Chan , A. B. (2014). 3d human pose estimation from monocular images with deep convolutional neural network. In Asian Conference on Computer Vision,
Li, S., Zhang, W., & Chan, A. B. (2015). Maximum-margin structured learning with deep networks for 3d human pose estimation. In Proceedings of the IEEE international conference on computer vision,
Li, S., Wang, L., Jia, W., Zhao, Y., & Zheng, L. (2022a). An iterative solution for improving the generalization ability of unsupervised skeleton motion retargeting. *Computers & Graphics*, *104*, 129-139.
Li, W., Liu, H., Ding, R., Liu, M., Wang, P., & Yang, W. (2022b). Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, *25*, 1282-1293.
Li, W., Liu, H., Tang, H., Wang, P., & Van Gool, L. (2022c). Mhformer: Multi-hypothesis transformer for 3d human pose estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Li, Z., Wang, X., Wang, F., & Jiang, P. (2019). On Boosting Single-Frame 3D Human Pose Estimation via Monocular Videos. In Proceedings of the IEEE/CVF International Conference on Computer Vision,
Liang, X., Xu, C., Shen, X., Yang, J., Liu, S., Tang, J., Lin, L., & Yan, S. (2015). Human parsing with contextualized convolutional neural network. Proceedings of the IEEE international conference on computer vision,
Lim, J., Chang, H. J., & Choi, J. Y. (2019). PMnet: Learning of Disentangled Pose and Movement for Unsupervised Motion Retargeting. BMVC,
Liu, J., Ding, H., Shahroudy, A., Duan, L.-Y., Jiang, X., Wang, G., & Kot, A. C. (2019). Feature Boosting Network For 3D Pose Estimation. *IEEE transactions on pattern analysis and machine intelligence*.
Liu, K., Ding, R., Zou, Z., Wang, L., & Tang, W. (2020a). A Comprehensive Study of Weight Sharing in Graph Networks for 3D Human Pose Estimation. In European Conference on Computer Vision,
Liu, K., Ding, R., Zou, Z., Wang, L., & Tang, W. (2020b). A comprehensive study of weight sharing in graph networks for 3d human pose estimation. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16,
Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.-c., & Asari, V. (2020c). Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.-c., & Asari, V. (2020d). Attention

mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.

Luvizon, D. C., Picard, D., & Tabia, H. (2018). 2D/3D Pose Estimation and Action Recognition Using Multitask Deep Learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Luvizon, D. C., Tabia, H., & Picard, D. (2019). Human Pose Regression by Combining Indirect Part Detection and Contextual Information. *Computers & Graphics*.

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017a). A simple yet effective baseline for 3d human pose estimation. Proceedings of the IEEE international conference on computer vision,

Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017b). A simple yet effective baseline for 3d human pose estimation. Proceedings of the IEEE International Conference on Computer Vision (ICCV),

Martinsson, J., & Mogren, O. (2019). Semantic segmentation of fashion images using feature pyramid networks. Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops,

McAuley, J., Targett, C., Shi, Q., & Van Den Hengel, A. (2015). Image-based recommendations on styles and substitutes. Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval,

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., & Theobalt, C. (2017). Monocular 3d human pose estimation in the wild using improved cnn supervision. 2017 international conference on 3D vision (3DV),

Moreno-Noguer, F. (2017). 3D Human Pose Estimation From a Single Image via Distance Matrix Regression. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Mori, G., & Malik, J. (2002). Estimating human body configurations using shape context matching. European conference on computer vision,

Nazir, A., Cheema, M. N., Sheng, B., Li, H., Li, P., Yang, P., Jung, Y., Qin, J., Kim, J., & Feng, D. D. (2020). OFF-eNET: An optimally fused fully end-to-end network for automatic dense volumetric 3D intracranial blood vessels segmentation. *IEEE Transactions on Image Processing*, *29*, 7192-7202.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In European conference on computer vision,

Nibali, A., He, Z., Morgan, S., & Prendergast, L. (2018). Numerical coordinate regression with convolutional neural networks. arXiv preprint arXiv:1801.07372.,

Park, S., Hwang, J., & Kwak, N. (2016). 3D Human Pose Estimation Using Convolutional Neural Networks with 2D Pose Information. In European

Conference on Computer Vision,
Park, S., & Kwak, N. (2018). 3d human pose estimation with relational networks. *arXiv preprint arXiv:1805.08961*.
Pavlakos, G., Zhou, X., Derpanis, K. G., & Daniilidis, K. (2017). Coarse-To-Fine Volumetric Prediction for Single-Image 3D Human Pose. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,
Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019a). 3D Human Pose Estimation in Video With Temporal Convolutions and Semi-Supervised Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019b). 3d human pose estimation in video with temporal convolutions and semi-supervised training. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Peng, J., Zhou, Y., & Mok, P. (2024). KTPFormer: Kinematics and Trajectory Prior Knowledge-Enhanced Transformer for 3D Human Pose Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,
Pishchulin, L., Andriluka, M., Gehler, P., & Schiele, B. (2013). Poselet conditioned pictorial structures. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),
Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
Ren, X., Berg, A. C., & Malik, J. (2005). Recovering human body configurations using pairwise constraints between parts. In Tenth IEEE International Conference on Computer Vision,
Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18,
Rumelhart, D. E., Hinton, G. E., & Williams , R. J. (1986). Learning representations by back-propagating errors. *nature*.
Sapp, B., Weiss, D., & Taskar, B. (2011). Parsing human motion with stretchable models. In CVPR 2011,
Shan, W., Lu, H., Wang, S., Zhang, X., & Gao, W. (2021a). Improving Robustness and Accuracy via Relative Information Encoding in 3D Human Pose Estimation. Proceedings of the 29th ACM International Conference on Multimedia,
Shan, W., Lu, H., Wang, S., Zhang, X., & Gao, W. (2021b). Improving robustness and accuracy via relative information encoding in 3d human pose estimation. In Proceedings of the 29th ACM International Conference on Multimedia,
Shan, W., Liu, Z., Zhang, X., Wang, S., Ma, S., & Gao, W. (2022). P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. European

Conference on Computer Vision,

Shan, W., Liu, Z., Zhang, X., Wang, Z., Han, K., Wang, S., Ma, S., & Gao, W. (2023). Diffusion-Based 3D Human Pose Estimation with Multi-Hypothesis Aggregation. *arXiv preprint arXiv:2303.11579*.

Sharma, S., Varigonda, P. T., Bindal, P., Sharma, A., & Jain, A. (2019). Monocular 3d human pose estimation by generation and ordinal ranking. In Proceedings of the IEEE/CVF International Conference on Computer Vision,

Sigal, L., Balan, A. O., & Black, M. J. (2010). Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, *87*(1-2), 4-27.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever , I., & Salakhutdinov , R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*.

Su, B., Ding, X., Wang, H., & Wu, Y. (2017). Discriminative dimensionality reduction for multi-dimensional sequences. *IEEE transactions on pattern analysis and machine intelligence*, *40*(1), 77-91.

Sun, K., Xiao, B., Liu, D., & Wang, J. (2019). Deep High-Resolution Representation Learning for Human Pose Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Sun, X., Shang, J., Liang, S., & Wei, Y. (2017). Compositional Human Pose Regression. Proceedings of the IEEE International Conference on Computer Vision,

Sun, X., Xiao, B., Wei, F., Liang, S., & Wei, Y. (2018). Integral human pose regression. Proceedings of the European Conference on Computer Vision (ECCV),

Tak, S., & Ko, H.-S. (2005). A physically-based motion retargeting filter. *ACM Transactions on Graphics (TOG)*, *24*(1), 98-117.

Tang, Z., Hao, Y., Li, J., & Hong, R. (2023a). FTCM: Frequency-temporal collaborative module for efficient 3D human pose estimation in video. *IEEE Transactions on Circuits and Systems for Video Technology*.

Tang, Z., Qiu, Z., Hao, Y., Hong, R., & Yao, T. (2023b). 3D Human Pose Estimation With Spatio-Temporal Criss-Cross Attention. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., & Fua, P. (2016a). Structured Prediction of 3D Human Pose with Deep Neural Networks. arXiv preprint arXiv:1605.05180.,

Tekin, B., Rozantsev, A., Lepetit, V., & Fua, P. (2016b). Direct Prediction of 3D Body Poses from Motion Compensated Sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

Tekin, B., Marquez-Neila, P., Salzmann, M., & Fua, P. (2017). Learning to fuse 2d and 3d image cues for monocular body pose estimation. Proceedings of the IEEE International Conference on Computer Vision (ICCV),

Toshev, A., & Szegedy, C. (2014). DeepPose: Human Pose Estimation via Deep Neural Networks. In Proceedings of the IEEE conference on computer vision and

pattern recognition,

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Veit, A., Kovacs, B., Bell, S., McAuley, J., Bala, K., & Belongie, S. (2015). Learning visual clothing style with heterogeneous dyadic co-occurrences. Proceedings of the IEEE International Conference on Computer Vision,

Verma, S., Anand, S., Arora, C., & Rai, A. (2018). Diversity in fashion recommendation using semantic parsing. 2018 25th IEEE International Conference on Image Processing (ICIP),

Villegas, R., Yang, J., Ceylan, D., & Lee, H. (2018). Neural kinematic networks for unsupervised motion retargetting. Proceedings of the IEEE conference on computer vision and pattern recognition,

Villegas, R., Ceylan, D., Hertzmann, A., Yang, J., & Saito, J. (2021). Contact-aware retargeting of skinned motion. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Vittayakorn, S., Yamaguchi, K., Berg, A. C., & Berg, T. L. (2015). Runway to realway: Visual analysis of fashion. 2015 IEEE Winter Conference on Applications of Computer Vision,

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, *37*(3), 328-339.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., & Wang, X. (2020a). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *43*(10), 3349-3364.

Wang, J., Yan, S., Xiong, Y., & Lin, D. (2020b). Motion guided 3d pose estimation from videos. European Conference on Computer Vision,

Wang, J., Yan, S., Xiong, Y., & Lin, D. (2020c). Motion guided 3d pose estimation from videos. In European Conference on Computer Vision,

Wang, M., Chen, X., Liu, W., Qian, C., Lin, L., & Ma, L. (2018). DRPose3D: Depth Ranking in 3D Human Pose Estimation. the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018),

Wren, C., Azarbayejani, A., Darrell, T., & Pentland, A. (1997). Pfinder: real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*.

Xie, Z., Zhang, W., Sheng, B., Li, P., & Chen, C. P. (2021). BaGFN: broad attentive graph fusion network for high-order feature interactions. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(8), 4499-4513.

Xu, T., & Takano, W. (2021). Graph stacked hourglass networks for 3d human pose estimation. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., & Wang, X. (2018). 3d human pose estimation in the wild by adversarial learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

Yang, Y., & Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In CVPR 2011,

Yang, Y., & Ramanan, D. (2012). Articulated Human Detection with Flexible Mixtures of Parts. *IEEE transactions on pattern analysis and machine intelligence*.

Yu, B. X., Zhang, Z., Liu, Y., Zhong, S.-h., Liu, Y., & Chen, C. W. (2023). Gla-gcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Yu, F., & Koltun, V. (2015). *Multi-Scale Context Aggregation by Dilated Convolutions*.

Zeng, A., Sun, X., Huang, F., Liu, M., Xu, Q., & Lin, S. (2020). Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. European Conference on Computer Vision,

Zeng, A., Sun, X., Yang, L., Zhao, N., Liu, M., & Xu, Q. (2021). Learning Skeletal Graph Neural Networks for Hard 3D Pose Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision,

Zhang, J., Chen, Y., & Tu, Z. (2022a). Uncertainty-aware 3D human pose estimation from monocular video. Proceedings of the 30th ACM International Conference on Multimedia,

Zhang, J., Tu, Z., Yang, J., Chen, Y., & Yuan, J. (2022b). Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Zhang, J., Weng, J., Kang, D., Zhao, F., Huang, S., Zhe, X., Bao, L., Shan, Y., Wang, J., & Tu, Z. (2023). Skinned Motion Retargeting with Residual Perception of Motion Semantics & Geometry. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Zhang, W., Doi, K., Giger, M. L., Nishikawa, R. M., & Schmidt, R. A. (1996). An improved shift-invariant artificial neural network for computerized detection of clustered microcalcifications in digital mammograms. *Medical Physics*, *23*(4), 595-601.

Zhao, L., Peng, X., Tian, Y., Kapadia, M., & Metaxas, D. N. (2019a). Semantic graph convolutional networks for 3d human pose regression. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Zhao, L., Peng, X., Tian, Y., Kapadia, M., & Metaxas, D. N. (2019b). Semantic graph convolutional networks for 3d human pose regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Zhao, Q., Zheng, C., Liu, M., Wang, P., & Chen, C. (2023). PoseFormerV2: Exploring Frequency Domain for Efficient and Robust 3D Human Pose Estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Zhao, W., Wang, W., & Tian, Y. (2022). Graformer: Graph-oriented transformer for 3d pose estimation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., & Ding, Z. (2021a). 3d human pose estimation with spatial and temporal transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., & Torr, P. H. (2021b). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., & Xiong, C. (2018). End-to-end dense video captioning with masked transformer. Proceedings of the IEEE conference on computer vision and pattern recognition,

Zhou, X., Sun, X., Zhang, W., Liang, S., & Wei , Y. (2016). Deep kinematic pose regression. In European Conference on Computer Vision,

Zhou, X., Huang, Q., Sun, X., Xue, X., & Wei, Y. (2017). Towards 3d human pose estimation in the wild: a weakly-supervised approach. Proceedings of the IEEE International Conference on Computer Vision (ICCV),

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.

Zhu, Y., Xu, X., Shen, F., Ji, Y., Gao, L., & Shen, H. T. (2021). PoseGTAC: Graph Transformer Encoder-Decoder with Atrous Convolution for 3D Human Pose Estimation. IJCAI,

Zou, Z., & Tang, W. (2021). Modulated graph convolutional network for 3D human pose estimation. Proceedings of the IEEE/CVF international conference on computer vision,