

國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

應用於 3D 人體姿態估計的全局與局部交替混合注意力  
模型

AMPose: Alternately Mixed Global-Local Attention  
Model for 3D Human Pose Estimation

林宏信

Hong-Xin Lin

指導教授: 吳沛遠 博士

Advisor: Pei-Yuan Wu, Ph.D.

中華民國 112 年 7 月

July, 2023



# Acknowledgements

感謝我的父母。





## 摘要

3D 人體姿態估計在復健、高爾夫和棒球等領域被廣泛的應用。過去研究分為從影片中的多張連續圖片或僅單張圖片來進行人體 3D 重建。圖卷積因可以定義人體的骨架關係來增強資料間的關聯，所以普遍被使用在 3D 人體姿態估計的領域，並且過去的研究與實驗結果證實圖卷積可以更精確地重建 3D 人體姿態。近年在多個電腦視覺的子領域發現自注意力機制之優越性，且在許多資料集取得優異的成果。然而，在 3D 的領域中，人體關節點間的關聯不盡然可以透過純粹的自注意力機制來表達，並且過去圖卷積已經提出非常多的方法來考慮人體關節點間之關聯。本研究主要在改善自注意力機制沒辦法完全的利用人體骨架的問題，並提升重建 3D 人體骨架的表現。我們藉由交替的混合自注意力機制和圖卷積的模型，來獲取局部和全局的關聯性來得到更全面的特徵向量，進而得到 3D 關節點位置。我們廣泛的測試模型可能的各種變因來證明所提模型之有效性，並且在公開資料集 Human3.6M 和 MPI-INF-3DHP 上都取得相當好的結果，並超越現有模型。

**關鍵字：**圖卷積、自注意力機制、3D 人體姿態

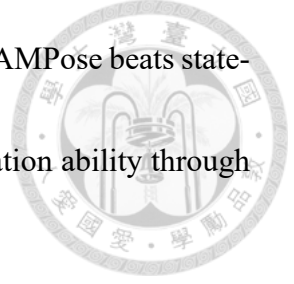




# Abstract

Single-image 3D human pose estimation (HPE) has many applications in rehabilitation, golf, and baseball fields. Over the past few years, much research has involved reconstructing the human skeleton from either a series of video frames or a single image. Previous studies have commonly discussed the utilization of graph convolutional networks (GCNs) as a means to address 3D HPE, and substantial experiments have verified the efficacy of GCNs for this purpose. Recently, Transformer-based models have attracted considerable interest because of their excellent capacity for relating multiple frames. Nevertheless, the pure Transformer method in the single-frame condition cannot exploit the characteristics of the human joints. To address this, we introduce AMPose as an innovative approach that combines Transformer and GCN blocks to capture global and local dependencies among human joints. By leveraging the strengths of both modules, AMPose achieves a comprehensive understanding of human joint interactions. In order to assess the effectiveness of AMPose, we conduct experiments using well-known public

datasets, including MPI-INF-3DHP and Human3.6M. Consequently, AMPose beats state-of-the-art models on both datasets, demonstrating superior generalization ability through cross-dataset comparisons.



**Keywords:** Graph convolution neural network, Transformer, 3D human pose



# Contents

	<b>Page</b>
<b>Acknowledgements</b>	<b>i</b>
<b>摘要</b>	<b>iii</b>
<b>Abstract</b>	<b>v</b>
<b>Contents</b>	<b>vii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
<b>Chapter 2 Related Works</b>	<b>5</b>
2.1 3D Human Pose Estimation . . . . .	5
2.2 Graph Convolutional Networks . . . . .	7
2.3 Transformer Encoder . . . . .	8
<b>Chapter 3 Methodology</b>	<b>11</b>
3.1 Overview . . . . .	11
3.2 Transformer Encoder . . . . .	12
3.3 GCN Blocks . . . . .	14
3.4 Loss Function . . . . .	15



<b>Chapter 4</b>	<b>Experiment</b>	<b>17</b>
4.1	Dataset . . . . .	17
4.2	Evaluation Protocol . . . . .	18
4.3	Implementation Details . . . . .	18
4.4	Comparison with the State-of-the-art . . . . .	19
4.5	Ablation Study . . . . .	22
4.6	Qualitative Results . . . . .	24
<b>Chapter 5</b>	<b>Conclusion</b>	<b>27</b>
<b>References</b>		<b>29</b>





# List of Figures

3.1	Overall process of the AMPose method. . . . .	11
4.1	Instantiation of the proposed AMPose method. . . . .	22
4.2	Variety of configurations of the GCN block. . . . .	22
4.3	Graphical display on Human3.6M. . . . .	25
4.4	Graphical display of indoor scenes from 3DHP. . . . .	25
4.5	Graphical display of outdoor scenes from 3DHP. . . . .	26





# List of Tables

4.1	MPJPE comparisons on Human3.6M. Top table: CPN with 17 human joints used as input. Bottom table: GT 2D with 17 human joints used as input. * indicates refinement. . . . .	19
4.2	P-MPJPE comparisons on Human3.6M. CPN with 17 human joints used as input. * indicates refinement. . . . .	19
4.3	MPJPE comparisons on Human3.6M. Top table: CPN with 16 human joints used as input. Bottom table: GT 2D with 16 human joints used as input. . . . .	19
4.4	Evaluation outcomes from conducting a cross-dataset comparison on 3DHP.	20
4.5	The comparison of results on the 3DHP dataset. . . . .	20
4.6	The outcomes obtained from the different variations of the GCN block designs. . . . .	21
4.7	Ordering Analysis of Transformer encoders and GCN blocks. . . . .	23
4.8	Ablation study for splitting neighbor nodes. . . . .	23
4.9	Quantifying the impact: Ablation study on depth and channel settings. . . . .	24





# Chapter 1 Introduction

3D human pose estimation (HPE) involves inferring the posture of the human body from images/videos taken from the camera, which has found many practical applications such as rehabilitation [8], golf [22], and baseball [15]. Most previous methods [1, 25, 34] estimated 3D keypoints by the lifting structure in which the images are fed into the 2D HPE model to extract 2D keypoints first, and the estimated 2D keypoints are used as the input of the lifting model to produce the 3D keypoints. The different characteristics of 2D and 3D data can be modeled by 2D HPE models and lifting models, respectively. On the other hand, 3D human mesh estimation contains both body shape and pose estimation [18]. The recent work [4] successfully applied the state-of-the-art multi-view 3D human pose model [10] to 3D human mesh estimation, leveraging the information of multiple 2D joints from different views to reconstruct accurate human pose and shape. Therefore, the progress in 3D HPE can benefit other fields.

The temporal methods [14, 25, 32, 36] have a longer history than the single-frame methods [20, 31, 34]. Temporal models take advantage of multiple frames in a video as input, enabling them to leverage the pair relations between 2D and 3D poses over longer sequences to predict 3D poses. However, it is important to note that temporal methods often require significant computational power and exhibit slower computation speeds. In view of the dire computation cost of temporal methods, this work focuses on the single-

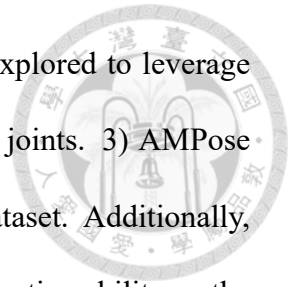
frame method, which is more suitable for real-time and resource-constrained scenarios.

Early studies [25, 31] have proven that the feature representation derived from 2D posture is effective towards 3D HPE. After that, the exploration of the powerful feature representation becomes the main target in the field of 3D HPE. To leverage the spatial information, graph convolutional networks (GCNs) are used to describe spatial relations in keypoints of the human body, which consider the joint as an independent dimension. A weakness of the GCNs is weight-sharing within the 1-hop range, which leads to unsatisfactory performance in 3D HPE since the equal weights cannot capture the various levels of flexibility in the human skeleton.

The previous works [1, 16, 29, 37] proposed to redesign the kernel within the 1-hop range to alleviate this issue. On the other hand, the Transformer-based method is frequently adopted to build 3D HPE models as in the other areas of computer vision. A crucial part of the Transformer encoder is the self-attention module, as this module can be used to model the global dependency among joints in 3D HPE. Although the global relation can be exploited via the Transformer, an interesting question arises as how the structure of the human skeleton can be incorporated for 3D HPE. The previous studies [34] adopted ChebNet [6] to link up the Transformer with the physical relations in the human skeleton. However, the ChebNet is weight-sharing within the same hop range, which may not capture the variety of human motions. Hence, we propose to combine the GCN block and Transformer encoder to exploit the local and global relationships among the human joints, respectively.

In brief, this work makes three contributions: 1) we introduce AMPose as a novel approach to 3D HPE that integrates Transformer encoders and GCN blocks using a global-

local attention structure. 2) the optimal design for GCN blocks is explored to leverage physical-connectivity features and enhance the local relation among joints. 3) AMPose beats state-of-the-art single-frame approaches on the Human3.6M dataset. Additionally, in cross-dataset evaluations, AMPose demonstrates excellent generalization ability on the MPI-INF-3DHP dataset.







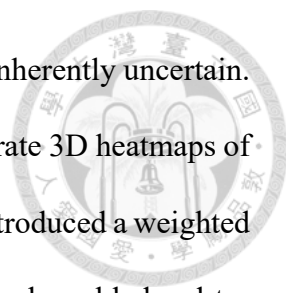


## Chapter 2 Related Works

### 2.1 3D Human Pose Estimation

The field of 3D HPE is advancing rapidly, with researchers presenting various methodologies to tackle this complex challenge. These methodologies can be organized into two primary groups: direct estimation and lifting estimation. The critical difference between the two groups is that lifting estimation uses existing 2D models as an intermediate step in the 3D HPE process, while direct estimation employs end-to-end learning.

In direct estimation methods, the models for 3D HPE are trained through end-to-end training, enabling the adjustment of the complete model weight. Park *et al.* [23] proposed to improve the feature representation obtained from images by fusing the 2D and 3D information via convolutional neural networks (CNNs). Pavlakos *et al.* [24] adopted a pre-trained 2D posture model as the backbone, and the output of the backbone is fed to two streams in which the 3D human posture and shape are predicted separately. The pre-trained 2D pose model can improve the data efficiency and alleviate the problem of scarcely labeled 3D data. There are abundant labels in public datasets for 2D HPE, and the pre-trained 2D pose models can be exploited to initialize weights of neural networks in 3D pose models.



In monocular 3D HPE, estimating 3D positions from images is inherently uncertain. Sun *et al.* [28] addressed this challenge by employing CNNs to generate 3D heatmaps of human joints. To compute the expectation of the 3D heatmaps, they introduced a weighted sum operation for each joint, replacing the max operation. This approach enabled end-to-end learning, effectively reducing the discrepancy between the 3D and 2D domains.

The lifting method has become increasingly prominent due to its numerous advantages, including privacy preservation, reduced computation, and superior generalization ability. The proposed method belongs to this category. Previous works [2, 25, 31, 32] have leveraged the lifting method by using a longer sequence as input, taking advantage of its reduced computational requirements.

Pavlo *et al.* [25] explored the capabilities of the lifting method by employing a long-range temporal sequence as input to output the 3D position of the central frame. Their lifting model embeds the entire joints of each frame in a feature representation, which a temporal dilated convolutional model then convolves to produce the feature representation of the center frame. Taking a different approach, Zeng *et al.*[31] considered the relations in the human skeleton by splitting the feature vector of human joints into different groups corresponding to the head, limbs, and torso. This technique enhances the local dependency among joints, and the groups are concatenated along the channel dimensions to transfer the feature vector after extracting the local information. In a separate study, Chen *et al.* [2] decomposed the HPE model into two streams, generating the bone length and direction to reconstruct the 3D pose. The two-stream method can model the property of bone length and direction with different networks. In the branch of bone directions, temporal dilated convolutions are applied to make predictions of the 3D bone directions for the central frame. Moreover, random samples from the input data are selected and fed

into a regression network to estimate the bone lengths for the corresponding target pose.



## 2.2 Graph Convolutional Networks

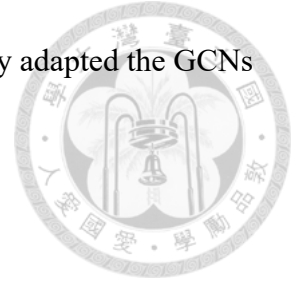
Vanilla GCNs aim to establish connections between input data through manually constructed graphs. In these models, the features of all nodes are transformed using the same projection matrix, and the features of neighboring nodes are added together to yield the convolved feature for the next layer in the network. However, this approach is inadequate for accurately modeling 3D HPE due to the intricate nature of human actions and the inherent flexibility of the human skeleton. To address this issue, prior works have proposed several solutions such as ST-GCN [30], Semantic GCN [37], and Modulated GCN [33].

Cai *et al.* [1] introduced ST-GCN to the field of 3D HPE. In the ST-GCN, the neighbor nodes of the hand-crafted graph are split into multiple groups according to their relative proximity from the root node. Then, the different transformations are applied to the groups rather than sharing the same transformation. By splitting the neighbor nodes based on the proximity to the root node, ST-GCN enhances the ability to capture fine-grained spatial relationships.

Semantic GCN improved the hand-crafted graph by multiplying it element-wise with a trainable weight matrix and then applying a softmax operation to obtain an enhanced graph. This empowers the model to grasp and exploit the semantic connections among human joints.

Modulation was introduced to Semantic GCN by Zou *et al.* [37] to improve the performance of GCNs. The shared transformation matrix is adapted by learning various modulation vectors for each joint, resulting in substantial enhancements in accuracy with a

reduced parameter count. Overall, these approaches have successfully adapted the GCNs for 3D HPE.

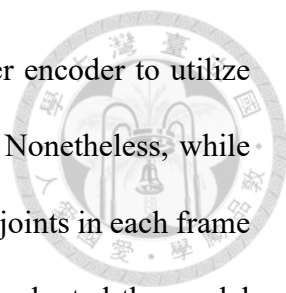


## 2.3 Transformer Encoder

In recent years, the Transformer encoder has gained widespread popularity in computer vision applications, mainly due to its self-attention mechanism of the Transformer. This mechanism has a large receptive field and adaptable attention weights, which have been thoroughly demonstrated to be effective through extensive experiments [14, 26, 32, 34, 35]. The Poseformer, developed by Zheng *et al.* [35], was the first to utilize a pure Transformer model in monocular 3D HPE. In the Poseformer framework, a spatial Transformer establishes intricate connections in the human skeletons. At the same time, a temporal Transformer links each frame with all other frames. This ingenious design led the Poseformer to achieve remarkable performance, drawing significant attention from the HPE community. Since the introduction of the Poseformer, Transformer-based approaches have been widely incorporated into many subsequent 3D HPE models.

Zhang *et al.* [32] made notable advancements in 3D HPE by modifying the architecture of Poseformer. They proposed to integrate spatial and temporal Transformers in an interleaved manner. This approach effectively captures spatial and temporal dependencies, improving accuracy in 3D HPE.

While the Transformer-based method is effective, it may not explicitly capture local relations in 3D HPE. Recent research [12, 26, 34] has shown that combining global and local information can address this issue. In 3D HPE, the temporal dependencies and spatial relations among joints can be enhanced by integrating global and local information. Li *et*



*al.* [12] integrated stride convolution into the temporal Transformer encoder to utilize both long-range and local context information in a video sequence. Nonetheless, while temporal information is considered, the spatial relationships between joints in each frame are not explicitly modeled. To address this issue, Shan *et al.* [26] adopted the model proposed by Li *et al.* [12] as the backbone and introduced a pre-training technique. In the initial training phase, the Transformer aims to recover the original 2D keypoints from the masked 2D keypoints. Following that, the learned weights are transferred to the 3D HPE model, which is then fine-tuned using ground truth 3D positions. To reinforce spatial relationships, the 2D pose of each frame is embedded using a multilayer perceptron.





## Chapter 3 Methodology

### 3.1 Overview

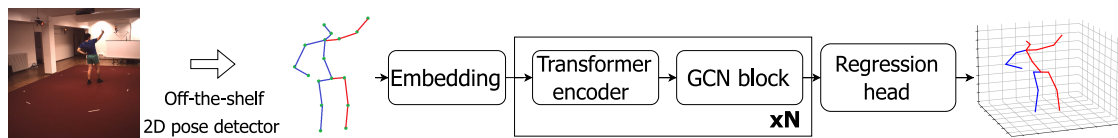


Figure 3.1: Overall process of the AMPose method.

Our objective is to precisely infer the 3D poses of human subjects from individual images. As depicted in Figure 3.1, the AMPose approach uses the 2D joint positions estimated by a readily available 2D pose model [3] as input to our model. We then apply a trainable linear projection layer to each 2D joint pose, transforming it into a feature representation with more channels, which serves as the starting point for the proposed lifting method.

To further reinforce the transformed feature representation, the Transformer encoder and the GCN block are alternately stacked to extract the global and local dependency, respectively. The Transformer encoder incorporates a self-attention mechanism, facilitating dynamic interaction among human joints. On the other hand, the GCN block consists of GCN layers followed by GCN layers with a residual connection. This block explicitly models the characteristics of the human skeleton.





## 3.2 Transformer Encoder

**Self-attention** is a generalized version of GCNs, with distinct differences in attention weight computation and receptive fields. In GCNs, attention weights are determined based on the reciprocal number of neighboring nodes, implying equal treatment of all neighbors. However, self-attention extends the receptive field to the global scope, relaxing the constraint of equal treatment for neighboring nodes. As a result, attention weights are determined by the input feature, leading to adaptable feature representations. Specifically, the input feature  $Z \in R^{N_j \times N_d}$ , where  $N_j$  is the number of joints and  $N_d$  is the number of channels, is transformed by three trainable matrices to produce the query, key, and value, namely

$$Q = Z\Theta^Q, K = Z\Theta^K, V = Z\Theta^V, \quad (3.1)$$

where  $\Theta^Q$ ,  $\Theta^K$ , and  $\Theta^V \in R^{N_d \times N_d}$  are the trainable matrices.

The pairwise similarity between the query matrix and the key matrix is computed, typically using a dot product. The softmax operation is applied to the similarity scores. In the final step of the self-attention operation, the resulting similarity scores are multiplied by the value matrix. This process allows for capturing global relations in the features. The self-attention mechanism can be mathematically formulated as follows:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{N_d}}\right)V. \quad (3.2)$$

**Multi-head self-attention** (MSA) serves the specific purpose of introducing diversification in attention weights. To achieve this, the query, key, and value matrices are partitioned

into  $N_h$  submatrices denoted as  $Q_i, K_i,$  and  $V_i \in N_j \times \frac{N_d}{N_h}$ , respectively, where  $N_h$  represents the number of attention heads. MSA captures diverse patterns and relationships in the input data by interconnecting the features from different heads using a trainable matrix  $\Theta^{out} \in R^{N_d \times N_d}$ . The MSA operation can be denoted as:

$$MSA(Q, K, V) = \text{concat}(head_1, \dots, head_{N_h}) \Theta^{out}, \quad (3.3)$$

where  $head_i = \text{Attention}(Q_i, K_i, V_i)$  for all  $i \in [1, \dots, N_h]$ .

**Multi-layer perceptron (MLP)** consists of two fully connected layers (FCs) in the AM-Pose. An activation function  $\sigma$  is applied between FCs. The MLP can be represented as follows:

$$MLP(Z) = FC(\sigma(FC(Z))), \quad (3.4)$$

The overall procedure can be expressed as follows:

$$Z' = Z_{in} + \Theta_{pos}, \quad (3.5)$$

$$Z'' = MSA(LN(Z')) + Z', \quad (3.6)$$

$$Z_{out} = MLP(LN(Z'')) + Z'', \quad (3.7)$$

where  $\Theta_{pos}$  is the positional embedding,  $LN(\cdot)$  denotes layer normalization function,  $Z_{in}$  denotes the input feature, and  $Z_{out}$  denotes the output feature of the Transformer encoder.



### 3.3 GCN Blocks

The fountainhead of GCN can be traced back to the spectral graph convolution proposed by Shuman *et al.* [27], which incorporates the input data with specific relation via a handcrafted graph. ChebNet [6] approximates the spectral graph convolution through Chebyshev polynomials, in which the same weights are applied to the same hop range to convolve the features. In the previous study [11], it is pointed out that multi-hop convolution may result in over-smoothing. Reducing the hop is a feasible method to overcome this problem since neural networks with multiple layers can propagate the information effectively. Therefore, Kipf *et al.* [11] proposed to approximate the spectral graph convolution with the first order, which limits the receptive field to the 1-hop range. The formula of the vanilla GCNs can be expressed as follows:

$$Z = D^{-0.5}AD^{-0.5}X\Theta, \quad (3.8)$$

where the adjacency matrix  $A \in \{0, 1\}^{N_j \times N_j}$  defines the relation in the input feature,  $X \in R^{N_j \times N_d}$  is the input feature, the diagonal matrix  $D \in R^{N_j \times N_j}$  represents the degree matrix of  $A$ ,  $\Theta \in R^{N_d \times N_D}$  indicates the filter,  $Z \in R^{N_j \times N_D}$  indicates the output feature,  $N_j$  denotes the number of nodes in the graph,  $N_d$  denotes the input dimension of the feature, and  $N_D$  indicates the dimension of the output feature.

Vanilla GCNs can solve the tasks in node and graph classification with fully weight-sharing [16], but the 3D HPE contains complicated motion and self-occlusion issues. To accommodate GCNs to 3D HPE, the previous research [1, 33] took account of the semantic relations among human joints. In our work, we adopt the technique introduced in ST-GCN [1] to model local dependency among human skeletons. The neighbor joint can



be split into three groups:

1. The center node itself.
2. Neighboring nodes which are closer to the hip node than the center node.
3. Neighboring nodes which are farther away from the hip node than the center node.

The formula of GCNs is modified correspondingly: Three transformation matrices are applied to the neighbor joints for each center joint, and then the neighbor features are added up to generate the convolved feature, namely

$$Z = \sum_{k=1}^3 D_k^{-0.5} A_k D_k^{-0.5} X \Theta_k, \quad (3.9)$$

where  $k$  indicates the index of various groups, the vertex matrix  $A \in \{0, 1\}^{N_j \times N_j}$  is disassembled into three sub-matrices  $A_k \in \{0, 1\}^{N_j \times N_j}$  in accord with the human joint groups which satisfy  $A_1 + A_2 + A_3 = A$ , the diagonal matrix  $D_k \in R^{N_j \times N_j}$  is the degree matrix of  $A_k$ , and  $\Theta_k$  is the filter for the  $k$ -th group.

### 3.4 Loss Function

To predict the 3D position with the AMPose, the mean square error is used as the loss function of our model:

$$Loss = \frac{1}{N} \sum_{i=1}^N \|X_i - \tilde{X}_i\|^2, \quad (3.10)$$

where the index  $i$  represents the human joint type,  $N$  signifies the count of joints,  $\tilde{X}_i$  represents the estimated 3D coordinates, and  $X_i$  represents the ground truth 3D coordinates.





# Chapter 4 Experiment

## 4.1 Dataset

The AMPose is tested on two public datasets: Human3.6M [9] and MPI-INF-3DHP (3DHP) [21].

**Human3.6M** is one of the most common datasets for 3D HPE, which is composed of 3.6 million images. The images are taken by four calibrated cameras from different angles. To preprocess the data, we follow [34, 37] and use the mid-hip joint as the root joint to localize the 3D human positions. The calibration process establishes the world coordinate system, while the ground truth (GT) 3D position in the camera-specific coordinate system is derived by applying translation and rotation to the GT 3D position in the world coordinate system using the camera’s extrinsic parameters. The AMPose is trained with subjects S1, S5, S6, S7, and S8, while tested on subjects S9 and S11.

**MPI-INF-3DHP** constitutes a comprehensive dataset encompassing a wide range of indoor and outdoor settings, intricate and infrequent actions, and varying camera viewpoints. In accordance with prior research [29, 37], we evaluate the generalization ability of the AMPose by training on Human3.6M and testing on the test set of 3DHP. Additionally, we train the AMPose model on the training set of 3DHP and conduct a comparative evaluation

against video-based methods [14, 26, 32] by evaluating it on the test set of 3DHP.



## 4.2 Evaluation Protocol

In our experiment, we use the mean per joint position error (MPJPE) as the performance metric for AMPose. MPJPE is computed as the mean Euclidean distance between the estimated 3D human pose and the GT 3D human pose. When it comes to aligning the estimated pose with the GT pose, a rigid transformation is applied, which is commonly referred to as the P-MPJPE.

On 3DHP, we employ two supplementary metrics to evaluate AMPose: the Percentage of Correct Keypoints (PCK) and the Area Under the Curve (AUC). PCK measures the correctness of the estimated keypoints based on a fixed threshold. AUC is calculated as the mean PCK across different thresholds. A prediction is considered successful regarding PCK if the MPJPE falls below the predefined threshold. In line with previous studies [16, 29, 37], we set the PCK threshold to 150 millimeters (mm).

## 4.3 Implementation Details

The AMPose is built upon a two-stage approach for 3D HPE, where a pre-trained 2D pose model first processes input images. For our experiments conducted on Human3.6M, we employ the cascaded pyramid network (CPN) [3] as our 2D backbone. The CPN is widely used in previous works for 2D HPE [31, 32, 34, 35, 37].

In our experiments, we set the depth  $N$  of the AMPose to 5 and the embedding channels to 512. The proposed model is trained for 60 epochs. The batch size is set to 128. The

Table 4.1: MPJPE comparisons on Human3.6M. Top table: CPN with 17 human joints used as input. Bottom table: GT 2D with 17 human joints used as input.  $\star$  indicates refinement.

CPN	WalkT.	WalkD.	Walk	Wait	SitD.	Smoke	Sit	Pose	Purchu.	Photo	Greet	Phone	Eat	Dire.	Discu.	Avg.
Ci <i>et al.</i> [5]	43.3	54.8	40.4	50.0	78.9	51.2	60.2	49.6	46.4	68.9	50.4	52.9	<b>44.7</b>	46.8	52.3	52.7
Pavlo <i>et al.</i> [25]	42.7	55.3	39.5	49.5	67.4	52.4	59.3	49.4	47.4	61.4	51.8	53.6	49.0	47.1	50.6	51.8
Cai <i>et al.</i> [1] $\star$	41.2	53.5	39.2	48.4	64.4	51.2	59.2	48.3	45.8	61.3	50.9	52.9	47.6	46.5	48.8	50.6
Lutz <i>et al.</i> [19]	42.4	53.8	39.4	47.1	67.1	51.2	59.6	47.0	46.7	60.1	49.4	53.2	46.6	45.0	48.8	50.5
Zou <i>et al.</i> [37] $\star$	40.8	52.2	38.9	46.6	<b>63.0</b>	49.7	57.5	47.9	46	58.2	49.4	<b>50.4</b>	45.7	45.4	49.2	49.4
Ours	40.6	52.9	39.0	46.4	66.5	49.9	57.1	47.8	<b>44.8</b>	58.6	48.8	51.3	45.2	44.9	49.3	49.5
Ours $\star$	<b>39.5</b>	<b>50.5</b>	<b>37.9</b>	<b>46.2</b>	63.9	<b>49.6</b>	<b>56.5</b>	<b>46.2</b>	44.9	<b>56.5</b>	<b>48.0</b>	51.0	45.1	<b>42.8</b>	<b>48.6</b>	<b>48.5</b>

GT	WalkT.	WalkD.	Walk	Wait	SitD.	Smoke	Sit	Pose	Purchu.	Photo	Greet	Phone	Eat	Dire.	Discu.	Avg.
Ci <i>et al.</i> [5]	34.2	38.2	31.3	38.4	<b>39.5</b>	34.4	36.2	39.8	32.5	42.5	37.8	34.6	29.7	36.3	38.8	36.3
Lutz <i>et al.</i> [19]	29.1	35.0	28.7	<b>33.5</b>	40.1	33.8	37.1	37.1	31.3	39.0	33.4	<b>33.5</b>	30.2	31.0	36.6	34.0
Zeng <i>et al.</i> [31]	<b>27.1</b>	34.4	<b>26.5</b>	34.9	45.9	33.3	38.9	<b>35.1</b>	<b>29.5</b>	42.5	<b>31.7</b>	<b>33.5</b>	<b>27.6</b>	32.9	<b>34.5</b>	33.9
Ours	28.5	<b>34.3</b>	27.5	34.5	41.8	<b>33.2</b>	<b>34.3</b>	37.1	31.9	<b>37.7</b>	33.2	33.9	30.0	<b>30.7</b>	35.9	<b>33.7</b>

Table 4.2: P-MPJPE comparisons on Human3.6M. CPN with 17 human joints used as input.  $\star$  indicates refinement.

CPN	WalkT.	WalkD.	Walk	Wait	SitD.	Smoke	Sit	Pose	Purch.	Photo	Greet	Phone	Eat	Dire.	Disc.	Avg.
Ci <i>et al.</i> [5]	37.0	43.5	32.2	39.9	62.1	43.1	49.1	38.2	37.6	51.1	41.0	41.9	38.0	36.9	41.6	42.2
Cai <i>et al.</i> [1] $\star$	34.7	42.7	31.0	36.8	51.7	41.3	47.6	37.9	35.6	46.8	41.7	40.7	38.2	36.8	38.7	40.2
Pavlo <i>et al.</i> [25]	34.8	43.1	<b>30.3</b>	36.9	53.4	41.4	46.8	37.1	35.4	45.9	41.7	40.1	38.0	36.0	38.7	40.0
Zou <i>et al.</i> [37] $\star$	33.9	41.7	30.7	35.6	<b>51.2</b>	40.5	<b>46.4</b>	37.0	35.4	<b>44.5</b>	40.5	<b>39.2</b>	<b>36.3</b>	35.7	38.6	39.1
Ours	<b>33.4</b>	<b>41.6</b>	30.4	<b>35.5</b>	52.9	<b>40.4</b>	46.8	<b>36.0</b>	<b>34.3</b>	44.7	<b>39.7</b>	39.6	36.7	<b>34.9</b>	<b>38.2</b>	<b>39.0</b>

Table 4.3: MPJPE comparisons on Human3.6M. Top table: CPN with 16 human joints used as input. Bottom table: GT 2D with 16 human joints used as input.

CPN	WalkTo.	WalkDo.	Walk	Wait	SitDo.	Smoke	Sit	Pose	Purchu.	Photo	Greet	Phone	Eat	Direc.	Discu.	Avg.
Liu <i>et al.</i> [16]	43.7	54.5	40.3	50.1	71.1	51.5	60.4	49.2	46.0	67.1	50.7	55.5	47.3	46.3	52.2	52.4
Xu <i>et al.</i> [29]	44.1	53.9	39.9	48.6	71.5	51.4	59.7	48.5	46.3	66.1	50.9	54.9	<b>47.5</b>	<b>45.2</b>	49.9	51.9
Zhao <i>et al.</i> [34]	43.1	54.1	<b>39.7</b>	48.7	70.0	51.6	60.2	48.0	47.1	65.0	<b>50.0</b>	54.9	48.0	<b>45.2</b>	50.8	51.8
Ours	<b>41.8</b>	<b>53.6</b>	39.9	<b>47.2</b>	<b>68.0</b>	<b>51.1</b>	<b>58.9</b>	<b>46.9</b>	<b>45.8</b>	<b>59.0</b>	<b>50.0</b>	<b>52.9</b>	47.8	45.8	<b>49.1</b>	<b>50.5</b>

GT	WalkTo.	WalkDo.	Walk	Wait	SitDo.	Smoke	Sit	Pose	Purchu.	Photo	Greet	Phone	Eat	Direc.	Discu.	Avg.
Liu <i>et al.</i> [16]	32.0	38.6	29.6	38.5	47.7	37.4	40.3	39.7	34.9	45.0	36.3	37.5	33.0	36.8	40.3	37.8
Xu <i>et al.</i> [29]	30.7	36.8	27.9	36.7	45.5	35.4	38.4	37.3	31.7	43.2	35.3	35.8	31.0	35.8	38.1	35.8
Zhao <i>et al.</i> [34]	30.6	36.1	<b>27.4</b>	35.7	46.2	34.2	38.0	<b>35.2</b>	<b>31.4</b>	43.3	34.4	34.7	30.4	32.0	38.0	35.2
Ours	<b>29.8</b>	<b>34.6</b>	28.2	<b>34.4</b>	<b>40.7</b>	<b>34.1</b>	<b>36.0</b>	38.2	32.0	<b>37.8</b>	<b>34.3</b>	<b>34.0</b>	<b>30.0</b>	<b>31.3</b>	<b>36.7</b>	<b>34.1</b>

learning rate is initially assigned a value of 0.000025 and exponentially decreased with a factor of 0.98 for each epoch. All experimental procedures are performed on a machine featuring a single NVIDIA RTX 2080 GPU.

## 4.4 Comparison with the State-of-the-art

Table 4.1, Table 4.2, and Table 4.3 provide comparisons between the AMPose and state-of-the-art models using CPN and GT keypoints as the input on Human3.6M. Despite the additional application of a pose refinement module [1] in some methods to enhance performance, the MPJPE of the AMPose prior to refinement is also reported to ensure fair





Table 4.4: Evaluation outcomes from conducting a cross-dataset comparison on 3DHP.

Method	Outdoor	noGS	GS	AUC	PCK
Ci <i>et al.</i> [5]	77.3	70.8	74.8	36.7	74.0
Zhao <i>et al.</i> [34]	74.1	77.9	80.1	43.8	79.0
Liu <i>et al.</i> [16]	80.1	80.5	77.6	47.6	79.3
Xu <i>et al.</i> [29]	75.2	81.7	81.5	45.8	80.1
Zou <i>et al.</i> [37]	85.7	86.0	<b>86.4</b>	53.7	86.1
Ours	<b>87.4</b>	<b>87.5</b>	86.1	<b>55.2</b>	<b>87.0</b>

Table 4.5: The comparison of results on the 3DHP dataset.

Method	PCK	AUC	FLOPs (M)	Param. (M)	MPJPE
Li <i>et al.</i> (9 frames) [14]	93.8	63.3	1030	18.9	58.0
Zhang <i>et al.</i> (1 frame) [32]	94.2	63.8	645	33.7	57.9
Zhang <i>et al.</i> (27 frames) [32]	94.4	66.5	645	33.7	54.9
Shan <i>et al.</i> (81 frames) [26]	97.9	75.8	493	<b>5.4</b>	32.2
Ours (1 frame)	<b>98.0</b>	79.1	<b>312</b>	18.3	30.8
Ours (9 frames)	<b>98.0</b>	81.2	<b>312</b>	18.4	28.0
Ours (27 frames)	<b>98.0</b>	<b>81.3</b>	313	18.4	<b>27.9</b>

comparisons. The quantitative outcomes depicted in Table 4.1, Table 4.2, and Table 4.3 demonstrate that AMPose outperforms all previous methods.

In Table 4.1, AMPose demonstrates superiority in most actions compared to previous methods when CPN is employed as the 2D detector. Specifically, our estimated 3D pose for the photographing and directing action shows 2.9% and 4.5% improvement of MPJPE over the previous best results, respectively. In addition, our method achieves accurate predictions for the photographing actions, even when using 2D GT poses as input. We also observe that the accuracy of photographing action has been improved by 3.3% in comparison to the previous leading approach.

The 3DHP dataset includes two distinct types of comparison methods. The single-frame methods are typically evaluated by cross-dataset, while the video-based methods are trained directly on 3DHP’s training set. To comprehensively assess the effectiveness of AMPose, we conduct both types of comparisons. Table 4.4 displays the AMPose method’s ability to generalize, given that it is trained solely on the Human3.6M dataset without

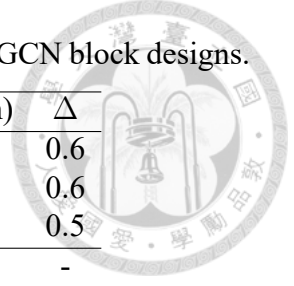


Table 4.6: The outcomes obtained from the different variations of the GCN block designs.

Method	FLOPs (M)	Param. (M)	MPJPE (mm)	$\Delta$
Two residual	312.2	18.5	50.1	0.6
Transformer	289.8	17.0	50.1	0.6
ConvNeXt	289.9	17.0	50.0	0.5
Ours	312.2	18.3	49.5	-

any fine-tuning for 3DHP. We provide distinct outcomes for various backgrounds: outdoor, green screen (GS), and noGS. Our model achieves an 87% PCK and 55.2% AUC, surpassing existing approaches in challenging conditions such as noGS and outdoor environments. Furthermore, AMPose is trained on 3DHP’s training set to compare it with temporal methods. Table 4.5 shows the superior performance of our method, even with fewer Floating Point Operations (FLOPs). The video-based methods have higher computational demands for generating 3D pose outputs, which becomes evident when considering FLOPs. Consequently, our proposed method is advantageous in real-time scenarios where computational efficiency is crucial. Additionally, we observe that video-based methods struggle to handle complex conditions, such as 3DHP’s test set.

According to [13], we extend our method into a temporal model by incorporating sequential data. In this method, the coordinates of the joints across different time frames are concatenated within their respective joints. Assuming we have  $T$  temporal sequences and  $J$  joint nodes, the input data can be interpreted as  $J$  individual tokens. In specific terms, the input  $X \in R^{T \times J \times 2}$  is reorganized as  $X' \in R^{J \times (2T)}$ . This allows for the integration of temporal data into single-frame models without the need for structural modifications. The outcome of this extension is presented in Table 4.5, where we observed an improvement in accuracy. Specifically, when  $T$  is set to 27, the MPJPE is reduced to 27.9mm.

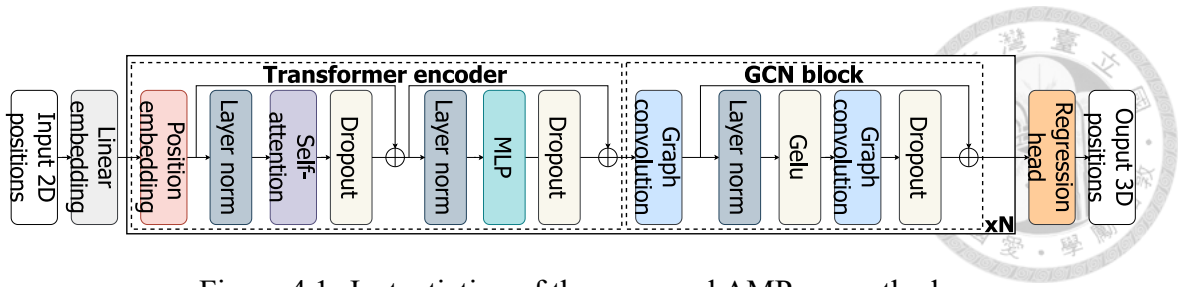


Figure 4.1: Instantiation of the proposed AMPose method.

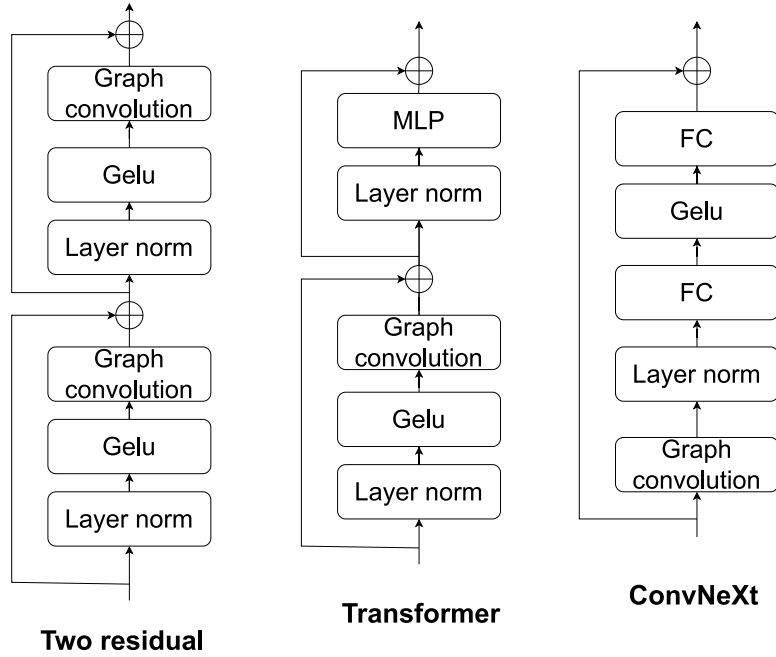


Figure 4.2: Variety of configurations of the GCN block.

## 4.5 Ablation Study

In the ablation study, we examine different configurations of the GCN block. In Figure 4.2, we illustrate the various designs of the GCN block, which integrate graph convolution into ConvNeXt [17] and the Transformer [7]. Table 4.6 presents the corresponding FLOPs and results. As shown in Figure 4.1, our proposed design, featuring a plain GCN followed by GCN with the residual, outperforms the other designs.

We conduct the ablation study on Human3.6M, using CPN with 16 keypoints as the input of our proposed model. In this study, we investigate the ramifications of varying the orders of Transformer encoders and GCN blocks. The model depth is consistently set to

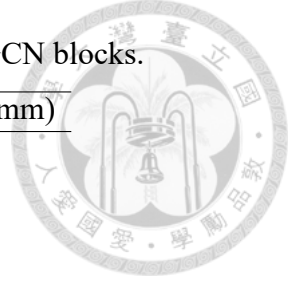


Table 4.7: Ordering Analysis of Transformer encoders and GCN blocks.

Order	MPJPE (mm)
5 Transformers followed by 5 GCN blocks	52.0
5 GCN blocks followed by 5 Transformers	57.2
Alternately (GCN blocks first)	50.9
Alternately (Transformers first)	50.5

Table 4.8: Ablation study for splitting neighbor nodes.

Group	MPJPE (mm)
4 groups {(center), (farther), (closer), (symmetry)}	51.1
3 groups {(center), (farther, closer), (symmetry)}	51.0
2 groups {(center), (farther, closer, symmetry)}	51.6
1 group {(center, farther, closer, symmetry)}	53.6
3 groups {(center), (farther), (closer)}	50.5
2 groups {(center), (farther, closer)}	50.8

5 in our experimental setup for this investigation. The results, presented in Table 4.7, reveal that the alternately mixed architecture demonstrates superior performance compared to other configurations. The impact of splitting the neighbor nodes is examined in Table 4.8. Our findings indicate that dividing the neighbor nodes into three groups, without considering symmetrically-related neighbor nodes, yields the most favorable outcome. The influence of different model sizes on our proposed method is evaluated and summarized in Table 4.9. Since Zhao *et al.* [34] emphasize their performance with fewer parameters, we use a similar size for a fair comparison, setting the depth and channel to 4 and 108, respectively. After adjustment, our model size is comparable to that of Zhao *et al.*'s approach. MPJPE of our method is 51.3mm, while Zhao *et al.*'s approach achieves 51.8mm. These results demonstrate our proposed method's effectiveness and efficiency, even when employing a smaller model size.

Additionally, we investigate the effect of adding the toe direction to the input data and using it as a positional encoding at the start of the Transformer encoder, resulting in an MPJPE of 50.7mm. While this is 0.2mm worse than the original MPJPE of 50.5mm

Table 4.9: Quantifying the impact: Ablation study on depth and channel settings.

Depth	Channel	Param. (M)	MPJPE (mm)
4	108	0.67	51.3
4	128	0.93	51.0
5	128	1.17	51.2
4	256	3.70	51.0
4	384	8.31	50.9
5	512	18.38	50.5
6	512	22.13	50.8



(refer to Table 4.3), our experiments reveal that enhancing the precision of the 2D pose is a more effective strategy, leading to a substantial reduction of the MPJPE to 34.1mm (refer to Table 4.3).

## 4.6 Qualitative Results

Visual representations of 3D human pose estimated by the AMPose method, trained on Human3.6M, are presented in Figure 4.3, Figure 4.4, and Figure 4.5. The 3DHP dataset’s test set comprises outdoor scenes and backgrounds without a green screen. By utilizing 2D pose models, the lifting method can effectively mitigate the impact of background noise. As depicted in Figure 4.4 and Figure 4.5, our proposed method can accurately predict 3D poses across various challenging scenarios.

In contrast to the Human3.6M dataset, we observe notable variations in keypoint definitions within the 3DHP dataset. Specifically, the nose appears flatter, and the shoulder exhibits a narrower appearance. These observed dissimilarities disclose the importance of accounting for dataset-specific variations when developing pose estimation models, as subtle differences in keypoint definitions can affect the accuracy of the predictions.

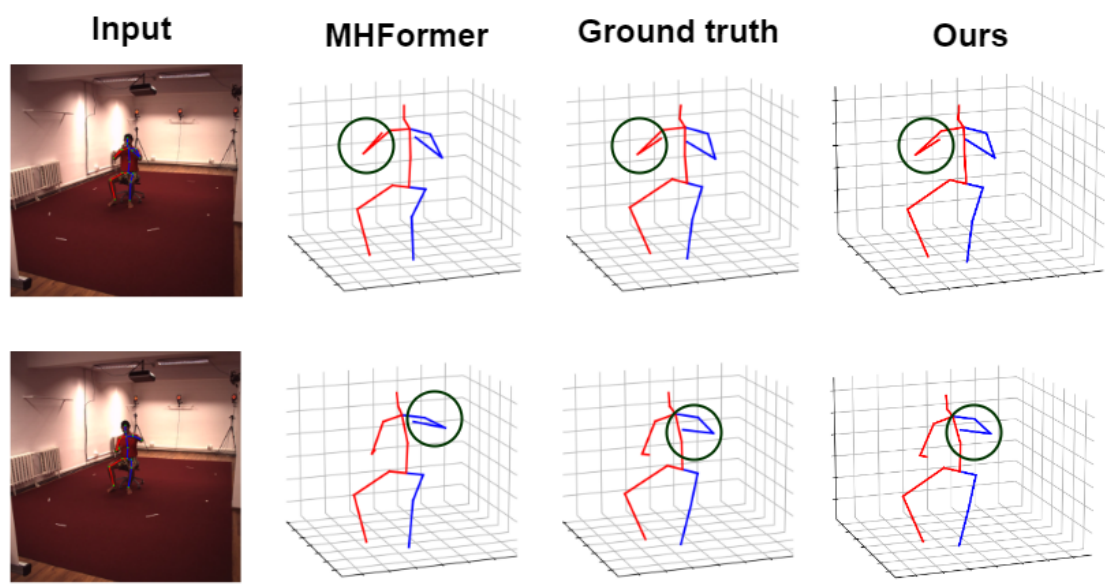


Figure 4.3: Graphical display on Human3.6M.

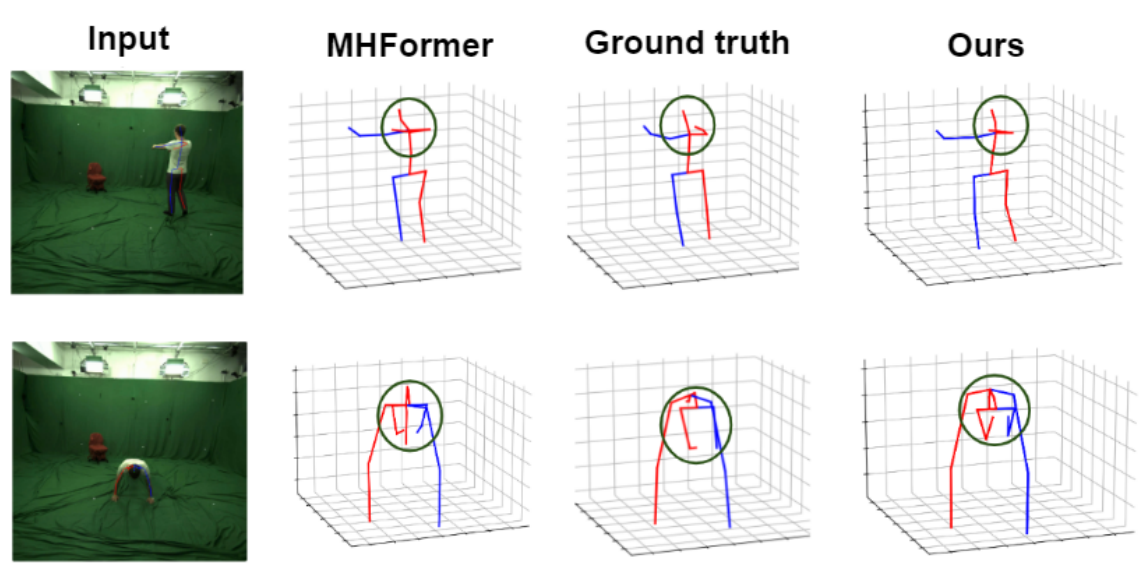


Figure 4.4: Graphical display of indoor scenes from 3DHP.

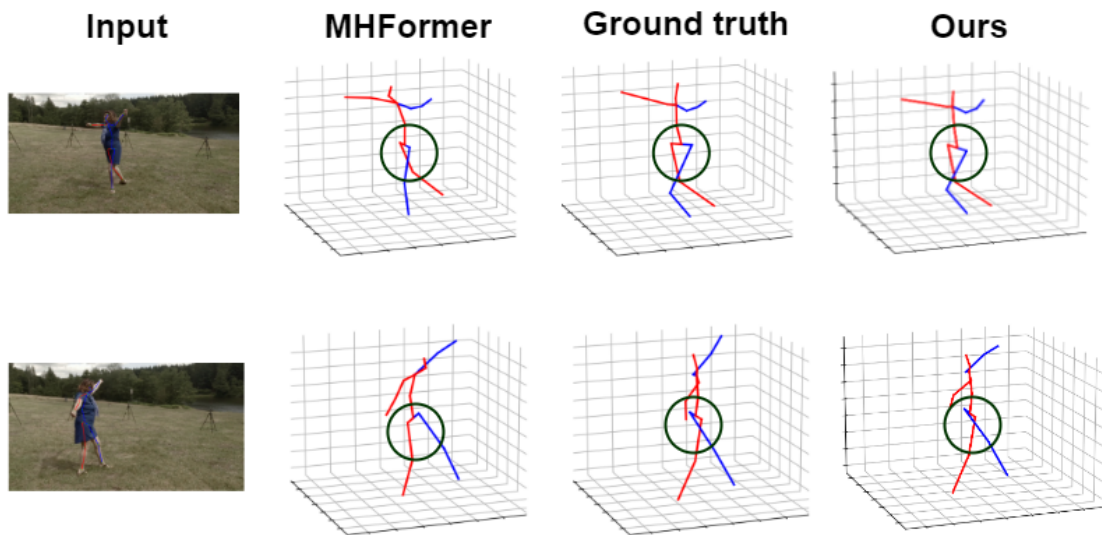


Figure 4.5: Graphical display of outdoor scenes from 3DHP.



## Chapter 5 Conclusion

In this work, we have presented a novel method called AMPose for 3D HPE, which alternately combines the Transformer encoder and GCN blocks. Our approach aims to modularize human joint relations into global and physically connected relations, which the Transformer encoder and GCNs can efficiently capture. The proposed method shows superior accuracy and generalization ability performance by comparing with state-of-the-art models on the Human3.6M and 3DHP datasets.

In the ablation study, we provide the design rationale behind AMPose and validate the advantages gained from combining the Transformer encoder and GCNs.

In future research, we propose investigating the application of the AMPose method to 3D human mesh estimation, which encompasses both pose estimation and body shape estimation. By harnessing the distinctive capabilities of the Transformer encoder and GCNs, our approach holds the potential for advancing the estimation of 3D human mesh.



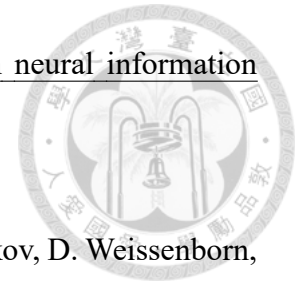




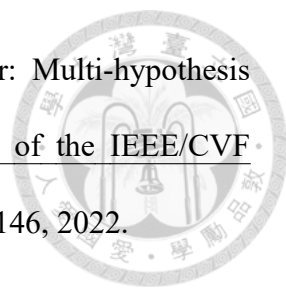
## References

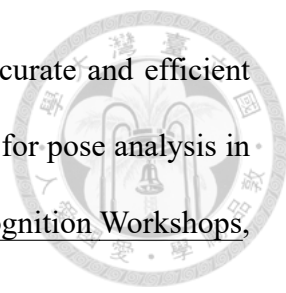
- [1] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2272–2281, 2019.
- [2] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, and J. Luo. Anatomy-aware 3d human pose estimation in videos. arXiv preprint arXiv:2002.10322, 2020.
- [3] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7103–7112, 2018.
- [4] S. Chun, S. Park, and J. Y. Chang. Learnable human mesh triangulation for 3d human pose and shape estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2850–2859, 2023.
- [5] H. Ci, C. Wang, X. Ma, and Y. Wang. Optimizing network structure for 3d human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2262–2271, 2019.
- [6] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks

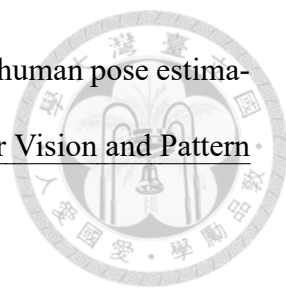
on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29, 2016.

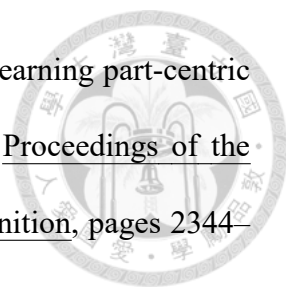


- [7] A. Dosovitskiy, L. Beyer, A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.
- [8] I. M. Hakim, H. Zakaria, K. Muslim, and S. I. Ihsani. 3d human pose estimation using blazepose and direct linear transform (dlt) for joint angle measurement. In International Conference on Artificial Intelligence in Information and Communication, pages 236–241, 2023.
- [9] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7):1325–1339, 2014.
- [10] K. Isakov, E. Burkov, V. Lempitsky, and Y. Malkov. Learnable triangulation of human pose. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 7718–7727, 2019.
- [11] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.
- [12] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, and W. Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. IEEE Transactions on Multimedia, 25:1282–1293, 2023.
- [13] W. Li, H. Liu, T. Guo, H. Tang, and R. Ding. Graphmlp: A graph mlp-like architecture for 3d human pose estimation. arXiv preprint arXiv:2206.06420, 2022.

- 
- [14] W. Li, H. Liu, H. Tang, P. Wang, and L. Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 13137–13146, 2022.
- [15] Y.-C. Li, C.-T. Chang, C.-C. Cheng, and Y.-L. Huang. Baseball swing pose estimation using openpose. In IEEE International Conference on Robotics, Automation and Artificial Intelligence, pages 6–9, 2021.
- [16] K. Liu, R. Ding, Z. Zou, L. Wang, and W. Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In Proceedings of the European conference on computer vision, pages 318–334, 2020.
- [17] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11966–11976, 2022.
- [18] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. ACM Trans. Graph., 34(6), nov 2015.
- [19] S. Lutz, R. Blythman, K. Ghosal, M. Moynihan, C. Simms, and A. Smolic. Jointformer: Single-frame lifting transformer with error prediction and refinement for 3d human pose estimation. arXiv preprint arXiv:2208.03704, 2022.
- [20] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 2640–2649, 2017.
- [21] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In International Conference on 3D Vision, pages 506–516, 2017.

- 
- [22] S. Park, J. Y. Chang, H. Jeong, J.-H. Lee, and J.-Y. Park. Accurate and efficient 3d human pose estimation algorithm using single depth images for pose analysis in golf. In IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 105–113, 2017.
- [23] S. Park, J. Hwang, and N. Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. arXiv preprint arXiv:1608.03075, 2016.
- [24] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 459–468, 2018.
- [25] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 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, pages 7753–7762, 2019.
- [26] W. Shan, Z. Liu, X. Zhang, S. Wang, S. Ma, and W. Gao. P-stmo: Pre-trained spatial temporal many-to-one model for 3d human pose estimation. In Proceedings of the European conference on computer vision, pages 461–478, 2022.
- [27] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83–98, 2013.
- [28] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei. Integral human pose regression. In Proceedings of the European conference on computer vision, pages 529–545, 2018.

- 
- [29] T. Xu and W. Takano. Graph stacked hourglass networks for 3d human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16100–16109, 2021.
- [30] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.
- [31] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, and S. Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. In Proceedings of the European conference on computer vision, pages 507–523, 2020.
- [32] J. Zhang, Z. Tu, J. Yang, Y. Chen, and J. Yuan. Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13222–13232, 2022.
- [33] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3420–3430, 2019.
- [34] W. Zhao, W. Wang, and Y. Tian. Graformer: Graph-oriented transformer for 3d pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20406–20415, 2022.
- [35] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding. 3d human pose estimation with spatial and temporal transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11636–11645, 2021.

- 
- [36] K. Zhou, X. Han, N. Jiang, K. Jia, and J. Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2344–2353, 2019.
- [37] Z. Zou and W. Tang. Modulated graph convolutional network for 3d human pose estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 11457–11467, 2021.