Primal-Dual Graph Attention Networks for 3D Human Pose Estimation

BY

HARSH YADAV
B.Tech., APJ Abdul Kalam Technical University, 2019

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2021

Chicago, Illinois

Defense Committee:

Wei Tang, Chair and Advisor
Xinhua Zhang
Cornelia Caragea

This thesis is dedicated to my parents. Without their unending support, this would have not been possible.

ACKNOWLEDGEMENTS

I would like to thank my advisor – Prof Wei Tang, for his constant support throughout my duration at UIC. His guidance has really helped me enter the world of computer science research, and this resulting thesis would not have been possible without him.

I would also like to express my gratitude towards Prof Zhang and Prof Caragea for taking the time out of their schedule to join the committee for the defense of this thesis.

My thanks to Zhiming for always being available for help with my research.

TABLE OF CONTENTS

CHAPTER **PAGE**

1. INTRODUCTION 1
   - 1.1. Human pose estimation: 1
   - 1.2. Motivation 1
     - 1.2.1. Video games and animation: 1
     - 1.2.2. Gesture recognition 2
     - 1.2.3. Security and threat detection 2
   - 1.3. Graph convolution networks 2
   - 1.4. Contributions 2
2. RELATED WORK 4
   - 2.1. 3D Human pose estimation 4
   - 2.2. Graph convolution networks 4
3. GRAPH CONVOLUTION NETWORKS 6
   - 3.1. Need for Neural Networks for graphs 6
   - 3.2. Graph networks in Human Pose Estimation 6
   - 3.3. Vanilla GCN 6
   - 3.4. SemGCN as a baseline 7
4. GRAPH ATTENTION NETWORK 9
5. CHANNEL ATTENTION 12
   - 5.1. Squeeze 12
   - 5.2. Excitation 12
6. PRIMAL DUAL GRAPH ATTENTION NETWORKS 14
   - 6.1. Intuition 14
   - 6.2. Primal branch 14
     - 6.2.1. Feature mask attention 15
     - 6.2.2. Feature aggregation 15
     - 6.2.3. Non-local layer 15
   - 6.3. Dual Branch 16
     - 6.3.1. Node-mask attention 16
     - 6.3.2. Non-local layer 17
   - 6.4. Primal-Dual Block 17

7. NETWORK ARCHITECTURE 19
8. 3D HUMAN POSE ESTIMATION 20
8.1. Problem formulation 20
8.2. Dataset 21
8.3. Loss Function 21
8.4. Evaluation metrics 22
9. IMPLEMENTATION DETAILS 24
9.1. Training 24
10. RESULTS 25
10.1. Ablation Study 25
10.1.1. Variants of Primal branch 25
10.1.2. Variants of Dual branch 26
10.1.3. Primal-Dual block study 26
10.2. Comparison with baseline 27
10.3. Comparison with the state of the art 27
10.4. Qualitative results 27
11. CONCLUSION 31
12. CITED LITERATURE 32
13. VITA 35

LIST OF TABLES

TABLE		PAGE
I.	COMPARISON OF DIFFERENT ELEMENTS OF THE PRIMAL BRANCH…….	25
II.	COMPARISON OF DIFFERENT ELEMENTS OF THE DUAL BRANCH…….……	26
III.	COMPARISON OF INDIVIDUAL BRANCHES OF PROPOSED NETWORK…….	26
IV.	COMPARISON OF PROPOSED NETWORK WITH BASELINE...	27
V.	QUANTITATIVE COMPARISONS UNDER PROTOCOL #1…..	28
VI.	QUANTITATIVE COMPARISONS UNDER PROTOCOL #2…..	29

TABLE OF FIGURES

FIGURE		PAGE
1.	Attention Coefficients in GAT…….	10
2.	Aggregation of features in GAT…..…….	10
3.	Squeeze and Excitation block……...	13
4.	Structure of the proposed Primal Branch…….	16
5.	Proposed Dual branch. ……...……..	17
6.	Proposed Primal-Dual block ……	18
7.	Model architecture……	19
8.	Human pose as a graph…….	21
9.	Samples from the Human3.6M dataset…….	22
10.	Qualitative results on Human3.6M dataset……	30

LIST OF ABBREVIATIONS

GCN	Graph Convolution Network
GAT	Graph Attention Network
DNN	Deep Neural Network
GNN	Graph Neural Network
HPE	Human Pose Estimation
SE	Squeeze and Excitation
ReLU	Rectified Linear Unit
MPJPE	Mean Per Joint Position Error
P-MPJPE	Mean Per Joint Position Error after Procrustes alignment

SUMMARY

We propose Primal-Dual Graph Attention networks, a novel model for the task of 3D pose estimation. It builds upon Graph Convolution Networks (GCNs) by tackling the shortcomings present in vanilla GCNs. The core of the network, the Primal-Dual block, employs the use of multiple attentions to obtain improved feature representations by modeling feature dependencies in both the spatial and the channel domain.

After conducting multiple ablation studies for the individual elements of the proposed network, it is found that the network is able to generate a superior model for 2D to 3D pose upliftment.

1. INTRODUCTION

1.1. Human pose estimation:

HPE is the task of finding the posture of the human body with given sensory information, usually an image or a video. It is one of the key areas of research in computer vision. With the recent rise in popularity of deep neural networks, the same has been applied to the task of pose estimation as well. Furthermore, 3D human pose estimation is concerned with predicting the pose in 3-dimensional space. This turns out to be a quite difficult problem as there can be multiple 3D poses from a given 2D pose.

1.2. Motivation

Research in the area of HPE is quite interesting, the reason being its varied applications. Perhaps one of the most popular implementations of this technology is in the Microsoft Kinect (Kinect, 2021) game console which lets players control the characters in the games using their body motions. Other areas of application include movies, video games animation, gesture control, security and threat detection, automotive control, etc.

1.2.1. Video games and animation:

Currently, most video games contain character models which are painstakingly animated by hand or require expensive motion capture equipment which is also a hassle to use. With proper pose detection and tracking, the animation could be self-generated by tracking an actor in plain clothes.

1.2.2. Gesture recognition

With the advent of fast and accurate pose tracking, devices like smartphones, TVs, etc. could be controlled without any sort of interaction medium like a touch screen or a remote.

1.2.3. Security and threat detection

With proper pose detection employed by security cameras, threats could be classified and alerts generated accordingly without human surveillance.

1.3. Graph convolution networks

Vanilla Convolution Neural Networks (CNNs) lack the ability to perform operations on data that is structured as a graph. To this endeavor, Graph Convolution Networks (GCNs) (Marco Gori, 2005) (Welling, 2017) (Franco Scarselli, 2009) have been proposed. GCNs are good for the task of pose estimation since the human body can be effectively modeled as a graph.

However, there are a few drawbacks of vanilla GCNs. Firstly, they have a fixed graph structure that is dictated by the affinity matrix, as such they are unable to learn the specific importance of different edges of the graph. Secondly, they only consider the immediate neighbors when performing the convolution operation. Due to this, they are unable to generate effective long-range interactions between the nodes. Thirdly, a vanilla GCN only aggregates features in the spatial domain but fails to capture relationships in the channel domain.

1.4. Contributions

To tackle the shortcomings of vanilla GCNs listed above, we propose a novel architecture called Primal-Dual Graph Attention Networks. These networks successfully

attempt to model the interaction of features both in the spatial and channel domain, while at the same time providing a flexible graph structure and effectively modeling long-range dependencies. The contributions of this research are twofold:

- We introduce a novel Primal-Dual block, which forms the backbone of our network and is responsible for tackling the problems of vanilla GCNs listed above.
- We demonstrate the effectiveness of the individual elements of the Primal-Dual block via three ablation studies and show how different attention mechanisms improve the performance of the network.

2. RELATED WORK

2.1. 3D Human pose estimation

One of the earliest attempts at 3D pose estimation from 2D was (Chen, 1985). Traditional methods involve using hand-crafted features (Triggs, 2006) (Catalin Ionescu F. L., 2011) (Gregory Rogez, 2008) or using nearest neighbors (Ankur Gupta, 2014) (Jiang, 2010) to predict 3D poses. However, with the recent boom in the capabilities of neural networks, deep learning methods for pose estimation have become quite popular. "Recently it has been proven that 2D pose information is crucial for 3D pose estimation" (Zhao, 2019). (Julieta Martinez, 2017) demonstrated a simple model for estimating the 3D pose solely from the 2D pose.

Some other methods employ the use of temporal information present in the input for generating 3D poses. However, for the purposes of this research, we are only concerned with single-frame 3D human pose estimation.

Graph-based skeleton models for pose estimation were first proposed by (Felzenszwalb, 2005). Such models were used by (Cao, 2017) for 2D and 3D pose estimation. Another type of model used for pose estimation is the contour-based model. Such models were used in (Ju, 1996) (T.F. Cootes, 1995). Volume-based models are also popular. They use geometric shapes, meshes, etc. to model the pose of the human body. Instances of their use are (Sidenbladh, 2000) (Anguelov, 2005) (Matthew Loper, 2015).

2.2. Graph convolution networks

Regular neural networks are unable to efficiently model graph data. To this goal, GNNs were formulated in (Marco Gori, 2005) (Welling, 2017) (Franco Scarselli, 2009). Similarly, GCNs are a generalization of CNNs for the purpose of computations on graph

data. Generally, there are two types of GCNs: spectral and spatial. GCNs have been very successful for computer vision applications such as object detection (Hang Xu, 2019), action recognition (Rui Zhao, 2019), tracking (Junyu Gao, 2019), modeling temporal information (Gupta, 2018) (Sijie Yan, 2018). Here we apply GCN for the task of 3D human pose estimation. (Sami Abu-El-Haija, 2019) makes use of a high-order GCN to compute relationships between nodes not in the direct vicinity, thus increasing the receptive field of the convolution. In contrast, we employ the use of non-local layers (He, 2018) to model long-range dependencies.

3. GRAPH CONVOLUTION NETWORKS

3.1. Need for Neural Networks for graphs

Vanilla neural networks are great for data where there is no structure between the input features. However, a large amount of data in the real world has an inherent structure, which needs to be modeled in the network. Examples of these are social networks, geographic locations, DNA sequences, webpages, etc. In these cases, the data is modeled in the form of a graph – with nodes containing information and edges connecting related nodes. For a neural network to work efficiently on this data, such a structure needs to be explicitly modeled in the network.

GCNs are used to apply convolution operations on general graphs instead of on images that have a fixed grid-like structure. GCNs are divided into two categories – spectral-based and non-spectral-based. For the purposes of our research, we are concerned with spatial GCNs.

3.2. Graph networks in Human Pose Estimation

The task of human pose estimation is very well suited to graph networks. The human body can be intuitively modeled as a graph where the nodes represent joints in the body and the edges correspond to bones or connected joints (Felzenszwalb, 2005). This type of representation has been extensively used previously for HPE (Cao, 2017).

3.3. Vanilla GCN

"Let G = {V, E} denote a graph where V is the set of K nodes and E are edges, while $\vec{x}_i^{(l)} \in \mathbb{R}^{D^{(l)}}$ and $\vec{x}_i^{(l+1)} \in \mathbb{R}^{D_{(l+1)}}$ are the representations of node i before and after the l-th convolution respectively" (Zhao, 2019). $D_{(l)}$ is the number of dimensions of the

input feature vector, and $D_{(l+1)}$ is the number of dimensions of the output feature vector. A weight matrix W transforms the feature vector to $D_{(l+1)}$ dimensions. Then, the features from neighboring nodes are aggregated using the adjacency matrix. This operation is followed by an activation function (ReLU (Hinton, 2010)).

The vanilla GCN updates the node features via:

$$X^{(l+1)} = \sigma\left(WX^{(l)}\tilde{A}\right) \quad 1$$

where $X^{(l)}$ and $X^{(l+1)}$ are the set of node features before and after the convolution operation respectively, $\tilde{A}$ is normalized from A, the adjacency matrix of the graph. A is K*K matrix, where K is the number of nodes in the graph. A(i,j) = 1 if node j is in the neighborhood of node i and A(i,i) = 1.

3.4. SemGCN as a baseline

Here we describe configuration 1 of the GCN model used by (Zhao, 2019). A new weight matrix $M \in \mathbb{R}^{K*K}$ is added to the Equation 1, which now becomes:

$$X^{(l+1)} = \sigma\left(WX^{(l)}\rho_i(M \odot A)\right) \quad 2$$

"where ρ_i is Softmax nonlinearity which normalizes the input matrix across all choices of node i, $\odot$ is an elementwise operation that returns m_{ij} if $a_{ij} = 1$ or negatives with large exponents saturating to zero after ρ_i" (Zhao, 2019).

Additionally, (Nanyang Wang, 2018) used separate transformation matrices for aggregations of the features for self-transformation and neighbor-transformation. We also employ this method in the actual implementation. Equation 1 now becomes:

$$X^{(l+1)} = \sigma\left(I \otimes W_0X^{(l)}\tilde{A} + (1-I) \otimes W_1X^{(l)}\tilde{A}\right) \quad 3$$

Where I is the identity matrix. Equation 2 is modified similarly:

$$X^{(l+1)} = \sigma\left(I \otimes W_0X^{(l)}\rho_i(M \odot A) + (1-I) \otimes W_1X^{(l)}\rho_i(M \odot A)\right) \quad 4$$

We adopt this method used in configuration 1 of (Zhao, 2019) as our baseline which we will refer to as SemGCN.

4. GRAPH ATTENTION NETWORK

In this section, we describe the "graph attentional layer" proposed by (Petar Velickovi ˇ c, 2018). The basis of the GAT layer is formed by the concept of 'edge-attention', where the connections between any two nodes are attended over. This type of attention helps to emphasize the impact of neighboring nodes. "As opposed to GCNs, [this] model allows for (implicitly) assigning different importances to nodes of a same neighborhood, enabling a leap in model capacity". Additionally, this attention method is independent of the structure of the graph and is applied individually to edges (Petar Velickovi ˇ c, 2018).

Given two nodes i and j, where x_i and x_j are the node representations of nodes i and j respectively, an attention coefficient can be computed as follows:

$$e_{ij} = a\left(W\vec{x}_i, W\vec{x}_j\right) \quad 5$$

where W is a transformation matrix and $a: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$ is a function to compute the attention coefficient between the two nodes of *D* dimensions each. Equation 5 gives the relationship between node j and node i. The coefficients are then normalized by using softmax operation:

$$\alpha_{ij} = softmax_j\left(e_{ij}\right) \quad 6$$

The function *a* is a "single-layer feedforward neural network parametrized by a weight vector $\vec{a} \in \mathbb{R}^{2D}$". The two nodes i and j are concatenated after transformation and a nonlinearity is applied. Finally, the weight vector $\vec{a}$ is applied and α_{ij} – the normalized attention coefficient- is calculated.

$$\alpha_{ij} = softmax_j\left(\vec{a}^T[W\vec{x}_i||W\vec{x}_j]\right) \quad 7$$

where T is the transpose function and || represents concatenation. Figures 1 and 2 show the working of the GAT layer.

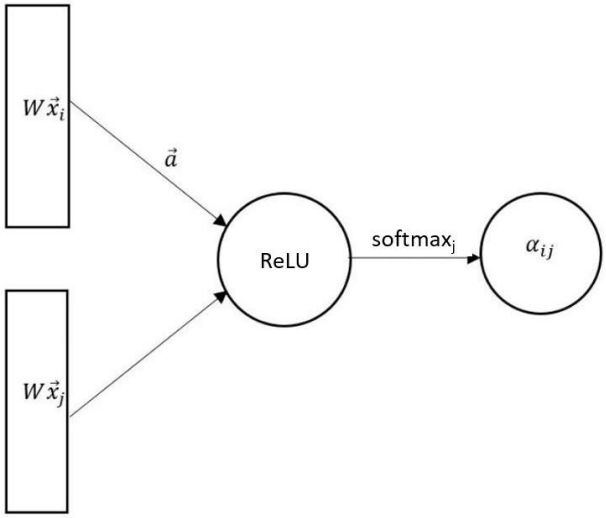

Figure 1 Attention Coefficients in GAT

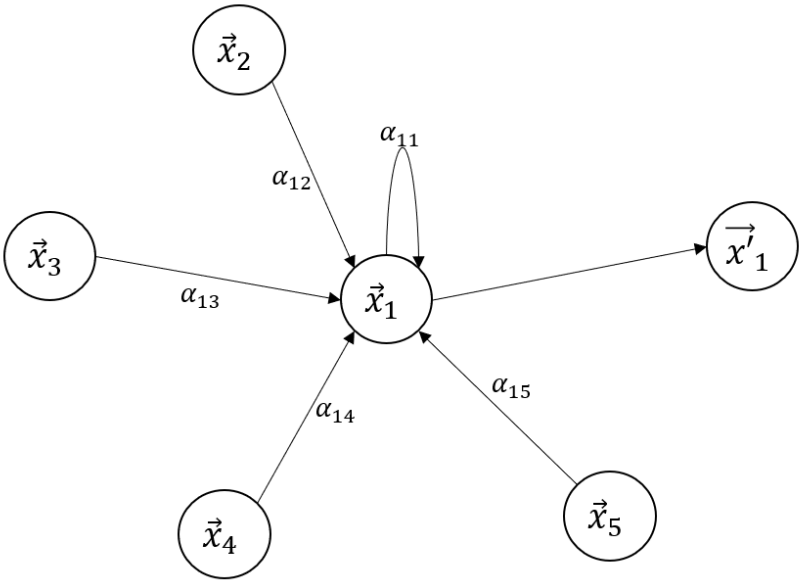

Figure 2 Aggregation of features in GAT

In the implementation of this function, we only compute the attention between the neighboring nodes. Thus, j is set to only take values in the neighborhood of i. By doing this, we encode the inherent structure of the graph by using the affinity matrix in the attention operation (masked attention).

Additionally, this GAT can also be viewed as a modification of the vanilla GCN. Instead of using the static affinity matrix of the graph in GCN, we encode the edge attention coefficients and use that as our affinity matrix, which now becomes dependent on the individual features of the node and its neighbors.

In (Petar Velickovi ˇ c, 2018), they also employ multiple attention heads for a single GAT layer. However, for the purposes of our research, we have fixed the number of heads to 1. We also only consider the immediate neighbors while computing the attention coefficients.

5. CHANNEL ATTENTION

In this section, we describe the squeeze-and-excitation block proposed by (Sun, 2018). This SE block employs channel attention to "[improve] the quality of representations produced by a network by explicitly modeling the interdependencies between the channels of its convolutional features" (Sun, 2018). The SE block consists of two parts. The squeeze operation takes the input features (K*D) and aggregates them across nodes. This aggregation is then followed by the excitation operation, which "aims to fully capture channel-wise dependencies" (Sun, 2018). This is implemented by applying some gating transformations to the embeddings created in the squeeze part, and then the original input features are scaled with the activations generated in the excitation step.

5.1. Squeeze

To model the dependencies in channels, we perform a squeeze operation to obtain condensed channel descriptors. This is done by average-pooling across the spatial dimensions. The result is a D dimensional embedding vector z.

$$z^c = \frac{1}{K}\sum_{i=1}^{K} x_i^c \qquad 8$$

where x_i^c is the c[th] feature of node i.

5.2. Excitation

Using the embeddings z created above, we apply transformations and non-linear activations to generate a gating vector. This is done by two weight matrices – $W_1 \in \mathbb{R}^{\frac{D}{r}\times D}$ and $W_2 \in \mathbb{R}^{D\times\frac{D}{r}}$ which form a bottleneck. This helps reduce the model size. r is the reduction ratio. The computation of gating vector s is as follows:

$$s = \sigma\big(W_2\delta(W_1 z)\big) \quad (9)$$

Here $s \in \mathbb{R}^D$ and δ is a ReLU non-linear activation function (Hinton, 2010). This is followed by a scaling operation which scales the input X by the vector s to generate new feature matrix X'.

$$x'^c = s^c x^c \quad (10)$$

The resulting feature matrix X' now contains the channel dependencies present in the original input. Figure 3 gives a pictorial representation of the SE block.

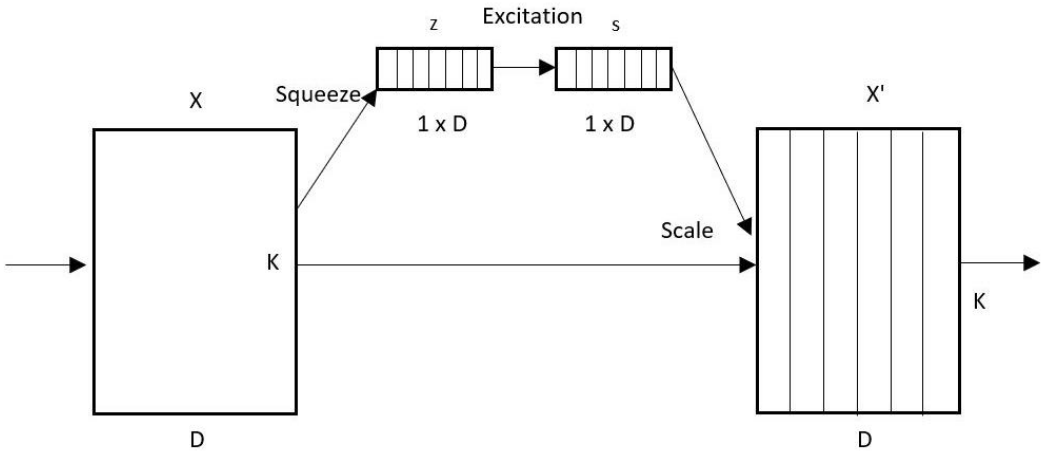

Figure 3. Squeeze and Excitation block

6. PRIMAL DUAL GRAPH ATTENTION NETWORKS

In this section, we describe the novel primal-dual block, that forms the backbone of the architecture we have created for the task of 3D human pose estimation.

6.1. Intuition

The idea behind this model is to have feature aggregations in both the spatial and channel dimensions. For this, we create two branches- the Primal branch and the Dual branch.

The Primal branch applies feature-mask attention along with feature aggregation and attention across body-joints. It utilizes the SE block (specified in chapter 5) for feature masking and performs feature aggregation with the GAT layer (specified in chapter 4). Finally, it applies a non-local layer to compute the dependencies of nodes that are not in the direct neighborhood.

The Dual branch is very similar to the Primal branch, but it is supposed to encode the interactions of elements in the feature dimension as opposed to the spatial dimension in the Primal branch. Essentially, it is a mirror of the Primal branch with certain modifications.

6.2. Primal branch

As mentioned above, the primal branch (shown in Figure 4) concentrates on feature aggregation in the spatial dimension. This branch is responsible for encoding the graph structure in the network with the affinity matrix. The three elements of the Primal branch are specified below:

6.2.1. Feature mask attention

Feature masking is done using the SE block. Given an input $X \epsilon \mathbb{R}^{K \times D}$ where K is the number of nodes in the graph and D is the number of the feature dimensions, the input to the primal branch $X_{primal} \epsilon \mathbb{R}^{K \times D}$ is $X_{primal} = X$. A mask is applied as follows:

$$X_{primal}{}' = X_{primal} \odot s$$

where $s \in \mathbb{R}^{D}$ is computed from Equations 8 and 9. X' now contains channel dependencies present in the input X.

6.2.2. Feature aggregation

This step performs the actual convolution operation where it uses the GAT layer to perform graph attention and convolution. This step is done by combining Equations 2 and 7. This step of feature aggregation computes the relations between neighboring nodes and also uses a flexible graph structure. Additionally, a batch normalization (Szegedy, 2015) layer and Relu (Hinton, 2010) non-linearity are applied here.

6.2.3. Non-local layer

In order to encode long-range dependencies, we need to consider features of nodes that are not in the direct vicinity. For this, we use the non-local layers specified in (He, 2018). The non-local operation is as follows:

$$\vec{x}_i^{(l+1)} = \vec{x}_i^{(l)} + \frac{W_x}{K} \sum_{j=1}^{K} f\left(\vec{x}_i^{(l)}, \vec{x}_j^{(l)}\right) . g\left(\vec{x}_j^{(l)}\right) \qquad 10$$

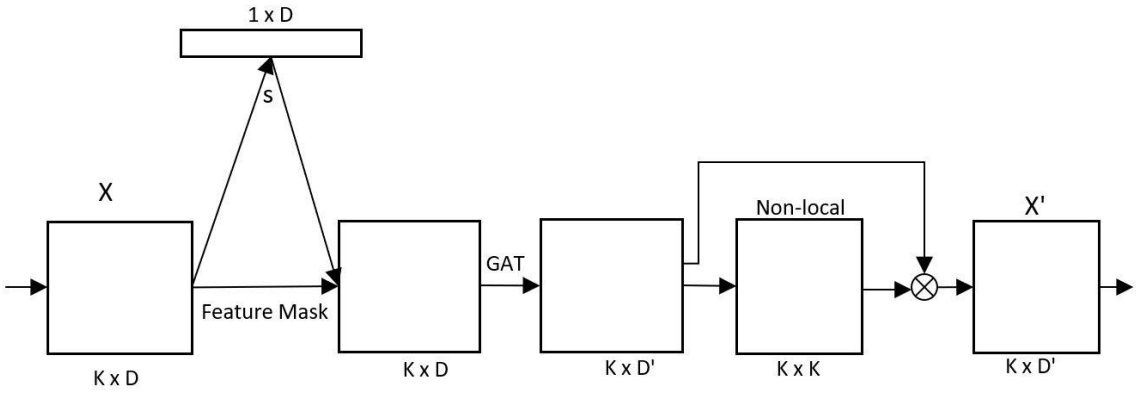

Figure 4. Structure of the proposed Primal Branch. The first part is the application of feature-mask with the SE (Sun, 2018) layers. This is followed by a GAT layer (Petar Velickovi ˇ c, 2018) which also transforms the dimensionality of the input. Finally, a non-local layer is applied and the output is obtained.

6.3. Dual Branch

The dual branch (shown in Figure 5) is responsible for computing the relations between the features in the channel dimension. It applies node-mask attention along with a linear transformation, and a non-local layer. The Dual branch is a sort-of replica of the Primal branch, with the key difference being that the input to the Dual branch is the transpose of the input to the Primal branch. Therefore, given an input graph $X \epsilon \mathbb{R}^{K \times D}$, the input to the Dual branch $X_{dual} \epsilon \mathbb{R}^{D \times K}$ is $X_{dual} = X^T$. The elements of the Dual branch are detailed below.

6.3.1. Node-mask attention

This is used to model the interactions between the nodes in the Dual branch. It is similar to the feature-mask attention of the Primal branch, but since the input is transposed, masking is done along the node dimension. The node-mask is as follows:

$$X_{dual}{}' = X_{dual} \odot s$$

Here $s \in \mathbb{R}^K$ is computed similarly from Equations 8 and 9.

6.3.2. Non-local layer

Again, this part is the same as the non-local layer in the Primal branch, but since the input to the Dual branch is the transpose of X, the non-local layer will calculate the long-range dependencies in the channel domain.

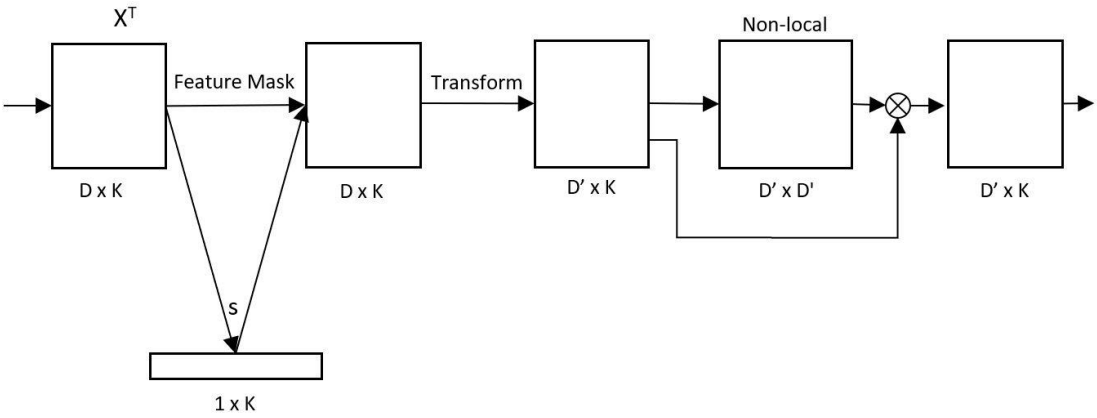

Figure 5 Proposed Dual branch. Operates similarly to the Primal branch, input to the branch is transposed.

6.4. Primal-Dual Block

Combining the Primal and the Dual branches, we have a Primal-Dual block. The two branches operate in parallel and after computation, the results of both the branches are fused by summation. The structure of the Primal-Dual block is shown in Figure 6.

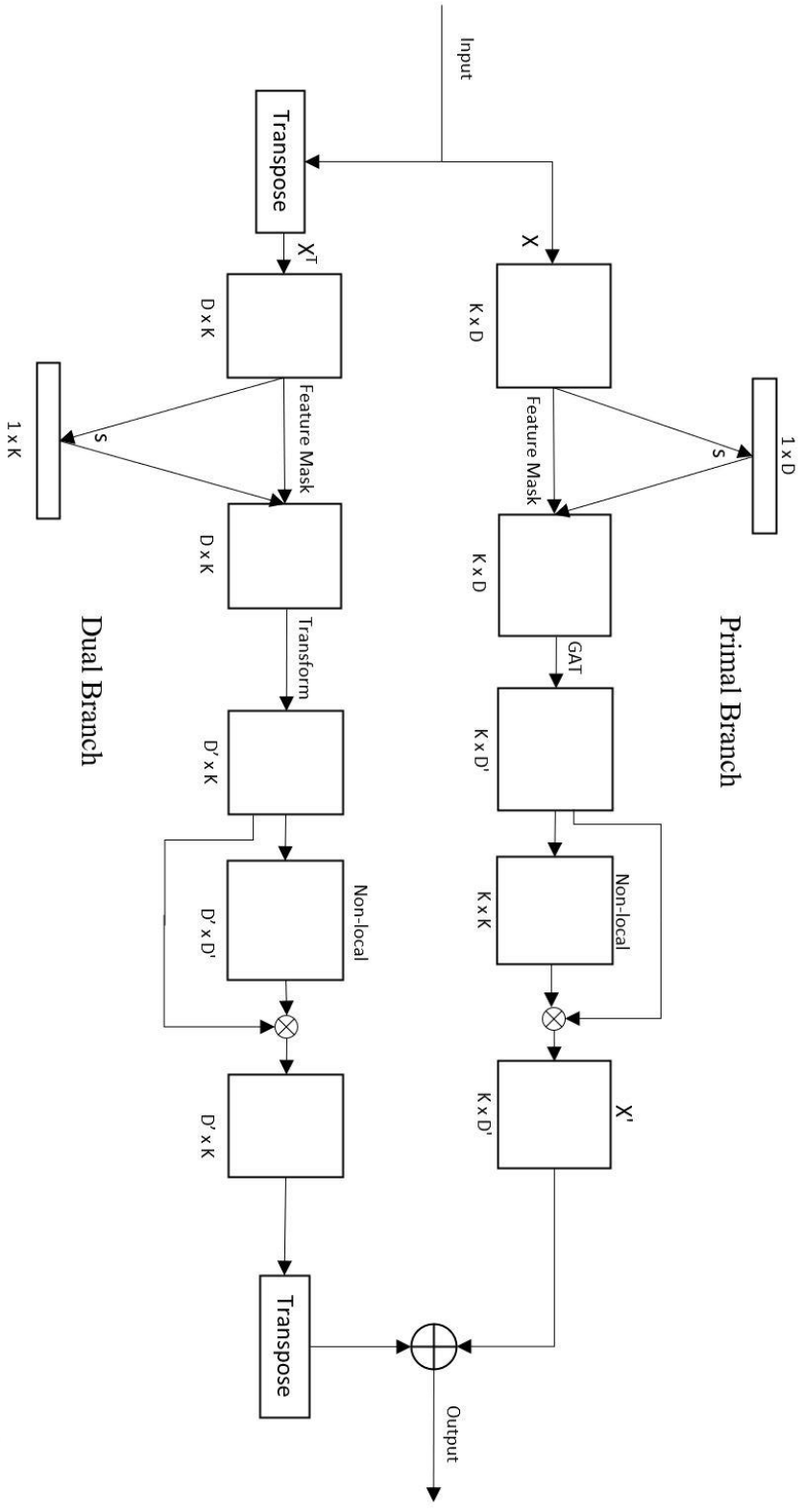

Figure 6 Proposed Primal-Dual block

7. NETWORK ARCHITECTURE

The Primal-Dual block specified in the previous section makes the backbone of our network architecture. Like (Julieta Martinez, 2017), we employ the use of residual blocks (Kaiming He, 2016) each of which consists of two Primal-Dual blocks. In the input of the network, we first have a Primal-Dual block which transforms the input graph into a higher dimensional feature space. Then we have four residual blocks. Finally, towards the output, we have a lone Primal block to generate the results in the output space. The architecture of the network is depicted in Figure 7.

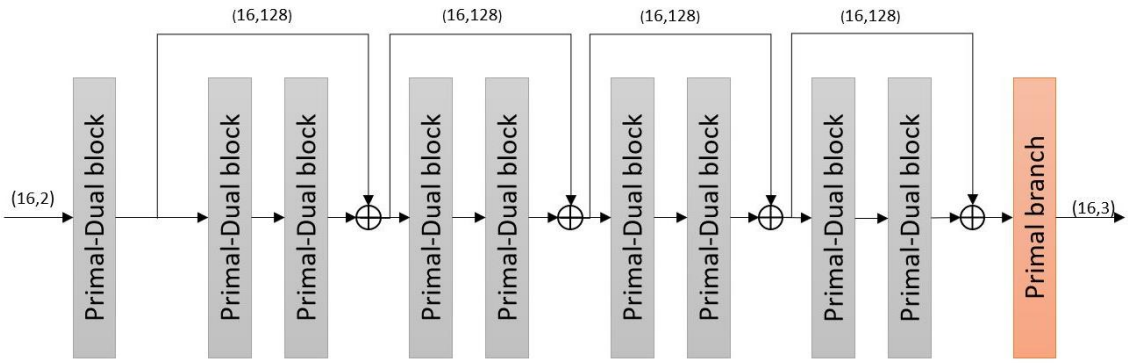

Figure 7 Model architecture

8. 3D HUMAN POSE ESTIMATION

In this chapter, we describe the usage of the proposed Primal-Dual graph attention network for the task of 3D human pose estimation

8.1. Problem formulation

The human skeleton can be intuitively represented as a graph. The nodes of the graph represent the joints of the body while the edges of the graph represent the connected joints or the bones. Such a representation has previously been employed in (Cao, 2017). The skeleton is made up of 16 joints with the pelvis as the root joint.

Many works in the past years have shown that highly accurate 3D pose can be estimated from 2D pose only (Julieta Martinez, 2017). Hence, we employ the Primal-Dual graph attention network to lift the pose from 2D to 3D in the camera coordinate system.

We have our input as a set of 2D joints $P \in \mathbb{R}^{K\times 2}$ and desired output as the ground-truth joint positions in 3D $J \in \mathbb{R}^{K\times 3}$. "The system aims to learn a regression function F^* which minimizes the following error over a dataset containing N human poses" (Zhao, 2019).

$$F^* = \underset{F}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(F(P_i), J_i) \qquad 11$$

The input 2D joint locations can either be ground-truth 2D locations or the estimated 2D locations from a 2D joint detector.

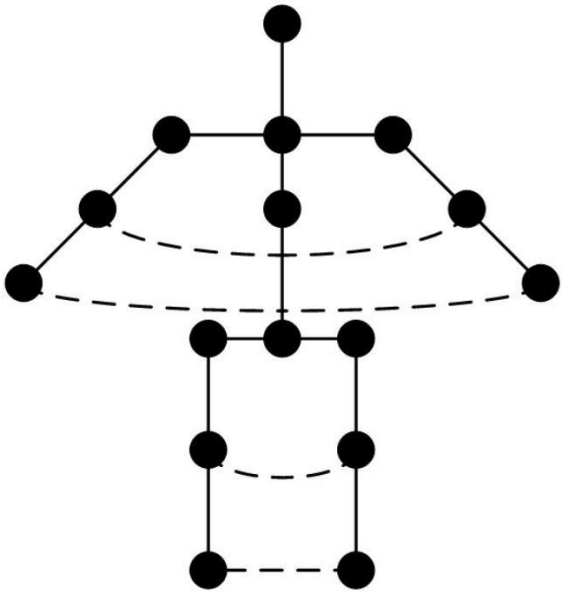

Figure 8 Human pose as a graph with nodes representing body joints and edges representing bones

8.2. Dataset

The Dataset used for the purpose of 3D human pose estimation is Human3.6M (Catalin Ionescu D. P., 2014). “The dataset consists of 3.6 million different human poses collected with 4 digital cameras. Data is organized into 15 training scenarios including walking, sitting and lying down, various types of waiting poses and so on” (Catalin Ionescu D. P., 2014). Similar to (Xingyi Zhou, 2017), the input poses are converted from 50fps to 10fps. Samples from the dataset are shown in Figure 9.

8.3. Loss Function

Following configuration 1 of (Zhao, 2019), the mean squared error loss function is employed. The error is calculated between the predicted and the ground truth 3D joint locations.

$$\mathcal{L}(J) = \sum_{i=1}^{K} \left|\left|\tilde{J}_i - J_i\right|\right|^2 \qquad 12$$

where J_i and $\tilde{J}_i$ are the ground truth and predicted 3D joint locations for joint i respectively.

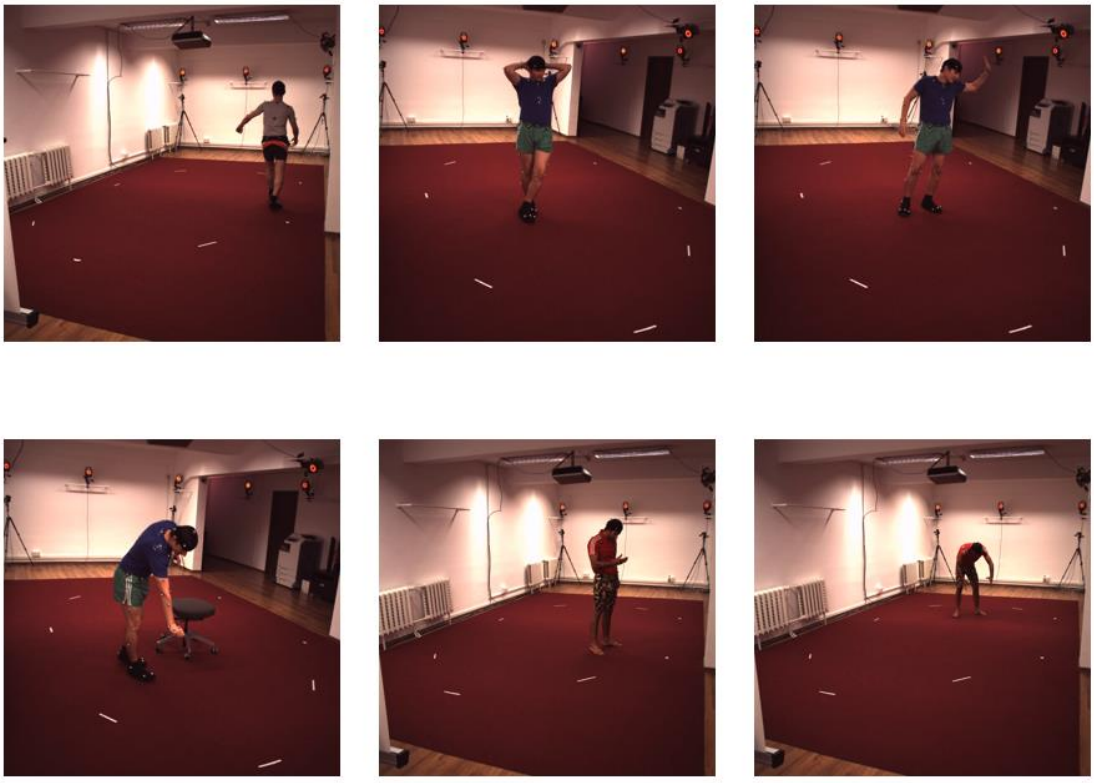

Figure 9 Samples from the Human3.6M dataset

8.4. Evaluation metrics

For Human3.6M (Catalin Ionescu D. P., 2014), two protocols are used for evaluation. Protocol #1 uses

> "all 4 camera views in subjects S1, S5, S6, S7 and S8 for training and the same 4 camera views in subjects S9 and S11 for testing. Errors are calculated after the ground truth and predictions are aligned with the root joint.
>
> [Protocol 2] makes use of six subjects S1, S5, S6, S7, S8 and S9 for training, and evaluation is performed on every 64th frame of S11. It also utilizes a rigid

transformation to further align the predictions with the ground truth." (Zhao, 2019)

For Protocol #1, we report the performance of our model using mean per-joint position error (MPJPE) as the metric. For protocol #2, the metric used is mean per-joint position error after Procrustes alignment (P-MPJPE). Both of these evaluation metrics measure the Euclidean distance in millimeters between the ground truth and the predicted 3D pose.

$$MPJPE = \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{K}\left|\left|(J_i^t - J_{root}^t) - (\tilde{J}_i^t - \tilde{J}_{root}^t)\right|\right|_2 \quad 13$$

9. IMPLEMENTATION DETAILS

Here we describe the specific implementation of our network for 3d human pose estimation. For the inference of positions of joints in 3D space, we use the camera coordinate system. As has been used in earlier work (Dario PavlloF, 2019), we use cascaded pyramid network (CPN) (Yilun Chen, 2018) as our 2D pose detector which takes images and estimates a pose in 2D. However, in our ablation study, we only use ground truth 2D poses to pass to our network.

9.1. Training

We train our network using Adam optimization (Ba, 2014). Via experimentation, we set the hyperparameters as follows: For the ground truth 2D pose, we use an initial learning rate of 0.001, with the learning rate decaying by a factor of 0.96 for every 4 epochs, and a batch size of 64. For the CPN 2D detections, we use an initial learning rate of 0.005, with the learning rate decaying by a factor of 0.65 for every 4 epochs, and a batch size of 128.

Training is done on an Nvidia RTX 2080Ti GPU with PyTorch. Weights are initialized by (Bengio, 2010).

10. RESULTS

10.1. Ablation Study

We conduct 3 ablation studies to demonstrate the effectiveness of our proposed model. For the purpose of the ablation study, we use the 2D ground-truth poses as input to the network. The 3 studies demonstrate the effectiveness of the individual modules in each of the following: Primal branch, Dual branch, Primal-Dual block.

10.1.1. Variants of Primal branch

In this, we consider network configuration with only the Primal branch. We remove individual elements from the Primal branch to analyze their impact on the performance. We demonstrate that the removal of each element from the Primal branch results in a drop in performance. Results are shown in TABLE I.

TABLE I
COMPARISON OF DIFFERENT ELEMENTS OF THE PRIMAL BRANCH

Method	#Parameters	MPJPE	P-MPJPE
Baseline SemGCN	0.27M	42.88 mm	33.89 mm
Primal (only GAT)	0.27M	42.36 mm	34.66 mm
Primal (only feature-mask)	0.29M	41.47 mm	32.56 mm
Primal (only non-local)	0.57M	40.84 mm	31.08 mm
Primal w/o feature-mask	0.57M	40.55 mm	31.38 mm
Primal w/o GAT	0.59M	39.39 mm	31.47 mm
Primal w/o non-local	0.29M	38.87 mm	31.43 mm
Primal branch	0.59M	38.37 mm	29.89 mm

10.1.2. Variants of Dual branch

In this, we consider network configuration with only the Dual branch. Individual elements are removed to analyze their impact on the performance. We show that each element on the Dual branch contributes to the improved performance. Results are shown in TABLE II.

TABLE II
COMPARISON OF DIFFERENT ELEMENTS OF THE DUAL BRANCH

Method	#Params	MPJPE	P-MPJPE
Dual w/o non-local	0.14M	58.94 mm	48.89 mm
Dual w/o node-mask	0.14M	44.02 mm	35.20 mm
Dual branch	0.14M	42.26 mm	34.90 mm

10.1.3. Primal-Dual block study

Finally, we study the effects of the combination of the Primal and Dual branches for the proposed Primal-Dual block. TABLE III shows that the Primal and Dual branches are complementary to each other and help in improving performance.

TABLE III
COMPARISON OF INDIVIDUAL BRANCHES OF PROPOSED NETWORK

Method	#Params	MPJPE	P-MPJPE
Primal branch only	2.33M	38.62 mm	30.12 mm
Dual branch only	0.54M	44.02 mm	35.20 mm
Primal-Dual graph attention network	2.87M	36.69 mm	29.37 mm

10.2. Comparison with baseline

For this, we change the number of channels in the baseline SemGCN to match the number of parameters in our proposed network. This proves that the performance boost in the proposed model is not simply due to an increased number of parameters. Results are shown in TABLE IV.

TABLE IV
COMPARISON OF PROPOSED NETWORK WITH BASELINE

Method	Channels	#Params	MPJPE	P-MPJPE
Baseline SemGCN	422	2.87M	40.37 mm	31.91 mm
Primal-Dual graph attention network	256	2.87M	36.69 mm	29.37 mm

10.3. Comparison with the state of the art

The quantitative results of the evaluation of our model and its comparison with some state-of-the-art approaches on the Human3.6M dataset (Catalin Ionescu D. P., 2014) are shown in TABLES V and VI. Some of these approaches use additional techniques which are complementary to our network. For example, (Xiao Sun, 2017), (Wei Yang, 2018) use complex loss functions.

10.4. Qualitative results

In Figure 10, we show examples of the resulting 3D pose estimated from image inputs. We see that our network is able to accurately predict the pose of people doing different activities.

TABLE V
QUANTITATIVE COMPARISONS UNDER PROTOCOL #1. ERRORS ARE IN MILLIMETER

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD.	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Mehta et al.	52.6	64.1	55.2	62.2	71.6	79.5	52.8	68.6	91.8	118.4	65.7	63.5	49.4	76.4	53.5	68.6
Martinez et al.	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.664.	62.3	59.1	65.1	49.5	52.4	62.9
Sun et al.	52.8	54.8	54.257.	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Yang et al.	51.5	58.9	50.4	57.1	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Pavlakos et al.	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Zhao et al.	47.3	60.7	51.4	60.5	61.1	49.9	47.3	69.1	86.2	55.0	67.8	61.0	42.1	60.6	45.3	57.6
Ours	45.7	52.4	48.4	51.4	55.9	67.6	48.4	48.5	62.1	73.5	52.5	50.6	56.2	40.1	42.58	53.07

TABLE VI
QUANTITATIVE COMPARISONS UNDER PROTOCOL #2. ERRORS ARE IN MILLIMETER

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch	Sit	SitD.	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Zhou et al.	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1	53.7	65.5	51.6	50.4	54.8	55.9	55.3
Martinez et al.	39.5	43.2	46.4	47.0	51.0	56.0	4.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Sun et al.	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
Fang et al.	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Pavlakos et al.	47.5	50.5	48.3	49.3	50.7	55.2	46.1	48.0	61.1	78.1	51.1	48.3	52.9	41.5	46.4	51.9
Hossain & Little	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Ours	36.2	40.9	39.1	41.4	42.4	52.0	36.9	37.8	49.9	58.5	42.0	38.8	44.5	31.8	36.1	41.9

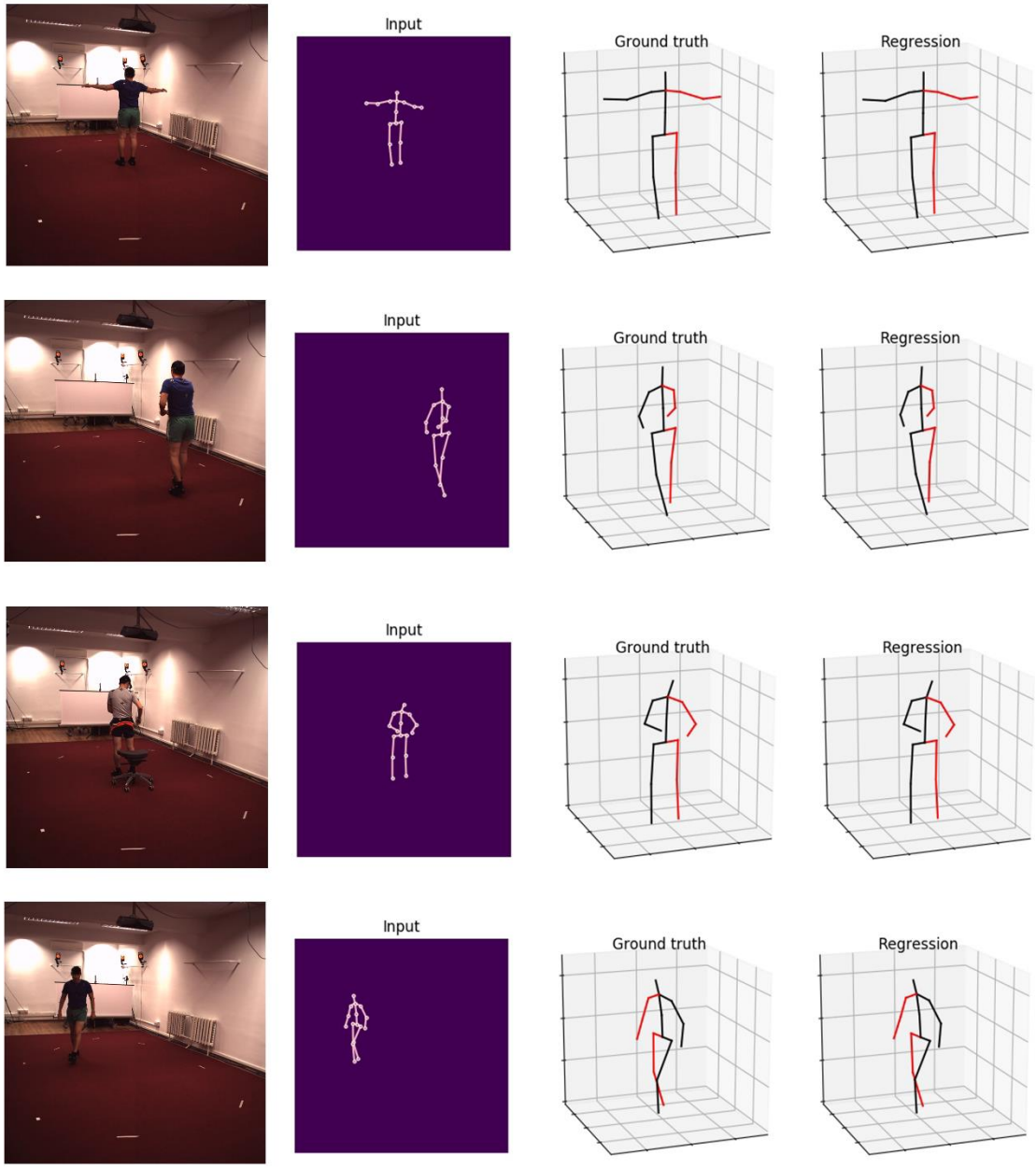

Figure 10 Qualitative results on Human3.6M dataset

11. CONCLUSION

In this thesis, we have studied the application of graph convolution networks for the task of 3D human pose estimation. We see how a graph neural network is able to effectively model the skeletal structure of the human body. We extensively study the effect of various attention mechanisms on the performance of pose estimation. Acknowledging the shortcomings of vanilla GCNs, we propose a novel model for 3D human pose estimation, the Primal-Dual Graph Attention Networks. This network incorporates a flexible graph structure along-with using the interdependencies in both the spatial and channel domain and also long-range dependencies. A comprehensive ablation study of the proposed modules of the network shows their individual effectiveness as well as their power when used together in the proposed network.

For future work, the network could be modified to make use of temporal information from videos and sequences. Additionally, the application of the Primal-Dual graph attention network in domains other than human pose estimation is left to be attempted.

12. CITED LITERATURE

Anguelov, D. a. (2005). SCAPE: Shape Completion and Animation of People. *Association for Computing Machinery.*

Ankur Gupta, J. M. (2014). 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. *CVPR*.

Ba, D. P. (2014). Diederik P. Kingma and Jimmy Ba. *ICLR.*

Bengio, X. G. (2010). Understanding the difficulty of training deep feedforward neural networks. *AISTATS, volume 9*, pp. 249–256.

Cao, Z. S. (2017). Realtime multi-person 2d pose estimation. *IEEE Conference on Computer Vision and Pattern*, (pp. 7291-7299).

Catalin Ionescu, D. P. (2014, July). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Catalin Ionescu, F. L. (2011). Latent structured models for human pose estimation. *ICCV*.

Catalin Ionescu, F. L. (2011). Latent Structured Models for Human Pose Estimation. *ICCV.*

Chen, H.-J. L. (1985). Determination of 3D HumanBody Postures From a Single View. *Computer Vision Graphics and Image Processing*, pp. 148-168.

Dario PavlloF, C. F. (2019). 3d human. *Conference on Computer Vision and Pattern Recognition (CVPR).*

Felzenszwalb, P. H. (2005). Pictorial Structures for Object Recognition. *International Journal of Computer Vision*, 55–79.

Franco Scarselli, M. G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*.

Gregory Rogez, J. R. (2008). Randomized trees for human pose detection. *CVPR*.

Gupta, X. W. (2018). Videos as space-time region graphs. *ECCV*.

Hang Xu, C. J. (2019). Spatial-aware graph relation network for large-scale object detection. *Proceedings of the IEEE Conference*.

He, X. W. (2018). Non-local Neural Networks. *CVPR*.

Hinton, V. N. (2010). Rectified linear units improve restricted boltzmann machines. *ICML*, (pp. 807-814).

Jiang, H. (2010). 3D Human Pose Reconstruction Using Millions of Exemplars. *ICPR*.

Ju, S. B. (1996). Cardboard people: A parameterized model of articulated image motion. *IEEE Conference on Automatic Face and Gesture Recognition.*

Julieta Martinez, R. H. (2017). A simple yet effective baseline for 3d human pose. *ICCV*.

Junyu Gao, T. Z. (2019). Graph convolutional tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kaiming He, X. Z. (2016). Deep residual learning for image recognition. *CVPR*.

Kinect. (2021). *URL https://developer.microsoft.com/en-us/windows/kinect (accessed on 2021)*.

Marco Gori, G. M. (2005). A new model for learning in graph domains. *IJCNN*, 729–734.

Matthew Loper, N. M.-M. (2015). SMPL: a skinned multi-person linear model. *Association for Computing Machinery*.

Nanyang Wang, Y. Z. (2018). Pixel2mesh: Generating 3d mesh. *ECCV.*

Petar Velickovi ˇ c, G. C. (2018). Graph Attentions Networks. *ICLR.*

Rui Zhao, K. W. (2019). Bayesian graph convolution lstm for skeleton based action recognition. *Proceedings of the IEEE International Conference on Computer Vision*.

Sami Abu-El-Haija, B. P. (2019). Mixhop: Higherorder graph convolution architectures via sparsified neighborhood mixing. *arXiv preprint arXiv:1905.00067*.

Sidenbladh, H. D. (2000). A framework for modeling the appearance of 3d articulated figures. *IEEE Conference on Automatic Face and Gesture Recognition.*

Sijie Yan, Y. X. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. *AAAI*.

Sun, J. H. (2018). Squeeze-and-Excitation Networks. *IEEE Conference on Computer Vision and Pattern Recognition*.

Szegedy, S. I. (2015). Batch normalization:. *ICML*.

T.F. Cootes, C. T. (1995). Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*.

Triggs, A. A. (2006). Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Wei Yang, W. O. (2018). 3d human pose estimation in the wild by adversarial learning. *Proceedings of*, 5255–5264.

Welling, T. N. (2017). Semi-supervised classification with graph convolutional networks. *ICLR*.

Xiao Sun, J. S. (2017). *Proceedings of the IEEE International Conference on Computer Vision*, 2602–2611.

Xingyi Zhou, Q. H. (2017). Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach. *ICCV.*

Yilun Chen, Z. W. (2018). Cascaded pyramid network for multi-person pose estimation. *Proceedings of the IEEE conference on computer vision and pattern recognition.*

Zhao, L. a. (2019). Semantic Graph Convolutional Networks for 3D Human Pose Regression. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 3425--3435).

13. VITA

Educational Background:

1. Bachelors of Technology (Computer Science and Engineering), APJ Abdul Kalam Technical University, Lucknow, UP, India.
2. MS in CS, University of Illinois at Chicago, Chicago, IL.