SGAT: Semantic Graph Attention for 3D human pose estimation

Luiz Schirmer*, Djalma Lucio*, Leandro Cruz‡, Alberto Raposo*, Luiz Velho† and Hélio Lopes*
*PUC-Rio - Pontifícia Universidade Católica do Rio de Janeiro, Brazil
†IMPA - Instituto Nacional de Matemática Pura e Aplicada, Brazil
‡Institute of Systems and Robotics - University of Coimbra, Portugal

***Abstract*—We propose a novel gating mechanism applied to Semantic Graph Convolutions for 3D applications, named Semantic Graph Attention. Semantic Graph Convolutions learn to capture semantic information such as local and global node relationships, not explicitly represented in graphs. We improve their performance by proposing an attention block to explore channel-wise inter-dependencies. The proposed method performs the unprojection of the 2D points (image) onto their 3D version. We use it to estimate 3d human pose from 2D images. Both 2D and 3D human poses can be represented as structured graphs, exploring their particularities in this context. The attention layer improves the accuracy of skeleton estimation using 58% fewer parameters than state-of-the-art.**

I. Introduction

Convolutional Neural Networks achieve the state of the art results in regular-structured problems. However, many data structures such as 3D Meshes and human skeletons can only be represented by irregular structures (graphs) where *CNNs* have limited applications. To tackle the limitations of common *CNNs*, *Graph Convolutional Networks (GCN)* [1]–[3] have been recently proposed. However, it still has some issues when inferring the 3D position of points from their projected 2D version—the conventional *GCNs* process nodes with arbitrary topologies. The learned kernel is shared for all edges. It only considers the first-order neighbors of each node (local operation). Hence, the global structure of the graph is not completely exploited [3].

Semantic Graph Convolutions (SGC) network [3] learns the semantic information encoded in a given graph. It learns channel-wise weights for the edges and combines them with kernel matrices, improving the power of graph convolutions. It tackles the limitation of the original GCN, allowing the unprojection from a 2D skeleton to a 3D one. However, it still has some limitations; for example, it does not consider features and channel-wise inter-dependencies.

In this paper, we propose a novel gating mechanism applied to *SGCs* named *Semantic Graph Attention (SGAT)*. We enhance the analysis of global correlation, which is crucial for understanding human actions [4]. Our new layer can learn both channel-wise weights for edges, combine them with kernel matrices, and features inter-dependencies over channels. This work also proposes a novel neural network architecture used for 3D pose estimation from 2D joints. The main feature of this approach is our attention layer combined with Semantic Graph Convolutions.

Given a 2D human pose as input, we predict the locations of its corresponding 3D joints in a 3D coordinate space. The proposed method achieves state-of-the-art performance for predicting a 3D human pose from their 2D skeleton (12% better than the previous works). Furthermore, it has 58% fewer parameters than the original SGC model. We evaluated our approach in the following three datasets: Human 3.6M [5], COCO [6] and MPI-INF-3DHP [7].

This approach can be used as a part of a 3D real-time pose estimation tool, useful for human animation. Since this method uses as input only 2D keypoints, captured from an RGB video using ordinary cameras, and it also demands fewer parameters than other known methods, it can be implemented in ordinary devices. In summary, our main contributions are:

- A novel attention layer for semantic graph convolutions based on a simple but effective gating mechanism;
- A novel lightweight architecture for 3D human pose estimation based SGCs with performance enhanced by our attention layer.

II. Related Work

A. Graph Neural Networks

Graph Neural Networks (*GNN*) is a framework to understand and explore graph structure relationships [8]–[10]. In *GNNs*, the node representation vector is computed by aggregating and transforming the data representation of its neighbors. As an evolution, *Graph Convolutional Networks* (*GCNs*) [1], [2] where introduced to deal with spectral and spatial problems. But, they also have limitations, such as the kernel of an operation is shared by all nodes. Zhao et al. [3] propose the so-called *Semantic Graph Convolutional Networks* (*SemGCN*) that captures the global and semantic information of nodes relationships. On the other hand, it aims to approximate convolutions by learning a channel-wise weighting vector, and each spatial kernel has a shared transformation matrix. Moreover, it does not consider inter-dependencies between channels.

Veličković et al. [11] present the *Graph Attention Networks* which operate on graph-structured data. They use self-attentional methods by stacking layers, implicitly enabling different weights to different nodes in a neighborhood. In this work, we additionally aim to learn independent weights for the edges.

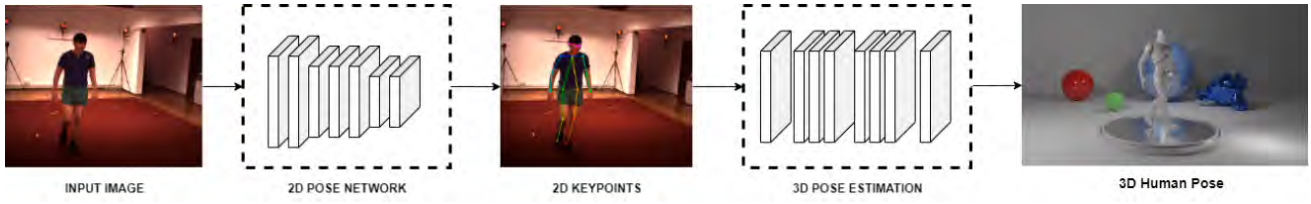

Fig. 1: Our proof of concept model for 3D human pose estimation and computer animation. Here we capture 2D keypoints e interactively regress them to a 3D domain. After we generate 3D motion files and 3D animations in an Open Source 3D creation suite. Also, the quality of the 3D pose output highly depends on the 2D inputs.

Considering traditional convolutional neural networks, several papers aim to model global context for feature extraction. An example is *Squeeze-and-Excitation* (*SE*) networks [12] that analyzes channel-wise feature responses by explicitly modeling inter-dependencies between channels. This approach can be remodeled and adapted to other neural network techniques. Another example is Global Context Networks [13] that adapt the *SE* block, in convolutional neural networks, for global-context modeling.

Our formulation surpasses previous works by reducing the reprojection error, drastically reducing the computational complexity, and the quantity of parameters (Section 5).

Fig. 2: Our model for the neural network to estimate the 3D keypoints. Note that we have here 2 internal blocks that uses semantic graph structures followed by an attention block. Also, at the end of each internal block we also have a residual operation.

B. Pose Estimation and Motion Capture

Wei et al. [14] introduced the *Convolutional Pose Machines* (*CPMs*) that combines the advantages of convolutional neural networks with pose machines. *CPMs* consist of a sequence of convolutional layers that repeatedly produce 2D confidence maps for the location of each body part. We use a *CPM* in our experiments to generate a 2D skeleton, the input for our 3D model.

Following the success of 2D pose estimation models, several papers propose an end-to-end model to predict the 3D human pose given in-the-wild images. One with the best accuracy results was obtained by Wang et al. [15]. They present a *3D Human Pose Machine* with self-supervised learning, where they developed a multi-stage system composed of three neural network models involving two dual learning tasks. They generate transformations for 2D-to-3D pose and 3D-to-2D pose projection. The 2D-to-3D pose model regresses intermediate 3D poses by transforming the pose representation from the 2D domain to the 3D, receiving the features extracted by a 2D pose machine as input. In contrast, the 3D-to-2D pose projection contributes to refine the intermediate 3D poses. However, their approach looks pretty costly, considering computational resources, even in a high-end GPU. Our approach only focuses on efficiently predict 3D poses, not using sophisticated frameworks as in this approach.

Mehta et al. [16] present the first real-time method to capture the 3D pose in a stable, temporally consistent manner using a single RGB camera, named *VNect*. Their formulation uses real-time regression for 2D to 3D projection and creates a kinematic skeleton fitting method for coherent kinematic analysis. *XNect* [17] is an evolution of this work that also predicts the 3D pose of Humans and even infer the bones rotations. They present a real-time approach for multi-person 3D motion capture at over 30 fps. They both present complex frameworks to generate 3D poses and motion capture information. Our model focuses on a lightweight approach to generate 3D poses from 2D data, which is similar to stage 2 from *XNect*, but surpasses its results considering the error metric of this stage.

In a different approach, Martinez et al. [18] propose a simple feed-forward neural network that receives the 2D joint locations and predicts 3D positions. They "lift" the ground-truth 2D joint locations to 3D space. However, this also has limitations, such as it does not maintain the bone proportion for all bodies. Zhao et al. [3] has expanded this work by using *Semantic Graph Convolutions*. Their architecture can also be extended to use attention modules. Our approach minimizes the reprojection error and network complexity using the attention layer, a way to model inter-dependencies in Semantic Graph Convolutions. Furthermore, we use bone vector constraints and joints measures in the loss function to reduce the reprojection error.

Kocabas et al. [19] propose the Video Inference for Body Pose and Shape Estimation (VIBE), which predicts the parameters of *SMPL* body model [20] for each frame of a video. It is a sophisticated adversarial learning framework to discriminate between real human motions and the results produced by temporal pose and shape regression networks.

In contrast with these previous works, we present a lightweight framework for computer animation using 3D human pose estimation. For this purpose, our model does

not need any specialized hardware or even high-end GPU configurations. Furthermore, our model only predicts the 3D keypoints location, leaving the rotation information for future work.

III. Graph Convolution

In this section, we present the main aspects of our new attention model for Graph Convolutional Neural Networks called *Semantic Graph Attention*. The primary motivation is to create a new network model that learns both channel-wise weights for edges and channel inter-dependencies. The edges are combined with kernel matrices allowing an understanding of the global channel inter-dependencies without non-local layers.

A. Graph Convolutional Networks

Following principles of regular *Convolution Neural Networks*, a *Graph Convolution Network* can be considered a way to deal with arbitrary graph structures [1], [2], [21]. This is highly related to our approach to analyze human pose as a structured graph.

Convolutional Graph Networks (GCNs) share the filter parameters in the graph. The *GCNs* training stage consists in learning structures capable of processing graph information from the node matrix $X \in R^{N \times D}$ (N nodes containing D features) and the adjacency matrix $A \in R^{\|N\| \times \|N\|}$ [1], [2].

Each layer is a non-linear function as follows:

$$\boldsymbol{H}^{(l+1)} = f(\boldsymbol{H}^l, \boldsymbol{A}), \tag{1}$$

where H is the output of each layer and $H^0 = X$. Rewriting this equation, we have:

$$f(\boldsymbol{H}^l, \boldsymbol{A}) = \sigma(\boldsymbol{A}\boldsymbol{H}^l\boldsymbol{W}^l), \tag{2}$$

where σ represents the *ReLU* activation function and W the weight matrix of the network layer. There are some limitations to this approach because the multiplication by the matrix A would only consider features from the neighborhood, but not from the node itself. This problem is addressed by adding the identity matrix to A ($A' = A + I$).

Furthermore, A' should be an unitary matrix to do not scale the vector of features. We reach it by normalizing A' rows using the *Normalized Laplacian Matrix* $D^{-1/2}AD^{-1/2}$, where D^{-1} is the inverse of the diagonal matrix with the degree of the graph nodes. We add these concepts to equation 2 as follows:

$$f(\boldsymbol{H}^l, \boldsymbol{A}) = \sigma(\boldsymbol{D}^{'-1/2}\boldsymbol{A}^{'}\boldsymbol{D}^{'-1/2}\boldsymbol{H}^l\boldsymbol{W}^l). \tag{3}$$

There are two clear disadvantages to make the graph convolution considering a regression to work on nodes with arbitrary topologies. The first one is the kernel matrix W is shared by all the edges. As a result, the relationships of neighboring nodes, i.e. internal structure, are not well explored. This is also a limiting factor because the receptive field is fixed with ones [3], the second disadvantage.

A *CNN* with a convolution kernel of size $k \times k$ learns k^2 different transformation matrices. The transformation matrices decode features within the kernel spatial dimension. This formulation can be approximated by learning a vector of weights $\vec{a}_i$ for each position of a pixel in an image or a graph node, and then combining them with a shared transformation matrix W [3].

We can transform an image to a graph by considering the pixels as nodes, and two neighbor pixels being connected by an edge (8-connect neighborhood). So, a kernel size k affects all pixels distant less than $d = \frac{k-1}{2}$. We can extend this approach for *GCNs* by considering that a convolution in a graph using a kernel of size d affects all nodes in a neighborhood of size d [3].

GCNs cannot handle directly with regression problems due the issue that convolution filter shares the same weight matrix for all edges. Furthermore, the filters just operate in a one step neighborhood. As a solution, Zhao et al. [3] propose to add the weight matrix M to the graph convolution process described by:

$$f(\boldsymbol{H}^l, \boldsymbol{A}) = \sigma(\phi(\boldsymbol{A}' \odot \boldsymbol{M})\boldsymbol{H}^l\boldsymbol{W}^l), \tag{4}$$

where the matrix M is a parameter to be learned on the network and ϕ is a *softmax* function that normalizes the entries of each node, $\odot$ is an element-wise multiplication (*Hadamard*) that returns m_{ij} if $a_{ij} = 1$ or negative values with large exponents after the *softmax*. In this approach, A works like a mask that forces this to the i node in the graph and σ is a *ReLU* activation. Also as proposed by Zhao et al. [3], this formulation can be extended to consider multiple channels as in traditional convolutions. In our experiments, we use *PreLU* instead of *ReLU* activation because it shows a performance improvement.

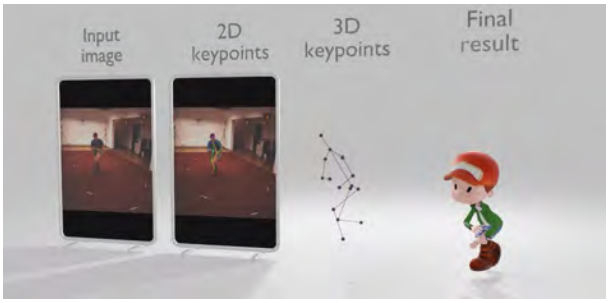

Fig. 3: An example of our 3D human pose aproach to generate animations with a single RGB camera. The 2D data is captured with a 2D pose network and after processed by the 3D neural network.At a final stage, we use the captured data in a 3D animation suite.

B. Attention block for SGC

This section presents our first contribution: we propose a method to enhance the runtime and reduce the complexity of Semantic Graph Convolution Networks called Semantic Graph Attention. The intuition here is to allow the neural network to perform feature recalibration, i.e., emphasize more relevant

features and suppress less meaningful information. In other words, to give weights to the features over channels of SCGs, in a similar way to the SE-NET [12]. This also can be related to *Global Context Networks* [13] when applying attention to graph networks. We aim to solve precision and computational complexity issues, considering both space storage and time complexity, from previous related works.

Considering the computation of features for each node, the idea of adding weights via element-wise multiplication is natural. We intend to identify inter-dependencies between node features. For this purpose, we propose the following gating mechanism for each channel after a regular *SGC*:

$$g = \boldsymbol{A}' \odot \phi(\boldsymbol{M}_1)\boldsymbol{W}_1^l\boldsymbol{H}^l, \tag{5}$$

where g is composed by a *softmax* ϕ function over the entries of Matrix M_1. This hidden layer performs a dimensionality reduction that reduces drastically the input space by a factor r, where the kernel size of $W_1^l \in R^{\frac{C}{r}\times C}$. The intuition behind this layer is similar to Principal Components Analysis. We perform a dimensionality reduction to a space that better represents the data given a new basis. Consider that our Graph Neural Network has input data contained in a 2D space. The neural network project the data into a frequency space in the first layer, as in a Fourier Transform. The first block of our attention module evaluates features in frequency space. It forces this neural network stage to consider the most relevant ones as in an orthogonal transformation. In a gating mechanism, the next block will use the data representation in this new space, expanding the data to the original size and given weights to the features as follows:

$$s(g) = \alpha(\boldsymbol{A}' \odot \phi(\boldsymbol{M}_2)\sigma_1(g)\boldsymbol{W}_2^l), \tag{6}$$

where kernels $W_2^l \in R^{C\times\frac{C}{r}}$, σ represent a *PRelu* function, α represents a *sigmoid* activation, C represents the number of features and r is defined empirically. In our experiments, we use a value $r = 16$, similarly to Hu et. al. [12]. With the output of function $s(g)$ for each channel, we perform an element-wise multiplication operation to give weights to the input data:

$$\boldsymbol{H}^{l+1} = \boldsymbol{H}^l \circ s(g). \tag{7}$$

At the final process, the channels are also concatenated. Such a gating operation allows us to consider more relevant features after each convolution operation, and thus refine our regression process for the following pose estimation case.

As we will see in our experiments, this formulation enhanced our neural network's overall performance and drastically reduced its complexity.

IV. 3D Pose Estimation framework

As a second contribution, we propose a 3D human pose estimation framework. The method presented in the previous section takes as input a 2D human skeleton. We can use any method to calculate these 2D joints from a single RGB image. It is noteworthy that we do not consider temporal coherence and rotation issue.

Each module is independent in our architecture in terms of video processing, inference of captured skeletons, information transmission, and 3D animation. Since all models are decoupled, we can use different 2D pose networks to extract the 2D keypoints. The user can choose the best model for the respective application. Figure 1 illustrates the elements of our framework.

The framework starts by calculating the 2D keypoints for each person in a given image using a 2D neural network. Our 3D model only needs the 2D keypoints as input. It makes our architecture flexible and not dependent on a specific 2D pose model to generate keypoints.

Afterward, our neural network exploits the power of the semantic graph convolution with a gating mechanism. Our model has an input layer with an *SGC* followed by batch normalization and a *PRelu* activation. The building blocks of our network's internal layers are composed of two *SGC* layers, also followed by batch normalization and *PRelu* activation. The output of the second *SGC* layer is used as the input for our gating mechanism. This is repeated twice, and the blocks also use residual connections. We consider 128 channels for 16 graph nodes in the internal layers, where each node represents a human keypoint. The output layer comprises an *SGC* layer with the 16 nodes and the 3D positions as output data. In the next section, we will show validation of this architecture via an ablation study. Figure 2 illustrates the design of our network that still has a residual layer for refinement purposes. Our 3D network model was trained over 100 epochs, using the *Adam optimizer* with a learning rate of $1e-3$, rate decay of 0.5, and batches of size 64. We also use the *Xavier* normal function to initialize the weights of each layer. Furthermore, we use a function based on the Mean per joint position error [5], [22] and its derivative [23] for our loss function as:

$$L(J) = \frac{1}{N}\sum_{i=1}^{N} \|(f(B_j) - B_j)\|_2^2 + \left\|(f'(B_j) - B'_j)\right\|_2^2, \tag{8}$$

where $f(B)$ are the 3D joints coordinates predicted by our neural network, B are the corresponding ground-truth.

We built our framework aiming to be lightweight and accessible. It does not need any specialized hardware or high-end GPUs. Thus, it allows the creation of experiences based on pose estimation in ordinary devices. Furthermore, it is straightforward and customizable. As proof of concept, we generate BVH files, which is a character animation file format, with the captured data. The data can be used in computer animation, where we export the captured information to commercial and Open Source 3D animation suites, as we can see in Figure 3.

V. Experimental Results

A. Datasets

Following the standard protocol, we evaluated the proposed method using the Human3.6M dataset for 3D human pose

Protocol	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD	Smoke	Wait	WalkD	Walking	WalkT	Average
Martinez et al. ICCV'17 [18]	51.8	56.2	58.1	59	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Yang et al. CVPR'18 [24]	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Mehta et al. SIGGRAPH'17 [16]	62.6	78.1	63.4	72.5	88.3	93.8	63.1	74.8	106.6	138.7	78.8	73.9	82.0	55.8	59.6	80.5
Hossain & Little ECCV'18 [25]	48.4	50.7	57.2	55.2	63.1	72.6	53.0	51.7	66.1	80.9	59.0	57.3	62.4	46.6	49.6	58.3
Pavllo et al. CVPR'19 [23]	45.1	47.4	42.0	46.0	49.1	56.7	44.5	44.4	57.2	66.1	47.5	44.8	49.2	32.6	34.0	47.1
Dabra et al ECCV'18 [26]	44.8	50.4	44.7	49.0	52.9	43.5	45.5	63.1	87.3	51.7	61.4	48.5	**37.6**	52.2	41.9	52.1
Zhao et al CVPR'19 (SH) [3]	48.2	60.8	51.8	64.0	64.6	53.6	51.1	67.4	88.7	57.7	73.2	65.6	48.9	64.8	51.9	60.8
Zhao et al CVPR'19 (GT) [3]	37.8	49.4	37.6	40.9	45.1	**41.4**	40.1	48.3	50.1	**42.2**	53.5	44.3	40.5	47.3	39.0	43.8
Our Model (SH)	43.45	49.16	51.18	50.51	63.56	62.54	43.31	45.50	70.12	87.78	56.58	49.04	53.15	40.96	46.01	54.19
Our Model (CPM)	52.3	62.8	60.4	62.14	87.73	79.76	58.33	60.44	85.42	88.64	69.82	64.69	66.67	52.92	55.19	69.5
Our Model (GT)	**34.01**	**40.18**	**31.86**	**35.91**	**38.55**	47.64	**39.4**	**34.03**	**44.51**	60.76	**37.27**	**37.86**	39.08	**29.51**	**31.67**	**38.82**

TABLE I: Results under Protocol 1 on Human3.6M (no rigid alignment in post-processing). We show the results of our model (trained and tested with ground truth data(GT) and 2D predictions from a Convolutional Pose Machine (CPM) and a Stacked Hourglass (SH). Note that, on average, our model surpasses the previous state-of-the-art approach considering GT and SH predictions. The results of all approaches are obtained from the original papers.

estimation. This dataset is publicly available, containing more than 3 million images and 3D data captured by a *MoCap* system and the calculated 2D joints. The dataset includes data from 7 people performing everyday activities such as walking, eating, discussing, etc.

We adopt in our model evaluation the two metrics proposed in the paper that originated the dataset Human3.6M considering different approaches to split the data for training, validation, and testing. The first protocol, called *Mean Per Joint Position Error (MPJPE)*, consider all four camera views for all subjects. We used five subjects for training (1, 5, 6, 7, and 8) and 2 for testing (9 and 11). Furthermore, we calculate the error of the predictions and the ground-truth after aligning them with the root joint in our experiments represented by the pelvis keypoint. The second protocol is called *Mean per-joint position error after rigid alignment (P-MPJPE)*, which differs from the first protocol only on the alignment. We used the same division mentioned above for training, validation, and testing.

Moreover, we utilize a rigid transformation to align the predictions with the ground-truth data. All errors were analyzed in millimeters. We also use the COCO dataset [6], a state-of-the-art dataset for 2D human pose estimation in the wild. We use this dataset to generate 2D input for our method in a qualitative evaluation.

In a second experiment, we use the MPI-INF-3DHP dataset. It was built over a state-of-the-art markerless motion capture system and provided ground truth 3D annotations for human poses. This can be used as an alternative dataset to Human3.6, where it offers an extensive range of human motions, interactions with objects, and more varied camera viewpoints. In addition to Human 3.6 and 2D pose neural networks, we test our approach with this dataset to evaluate the accuracy and generalizability of our learned model. We also use the 3D Percentage of Correct Keypoints (PCK) as an evaluation metric. As proposed by Mehta et al. [7] , we pick a threshold of 150mm for the error, corresponding to roughly half of the head keypoint size.

B. 2D to 3D keypoints

Our method can be seen as an unprojection of 2D joint locations to 3D positions. First, we train our network considering the ground truth for 2D and 3D joint positions from the Human3.6M. However, for a fair evaluation of our method, we also train our network with 2D predictions from a *Convolutional Pose Machine* [14], and we also test a Stacked HourGlass pre-trained with the MPII dataset [27]. It is natural to say that our model depends on the quality of the output of a 2D pose detector and achieves the best results when we use as input the ground-truth 2D joint locations.

We use two pre-trained networks in the tests: a Stacked HourGlass trained on the MPII dataset [28] and a *CPM* with the COCO dataset [6]. The COCO dataset skeleton has a different configuration for the human body structure, following the order of keypoints compared with Human3.6M. We convert the output dictionary of this model to the Human3.6M and train our 3D network. We use the Stacked HourGlass [27] to a fair experiment to evaluate the 3D output for Human3.6 since the original SCG [3] and Martinez et al. [18], also use this architecture.

All 2D keypoints were previously generated in this process. The COCO skeleton has 18 joints considering five joints in the head. The Human3.6M skeleton consists of 16 joints, and we define the spine joint as the root joint. To convert to Human 3.6M and create the spine point, we consider the midpoint between the *lheap* and *rheap* of COCO, and we discoursed the thorax joint.

Also, in a second approach, we train our network on MPI-INF-3DHP pose dataset [7], and we compare the performance of our approach with different network architectures. This dataset is more complex considering poses, clothing, and skeleton structure. We use these data to evaluate the robustness and generalizability of our method.

C. Ablation Study and Network evaluation

We have also analyzed the impact of the chosen hyper-parameters and architecture on the final result in testing. We trained our network with different configurations and compared it to the baseline for *SGCs* [3] and the baseline for 3D Pose Estimation [18]. We considered the error analysis for protocol 1. In the first test, our models were trained for over 100 epochs under three configurations, as we can see in Table III. We first evaluate different design choices on a separate validation set for Human3.6, and then, we use the best option to compare to SGC [3] and Martinez et al. [18] on a test set. The first is a model with two internal blocks and 64 channels,

Protocol	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD	Smoke	Wait	WalkD	Walking	WalkT	Average
Martinez et al. ICCV'17 [18]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	38.0	47.7
Yang et al. CVPR'18 [24]	26.9	30.9	36.3	39.9	43.9	47.4	28.8	29.4	36.9	58.4	41.5	30.5	29.5	42.5	32.2	37.7
Hossain & Little ECCV'18 [25]	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
Pavllo et al. CVPR'19 [23]	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Dabra et al. ECCV'18 [26]	28.0	**30.7**	39.1	34.4	37.1	**28.9**	31.2	39.3	60.6	**39.3**	44.8	**31.1**	**25.3**	37.8	28.4	36.3
Our Model (2 blocks, 128 ch)	**27.95**	34.19	**30.88**	**30.23**	**32.70**	39.90	**31.03**	**30.33**	**42.01**	47.89	**33.50**	31.99	33.29	**25.40**	**26.92**	**33.22**

TABLE II: Results of protocol 2 on Human3.6M under rigid alignment in post-processing. Tests where performed considering model trained with ground truth data. Note that in most cases, our model surpasses the previous works. The results of all approaches are obtained from the original papers.

the second is our regular model with two inner blocks, and in the third, we use a model with four internal blocks and 128 channels. Table III shows that with the second configuration, our model performs better than the baseline algorithm from Zhao et al. [3] and the other two configurations.

In the second test, we compare our model with the state-of-art approaches in a test set, for 2D joints to 3D pose regression, following two configurations: with and without the attention layer. Table IV reports the result. Also, we analyze the impact of using as input 2D prediction from a *CPM*. We use a network configuration with attention layers and two blocks, and 128 channels per layer. Table V shows the quality of 3D predictions highly depends on the input since the error increases when we use a pre-trained *CPM* in the COCO dataset. We compare our technique with 2 state-of-the-art frameworks for 3D pose and shape estimation: XNect [17] and VIBE [19]. These projects have a different research target than ours. Still, we decided to compare our method due to evaluation completeness as we can see our model surpass their results when considering the ground truth for the Human3.6 dataset. However, as said before, our model depends highly on the quality of 2D inputs; when we use the CPM predictions, our performance is reduced.

Model	# Parameters	MPJPE (mm)
2 blocks and 64 channels	0.06 M	43.88
2 blocks and 128 channels	0.18 M	38.82
4 blocks and 128 channels	0.36 M	41.04

TABLE III: Evaluation of our parameters for the 3D pose estimation model. The error is computing in the testing dataset. As we can see, our best configuration has approximately 58% fewer parameters than the baseline achieving the state-of-art performance.

Model	# of Parameters	MPJPE (mm)
SGC [3]	0.43 M	43.8
Martinez et al. [18]	4.29 M	45.5
Ours without attention (2 blocks and 128 ch)	0.16 M	46.71
Ours with attention (2 blocks and 128 ch)	0.18 M	**38.82**

TABLE IV: 3D pose regression errors and the parameter numbers of our networks with different settings on Human3.6M. For each technique, we use the 2D ground truth data for the training and evaluation.

We evaluated our 3D unprojection model following the dataset Human3.6M. Table I shows the result using 2D ground-truth of Human3.6M, CPM, and Stacked HourGlass predictions for testing. The results are competitive and, on average,

Model	MPJPE (mm)	P-MPJPE (mm)
Ours (Ground Truth)	**38.82**	**33.22**
Ours (CPM detections)	69.5	54.09
Xnect [17]	63.6	-
VIBE [19]	65.6	41.4

TABLE V: 3D pose regression errors with different inputs. We use 2D ground-truth from Human3.6M and 2D predictions from a CPM. We compare our results with the stage 2 output of Xnect [17]. The metrics for Xnect and Vibe were obtained from the original papers.

Model	MPJPE (mm)	3D PCK
Vnect [16]	124.7	76.7
M3DHP [7]	117.6	75.7
Mehta [29]	122.2	75.2
Xnect (stage 2) [17]	98.4	82.8
Xnect (stage 3) [17]	115.0	77.8
Kanazawa [20]	124.2	72.9
Kundu [30]	103.8	82.1
VIBE [19]	96.6	89.3
Ours (CPM)	105.17	81.27
Ours (GT H3.6)	80.28	91.0
Ours (GT)	**76.39**	**92.0**

TABLE VI: Comparison on the single person MPI-INF-3DHP dataset. Top part are methods designed and trained for single-person capture.The Xnect is multi-person method, however we evaluate only single person predictions. We tested models trained over the 2D ground-truth from Human3.6M, 2D predictions from a CPM, and ground truth data of MPI-INF-3DHP.

our performance is better than the state-of-the-art. We consider our model trained and tested with ground-truth data as an upper bound of our method since it uses only 2D ground truth (GT) as the input. Our technique outperforms the state-of-the-art *SGC*, for 3D pose regression, [3] by 12.83% considering Ground Truth and 12.19% considering a 2D Stacked Hourglass [27] predictions. Also, our model with attention layers surpasses the model only with regular *SGCs* (without attention) by almost 17%. It is noteworthy that our approach has much fewer parameters, meaning that using the attention module drastically reduces the network's computational complexity and improves the overall performance. We have approximately 58% fewer parameters than the baseline *SGC* [3] and 95% fewer parameters than the model from Martinez et al. [18].

Fig. 4: Visual results of our method on in-the-wild images from COCO dataset [6] . In most cases, our technique can effectively predict 3d joints in different situations. Small errors can be seen considering the image scale and camera projection. In the last row, in an example with self-occlusion, our model cannot predict data from incomplete data.

Most methods have sophisticated frameworks [23]–[25] or learning strategies. They were trained and focus on in-the-wild images, propose end-to-end frameworks to generate the 3D pose directly from images, consider temporal information and also use complex loss functions [23], [26]. Due to more data variability and their proposed constraints to reduce prediction error, they were expected to have better performance, including ground truth. However, this is not true. Our model surpasses the previous works, considering the *MPJPE* and *P-MPJPE* for ground truth, proving the potential of the attention layer. The tests consider each action of the motion capture dataset. Table I shows the error in millimeters for each step following protocol 1 *MPJPE*.

Our results on Human3.6M under protocol 2 (using a rigid alignment with the ground-truth), are shown in Table II. In most cases, our method surpasses the previous works and has better performance on average. Note that in some cases, our model has similar performance or worse than the model from Dabra et al. [26]. However, our approach has fewer parameters to compute and does not need complex anatomically loss functions or a sophisticated, weakly supervised learning framework. Also, on average, our model outperforms Dabra et al. [26] by 8.3%. In Table VI, we compare the 3D pose output on the MPI-INF-3DHP dataset [7] using the 3D Percentage of Correct Keypoints (3DPCK - higher is better) and MPJPE. We prove the robustness of our method, where for both metrics, we surpass the previous state-of-the-art approaches when using 2D ground truth data. Again, note that most of these methods are built over sophisticated frameworks and can predict multi-person poses and shapes. In contrast, our method can be seen as a 2D to 3D unprojection. Also, this test confirms the hypothesis that our prediction quality highly depends on 2D inputs. As we can see when considering the CPM detection, our performance is reduced but still competitive. We also tested our model trained on 2D ground truth data for Human 3.6 on MPI-INF-3DHP test data. As we can see, as said before, we prove the robustness of our technique, outperforming the previous works.

Moreover, considering runtime performance, our 3D network took, on average, 10 seconds to evaluate 1062 poses. The tests were performed in a GPU Nvidia RTX 2060 with 6GB of memory, where we repeat each test 1000 times. In terms of the number of parameters, our network has 0.18M, while the model proposed by Zhao et al. [3] has 0.43M. This means that our network is lightweight and could be part of a complete system that infers 3D human pose in real-time.

D. Qualitative and Visual Analysis

Figure 4 illustrates some results generated using images from COCO dataset [6]. Our model can accurately predict 3D poses from these images indicating that it effectively encodes relationships among body joints and can generalize the results to different situations. The input of the method is the 2D joints generated using a *Convolutional Pose Machine*. However, our model also has some limitations, as we can see in the last row of figure 4. For example, when using data predicted by a *CPM*, if the 2D detector output fails to detect all body keypoints, our model can't recover the missing information. Also, it is not uncommon to see images with occluded or incomplete poses for in the wild examples. Our model has difficulty dealing with these cases. Both approaches proposed by Martinez et al. [18] and Zhao et al. [3] have the same issue.

Figure 5 shows the results of our technique applied on Human3.6M. In another approach, the input is generated by the method from Schirmer et. al. [31], and from the output we created a *BVH* model to generate the animation.

VI. Conclusions and Future Work

We present a novel model for attention layers in Semantic Graph Convolutions. With this approach, we build a lightweight 3D human pose estimation model to project 2D keypoints from the output of a convolutional pose machine in a 3D space. Our model can be seen as an unprojection from 2D to 3D keypoints. The combination of *SGCs* with attention layers improves the performance and reduces the overall complexity of our model, and we achieve state-of-the-art performance with 58% fewer parameters. However, as said before, the prediction quality depends on the 2D inputs. If the 2D predictor fails in generating the correct input data, our model will also fail to regress the data.

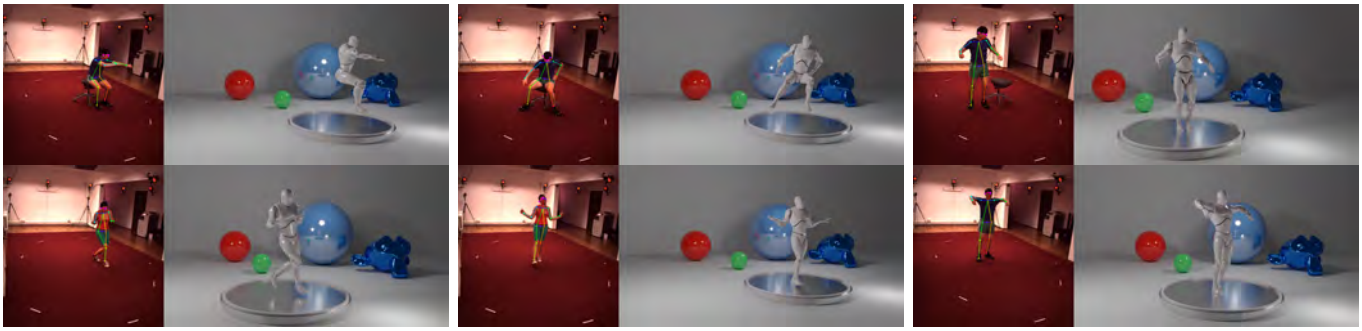

Fig. 5: Visual results of our method on Human3.6M [5]. As we can see, our method is robust but still has minor issues considering joint rotations. As we said before, our model focus only on project the human keypoints in a 3D space.

Since this method uses a small set of parameters, we intend to adapt it for applications in edge devices as future works. We believe that our pose estimation model can be handy for people to easily create 3D animations without any specialized hardware.

REFERENCES

[1] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[3] 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 CVPR*, 2019, pp. 3425–3435.

[4] Q. Huang, F. Zhou, J. He, Y. Zhao, and R. Qin, "Spatial–temporal graph attention networks for skeleton-based action recognition," *Journal of Electronic Imaging*, vol. 29, no. 5, p. 053003, 2020.

[5] 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 TPAMI*, vol. 36, no. 7, pp. 1325–1339, 2013.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[7] 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 *2017 Proceedings of 3DV*. IEEE, 2017, pp. 506–516.

[8] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[9] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[10] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[12] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE CVPR*, 2018, pp. 7132–7141.

[13] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Gcnet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[14] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE CVPR*, 2016, pp. 4724–4732.

[15] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3d human pose machines with self-supervised learning," *IEEE TPAMI*, vol. 42, no. 5, pp. 1069–1082, 2019.

[16] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," *ACM Transactions on Graphics*, 2017.

[17] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, M. Elgharib, P. Fua, H.-P. Seidel, H. Rhodin, G. Pons-Moll, and C. Theobalt, "Xnect: Real-time multi-person 3d motion capture with a single rgb camera," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 82–1, 2020.

[18] 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 ICCV*, 2017, pp. 2640–2649.

[19] M. Kocabas, N. Athanasiou, and M. J. Black, "Vibe: Video inference for human body pose and shape estimation," in *Proceedings of the IEEE/CVF CVPR*, 2020, pp. 5253–5263.

[20] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proceedings of the IEEE CVPR*, 2018.

[21] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[22] A.-I. Popa, M. Zanfir, and C. Sminchisescu, "Deep multitask architecture for integrated 2d and 3d human sensing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6289–6298.

[23] 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 CVPR*, 2019, pp. 7753–7762.

[24] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *Proceedings of the IEEE CVPR*, 2018, pp. 5255–5264.

[25] M. Rayat Imtiaz Hossain and J. J. Little, "Exploiting temporal information for 3d human pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 68–84.

[26] R. Dabral, A. Mundhada, U. Kusupati, S. Afaque, A. Sharma, and A. Jain, "Learning 3d human pose from structure and motion," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 668–683.

[27] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

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

[29] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *2018, Proceedings of 3DV*. IEEE, 2018, pp. 120–130.

[30] J. N. Kundu, S. Seth, V. Jampani, M. Rakesh, R. V. Babu, and A. Chakraborty, "Self-supervised 3d human pose estimation via part guided novel image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6152–6162.

[31] L. J. S. Silva, D. L. S. da Silva, A. B. Raposo, L. Velho, and H. C. V. Lopes, "Tensorpose: Real-time pose estimation for interactive applications," *Computers & Graphics*, 2019.