Occlusion Handling in 3D Human Pose Estimation with Perturbed Positional Encoding

Niloofar Azizi[1], Mohsen Fayyaz[2], and Horst Bischof[1]

[1] Graz University of Technology, Graz, Austria {azizi, bischof}@tugraz.at
[2] Microsoft
mohsenfayyaz@microsoft.com

Abstract. Understanding human behavior fundamentally relies on accurate 3D human pose estimation. Graph Convolutional Networks (GCNs) have recently shown promising advancements, delivering state-of-the-art performance with rather lightweight architectures. In the context of graph-structured data, leveraging the eigenvectors of the graph Laplacian matrix for positional encoding is effective. Yet, the approach does not specify how to handle scenarios where edges in the input graph are missing. To this end, we propose a novel positional encoding technique, PerturbPE, that extracts consistent and regular components from the eigenbasis. Our method involves applying multiple perturbations and taking their average to extract the consistent and regular component from the eigenbasis. PerturbPE leverages the Rayleigh-Schrodinger Perturbation Theorem (RSPT) for calculating the perturbed eigenvectors. Employing this labeling technique enhances the robustness and generalizability of the model. Our results support our theoretical findings, e.g. our experimental analysis observed a performance enhancement of up to 12% on the Human3.6M dataset in instances where occlusion resulted in the absence of one edge. Furthermore, our novel approach significantly enhances performance in scenarios where two edges are missing, setting a new benchmark for state-of-the-art.

1 Introduction

Estimating the 3D pose of the human skeleton is crucial for understanding human motion and behavior, which facilitates high-level computer vision tasks such as action recognition [26] and augmented and virtual reality [14]. Nevertheless, estimating 3D human joint positions presents considerable obstacles. First, there is a scarcity of labeled datasets, as acquiring 3D annotations is costly. Additionally, challenges such as self-occlusions, complex joint inter-dependencies, and small and barely visible joints further complicate the estimation process.

To address the challenge of estimating 3D human poses, several strategies have been explored, including leveraging multi-view setups [42], utilizing synthetic data [39], or incorporating motion analysis [45]. However, these methods can be cost-prohibitive, with multi-view configurations being impractical for real-world applications and the analysis of temporal data requiring significant resources. A more resource-efficient approach is lifting 2D-to-3D skeletons. The 2D human skeleton is a graph-structured data and thus Graph Convolutional Networks (GCNs) achieve state-of-the-art performance for 2D-to-3D human pose estimation by reducing the number of parameters in the order of magnitude, e.g. [2].

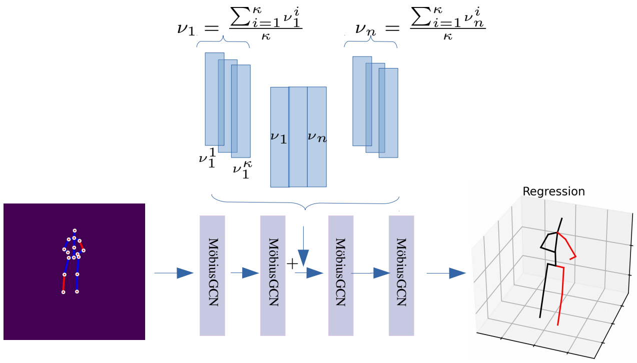
$\nu_1 = \frac{\sum_{i=1}^{\kappa} \nu_1^i}{\kappa}$
$\nu_n = \frac{\sum_{i=1}^{\kappa} \nu_n^i}{\kappa}$
ν_1^1
ν_1^κ
ν_1
ν_n
MöbiusGCN
MöbiusGCN
+
MöbiusGCN
MöbiusGCN
Regression

Fig. 1: Given a human skeleton graph with only blue edges provided and red edges absent, we calculate the corresponding graph Laplacian matrix. By selectively removing several edges at random (the selected elements in the graph Laplacian matrix), we then compute the perturbed eigenvectors (ν_i), which are subsequently averaged to identify consistent eigenbasis ($\frac{\sum_{i=1}^{\kappa} \nu_i}{\kappa}$). Finally, we integrate the perturbed positional features into a graph neural network architecture. The perturbed eigenvectors are computed with Rayleigh-Schrödinger Perturbation Theorem (RSPT).

Nevertheless, GCNs have limited expressivity. The seminal work [32] shows that GCNs are as expressive as 1-Weisfeiler-Lehman (1-WL) [51]. Efforts aimed at augmenting the expressiveness of GCNs have pursued four main directions: aligning with the k-WL hierarchy [28], enriching node features with identifiers, or exploiting structural information that cannot be captured by the WL test [4]. Subgraph GNNs [3] have emerged as a potential solution, enriching GNN features by encoding extracted subgraphs as novel features and incorporating them into the GNN architecture. One line of work addresses the limited expressivity of GCNs by positional encoding which utilizes the eigenbasis of graph Laplacian matrix [10, 21]. However, if some edges are missing these methods are not applicable. This scenario may occur in human pose estimation when the subject is not completely visible due to obstructions, e.g., occlusion.

To address this problem, we propose a novel perturbed **P**ositional **E**ncoding (PerturbPE), where we label the nodes by considering that graph Laplacian matrix has regular and irregular parts [23]. We extract the regular part of the graph Laplacian eigenbasis for the positional encoding, by applying multiple perturbations (, i.e., removing different sets of edges each time) to the initial graph Laplacian matrix. After computing the perturbed eigenvector each time, we average over the perturbed eigenvectors to extract the consistent part of the graph Laplacian eigenbasis with missed edges. Employing this labeling technique enhances the robustness and generalizability of the model. In other words, the eigenvectors can well reflect network structural features. As proved in [23], the regularity of the network is reflected in the consistency of structural features before and after a random removal of a small set of links.

We summarize our main contributions as follows:

- We provide a novel positional encoding, PerturbPE, to extract the regular part of the graph Laplacian eigenbasis for scenarios where some edges are missing in the input graph, e.g., scenarios where occlusion happens in the input human 2D skeleton.We utilize Rayleigh-Schrödinger Perturbation Theorem (RSPT) to compute the eigenpairs. Its primary advantage is that it eliminates the need to calculate the entire eigenbasis.
- We attain state-of-the-art by while training only one neural network to adeptly manage all scenarios where specific edges (specific parts of the 2D human skeleton) are missing.
- We also provide the state-of-the-art results of positional encoding for the task of 3D human pose estimation, achieving state-of-the-art 3D human pose estimation results, despite requiring the same number of model parameters as MöbiusGCN [2] on benchmark datasets.

2 Related Work

3D Human Pose Estimation Classical methods for 3D human pose estimation primarily rely on had-engineered features and incorporate prior knowledge (*e.g.,* [16, 41, 47]). While these techniques have achieved commendable outcomes, their main limitation is their inability to generalize. Current state-of-the-art in computer vision, including 3D human pose estimation, primarily utilize Deep Neural Networks (DNNs) [19, 25, 27]. These models operate under the assumption that input data exhibits characteristics of locality, stationarity, and multi-scalability. While DNNs excel in areas defined by Euclidean geometry, many challenges in the real world do not conform to this Euclidean framework.

Graph Convolutional Networks (GCNs) have emerged as a powerful solution for these problem categories, encompassing two main types: spectral and spatial GCNs. Spectral GCNs operate on the principles of the Graph Fourier Transform to analyze graph signals via the graph's Laplacian matrix vector space. On the other hand, spatial GCNs focus on transforming features and aggregating neighborhood information directly on the graph, *e.g.,* Message Passing Neural Networks [12] and GraphSAGE [13].

GCNs stand out for delivering high-quality results with a relatively small number of parameters in the task of 3D human pose estimation. Various studies [22, 53, 57], have explored pose estimation using GCNs. Xu and Takano [53] introduced the Graph Stacked Hourglass Networks (GraphSH), a method that processes graph-structured features across multiple levels of human skeletal structures. Similarly, Liu et al. [22] delved into diverse strategies for feature transformation and the aggregation of neighborhood spatial features, highlighting the advantage of assigning unique weights to enhance a node's self-information. Zhao et al. [57] developed the semantic GCN (SemGCN) focusing on the concept of learning the adjacency matrix to capture semantic relationships between graph nodes. MöbiusGCN [2], dramatically reduced parameter count to just 0.042M by leveraging the Möbius transformation to explicitly define joint transformations, showcasing an even more efficient model architecture for pose estimation.

GCN Expressivity Nonetheless, the expressiveness of GCNs is constrained, and to surmount this challenge, four principal approaches have been proposed. Evaluating the expressive capacity of GNNs requires dealing with the graph isomorphism, which has no P solution (NP-intermediate). The 1-WL test [51], effectively resolves graph isomorphism for a vast majority of graph-structured data. However, as indicated in research by Li and Leskovec [20], Morris et al. [32], Xu et al. [52], the expressivity of MPNNs is limited to that of the 1-WL test. This limitation becomes particularly significant in real-world applications involving graph structures, as the 1-WL test cannot distinguish between certain graph features. Specifically, it fails in differentiating isomorphism in attributed regular graphs, measuring distances between nodes, and counting cycles within graphs [20]. Addressing these limitations, recent studies have provided four different approaches. The first approach involves adding random attributes to nodes with identical substructures. This method aims to provide uniqueness to these nodes, enhancing their capability to be differentiated. However, this advantage comes at the cost of reduced deterministic predictability, potentially leading to difficulties during inference [43]. Furthermore, deterministic positional features (*e.g.,* [56]) argue that the incapability of the GNNs to encode the distance between nodes in the input graph raises the above issues and addresses them by injecting deterministic distance attributes. However, these methods assign different node features to isomorphic graphs and, thus, are not generalizable in inference time [20]. The third strategy involves developing higher-order GNNs [28] to surpass the 1-WL test's expressivity limits. The fourth is adopting subgraph-based methods like ESAN [3], which enhance expressivity through selected subgraphs.

Nevertheless, if some edges are missing, none of the previously mentioned methods can be applied to enhance expressivity. In the realm of 2D human pose estimation, this significant challenging scenario can happen when occlusion happens. When parts of the human body are occluded or hidden from view, accurately estimating the 3D human pose becomes more difficult. This problem is prevalent in real-world scenarios where objects may obstruct the view, or the camera angle limits visibility.

To address this problem we propose a novel positional encoding. In the seminal work [10], it was proposed that incorporating graph Laplacian eigenvectors as positional features within the Graph Neural Network (GNN) architecture could improve its effectiveness, although this approach encountered obstacles such as sign ambiguity and eigenvalue multiplicity. To address these challenges, SignNet/BasisNet [21] was developed. A Multilayer Perceptron (MLP) was employed to deal with the issue of multiplicity, while an approach using an even function—by amalgamating the function with its inverse—was implemented to tackle problems related to sign ambiguity. However, to the best of our knowledge, none of the previous works provided a solution in case some edges missing, which can be reduced to subgraph matching which is NP-complete. We propose extracting the consistent and robust part of the graph Laplacian matrix's eigenbasis by labeling the joints (graph nodes) utilizing the perturbed eigenbasis of the graph Laplacian matrix. To compute the perturbed eigenbasis, we employ the Rayleigh Schrodinger Perturbation Theory (RSPT).

Data Reduction Semi-supervised techniques are favored due to the tedious and costly nature of data annotation. MöbiusGCN [2] stands as the most efficient framework for 3D human pose estimation tasks to date, thereby diminishing the need for extensive annotated data for training. The advantage of light architectures is their ability to be trained with less data. Our novel perturbed positional encoding architecture, which does not add additional training parameters, maintains the framework's efficiency, enabling it to achieve leading results with a minimal amount of labeled data.

3 Preliminaries

To compute the consistent part of the graph Laplacian eigenbasis for the positional encoding, we use the perturbed eigenvectors. This process involves using the Rayleigh-Schrödinger Perturbation Theorem (RSPT) to determine the perturbed eigenvectors of the graph Laplacian matrix. Hence, we provide a concise overview of the RSPT. Moreover, while PerturbPE can enhance any GNN framework, our experiments are conducted using MöbiusGCN, a model developed specifically for the task of 3D human pose estimation. Therefore, we also briefly overview the MöbiusGCN.

3.1 Rayleigh-Schrödinger Perturbation Theory

In mathematical terms, we are dealing with a discretized Laplacian-type operator represented by a real symmetric matrix that undergoes a minor symmetric linear perturbation.

$$\mathbf{A}(\epsilon) = \mathbf{A}_0 + \epsilon \mathbf{A}_1. \tag{1}$$

The Rayleigh-Schrödinger Perturbation Theory (RSPT) [30] provides estimates for the eigenvalues and eigenvectors of the matrix $\mathbf{A}$ through a series of progressively higher-order adjustments to the eigenvalues and eigenvectors of the matrix $\mathbf{A_0}$. $\mathbf{A_0}$ is likewise real and symmetric but may possess multiple eigenvalues.

An advantage of RSPT is that it doesn't necessitate the full set of $\mathbf{A_0}$ and can be approached using the Moore-Penrose Pseudoinverse. The pseudoinverse need not be explicitly calculated since only pseudoinverse vector products are required. These may be efficiently be calculated by a combination of QR-factorization and Gaussian elimination. Since we are only concerned with real-symmetric matrices, the existence of a complete set of orthonormal eigenvectors is assured.

Reconstruction of Perturbed Matrix To compute the eigenpairs of $\mathbf{A}$ possessing respective perturbation expansions

$$\lambda_i(\epsilon) = \sum_{k=0}^{\infty} \epsilon^k \lambda_i^{(k)} \quad \mathbf{v}_i(\epsilon) = \sum_{k=0}^{\infty} \epsilon^k \mathbf{v}_i^{(k)}, \tag{2}$$

where $(i = 1, \ldots, n)$ for sufficiently small ϵ. Experimentally, we set both ϵ and k to 1.

Considering the eigenvalue problem and and taking into account the equations 1 and 2, we have to solve the following recurrence relation

$$(\mathbf{A}_0 - \lambda_i^{(0)} \mathbf{I}) \mathbf{x}_i^{(k)} = -(\mathbf{A}_1 - \lambda_i^{(1)} \mathbf{I}) \mathbf{x}_i^{(k-1)} + \sum_{j=0}^{k-2} \lambda_i^{k-j} \mathbf{x}_i^{(j)} \tag{3}$$

for $(k = 1, \ldots, \infty; i = 1, \ldots, n)$.

The solution is either degenerate or non-degenrate.

Nondegenerate Case By assuming all the eigenvalues are distinct the eigenpairs are computed as follows.

If j is odd, then

$$\lambda_i^{2j+1} = \langle \mathbf{x}_i^{(j)}, \mathbf{A}_1 \mathbf{x}_i^{(j)} \rangle - \sum_{\mu=0}^{j} \sum_{\nu=1}^{j} \lambda_i^{(2j+1-\mu-\nu))} \langle \mathbf{x}_i^{(\nu)}, \mathbf{x}_i^{(\mu)} \rangle. \tag{4}$$

If j is even, then

$$\lambda_i^{2j} = \langle \mathbf{x}_i^{(j-1)}, \mathbf{A}_1 \mathbf{x}_i^{(j)} \rangle - \sum_{\mu=0}^{j} \sum_{\nu=1}^{j} \lambda_i^{(2j-\mu-\nu))} \langle \mathbf{x}_i^{(\nu)}, \mathbf{x}_i^{(\mu)} \rangle. \tag{5}$$

The corresponding eigenvector is computed as follows

$$\mathbf{x}_i^{(k)} = (\mathbf{A}_0 - \lambda_i^{(0)} \mathbf{I})^{\dagger} [-(\mathbf{A}_1 - \lambda_i^{(1)} \mathbf{I}) \mathbf{x}_i^{(k-1)} + \sum_{j=0}^{k-2} \lambda_i^{(k-j)} \mathbf{x}_i^{(j)}] \tag{6}$$

The unperturbed eigenvectors are assumed to have been normalized to unity so that $\lambda_i^{(0)} = \langle \mathbf{x}_i^{(0)}, \mathbf{A}_0 \mathbf{x}_i^{0} \rangle$.

Degenerate Case For the calculation of perturbed eigenpairs in scenarios involving multiplicity, $\lambda_1^{(0)} = \lambda_2^{(0)} = \cdots = \lambda_m^{(0)} = \lambda^{(0)}$, accompanied by m known orthonormal eigenvectors $\mathbf{v}_1^{(0)}, \ldots, \mathbf{v}_m^{(0)}$ and the assumption of first-order degeneracy which ensures the uniqueness of the first-order eigenvalues, the calculation of these perturbed eigenvalues and their corresponding degenerate eigenvectors are achieved by determining appropriate linear combinations of

$$\mathbf{y}_i^{(0)} = a_1^{(i)} \mathbf{v}_1^{(0)} + a_2^{(i)} \mathbf{v}_2^{(0)} + a_3^{(i)} \mathbf{v}_3^{(0)} + \cdots + a_m^{(i)} \mathbf{v}_m^{(0)}, \tag{7}$$

To have a solution for Equation (3) in this scenario, it is necessary and sufficient that for each fixed i,

$$\langle x_\mu^{(0)}, (\mathbf{A}_1 - \lambda_i^{(1)} \mathbf{I}) \mathbf{y}_i^{(0)} \rangle = 0 \quad (\mu = 1, \ldots, m) \tag{8}$$

By replacing Equation (7),

$$\begin{bmatrix} \langle \mathbf{x}_1^{(0)}, \mathbf{A}_1 \mathbf{x}_1^{(0)} \rangle & \ldots & \langle \mathbf{x}_1^{(0)}, \mathbf{A}_1 \mathbf{x}_m^{(0)} \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{x}_m^{(0)}, \mathbf{A}_1 \mathbf{x}_1^{(0)} \rangle & \ldots & \langle \mathbf{x}_m^{(0)}, \mathbf{A}_1 \mathbf{x}_m^{(0)} \rangle \end{bmatrix} \begin{bmatrix} a_1^{(i)} \\ \vdots \\ a_m^{(i)} \end{bmatrix} = \lambda_i^{(1)} \begin{bmatrix} a_1^{(i)} \\ \vdots \\ a_m^{(i)} \end{bmatrix}$$

the corresponding eigenvector of each $\lambda_i^{(1)}$ becomes $[a_1^{(i)}, \ldots, a_m^{(i)}]^\top$.

3.2 Spectral Graph Convolutional Network

Graph Definitions Given a graph $\mathcal{G}(V, E)$ with vertices $V = \{\upsilon_1, \ldots, \upsilon_N\}$ and edges $E = \{e_1, \ldots, e_M\}$, where $e_j = (\upsilon_i, \upsilon_k)$ with $\upsilon_i, \upsilon_k \in V$. The adjacency matrix $\mathbf{A}$ marks 1 for connected vertices and 0 otherwise. The degree matrix $\mathbf{D}$ is diagonal, listing vertex degrees $\mathbf{D}ii$ for υi. Graph $\mathbf{A}$ is symmetric for undirected graphs. The graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and its normalized form $\bar{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, with $\mathbf{I}$ the identity matrix, are key in analyzing graph structure. $\bar{\mathbf{L}}$ is symmetric and positive semi-definite with ordered, real, non-negative eigenvalues λ_i and orthonormal eigenvectors $\mathbf{u}_i$. A graph signal $\mathbf{x} \in \mathbb{R}^N$ assigns values to vertices, and $\mathbf{X} \in \mathbb{R}^{N\times d}$ represents a d-dimensional signal on $\mathcal{G}$ [46].

Graph Fourier Transform Graph signals $\mathbf{x} \in \mathbb{R}^N$ admit a graph Fourier expansion $\mathbf{x} = \sum_{i=1}^{N} \langle \mathbf{u}_i, \mathbf{x} \rangle \mathbf{u}_i$, where $\mathbf{u}_i, i = 1, \ldots, N$ are the eigenvectors of the graph Laplacian [46]. Eigenvalues and eigenvectors of the graph Laplacian matrix are analogous to frequencies and sinusoidal basis functions in the classical Fourier series expansion.

Spectral Graph Convolutional Network Spectral GCNs [5] build upon the graph Fourier transform. Let $\mathbf{x}$ be the graph signal and $\mathbf{y}$ be the graph filter on graph $\mathcal{G}$. The graph convolution $*_{\mathcal{G}}$ can be defined as:

$$\mathbf{x} *_{\mathcal{G}} \mathbf{y} = \mathbf{U}\,\mathrm{diag}(\mathbf{U}^\top \mathbf{y})\mathbf{U}^\top \mathbf{x}, \tag{9}$$

where the matrix $\mathbf{U}$ contains the eigenvectors of the normalized graph Laplacian and $\odot$ is the Hadamard product. This can also be written as

$$\mathbf{x} \ *_{\mathcal{G}} \ g_\theta = \mathbf{U} g_\theta(\mathbf{\Lambda})\mathbf{U}^\top \mathbf{x}, \tag{10}$$

where $g_\theta(\mathbf{\Lambda})$ is a diagonal matrix consisting of the learnable parameters, and is a function of the eigenvalues $\mathbf{\Lambda}$. We utilize MöbiusGCN [2], which defines the function g_θ to be Möbius transformation.

Thus a MöbiusGCN block is

$$\mathbf{Z} = \sigma(2\Re\{\mathbf{U}\,\text{Möbius}(\mathbf{\Lambda})\mathbf{U}^\top \mathbf{X}\mathbf{W}\} + \mathbf{b}), \tag{11}$$

where $\mathbf{Z} \in \mathbb{R}^{N\times F}$ is the convolved signal matrix, σ is a nonlinearity (*e.g*. ReLU [33]), and $\mathbf{b}$ is a bias term and the graph signal matrix $\mathbf{X} \in \mathbb{C}^{N\times d}$ with d input channels (*i.e*. a d-dimensional feature vector for every node) and $\mathbf{W} \in \mathbb{C}^{d\times F}$ feature maps.

4 Method

Eigenvector Positional Encoding We aim to compute the consistent part of graph Laplacian eigenbasis to enhance generalizability, specifically in cases where the occlusions occur and the complete 2D human skeleton graph is not provided as an input to the architecture. To compute the positional encoding in such scenarios, we apply the Rayleigh-Schrödinger Perturbation Theory (RSPT) multiple times (κ-times) independently. During each iteration, we randomly eliminate a number of edges from the graph's Laplacian matrix (denoted as $\mathbf{A}_0$ in Equation (1)), and then compute the perturbation with respect to it (we update $\mathbf{A}_1$ with eliminated edges). By executing the RSPT for each of these iterations and averaging over them, the regular part of the κ eigenvectors can be extracted with

$$\mathbf{p} = \frac{\sum_{i=1}^{\kappa} \mathbf{v}_i}{\kappa}, \tag{12}$$

where $\mathbf{v}_i$ is the i^{th} perturbed eigenvector.

After executing the algorithm κ times and averaging the outcomes to isolate the stable elements of the graph Laplacian matrix's eigenbasis, we employ the following positional encoding technique to incorporate the features into the architecture.

Positional Features We incorporate positional features, similar to [10], into the architecture through the subsequent method.

$$\mathbf{X}^{\ell} = \sigma(f(\mathbf{Z}^{\ell} + \mathbf{P})) \tag{13}$$

where $\mathbf{P} \in \mathbb{R}^{N \times N}$ is the PerturbPE positional encoding computed with the RSPT. For each vector, we define the positional features, denoted as $\mathbf{p}$, as the mean of the perturbed eigenvectors where it contains the consistent regular part of the graph Laplacian eigenbasis. The function f represents a Multilayer Perceptron (MLP) that is applied to both node and positional features. Therefore, Equation 11 becomes

$$\mathbf{Z}^{\ell+1} = \sigma(2\Re\{\mathbf{U}\,\text{Möbius}(\mathbf{\Lambda})\mathbf{U}^{\top}\sigma(f(\mathbf{Z}^{\ell} + \mathbf{P}))\mathbf{W}^{\ell+1}\} + \mathbf{b}). \tag{14}$$

Masked Condition Strategy In our experiments, we operate under the assumption that each sample may lack certain components of the human skeleton, specifically that up to two edges between joints might be missing randomly during both testing and training phases. However, we assume that although some edges are missed the total number of joints is known.

5 Experimental Results

5.1 Datasets and Evaluation Protocols

We employ the widely recognized Human3.6M motion capture dataset for our study [16]. This extensive dataset encompasses over 3.6 million images, collected from 11 participants engaging in 15 distinct activities, captured through four calibrated RGB cameras. This setup was meticulously designed to ensure a comprehensive capture of each subject's movements, both during the training and testing phases. In alignment with previous works (*e.g.,* [2, 29, 37, 44, 48, 49, 53, 57]), our experimental framework utilizes the data from five subjects (S1, S5, S6, S7, S8) for model training purposes, while reserving two subjects (S9 and S11) for the testing phase. Each sample is independently analyzed, reflecting the unique viewpoints provided by each camera.

To evaluate our model's ability to generalize, we employ the MPI-INF-3DHP dataset [31]. This dataset features six subjects tested across three distinct settings: a studio with a green screen (GS), a studio lacking a green screen (noGS), and an outdoor environment (Outdoor). It's important to mention that for the experiments conducted using the MPI-INF-3DHP dataset, training was only performed on the Human3.6M dataset.

Following the methodology of prior research [29, 48, 49, 53, 57], we adopt the MPJPE protocol, Protocol #1. The MPJPE is the mean per joint position error in millimeters between predicted joint positions and ground truth joint positions after aligning the pre-defined root joints (*i.e.,* the pelvis joint). Note that some works (*e.g.,* [22, 38]) use the P-MPJPE metric, which reports the error after a rigid transformation to align the predictions with the ground truth joints. However, we deliberately chose the standard MPJPE metric as it is more challenging and the more equitable basis it provides for comparing our work with previous research.

In assessing performance on the MPI-INF-3DHP test set, in alignment with prior research [24, 53], we adopt the 3D Percentage of Correct Keypoints (3D PCK) with a 150mm threshold [31]. This measure allows us to accurately gauge the accuracy of our 3D joint predictions within a specified error margin, offering a comprehensive view of our model's performance in comparison to the benchmarks set by preceding studies.

5.2 Implementation Details

2D Pose Estimation. PerturbPE receives 2D joint positions as inputs, which are independently estimated from the RGB images captured by all four cameras. PerturbPE operates independently from any off-the-shelf architecture employed for the estimation of 2D joint positions. Although CPN [7] provides better 2D human skeleton estimation, similar to previous works [29, 57], we use the stacked hourglass architecture [34] to estimate the 2D joint positions. The Hourglass architecture is a type of autoencoder architecture that incorporates multiple skip connections at various intervals. In line with [57], the stacked hourglass network undergoes initial pre-training on the MPII dataset [1], followed by subsequent fine-tuning using the Human3.6M dataset [16]. As detailed by Pavllo et al. [38], the input joints are adjusted to fit within image coordinates and are normalized to the range of $[-1, 1]$.

3D Pose Estimation. The Human3.6M dataset [16] provides ground truth 3D joint positions in world coordinates. To align with previous works [2, 57], we utilize camera calibration parameters to transform these joint positions into camera space. Furtheremore, when training the pipeline, akin to previous studies [2, 29]a predefined joint (the pelvis joint) as the center of the coordinate system is selected.

We trained PerturbPE using Adam optimizer [17] with an initial learning rate of 0.001 and mini-batches of size 64. Our neural network pipeline, which operates in the complex-valued domain, is built upon the PyTorch framework [35], which leverages Wirtinger calculus [18] to enable backpropagation within the complex-valued domain.

We adopt MöbiusGCN as our baseline architecture due to its lightweight nature and impressive accuracy. In our experiments, we utilize eight MöbiusGCN blocks. Each block, excluding the first and last blocks with input and output channels set to 2 and 3 respectively, consists of either 128 channels (yielding 0.16 million parameters) or 192 channels (resulting in 0.66 million parameters). Furthermore, we incorporate positional encoding features in the BLA block, followed by a subsequent linear layer.

Same as [29, 57], we predict the normalized locations [22, 29, 36, 57] of 16 joints (*i.e.,* without the 'Neck/Nose' joint) in 3D and use the mean squared error (MSE) loss between the 3D ground truth joint locations $\mathcal{Y}$ and our predictions $\hat{\mathcal{Y}}$

$$\mathcal{L}(\mathcal{Y}, \hat{\mathcal{Y}}) = \sum_{i=1}^{k} (\mathcal{Y}_i - \hat{\mathcal{Y}}_i)^2, \tag{15}$$

where k is the number of joints [29, 38]. Furthermore, similar to [2, 40], to let the architecture differentiate between different 3D poses with the same 2D pose, the center of mass of the subject is provided as an additional input. Please note that during inference the scale of the outputs is calibrated by forcing the sum of the length of all 3D bones to be equal to a canonical skeleton [2, 59, 60].

All of our experiments were conducted using a PyTorch framework [35] on an NVIDIA GeForce RTX 2080 GPU.

RSPT Perturbed Eigenvectors In our experiments we consider computing the perturbed eigenvectors with different scenarios. We consider scenarios where zero, one, or two random edges are removed for computing the perturbed eigenvectors from the input occluded 2D human skeleton graph. Specifically, in our experiments, we consider graph Laplacian eigenvectors [10] positional encoding, RSPT eigenbasis with zero edge is missed for perturbation, one edge is missed for perturbation, or two edges are missed for computing the perturbation with RSPT.

Protocol #1	# Param.	Dir.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	**Average**
Martinez et al. [29]	4M	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Tekin et al. [49]	n/a	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
Sun et al. [48]	n/a	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Yang et al. [55]	n/a	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
Hossain and Little [15]	16.96M	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
Fang et al. [11]	n/a	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Pavlakos et al. [37]	n/a	48.5	54.4	54.5	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
SemGCN Zhao et al. [57]	0.43M	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
Sharma et al. [44]	n/a	48.6	54.5	54.2	55.7	62.2	72.0	50.5	54.3	70.0	78.3	58.1	55.4	61.4	45.2	49.7	58.0
GraphSH [53] ∗	3.7M	**45.2**	**49.9**	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
MöbiusGCN [2] (HG)	**0.16M**	46.7	60.7	47.3	50.7	64.1	61.5	46.2	45.3	67.1	80.4	54.6	51.4	55.4	43.2	48.6	52.1
Ours (HG)	0.66M	45.9	50.1	**41.2**	**43.2**	**52.7**	57.4	**43.0**	38.4	**55.4**	61.8	**45.8**	**46.8**	48.5	**38.9**	**42.8**	**50.8**
Liu et al. [22] (GT)	4.2M	36.8	40.3	33.0	36.3	37.5	45.0	39.7	34.9	40.3	47.7	37.4	38.5	38.6	29.6	32.0	37.8
GraphSH [53] (GT)	3.7M	35.8	38.1	31.0	35.3	**35.8**	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	**30.7**	35.8
SemGCN [57] (GT)	0.43M	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
MöbiusGCN [2] (GT)	**0.16M**	31.2	46.9	32.5	31.7	41.4	44.9	33.9	30.9	49.2	55.7	35.9	36.1	37.5	29.07	33.1	36.2
Ours (GT)	0.66M	**30.1**	**35.3**	**30.6**	**27.6**	36.2	**38.4**	**30.7**	**30.3**	**35.9**	**40.7**	**32.9**	**34.9**	**35.2**	**27.2**	32.0	**32.7**

Table 1: Quantitative Evaluation Using MPJPE (mm) on the Human3.6M [16] Dataset under Protocol #1, Highlighting Leading Performances. In the upper section, methods utilize stacked hourglass (HG) 2D estimates [34], with the exception of one approach using CPN [6] (denoted by ∗). The lower section compares methods based on 2D ground truth (GT) inputs. Best results are highlighted in **bold**, and the second-best are underlined. Lower is better.

5.3 Complete 2D Human Skeleton

In this study, we present a comparative analysis of PerturbPE's performance using a complete 2D human skeleton against the former leading techniques in 3D human pose estimation on the Human3.6M and MPI-INF-3DHP datasets. Our comparison utilizes two types of inputs: a) 2D poses estimated through the stacked hourglass architecture (HG) [34] and b) the 2D ground truth (GT).

Comparison on Human3.6M. Table 1 presents a comparison between our PerturbPE method and the leading techniques as per Protocol #1 in the Human3.6M dataset. Utilizing the eigenvector of the graph Laplacian matrix for positional encoding leads to enhanced performance, reducing the error rate from 34.1mm to 33.4mm without an increase in model complexity. By introducing perturbed eigenvectors to tackle the multiplicity issue, we improved further, lowering the error rate to 32.7mm. Notably, these advancements in positional encoding are achieved without the addition of extra parameters, ensuring the model remains efficient while benefiting from these enhancements. To evaluate the efficacy of our PerturbPE, we conducted experiments using SemGCN [57]. Please refer to the supplementary material in **??** for more details.

Comparison on MPI-INF-3DHP. To assess the adaptability and robustness of our method, we conducted evaluations using the MPI-INF-3DHP dataset, despite our model, PerturbPE, being exclusively trained on the Human3.6M dataset. This choice allowed us to test the generalizability of our approach beyond the conditions for which it was directly trained. In a comparative analysis with MöbiusGCN, PerturbPE exhibited outstanding performance, marking a notable improvement in the key metric of evaluation, PCK. With an identical configuration in terms of the number of parameters between the two models, our method demonstrated an overall enhancement in the PCK metric across various testing scenarios, improving the initial score of 80.0 to an improved score of 82.0. The significance of this improvement was even more pronounced in the most challenging conditions presented by outdoor scenarios, where our model achieved a state-of-the-art PCK score of 84.0. This result underscores the efficacy of PerturbPE in handling complex real-world situations. The results are in Table 2.

Method	# Parameters	GS	noGS	Outdoor	All(PCK)
Martinez et al. [29]	4.2M	49.8	42.5	31.2	42.5
Mehta et al. [31]	n/a	70.8	62.3	58.8	64.7
Luo et al. [24]	n/a	71.3	59.4	65.7	65.6
Yang et al. [55]	n/a	-	-	-	69.0
Zhou et al. [60]	n/a	71.1	64.7	72.7	69.2
Ci et al. [8]	n/a	74.8	70.8	77.3	74.0
Zhou et al. [58]	n/a	75.6	71.3	80.3	75.3
GraphSH [53]	3.7M	**81.5**	**81.7**	75.2	80.1
MöbiusGCN [2]	**0.16M**	79.2	77.3	83.1	80.0
Ours	**0.16M**	80.0	79.0	**84.0**	**82.0**

Table 2: Results on the MPI-INF-3DHP test set [31]. Best in bold, second-best underlined. All methods use 2D ground truth as input. Lower is better.

Comparison to Previous GCNs. Table 3 showcases how our approach stands up against previously established GCN models. By augmenting the channel count in each block of the MöbiusGCN architecture from 128 to 192, we observed an enhancement in the MPJPE metric by 2.1mm. Building on this improvement, our novel PerturbPE technique further advances MPJPE performance by an additional 1.4mm, all while maintaining an equivalent number of parameters to MöbiusGCN. This strategic enhancement effectively reduces the MPJPE from 34.1mm down to 32.7mm, illustrating the efficacy of our modifications in refining pose estimation accuracy.

Method	# Parameters	MPJPE
Liu et al. [22]	4.20M	37.8
GraphSH [53]	3.70M	35.8
Liu et al. [22]	1.05M	40.1
GraphSH [53]	0.44M	39.2
SemGCN [57]	0.43M	43.8
Yan et al. [54]	0.27M	57.4
Veličković et al. [50]	**0.16M**	82.9
MöbiusGCN [2]	**0.16M**	36.2
MöbiusGCN [2]	0.66M	$\underline{34.1}$
Ours	0.66M	**32.7**

Table 3: Supervised quantitative comparison between GCN architectures on Human3.6M [16] under Protocol #1. Best in bold, second-best underlined. All methods use 2D ground truth as input. Lower is better.

PerturbPE with Reduced Dataset. PerturbPE adeptly mirrors the parameter efficiency of the lightweight MöbiusGCN model, reaping the advantages of a compact design that necessitates a smaller volume of training data. By incorporating our novel positional encoding technique, PerturbPE dramatically refines its performance from 44.7mm to 42.9mm with MPJPE metric, achieved by analyzing data from just three subjects. This is particularly advantageous given the significant expense involved in acquiring 3D ground truth annotations. Moreover, we demonstrate that by further reducing the subject count to two and then to one, the results still show an improvement from 50.9mm to 48.9mm and from 67.4mm to 66.4mm, respectively. Results are detailed in Table 4. All experiments were conducted using ground truth 2D human skeleton data as the input.

Subject	# Parameters	MöbiusGCN [2]	PerturbPE
S1	0.15M	44.7	**42.9**
S1 S5	0.15M	50.9	**48.9**
S1 S5 S6	0.15M	67.4	**66.4**

Table 4: Evaluating the effects of using fewer training subjects on Human3.6M [16] under Protocol #1 (given 2D GT inputs). Lower is better.

5.4 Recover 3D pose from partial 2D observation

This section explores the impact of applying positional encoding through PerturbPE in scenarios where different numbers of edges are missing (*i.e.,* the problem reduced to subgraph matching). Initially, we examine scenarios with the absence of a single edge. Subsequently, we delve into the more challenging scenarios involving the omission of two edges. In the real world, these scenarios frequently arise since incomplete 2D pose estimates frequently occur in 2D human pose estimation. This typically happens when body parts are outside of the camera view or obscured by objects within the scene.

2D Human Skeleton with One Random Edge Missing In this first experiment, we eliminate one random edge from the input 2D human skeleton. We experiment the PerturbPE positional encoding under three conditions: no edge, one edge, and two edges missing for computing the perturbed eigenvectors.

Initially, when we compute the perturbed eigenvectors with no edge missing, which leads to addressing multiplicity, the results enhance accuracy from 55.0mm to 51.4mm. Further improvements achieved by averaging results from two applications of one-time perturbations, reducing the error to 49.0mm. Doubling the perturbation with two missing edges further refined accuracy to 48.0 mm, validating our theoretical approach.

This experiment demonstrates that increasing the frequency of perturbations and removing a greater number of edges leads to the extraction of the most robust components of the graph Laplacian eigenbasis. Consequently, leading to more consistent labeling, which in turn improves the outcomes during inference. However this improvement comes with the computational expense.

The results are demonstrated in Table 5, showcasing the effect of PerturbPE positional encoding in partially observed 2D human skeleton input graphs when one edge is missing, also compared to eigenvector labeling [10].

Method	MPJPE
PerturbPE Label w. Eigenvector	55.0
PerturbPE Label w. Perturbed Eigenvector w. multiplicity	51.4
PerturbPE Label w. Perturbed Eigenvector w. 1-edge perturb	49.0
PerturbPE Label w. Perturbed Eigenvector w. 2-edge perturb	**48.0**

Table 5: Evaluating the effects of positional encoding with one edge missing on Human3.6M [16] under Protocol #1 (given 2D GT inputs). Lower is better.

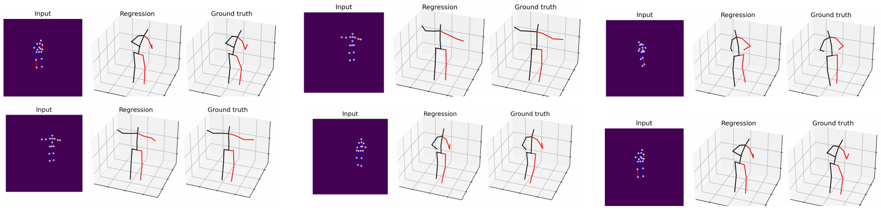

Fig. 2: Qualitative self-occlusion results of PerturbPE on Human3.6M [16]. This figure illustrates the PerturbPE's performance when trained to cope with the absence of any two arbitrary edges in the 2D human skeleton input. The network is trained for any two random edges missing in the input 2D human skeleton. A human skeleton graph with only blue edges provided and red edges absent.

2D Human Skeleton with Two Random Edges Missing In this experiment, we increase the difficulty of the task by removing two edges from the input 2D human skeleton. We experiment the PerturbPE positional encoding mainly under one condition: no edge is missing for computing the perturbed eigenvectors. In this scenario, we observe that when assigning labels by addressing multiplicity issues—the performance notably improves by approximately 10% (decreasing the MPJPE from 60.00mm to 54.00mm). The outcomes of this study are depicted in Table 6.

Further, we compare our results with GFPose [9]. While GFPose adopts a strategy of training distinct networks tailored to specific instances of missing body parts, our approach differs by training a single network that can handle any combination of missing edges. Despite this, our novel PerturbPE architecture significantly surpasses GFPose's performance, demonstrating the superior efficacy of our approach. Quantitative and qualitative results are shown in Table 7 and Figure 2, respectively. In experiments with a 2D human skeleton missing arms, accuracy improved from 60.0mm to 58.6mm. The absence of both legs further demonstrated the effectiveness of our PerturbPE method, enhancing performance to 52.4mm. PerturbPE outperforms GFPose when either the right or left leg and arm are missing and shows better or similar results when any two edges are absent.

Method	MPJPE
MöbiusGCN	60.0
PerturbPE w. Perturbed Eigenvector w. 0-edge missed	54.0

Table 6: Assessment of PerturbPE positional encoding impact on Human3.6M [16] dataset performance with two edges removed, utilizing Protocol #1 with given 2D ground truth inputs. Lower is better.

Time Complexity The computational complexity of the RSPT algorithm is $\mathcal{O}(n^3)$. Nevertheless, this level of complexity becomes acceptable for our purposes by taking into account the number of nodes present in a human skeleton and limiting ourselves to one order of perturbation (MöbiusGCN has an inference time of 0.009 seconds per sample, while our method takes 0.010 seconds, demonstrating similar performance).

Occ. Body Parts	Ours	GFPose [9]
2 Legs	**52.4**	53.5
2 Arms	**58.6**	60.0
Left Leg + Left Arm	**48.8**	54.6
Right Leg + Right Arm	**44.6**	53.1
PerturbPE(Any Two Edges Missed)	**54.0**	

Table 7: Recover 3D pose from partial 2D observation: We train one model for two random missing edges with a masking strategy on Human3.6M dataset [16] under Protocol #1 (given 2D GT inputs). Lower is better.

6 Conclusions

In this paper, we introduced the PerturbPE technique, a novel positional encoding method that leverages the Rayleigh-Schrodinger Perturbation Theorem (RSPT) to compute perturbed eigenvectors. This technique enables the extraction of consistent and regular components from the eigenbasis in cases where the input graph has missing edges, thereby enhancing the model's robustness and generalizability. Our empirical evidence strongly supports our theoretical claims. Notably, we witnessed an improved performance of up to 12% on the Human3.6M dataset when occlusion led to the absence of an edge. The performance improvement was even more significant in scenarios where two edges were missing, setting a new state-of-the-art benchmark. While the initial results are promising, potential future work could involve refining the PerturbPE technique, investigating other potential applications of the RSPT in graph-structured data analysis, or exploring the scalability of our proposed method for larger datasets. Ultimately, this paper presents a novel encoding approach that boosts the capabilities of Graph Convolutional Networks (GCNs) in handling missing edges in the input graph-structured data, marking a significant stride in the field of 3D human pose estimation.

Bibliography

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *CVPR*, 2014. 9

[2] Niloofar Azizi, Horst Possegger, Emanuele Rodolà, and Horst Bischof. 3D Human Pose Estimation Using Möbius Graph Convolutional Networks. In *ECCV*, 2022. 2, 3, 4, 7, 8, 9, 10, 11, 12

[3] Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M. Bronstein, and Haggai Maron. Equivariant Subgraph Aggregation Networks. In *ICLR*, 2022. 2, 4

[4] Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein. Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks. In *ICML*, 2021. 2

[5] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral Networks and Locally Connected Networks on Graphs. In *ICLR*, 2014. 7

[6] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded Pyramid Network for Multi-Person Pose Estimation. In *CVPR*, 2018. 10

[7] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded Pyramid Network for Multi-person Pose Estimation. In *CVPR*, 2018. 9

[8] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing Network Structure for 3d Human Pose Estimation. In *ICCV*, 2019. 11

[9] Hai Ci, Mingdong Wu, Wentao Zhu, Xiaoxuan Ma, Hao Dong, Fangwei Zhong, and Yizhou Wang. GFPose: Learning 3D Human Pose Prior With Gradient Fields. In *CVPR*, 2023. 13, 14

[10] Vijay Prakash Dwivedi and Xavier Bresson. A Generalization of Transformer Networks to Graphs. *AAAI Workshop*, 2020. 2, 4, 8, 10, 13

[11] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning Pose Grammar to Encode Human Body Configuration for 3d Pose Estimation. In *AAAI*, 2018. 10

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural Message Passing for Quantum Chemistry. In *ICML*, 2017. 3

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NeurIPS*, 2017. 3

[14] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An Image-based Virtual Try-on Network. In *CVPR*, 2018. 1

[15] Mir Rayat Imtiaz Hossain and James J Little. Exploiting Temporal Information for 3d Human Pose Estimation. In *ECCV*, 2018. 10

[16] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE TPAMI*, 36(7):1325–1339, 2014. 3, 8, 9, 10, 12, 13, 14

[17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015. 9

[18] Ken Kreutz-Delgado. The Complex Gradient Operator and the CR-calculus. *arXiv preprint arXiv:0906.4835*, 2009. 9

[19] Chen Li and Gim Hee Lee. Generating Multiple Hypotheses for 3d Human Pose Estimation with Mixture Density Network. In *CVPR*, 2019. 3

[20] Pan Li and Jure Leskovec. The Expressive Power of Graph Neural Networks. *Graph Neural Networks: Foundations, Frontiers, and Applications*, 2022. 4

[21] Derek Lim, Joshua David Robinson, Lingxiao Zhao, Tess Smidt, Suvrit Sra, Haggai Maron, and Stefanie Jegelka. Sign and Basis Invariant Networks for Spectral Graph Representation Learning. In *ICLR*, 2023. 2, 4

[22] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A Comprehensive Study of Weight Sharing in Graph Networks for 3d Human Pose Estimation. In *ECCV*, 2020. 3, 9, 10, 12

[23] Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H Eugene Stanley. Toward Link Predictability of Complex Networks. *National Academy of Sciences*, 2015. 2

[24] Chenxu Luo, Xiao Chu, and Alan Yuille. A Fully Convolutional Network for 3D Human Pose Estimation. In *BMVC*, 2018. 9, 11

[25] Dingli Luo, Songlin Du, and Takeshi Ikenaga. Multi-task Neural Network with Physical Constraint for Real-time Multi-person 3D Pose Estimation from Monocular Camera. *Multimed. Tools. Appl.*, 80:27223–27244, 2021. 3

[26] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d Pose Estimation and Action Recognition Using Multitask Deep Learning. In *CVPR*, 2018. 1

[27] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Hai Ci, and Yizhou Wang. Context Modeling in 3D Human Pose Estimation: A Unified Perspective. In *CVPR*, 2021. 3

[28] Haggai Maron, Heli Ben-Hamu, Nadav Shamir, and Yaron Lipman. Invariant and Equivariant Graph Networks. In *ICLR*, 2019. 2, 4

[29] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A Simple Yet Effective Baseline for 3d Human Pose Estimation. In *ICCV*, 2017. 8, 9, 10, 11

[30] Brian J McCartin. *Rayleigh-Schrödinger Perturbation Theory: Pseudoinverse Formulation*. Springer, 2009. 5

[31] 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 *3DV*, 2017. 9, 11

[32] Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks. In *AAAI*, 2019. 2, 4

[33] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. ICML*, 2010. 7

[34] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked Hourglass Networks for Human Pose Estimation. In *ECCV*, 2016. 9, 10, 11

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and

Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019. 9, 10

[36] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine Volumetric Prediction for Single-image 3D Human Pose. In *CVPR*, 2017. 10

[37] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal Depth Supervision for 3d Human Pose Estimation. In *CVPR*, 2018. 8, 10

[38] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d Human Pose Estimation in Video with Temporal Convolutions and Semi-supervised Training. In *CVPR*, 2019. 9, 10

[39] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly Optimize Data Augmentation and Network Training: Adversarial Data Augmentation in Human Pose Estimation. In *CVPR*, 2018. 1

[40] Georg Poier, David Schinagl, and Horst Bischof. Learning Pose Specific Representations by Predicting Different Views. In *CVPR*, 2018. 10

[41] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d Human Pose from 2d Image Landmarks. In *ECCV*, 2012. 3

[42] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning Monocular 3d Human Pose Estimation from Multi-view Images. In *CVPR*, 2018. 1

[43] Ryoma Sato, Makoto Yamada, and Hisashi Kashima. Random Features Strengthen Graph Neural Networks. In *SDM*, 2021. 4

[44] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d Human Pose Estimation by Generation and Ordinal Ranking. In *ICCV*, 2019. 8, 10

[45] Matthew Shere, Hansung Kim, and Adrian Hilton. Temporally Consistent 3D Human Pose Estimation Using Dual 360deg Cameras. In *ICCV*, 2021. 1

[46] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The Emerging Field of Signal Processing on Graphs: Extending High-dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Process. Mag.*, 30(3):83–98, 2013. 7

[47] Cristian Sminchisescu. 3D Human Motion Analysis in Monocular Video Techniques and Challenges. In *AVSS*, 2006. 3

[48] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional Human Pose Regression. In *ICCV*, 2017. 8, 9, 10

[49] Bugra Tekin, Pablo Marquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *ICCV*, 2017. 8, 9, 10

[50] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018. 12

[51] Boris Weisfeiler and Andrei Leman. The Reduction of a Graph to Canonical Form and the Algebra which Appears Therein. *nti, Series*, 1968. 2, 4

[52] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? In *ICLR*, 2019. 4

[53] Tianhan Xu and Wataru Takano. Graph Stacked Hourglass Networks for 3D Human Pose Estimation. In *CVPR*, 2021. 3, 8, 9, 10, 11, 12

[54] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial Temporal Graph Convolutional Networks for Skeleton-based Action Recognition. In *AAAI*, 2018. 12

[55] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d Human Pose Estimation in the Wild by Adversarial Learning. In *CVPR*, 2018. 10, 11

[56] Muhan Zhang and Yixin Chen. Link Prediction based on Graph Neural Networks. In *NeurIPS*, 2018. 4

[57] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *CVPR*, 2019. 3, 8, 9, 10, 11, 12

[58] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3D human pose estimation. In *ICCV*, 2019. 11

[59] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular Human Motion Capture using a cnn Coupled with a Geometric Prior. *IEEE TPAMI*, 41(4): 901–914, 2018. 10

[60] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D Human Pose Estimation in the Wild: a Weakly-supervised Approach. In *ICCV*, 2017. 10, 11