J. Vis. Commun. Image R. 118 (2026) 104829

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Full Length Article

ADGT: Enhancing 3D human pose estimation with attention-driven graph-transformers☆

Shuo Yang [a,d,*], Anh Tuan Luu [b], Xuan Son Nguyen [a], Aymeric Histace [a], Bart Jansen [c,d], Hichem Sahli [c,d]

[a] ETIS Laboratory UMR 8051, CY Cergy Paris University, ENSEA, CNRS, France
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[c] Interuniversity Micro-Electronics Center (IMEC), 3001 Leuven, Belgium
[d] Vrije Universiteit Brussel (VUB), Electronics and Informatics Department (ETRO), 1050 Brussels, Belgium

ARTICLE INFO

Keywords:
3D HPE
Graph convolution
Transformer

ABSTRACT

2D-to-3D lifting is a fundamental approach in 3D human pose estimation (3DHPE). This task is crucial in applications, including motion analysis and virtual reality. While Graph Convolutional Networks (GCNs) have demonstrated effectiveness in capturing spatial relationships in human skeletons, they suffer from over-smoothing and limited receptive fields. Transformer-based models provide global context but struggle with local feature extraction and computational efficiency. To address these challenges, we propose ADGT, a novel parallel GCN-transformer architecture combining the strengths of both approaches. Our method introduces three key innovations: Hop-Wise Scalable Adaptive GCN to refine local feature extraction, Attention-Based Local Feature Extractor to enhance the integration of local and global representations, and Register-Based Transformer Enhancement to improve feature separation. Extensive experiments on Human3.6M and MPI-INF-3DHP datasets demonstrate ADGT achieves state-of-the-art performance among frame-based methods while maintaining computational efficiency. These results highlight the potential of ADGT for real-time applications requiring accurate and efficient 3DHPE. The code is available at https://github.com/sYANGunique1111/ADGT.

1. Introduction

3D human pose estimation (HPE) is a fundamental task in computer vision that aims to reconstruct 3D joint positions from 2D input data such as images or videos. Accurate pose estimation is crucial in various applications, including action recognition [1–4], motion capture, virtual reality, and human–computer interaction. While early approaches relied on conventional deep learning models, recent advancements have demonstrated the benefits of leveraging graph-based methods and Transformer-based architectures to enhance spatial and temporal feature extraction.

A key challenge in 3D HPE is the 2D-to-3D lifting problem, where detected 2D joint locations are mapped into a 3D space. Existing methods can be broadly categorized into frame-based approaches and video-based approaches. Frame-based approaches [5–9] process one frame at a time and aim to estimate 3D poses by exploring spatial features of 2D poses. In contrast, video-based approaches [10–13] analyze sequences of consecutive frames, incorporating both spatial and temporal dependencies for improved depth estimation. While video-based methods generally achieve better accuracy, they require larger models and higher computational resources, which can limit their feasibility in real-time applications.

GCNs have been widely adopted in frame-based HPE due to their ability to model human skeletal structures as graph-structured data. These methods aggregate spatial features from neighboring joints, enabling effective pose estimation. However, deep GCNs often suffer from over-smoothing [14] and over-squashing [15], leading to feature indistinguishability in deeper layers. Kreuzer et al. [16] point out that the inherent limitations, e.g., over-smoothing and over-squashing, of deep GCNs could be alleviated by allowing graph nodes to attend global attention. Meanwhile, recent methods [13,17–19] in pose estimation attempt to integrate Transformers to offer global receptive fields and dynamic attention mechanisms. Nevertheless, naive combinations of GCNs and Transformers risk receptive field mismatches and ineffective feature fusion, limiting performance gains.

☆ This paper has been recommended for acceptance by Junsong Yuan.

* Corresponding author.
E-mail addresses: shuo.yang@ensea.fr (S. Yang), anhtuan.luu@ntu.edu.sg (A.T. Luu), xuan-son.nguyen@ensea.fr (X.S. Nguyen), aymeric.histace@ensea.fr (A. Histace), Bart.Jansen@vub.be (B. Jansen), Hichem.Sahli@vub.be (H. Sahli).

https://doi.org/10.1016/j.jvcir.2026.104829
Received 10 March 2025; Received in revised form 12 August 2025; Accepted 26 April 2026
Available online 28 April 2026

As shown in Fig. 1, existing Graph-Transformer architectures can be classified into sequential and parallel architectures. In sequential architectures, such as those proposed by Zheng et al. [13], GCN layers extract spatial features first, which are then refined by Transformer layers to enhance global relations. However, this approach can introduce feature misalignment due to differences in receptive fields. Parallel architectures, such as the work by Kreuzer et al. [16], attempt to mitigate over-smoothing and over-squashing issues by allowing GCNs and Transformers to operate simultaneously on the same hidden states. Despite their advantages, the use of parallel architectures in HPE has been underexplored, motivating our proposed solution.

To address key limitations such as receptive field mismatch in sequential architectures, over-smoothing and over-squashing in deep GCNs, and inefficient fusion of local and global features, we propose ADGT. Inspired by [16], ADGT adopts a parallel GCN-transformer architecture to ensure human joints participate simultaneously in local and global feature extraction. The global features are then combined with local features to alleviate the aforementioned challenges in deep GCNs. Unlike sequential architectures, such a parallel structure ensures that GCNs and Transformers operate on shared hidden states, preventing feature degradation due to receptive field disparities. However, naïvely combining these embeddings through summation or concatenation can lead to misaligned receptive fields. To overcome this, we introduce an attention-driven interaction mechanism (ALFE) that dynamically refines global embeddings by selectively integrating relevant local features, leading to a more effective and interpretable fusion of spatial and contextual information. Key components of our approach include:

- **Hop-Wise Scalable GCN (HS-GCN)**: A mechanism to extract multi-hop local features with adaptive scaling.
- **Attention-Based Local Feature Extractor (ALFE)**: A module that dynamically refines global embeddings by selectively incorporating relevant local features.
- **Register-Based Enhancement for Transformer (RET)**: Inspired by memory tokens [20–22], we integrate register tokens within the Transformer to facilitate the separation of global and local feature representations, improving accuracy on high-variance joints such as hands and feet. To the best of our knowledge, our approach is the first to introduce registers in 3D HPE to predict more accurate joint locations.

Extensive evaluations on the Human3.6M [23] and MPI-INF-3DHP [24] datasets demonstrate that our approach achieves state-of-the-art performance while maintaining computational efficiency. By effectively integrating graph-based local feature learning with attention-driven global reasoning, ADGT provides a promising direction for advancing 3D human pose estimation.

2. Related works

3D human pose estimation (HPE) has seen significant advancements, primarily driven by two key methodological paradigms: GCN-based
Approaches [12,25–27] and Graph-Transformer-based Approaches [13,17–19]. While graph convolutional networks (GCNs) leverage the natural skeletal structure of the human body, Transformers have demonstrated the ability to capture long-range dependencies. The combination of these methods has motivated various architectural innovations, yet challenges persist in balancing receptive fields, mitigating over-smoothing, and ensuring computational efficiency. Moreover, 3D HPE serves as a critical foundation for a wide range of skeleton-based tasks. For instance, Cao et al. [28] propose a Hierarchical Spatial-Temporal Masked Contrast (HSTM) framework for unsupervised skeleton-based action recognition. On top of attention-based Deep Convolutional GRU (DC-GRU) [29], Dey et al. [30] extend this approach with an Attention-driven Residual DC-GRU model tailored for workout action recognition.

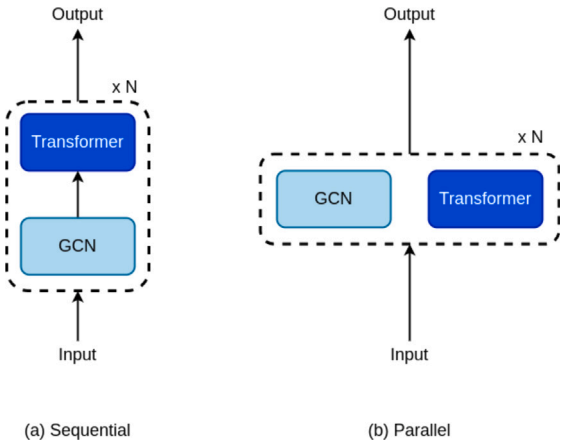

Fig. 1. Two types of graph-transformer architectures.

2.1. GCN-based approaches

Graph convolutional networks (GCNs) have been widely employed in HPE due to their ability to process structured skeleton data. They operate by aggregating spatial information from neighboring joints, making them effective for frame-based pose estimation. Early works, such as Kipf and Welling [31], established the foundation for graph convolutions, inspiring later applications in pose estimation [9,19,25,26].

Zhao et al. [12] introduced Semantic Graph Convolution Networks (SemGCN), which leverage learnable adjacency matrices to capture semantic relationships between joints. Similarly, Ci et al. [26] proposed a locally connected network that integrates a structure matrix and a weight matrix to enhance joint relationships. Zhou et al. [32] extended this by introducing high-order adjacency matrices, enabling a more refined extraction of multi-hop dependencies. Xie et al. [33] introduce HOGFormer, a novel architecture for 3D HPE that combines high-order GCNs and transformers. Hong et al. [34] present a Stacked Capsule Graph Convolutional Network for 3D HPE, aiming to exploit better the hierarchical and part-aware structure of the human body.

Despite their effectiveness, GCNs face critical challenges such as over-smoothing, where node features become indistinguishable as layers deepen [14]; over-squashing, where global context is compressed in deep networks [15]. These issues have led to hybrid approaches integrating global attention mechanisms to complement GCNs. Our proposed ADGT model addresses these limitations by incorporating a Hop-wise Scalable GCN (HS-GCN), which dynamically refines local features at different hop levels, effectively mitigating over-smoothing while enhancing local feature representations. Unlike [32], which employs a high-order convolution operator combined with an adaptive filtering module to adjust convolution weights based on local graph structures dynamically, our proposed HS-GCN introduces an additional learnable scaling mechanism that adaptively modulates the influence of different hop distances, ensuring a more flexible and context-aware feature propagation. This allows HS-GCN to selectively amplify or suppress multi-hop dependencies based on the structural and feature-based significance of each joint, leading to improved representation learning for human pose estimation.

2.2. Graph-transformer-based approaches

To mitigate the limitations of GCNs, recent efforts have integrated Transformer architectures, benefiting from their global receptive fields and dynamic attention mechanisms. The Transformer model, originally introduced by Vaswani et al. [35], has been adapted for pose estimation in several ways [13,17–19].

Zhu et al. [36] developed PoseGTAC, which alternates between graph convolutions and Transformer layers to balance local and global feature extraction. Chen et al. [17] proposed DGFormer, which dynamically updates adjacency matrices using a Transformer-based feature refinement process. Meanwhile, Mehraban et al. [18] designed MotionAGFormer, a parallel framework incorporating GCNs and Transformers to model spatial and temporal pose dynamics. Observing the limited capacity of transformers to process channel information as the number of attention heads increases, Huang et al. [37] propose a Channel MLP module, which significantly improves the performance of TokenPose [38].

The challenge in combining these architectures lies in their receptive field mismatch: GCNs emphasize local relationships, whereas Transformers capture global dependencies. Simple sequential arrangements [11,13] can lead to ineffective fusion, while naive parallelization [16] may not fully exploit cross-modal information. Moreover, introducing attention mechanisms must be carefully structured to avoid redundant computations and over-parameterization. Unlike existing methods, our ADGT model leverages a parallel fusion strategy with an attention-based local feature extractor (ALFE). This ensures that local embeddings dynamically enhance global representations, leading to more accurate 3D pose estimations.

2.3. Memory tokens in transformer architectures

Memory tokens, also called register tokens, have recently been introduced as a mechanism to store intermediate representations within Transformer models, enhancing their ability to process structured and sequential data [20–22]. Originally proposed for NLP tasks, memory tokens have found applications in computer vision and 3D pose estimation by enabling Transformers to retain essential information while reducing redundant computation.

Darcet et al. [21] demonstrated that memory tokens can improve vision Transformers by acting as a persistent memory unit that stores critical information across layers. Rao et al. [22] further explored token sparsification strategies, ensuring that specific tokens are dedicated to capturing local or global dependencies separately. In the context of HPE, these memory tokens can alleviate information loss in deep Transformer stacks by storing excess global context and allowing other tokens to focus on refining local feature representations.

By leveraging memory tokens, our approach enables Register-Based Enhancement for Transformers, ensuring that global embeddings are appropriately distributed while allowing finer local feature extraction. This strategy particularly benefits high-variance joint locations such as hands and feet, where detailed pose estimation is critical.

2.4. Bridging the gap: Our approach

To address the above-discussed challenges, we propose ADGT, a novel parallel GCNs-Transformers framework for 2D-to-3D lifting that balances the advantages of both architectures:

- **Parallel GCN-transformer Fusion**: Unlike sequential designs [46], our method processes spatial embeddings simultaneously in both GCN and Transformer branches.
- **Hop-Wise Scalable GCN (HS-GCN)**: Extending the idea of high-order adjacency matrices [32], we introduce adaptive scaling factors to refine local features at multiple hop levels dynamically.
- **Attention-Based Local Feature Extractor (ALFE)**: Instead of directly combining GCN and Transformer outputs, we introduce a query-key mechanism that enables global embeddings to selectively extract useful local features, inspired by the query-key paradigm in Transformers [35].
- **Register-Based Enhancement for Transformers**: Inspired by memory tokens [20–22], we incorporate registers to store excessive global information, allowing other Transformer tokens to focus on extracting refined local details. This approach enhances high-variance joint predictions (e.g., hands, feet), which are often misestimated in prior works.

3. Methodology

3.1. Preliminary

Given 2D joint detections $\mathbf{X}_{2D} \in \mathbb{R}^{N\times 2}$, where N is the number of joints, the objective is to predict the corresponding 3D joint positions $\mathbf{X}_{3D} \in \mathbb{R}^{N\times 3}$. This problem is framed as a 2D-to-3D lifting task, where spatial relationships between 2D joint locations must be leveraged to infer their depth and 3D structure.

To achieve this, we propose ADGT, a parallel GCN-transformer architecture that simultaneously processes local and global representations. As illustrated in Fig. 2, our framework consists of three core components:

- Hop-wise Scalable GCN (HS-GCN): A graph-based module that extracts multi-hop local features with adaptive scaling.
- Register-Based Transformer Enhancement (RET): A Transformer-based mechanism incorporating register tokens to improve the separation of global and local representations, enhancing accuracy in high-variance joints.
- Attention-Based Local Feature Extractor (ALFE): A module that dynamically integrates local and global embeddings, allowing global features to selectively refine themselves using relevant local information.

Both HS-GCN and RET operate in parallel on shared hidden states, ensuring efficient feature extraction while preventing receptive field mismatches that typically arise in sequential architectures. The ALFE module further refines this fusion by dynamically attending to local features, improving the expressiveness of global embeddings. Through this parallel processing and adaptive feature integration, ADGT achieves more accurate and robust 3D human pose estimation, as demonstrated in subsequent sections.

3.2. ADGT

While [13] alternatively combines GCNs and Transformers, which could lead to unmatched receptive fields. As shown in Fig. 2, our proposal adopts a parallel architecture to alleviate the receptive-fields problem and dynamically aggregate derived from the two mechanisms.

Given 2D detections $\mathbf{X}_{2D} \in \mathbb{R}^{N\times 2}$, we first map them to C-dimensional hidden states $H_{in} \in \mathbb{R}^{N\times C}$, with linear projection $f_{embed}(.) : 2 \to C$:

$$H_{in} = f_{embed}(\mathbf{X}_{2D}), H_{in} \in \mathbb{R}^{N\times C}, \tag{1}$$

The H_{in} embeddings are input to ADGT layers. In each ADGT layer, the hidden states are simultaneously forwarded to the Hop-wise Scalable GCN (HS-GCN) and the Register Enhanced Transformers (RET) to compute hop-wise local embeddings $H_{hs} \in \mathbb{R}^{N\times C}$ and register-enhanced global embeddings $H_{ret} \in \mathbb{R}^{N\times C}$. Given $H_{in}^{\ell} \in \mathbb{R}^{N\times C}$ as input to ℓ-th ADGT layer, the local and global embeddings at ℓ-th layer are generated as:

$$H_{hs}^{\ell} = HS\text{-}GCN^{\ell}(H_{in}^{\ell}), \tag{2}$$

$$H_{ret}^{\ell} = RET^{\ell}(H_{in}^{\ell}), \tag{3}$$

where $\ell \in \{0, 1, \ldots, L\}$ stands for layer indices. As the two embeddings have different receptive fields, a simple combination might comprise excessive information. To mitigate the impact of receptive-field differences, we consider local embeddings as auxiliaries to provide additional local information and introduce the ALFE module that enables global embeddings H_{ret} to selectively extract local embeddings from H_{hs}. The extracted local embeddings at ℓ-th are conducted following:

$$H_{ex}^{\ell} = ALFE^{\ell}(H_{hs}^{\ell}, H_{ret}^{\ell}), H_{ex} \in \mathbb{R}^{N\times C}. \tag{4}$$

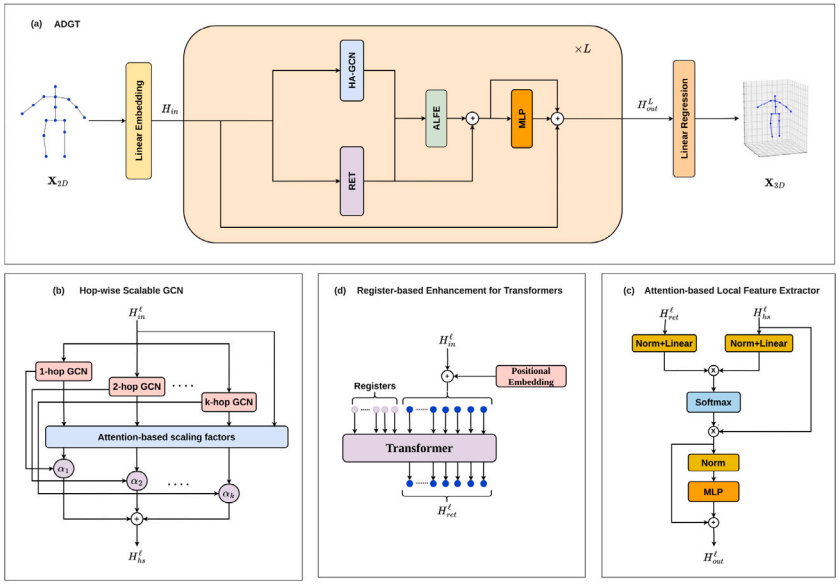

Fig. 2. The architecture of the ADGT model. (a) is the workflow of our method. (b) is the architecture of HS-GCN module. (c) illustrates how we implement registers in transformer architectures. (d) is the ALFE module, where we extract useful local information for global embeddings.

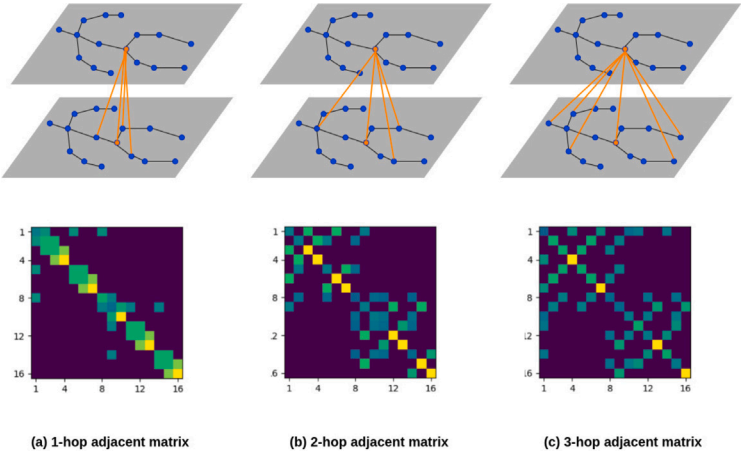

Fig. 3. Hop-wise adjacent matrices. The top part illustrates how node features are aggregated with different adjacent matrices. The bottom part illustrates normalized adjacent matrices at hops of 1,2, and 3.

To effectively integrate local and global embeddings, we concatenate them and pass the combined representation through a Multi-Layer Perceptron (MLP), enabling comprehensive feature aggregation:

$$H_{com}^{\ell} = MLP^{\ell}(Concat(H_{ex}^{\ell}, H_{ret}^{\ell})), H_{com}^{\ell} \in \mathbb{R}^{N \times C}. \tag{5}$$

To ensure stable feature representations during forward propagation, we incorporate a skip connection in each ADGT layer, updating the joint features at layer $\ell + 1$ as follows:

$$H_{in}^{\ell+1} = H_{com}^{\ell} + H_{in}^{\ell}. \tag{6}$$

To predict 3D joint locations, we employ a linear regressor, $f_{regress}(.) : C \rightarrow 3$, that directly maps the final hidden states H_{out}^{L} to 3D coordinates, $\mathbf{X}_{3D}$, eliminating the need for additional graph filters:

$$\mathbf{X}_{3D} = f_{regress}(H_{out}^{L}). \tag{7}$$

The implementation is designed with the assumption that the output hidden states H_{out}^{L} represent locally enhanced global features. Each joint token vector is deemed sufficiently expressive for accurate 3D joint location regression, eliminating the need for additional information extraction from neighboring joints during regression.

3.3. Hop-wise scalable GCN (HS-GCN)

To effectively complement Transformer-based global feature extraction, the GCN module must capture comprehensive local features. While Graph Attention Networks (GAT) [39] achieve this by constructing dynamic node connections with global attention, their functionality overlaps significantly with Transformers, making them less suitable as auxiliary modules. An alternative approach, as proposed in [32], employs high-order adjacency matrices A^n to capture multi-hop dependencies, enhancing local feature extraction. However, expanding the receptive field through higher-order neighbors can lead to excessive information aggregation at central joints, potentially degrading feature distinctiveness.

To address this, we introduce Hop-wise Scalable GCN (HS-GCN), which performs graph convolutions at multiple hop levels while dynamically adjusting feature importance. Instead of relying on predefined adjacency matrices, our method generates distinct hop-wise adjacency matrices that explicitly define connections between joints and their k-hop neighbors, see Fig. 3. Furthermore, we introduce learnable scaling factors that adaptively modulate the influence of different hop distances

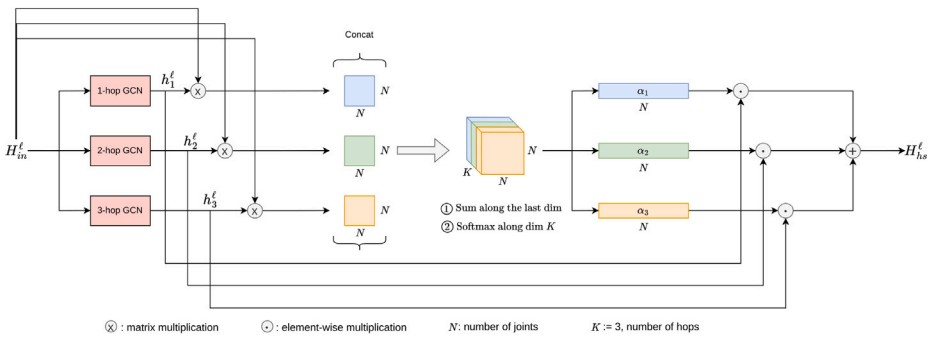

Fig. 4. An illustration of HS-GCN with three GCN layers. The scales of the GCN layers are dynamically controlled through attention between the input features and the embeddings learned within the GCN modules. The scaling weights $(\alpha_1, \alpha_2, \alpha_3)$ are applied in a joint-wise manner, enabling fine-grained modulation of features across different joints.

based on the input features, ensuring a more flexible and context-aware local feature representation. One example of HS-GCN pipeline is presented in Fig. 4. Given k for hop indices, and H_{in} as input hidden states, the k-hop graph convolution at ℓ-th layer is defined as:

$$h_k^\ell = \tilde{D}_k^{-\frac{1}{2}} \tilde{A}_k \tilde{D}_k^{-\frac{1}{2}} H_{in}^\ell W_k^\ell,\ h_k^\ell \in \mathbb{R}^{N\times C} \tag{8}$$

$$H_{hop}^\ell = Concat(h_1^\ell, h_2^\ell, \ldots, h_k^\ell),\ H_{hop}^\ell \in \mathbb{R}^{K\times N\times C} \tag{9}$$

where $\tilde{A}_k \in \mathbb{R}^{N\times N}$ is k-hop adjacent matrix; $\tilde{D}_k \in \mathbb{R}^{N\times N}$ is k-hop degree matrix; $W_k^\ell \in \mathbb{R}^{N\times C}$ is weight matrix for kth hop-wise graph convolution. To combine different hop features, we introduce scaling factors $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, where $\boldsymbol{\alpha} \in \mathbb{R}^{K\times N}$ and $\alpha_i \in \mathbb{R}^N$ to modulate H_{hop}. To derive scaling factors at ℓ-th layer, $\boldsymbol{\alpha}^\ell$, we first compute attention maps S^ℓ with H_{hop}^ℓ and H_{in}^ℓ:

$$\mathrm{S}^\ell = H_{hop}^\ell {H^l}_{in}^T, \tag{10}$$

where $\mathrm{S}^\ell \in \mathbb{R}^{K\times N\times N}$ contains K attention maps. To compute scaling factors $\boldsymbol{\alpha}^\ell$ from the attention maps S^ℓ, we first reduce the last dimension of S^ℓ with summation, then apply a $Softmax(.)$ function to normalize the summed S^ℓ along K dimension:

$$\boldsymbol{\alpha}^\ell = Softmax_K\left(\sum_n \mathrm{S}_{:,:,n}^\ell\right). \tag{11}$$

The hop features H_{hop}^ℓ are scaled using an element-wise product with $\boldsymbol{\alpha}^\ell$:

$$H_{scale}^\ell = \boldsymbol{\alpha}^\ell \odot H_{hop}^\ell,\ H_{scale}^\ell \in \mathbb{R}^{K\times N\times C}. \tag{12}$$

Last, H_{hs}^ℓ is deduced with summation of H_{scale}^ℓ along K dimension.

$$H_{hs}^\ell = \sum_k H_{scale,k,:,:}^\ell,\ H_{hs}^\ell \in \mathbb{R}^{N\times C}. \tag{13}$$

The method is able to dynamically tune importance node features at different hop levels with interaction between local and input embeddings.

To enhance local–global interaction, we attempt another method to derive scaling factors from attention between H_{hop} and H_{ret}. This variant is denoted as HGS-GCN. The architectures of HGS-GCN and HS-GCN are shown in Fig. 5. Experiments to compare and discuss the two methods are presented in an ablation study.

3.4. Register-based enhancement for transformers

Registers are extra tokens for transformer architectures to store specific information. The learned features by register tokens are not forwarded to the next layer. This idea was first proposed by Burtsev et al. [20] as memory tokens to store excessive information, Darcet et al. [21] further illustrated the functionality of register tokens in capturing low-informative features for Vision Transformers [40,41] in regressive tasks, such as image classification. Inspired by the two prior works, we attempt to deploy registers as additional tokens to register excessive global information, thus other tokens would focus on more local information. Our motivation for introducing registers in transformer architectures is to alleviate transformer tokens extracting excessive global information, which might impact prediction accuracy for 3D joint locations. Ablation studies are launched to study the information stored in register tokens and to evaluate their effectiveness.

Specifically, our register tokens are randomly initialized following a standard normal distribution. These tokens are learnable and are concatenated with the input tokens, participating in all operations within attention mechanisms (e.g., generating representations, and conducting self-attention [35]). The difference from input tokens is that the register tokens are not the output of transformer architectures. Given input hidden states at ℓ-th layer as H_{in}^ℓ, we first concatenate H_{in}^ℓ with M randomly generated register tokens, denoted as $H_{reg}^\ell \in \mathbb{R}^{M\times C}$:

$$H_{in+reg}^\ell = [H_{in}^\ell; H_{reg}^\ell] \in \mathbb{R}^{(N+M)\times C}. \tag{14}$$

Instead of directly forwarding the combined H_{in+reg}^ℓ to Transformers, we first project the H_{in+reg}^ℓ to generate the 3 representations Query $Q^\ell \in \mathbb{R}^{N\times D}$, Key $K^\ell \in \mathbb{R}^{N\times D}$, Value $V^\ell \in \mathbb{R}^{N\times C}$ with the matrices $W_q^\ell \in \mathbb{R}^{C\times D}, W_k^\ell \in \mathbb{R}^{C\times D}, W_v^\ell \in \mathbb{R}^{C\times C}$. The representations are input into Transformers to conduct an attention operation:

$$H_{attn}^\ell = Attention(Q^\ell, K^\ell, V^\ell),\ H_{attn}^\ell \in \mathbb{R}^{(N+M)\times C}. \tag{15}$$

Since the H_{attn}^ℓ is derived from H_{in+reg}^ℓ, it contains N input tokens and M register tokens. As the register tokens are introduced only to store global information, it is not necessary to forward them to the next layer. Therefore, we select the first N tokens from H_{attn}^ℓ to output, and discard the remaining M register tokens. The output token embeddings from the RET module at ℓ-th layer are denoted as H_{ret}^ℓ.

3.5. Attention-based local feature extractor (ALFE)

Existing approaches often combine GCN and Transformer embeddings through simple summation or concatenation [16], which may lead to suboptimal fusion due to mismatched receptive fields. To address this, we introduce the Attention-based Local Feature Extractor (ALFE), which enables global embeddings to selectively extract relevant local features through a query-key attention mechanism, enhancing the integration of spatial and contextual information.

Specifically, the ALFE module first projects H_{hs} and H_{ret} to key-query pairs, then computes their similarities. the matrix is passed to a $Softmax(.)$ function to generate weights for aggregating information from H_{hs}. Denoting the extracted local information by H_{lf}, the detailed extraction process in ℓ-th layer is computed as:

$$H_{lf}^\ell = Softmax\left(\frac{H_{ret}^\ell (H_{hs}^\ell)^T}{\sqrt{d}}\right) H_{hs}^\ell, \tag{16}$$

where d refers to the number of channels in H_{hs} and H_{ret}. The aggregated features H_{lf} are forwarded into an MLP layer with a residual connection. Function $LayerNorm(.)$ is employed before the MLP layer to stabilize the input data as:

$$H_{ex}^{\ell} = MLP(LayerNorm(H_{lf}^{\ell})) + H_{lf}^{\ell}. \tag{17}$$

The extracted local features H_{ex}^{ℓ} will be compensated to H_{ret} to enhance global embeddings.

3.6. Loss function

Our loss function aims to minimize the errors between the predicted and ground-truth poses:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\left\|J_i^{3D} - \hat{J}_i^{3D}\right\|_2\right), \tag{18}$$

where J_i^{3D} and $\hat{J}_i^{3D}$ are ground-truth and estimated 3D joint positions of the ith joint respectively, N indicates the number of joints in a mini-batch.

4. Experiments

We first give experimental settings and implementation details of ADGT model in Section 4.1. Our experimental results and comparisons with state-of-the-art methods are reported in . Finally, some ablation studies are presented in Section 4.3.

4.1. Experimental settings

Datasets. We evaluate our method on two datasets, i.e., Human3.6M [23] and MPI-INF-3DHP [24]. The Human3.6M dataset is currently the most commonly accepted indoor-scene dataset for HPE with more than 3.6 million frames. It records 11 subjects with 15 assigned actions and detects human joint positions with 4 cameras at different viewpoints. Following previous work [13], we train our model with subjects of S1, S5, S6, S7, and S8 and evaluate on subjects of S9 and S11. A single pose in Human3.6M comprise 32 joints, we follow previous works [13] to select 16 joints (Hip, Right hip, Right knee, Right ankle, Left hip, Left knee, Left ankle, Spine, Thorax, Head top, Left shoulder, Left elbow, Left wrist, Right shoulder, Right elbow, Right wrist) for pose estimation. MPI-INF-3DHP dataset, on the contrary, contains not only indoor scenes but also complex outdoor scenes. It is widely adopted to assess the generalizability of HPE methods. We use the test set of MPI-INF-3DHP [24] to validate our method. MPI-INF-3DHP contains 17 joints. To ensure consistency with the Human3.6M setting, we omit the "Nose" joint during evaluation.

Evaluations. For the Human3.6M dataset, we apply the standard Mean Per Joint Position Error (MPJPE) [23] and Procrustes-aligned Mean Per Joint Position Error (P-MPJPE) [42] metrics, referred to as Protocol #1 and Protocol #2, respectively. MPJPE computes the direct Euclidean distance between the ground-truth and predicted joints, using the same formulation as in Eq. (18). In contrast, P-MPJPE calculates this error after applying a rigid Procrustes alignment between the predicted and ground-truth poses. We use the Area Under the Curve (AUC) and the Percentage of Correct Keypoints (PCK) metrics for evaluations on the MPI-INF-3D dataset as suggested in [24]. The PCK metric is defined as:

$$\mathrm{PCK}(\tau) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}\left(\left\|J_i^{3D} - \hat{J}_i^{3D}\right\|_2 \le \tau\right), \tag{19}$$

where τ is the threshold to determine valid detections. The AUC metric computes a mean over PCK values at multiple thresholds. Following previous works [11,12], we evaluate on the PCK metric with a threshold of $\tau = 150$ mm and on the AUC metric for a range of PCK thresholds.

Implementation Details. We adopt 5 ADGT layers to train and evaluate, each layer has $C = 96$ dimensions. We adopt $K = 3$ for HS-GCNs and $M = 8$ registers for Transformers. The settings are attained by trial and error. Following the work in [9], we take both ground-truth and detected 2D poses from the cascaded pyramid network (CPN) [25] to train our model. Our model is trained with 100 epochs on both ground-truth and detected 2D poses. No extra datasets or data augmentation are utilized in training. We adopt Adam optimizer [43] with an initial learning rate of 0.001, and a batch size of 256 when training with ground-truth 2D poses. The learning rate decays every 2 epochs with a decaying rate of 0.9. When detected, 2D poses obtained from CPN are used; the model is trained with an initial learning rate of 0.0008 and a decaying rate of 0.95 for every 2 epochs. Our model is developed using the PyTorch framework and trained on one NVIDIA Quadro RTX 8000 GPU.

4.2. Experimental results and comparisons

We compare our method with frame-based approaches [7,12,13,17,26,44–47] on the Human3.6M using detected 2D key points as input and MPJPE as evaluation metric. The results are shown in Table 1. ADGT achieves 50.1 mm on average, which significantly outperforms most of the competing methods and attains 1.5 mm improvement compared to the method on the second rank [17]. We also compare the Procrustes-aligned accuracy (Protocol #2) with other state-of-the-art works on detected 2D poses in Table 2. Our method achieves 39.2 mm on average with 0.8 mm lower than the 2nd best method. Despite high uncertainties from detected 2D key points, our method achieves comparatively higher accuracy than other state-of-the-art methods under the two metrics, which validates the effectiveness and robustness of our model. Even though our method achieves the best performance in terms of action-wise average losses, it does not outperform all other methods on every individual action. This is primarily because certain actions, e.g., "Sitting" and "SittingD". often involve greater self-occlusion and more complex body configurations. Detected 2D poses from these actions typically introduce higher uncertainty. Our method is designed for general-purpose 3D pose estimation rather than being specifically tailored to handle actions with high ambiguity and complexity. As a result, the ADGT model does not achieve the best results across all action categories.

To further evaluate our approach, we benchmark against state-of-the-art video-based methods [18,48–52] under the same experimental settings as the frame-based methods. The results on MPJPE and P-MPJPE are in Tables 1 and 2. This comparison provides insights into how competitive our method is relative to video-based techniques, while also facilitating a discussion on computational efficiency and trade-offs in the ablation study.

4.3. Ablation study

Cross-dataset results on MPI-INF-3DHP. In order to validate the generalization of our network, we train the ADGT on the Human3.6M dataset and evaluate it on the test set of the MPI-INF-3DHP dataset. The results are shown in Table 3, where our method outperforms the second-rank method [54] over the AUC and PCK metrics by margins of 6.1 and 0.8 percent respectively. The competitive performance of our method compared to the others demonstrates its generalizability to unseen scenarios and data distributions.

Comparison of computational efficiency with other frame and video-based methods. Apart from prediction accuracies, we compare our method with state-of-the-art methods on computational cost and model size as well to evaluate its feasibility for engineering implementations; the results are shown in Table 4. Our method achieves the best performance for MPJPE and P-MPJPE metrics among all frame-based methods. Meanwhile, our method requires only 0.9 M parameters and 27.8 M FLOPs for computational cost, which is much lower than

Table 1
Quantitative comparison with other state-of-the-art models on Human3.6M under MPJPE metric (known as Protocol #1), using CPN [25] detected 2D key points as input (results are measured in millimeters). Some methods are evaluated with Stacked Hourglass networks (SH) [27] detected 2D key points marked by (SH). (+) and (†) indicate that the corresponding method uses extra data from MPII [53] and temporal information, respectively. The best results are marked in bold.

Video-based Methods																
Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Mehraban et al. [18] (†)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	38.4
Xu et al. [48] (†)	37.4	43.5	42.7	42.7	46.6	59.7	41.3	45.1	52.7	60.2	45.8	43.1	47.7	33.7	37.1	45.6
Yu et al. [49] (†)	41.3	44.3	40.8	41.8	45.9	54.1	42.1	41.5	57.8	62.9	45.0	42.8	45.9	29.4	29.9	44.4
Zhang et al. [50] (†)	37.6	40.9	37.3	39.7	42.3	49.9	40.1	39.8	51.7	55.0	42.1	39.8	41.0	27.9	27.9	40.9
Zhao et al. [51] (†)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	47.6
Zheng et al. [52] (†)	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Frame-based Methods																
Protocol #1	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Chen et al.[17]	47.5	50.4	47.0	50.8	53.5	60.6	50.2	46.9	**58.5**	66.6	52.2	48.9	54.9	41.4	43.6	51.6
Ci et al. [26] (+)	46.8	52.3	**44.7**	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Liu et al. [44]	46.3	52.2	47.3	50.7	55.5	67.1	49.2	**46.0**	60.4	71.1	51.5	50.1	54.5	40.3	43.7	52.4
Martinez et al. [45] (SH)	51.7	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
Pavlakos et al. [7] (+)	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
Yang et al. [47] (+)	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
Xu et al. [46]	**45.2**	49.9	47.5	50.9	54.9	66.1	48.5	46.3	59.7	71.5	51.4	48.6	53.9	**39.9**	44.1	51.9
Zhao et al. [12]	47.3	60.7	51.4	60.5	61.1	**49.9**	**47.3**	68.1	86.2	**55.0**	67.8	61.0	**42.1**	60.6	45.3	57.6
Zhao et al. [13]	49.3	53.9	54.1	55.0	63.0	69.8	51.1	53.3	69.4	90.0	58.0	55.2	60.3	47.4	50.6	58.7
Ours	46.4	**49.5**	44.9	**48.8**	**50.4**	59.5	50.1	47.2	59.6	64.7	**49.9**	**48.3**	52.4	**39.9**	**41.3**	**50.1**
Ours (GT)	28.8	34.9	28.5	31.4	35.0	40.8	34.7	29.2	35.3	43.1	31.6	33.2	34.6	26.1	28.9	33.1

Table 2
Quantitative comparison with other state-of-the-art models on Human3.6M under P-MPJPE metric (known as Protocol #2), using CPN [25] detected 2D key points as input (results are measured in millimeters). Some methods are evaluated with SH [27] detected 2D key points marked by (SH).(†) indicate that the corresponding method temporal information. The best results are marked in bold.

Video-based Methods																
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Mehraban et al. [18] (†)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	32.6
Xu et al. [48] (†)	31.0	34.8	34.7	34.4	36.2	43.9	31.6	33.5	42.3	49.0	37.1	33.0	39.1	26.9	31.9	36.2
Yu et al. [49] (†)	32.4	35.3	32.6	34.2	35.0	42.1	32.1	31.9	45.5	49.5	36.1	32.4	35.6	23.5	24.7	34.8
Zhang et al. [50] (†)	30.8	33.1	30.3	31.8	33.1	39.1	31.1	30.5	42.5	44.5	34.0	30.8	32.7	22.1	22.9	32.6
Zhao et al. [51] (†)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	37.3
Zheng et al. [52] (†)	32.5	34.8	32.6	34.6	35.3	39.5	32.1	32.0	42.8	48.5	34.8	32.4	35.3	24.5	26.0	34.6
Frame-based Methods																
Protocol #2	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Chen et al.[17]	36.5	39.1	36.4	**40.6**	41.0	45.4	**37.7**	35.7	**46.5**	53.1	**41.5**	**36.9**	42.6	31.9	34.3	40.0
Liu et al. [44]	36.7	41.3	38.8	42.6	43.7	52.1	38.8	39.1	50.0	58.7	42.4	39.5	43.9	32.4	38.5	42.5
Martinez et al. [45] (SH)	44.8	52.0	44.4	50.5	61.7	59.4	45.1	41.9	66.3	77.6	54.0	58.8	49.0	35.9	40.7	52.1
Xu et al. [46]	**35.4**	40.6	38.1	41.9	43.0	51.4	38.1	38.4	49.3	58.4	41.7	38.8	43.2	31.7	37.5	41.8
Zhao et al. [13]	38.6	43.5	41.3	44.7	46.8	54.6	41.3	41.5	51.9	60.7	44.3	41.6	46.1	34.9	40.3	44.8
Ours	35.8	**38.6**	**35.3**	40.8	**38.7**	**42.1**	38.9	**35.8**	46.7	**52.9**	41.6	37.2	**40.8**	**30.5**	**34.2**	**39.2**
Ours (GT)	22.3	28.7	23.4	26.7	27.3	33.3	28.0	23.9	30.1	36.5	26.2	26.6	29.1	21.1	24.3	27.1

Table 3
Quantitative comparison on the test set of MPI-INF-3D dataset with models trained on Human3.6M dataset. The best results are marked in bold.

Methods	AUC ↑ (%)	PCK ↑ (%)
Martinez et al. [45]	17.0	42.5
Ci et al. [26]	36.7	74.0
Zeng et al. [55]	43.8	77.6
Zhao et al. [13]	43.8	79.0
Liu et al. [44]	47.6	79.3
Xu et al. [46]	45.8	80.1
Nie et al. [54]	45.9	83.5
Ours	**52.0**	**84.3**

the second-best model, DGFormer [17], whose model size and computational cost are recorded at 4.3 M and 159.8 M, respectively. The compact parameter size and low computational cost attributes to a lightweight configuration, using a model dimension of 96 and a depth of 5 layers. On the other hand, our method achieves competitive performance compared with video-based methods, as ours attains 50.1 mm against 49.7 mm of PoseFormer [52] (3 frames) on MPJPE. The performance of our method is considerably close to PoseFormer, yet our model size is almost 10 times smaller than theirs, and the computational cost (FLOPs) is just half that of theirs. To evaluate its real-time performance, we measure the Frames Per Second (FPS). Since FPS can vary across runs and hardware configurations, particularly when using GPUs. We report the average FPS over 10 runs on a single NVIDIA Quadro RTX 8000 GPU to ensure fair and reproducible comparison. Under this setting, our model achieves an average FPS of 14,325.

Comparison of aggregation methods. This experiment evaluates the effectiveness of the ALFE module compared to simple aggregation methods such as summation and concatenation of global and local embeddings. The experimental results are given in Table 5, where the embeddings combined with the ALFE module achieve 50.1 mm and 39.2 mm on MPJPE and P-MPJPE metrics, respectively, which are better than the results obtained by the other two combination methods (summation and concatenation). This outstanding performance validates the effectiveness of the proposed ALFE module.

Comparison of modeling branches. This experiment examines the contribution of combining GCN and Transformer components within

Table 4
Quantitative comparison with other frame and video based methods on Human3.6M dataset (in mm). Frames: the number of frames as input. Params: tunable parameters for a model. FLOPs: floating point operations per second. M and G stand for million and billion FLOPs respectively. Others are frame-based methods.

Methods	Frames	Params	FLOPs	MPJPE [23]↓	P-MPJPE [42]↓
PoseFormer [52]†	3	9.5 M	60.6 M	49.7	39.2
PoseFormer [52]†	81	9.6 M	1634 M	47.0	37.0
PoseFormerV2 [51]†	243	18.9 M	78.2 G	40.5	31.8
MixSTE [50]†	243	33.6 M	278.0 G	40.9	32.6
MotionAGF [18]†	243	11.7M	96.6 G	39.5	33.1
Liu et al. [44]	1	2.1 M	0.72 M	52.4	41.4
ModulatedGCN [56]	1	0.3 M	9.7 M	54.7	43.5
Xu et al. [46]	1	3.6 M	1.4 M	51.9	–
Graformer [13]	1	0.7 M	33.0 M	58.7	44.8
DGFormer [17]	1	4.3 M	159.8 M	51.6	40.0
Ours	1	0.9 M	27.8 M	50.1	39.2

† Indicates usage of temporal information.

Table 5
Quantitative comparison of data aggregation approaches. Num-Channel refers to the number of projected dimensions by the skeleton embedding illustrated in Fig. 2. Num-Layer indicates the layers of ADGT modules.

Aggre.	Num-Channel	Num-Layer	MPJPE ↓ (mm)	P-MPJPE ↓ (mm)
Sum.	96	5	50.9	40.4
Concat.	96	5	51.8	41.0
ALFE	96	5	**50.1**	**39.2**

Table 6
Quantitative comparison with only GCN or Transformer branches. Num-Channel refers to the number of projected dimensions by the skeleton embedding illustrated in Fig. 2. Num-Layer indicates the layers of AttnGCN modules.

Method	Num-Channel	Num-Layer	MPJPE [23] ↓ (mm)	P-MPJPE [42] ↓ (mm)
GCN	96	5	62.5	46.4
Transformer	96	5	53.6	40.9
ADGT	96	5	**50.1**	**39.2**

the proposed ADGT architecture. Specifically, we compare the full ADGT model against its individual GCN-only and Transformer-only branches, all configured with 96 channels and 5 layers. As shown in Table 6, ADGT achieves 50.1 mm MPJPE and 39.2 mm P-MPJPE, outperforming both the GCN branch (62.5/46.4) and the Transformer branch (53.6/40.9). These results demonstrate that integrating GCN and Transformer mechanisms leads to more accurate 3D pose estimation by effectively capturing both local and global joint dependencies.

Comparison of GCN modules. This experiment compares our proposed HS-GCN module to state-of-art GCN-based modules [31,39,44,57]. We further discuss scaling factors generated with the two proposed methods: HS-GCN and HGS-GCN. In Table 7, we witness both HGS-GCN and HS-GCN outperform other GCN modules. Our method leads to 1.4 mm and 1.7 mm against the second-best method [44]. The results validate our choice of GCN architectures as auxiliary modules. On the other hand, HS-GCN achieves marginal improvements by 0.7 mm and 0.3 mm on MPJPE and PMPJPE metrics against HGS-GCN. As hop features H_{hop} are generated from input embeddings H_{in}, the scaling factors in HS-GCN derived from the attention between H_{hop} and H_{in}, are considered to have a higher correlation than the pair of H_{hop} and H_{ret}. Taking this into account, the different performance indicates that attention between similar feature representations can produce more accurate weights for scaling hop features.

Study of information stored in registers. Following the method proposed by Darcet et al. [21], we measure the information contained in transformer tokens by witnessing the shifts of distributions of feature norms in those tokens. The distributions of norms are divided by a threshold to distinguish inliers from outliers. The quantity of outliers represents the amount of global information, and the quantity of inliers represents local information. Following this method, we first find a threshold by computing the distribution of feature norms of a high-accurate joint in one layer of our model, which is shown in Fig. 6a. We could witness in the figure that most of the norms are distributed on the left side of **15**, thus we take this value as a threshold.

Then we compute the distribution of a high-loss joint and measure it with this threshold. The results are given in Fig. 6b, where we witness that almost half of the norms are distributed over the value of 15. To study the efficacy of the registers in storing global information, we compute the distributions of the same joint with 8 register tokens added in Transformers, the results are shown in Fig. 6c. By comparing Fig. 6b and Fig. 6c, we notice that more norm values are shifted to the left side of the threshold with the incorporation of registers. The shift of norm values validates that the registers can assist other transformer tokens in concentrating more on local features.

Study of registers in improving high-loss joints. We further study the effectiveness of registers on joint errors. By trial and error, we refer to the joints of feet, hands, and elbows as high-loss joints, and those that are close to the center joint (pelvis) as low-loss joints (e.g., the two hips). The results are given in Table 8, where some of the high-loss joints obtain significant improvements in accuracy. Specifically, improvements on the joints of the left and right foot arrive at 4.6 mm and 6.4 mm, respectively, and those on the joints of the left and right hand are 1 mm and 1.4 mm, respectively. On the contrary, the registers cause higher prediction losses for low-loss joints. Given the fact that the registers can strengthen the transformer tokens in exploiting local features, we could conclude that high-loss joints require more precise local information to predict accurate 3D joint locations. On the other hand, excessive local information might hinder our network from predicting accurate 3D locations of low-loss joints.

Number of registers. The results are shown in Table 9. The first row reports the results of ADGT trained without a register. The best

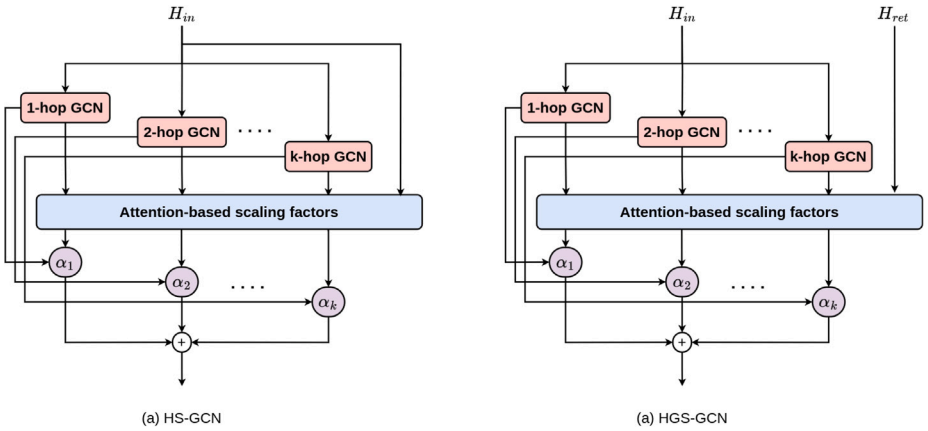

Fig. 5. Two methods to compute adaptive factors. (a) computes adaptive factors with H_{in} embeddings. (b) computes adaptive factors with H_{ret} embeddings.

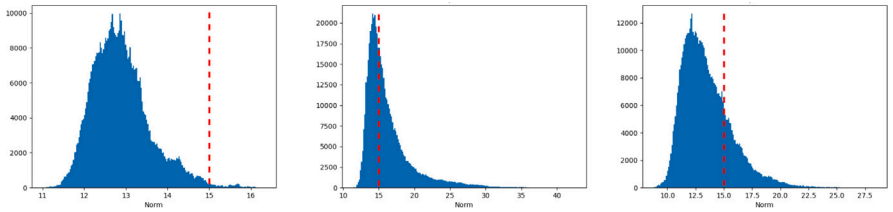

Fig. 6. Distributions of computed norms along the vectors of joint features. (a) The distribution of computed norms from a joint with high accuracy; (b) The distribution of norms from a joint with low accuracy; (c) The distribution of norms after adding registers. The distributions in (b) and (c) are from the same low-accuracy joint.

Table 7
Quantitative comparison of learnable and fixed graphs of GCN modules deployed to learn local embeddings. Each module is tested on ADGT with 5 layers and 96 channels. The differences between HS-GCN and HGS-GCN are shown in Fig. 5.

Module	Params	MPJPE [23] ↓ (mm)	P-MPJPE [42] ↓ (mm)
ChebyNet [57]	0.92M	52.1	41.7
GCN [31]	0.85M	52.8	42.1
Weightunshare [44]	1.78M	51.5	40.9
GAT [39]	0.95M	52.2	41.8
HGS-GCN	0.92M	50.8	39.5
HS-GCN	0.92M	**50.1**	**39.2**

Table 8
Quantitative comparison of the effectiveness of registers on joint errors. Both high-loss and low-loss joints are defined by comparing the losses with the average value. The joints with losses higher than the average value are considered as high-loss, otherwise low-loss. The ✓and ✗ signify appending register tokens in Transformers or not. The losses are evaluated with the MPJPE metric.

Registers	High-loss joints						Low-loss joints	
	left foot	right foot	left hand	right hand	left elbow	right elbow	left hip	right hip
✗	50.5 mm	58.3 mm	55.1 mm	56.9 mm	40.4 mm	46.7 mm	16.3 mm	16.3 mm
✓	45.9 mm	51.9 mm	54.1 mm	55.5 mm	40.2 mm	47.4 mm	16.9 mm	16.9 mm

accuracies are achieved at 50.1 mm and 39.2 mm by appending 8 registers. We can observe that the performance does not improve along with the increase in the number of registers. The losses with 8 registers are higher than those with fewer registers.

Qualitative comparison of visualized accuracy. We compare visualized predictions of our method and Graformer [13], the results are shown in Fig. 7. Our method exhibits improved alignment with ground-truth 3D poses, in comparison with the predicted 3D poses by Graformer. Meanwhile, more estimated 3D poses on Human36M [23] are presented in Fig. 8.

5. Limitations

Despite the effectiveness of our proposed approach, several limitations remain. First, our method is designed for frame-based 3D human pose estimation and does not explicitly incorporate temporal information. As a result, it cannot leverage motion continuity or handle sequential inputs, which are often beneficial for improving pose consistency in video-based applications. Second, the current design targets single-person pose estimation. In multi-person scenarios, interactions, occlusions, and overlapping body parts are common. Our model is

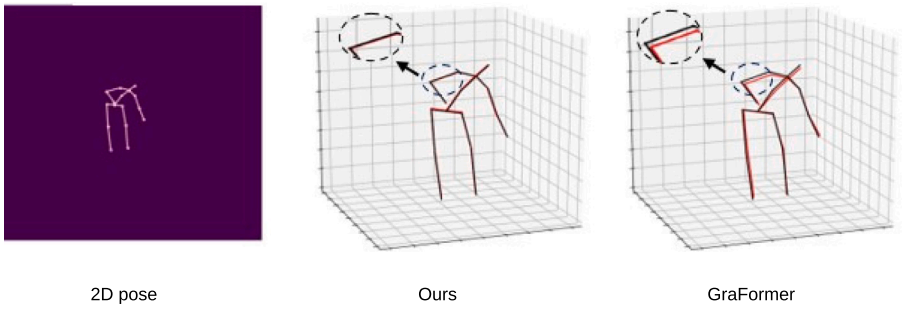

Fig. 7. Visualized comparison with GraFormer. The black curves are ground-truth 3D poses and the red curves are predicted 3D poses.

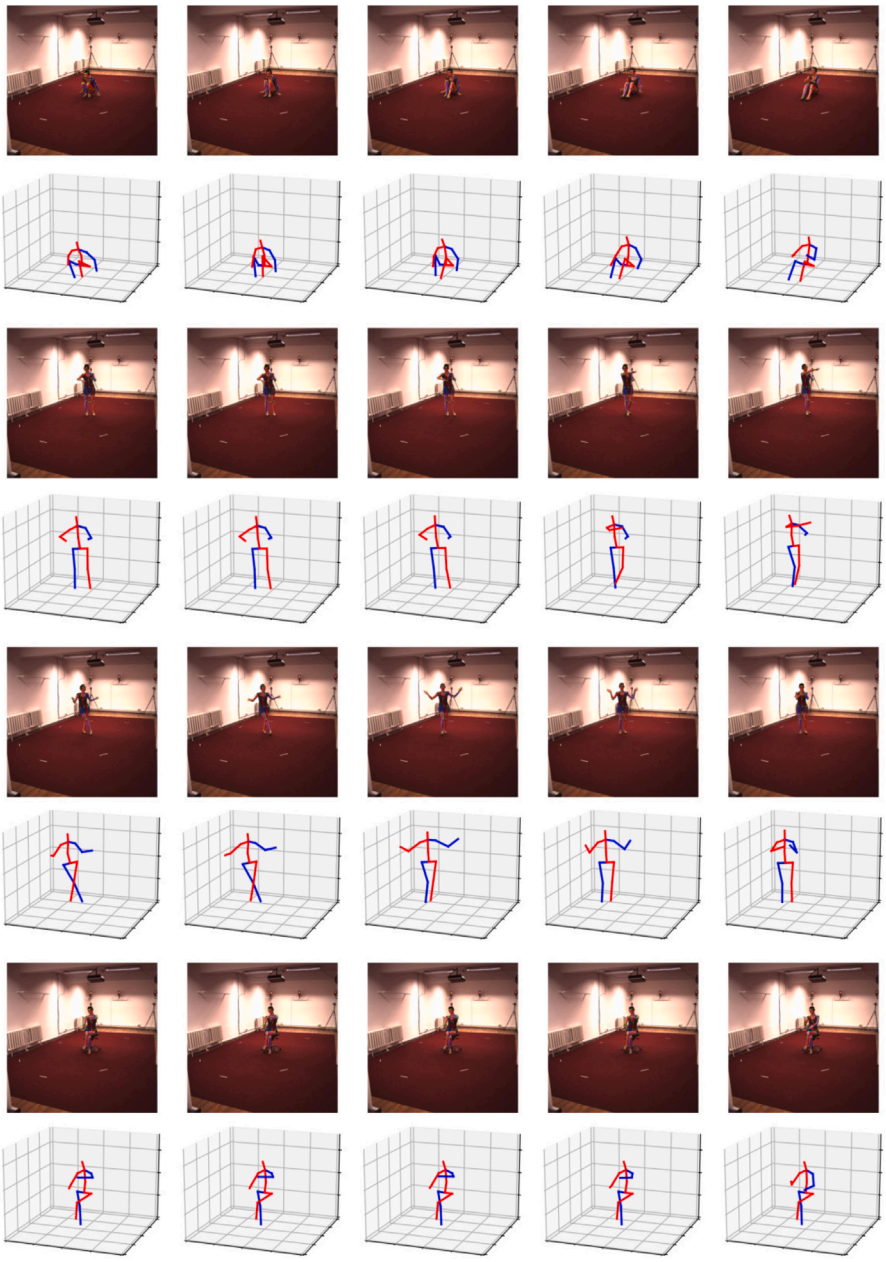

Fig. 8. Qualitative results of the proposed method on Human3.6M dataset.

not directly applicable without significant modification or the integration of person-specific disentanglement mechanisms. Third, while our approach is aimed at general-purpose 3D human pose estimation and performs robustly on action-wise average losses, it may struggle

Table 9
Quantitative comparison of the number of registers appended to Transformers. All the models are trained and evaluated with 5 layers of ADGT model and 96 channels.

Num-register	MPJPE [23] ↓ (mm)	P-MPJPE [42] ↓ (mm)
0	50.9	40.2
2	50.7	39.9
4	50.4	39.7
6	50.4	39.4
8	**50.1**	**39.2**
10	50.8	39.4

with highly complex poses involving severe self-occlusion. For example, actions such as "Sitting" or "SittingD" often result in entangled limb configurations that are inherently difficult to resolve from monocular 2D inputs, leading to a drop in performance on these categories.

6. Conclusion

In this work, we introduced ADGT, a novel parallel GCN-transformer architecture designed for efficient and accurate 2D-to-3D human pose estimation. Our approach effectively addresses key limitations in existing methods, including receptive field mismatch in sequential architectures, over-smoothing [14] in deep GCNs, and inefficient fusion of local and global features. By integrating Hop-wise Scalable GCN (HS-GCN) for enhanced local feature extraction, Attention-Based Local Feature Extractor (ALFE) for selective refinement, and Register-Based Transformer Enhancement for improved feature separation, our model ensures robust and efficient pose estimation.

Extensive experiments on Human3.6M [23] and MPI-INF-3DHP [24] datasets demonstrate that ADGT achieves state-of-the-art performance among frame-based methods while maintaining a significantly smaller model size and computational cost compared to video-based approaches. Our method outperforms prior GCN-based and Transformer-based techniques in key evaluation metrics such as MPJPE [23] and P-MPJPE [42] while also generalizing well to unseen data.

These findings highlight the potential of ADGT as an effective solution for real-time 3D human pose estimation in various applications, including motion analysis, virtual reality, and autonomous systems. Future work will focus on further optimizing computational efficiency and extending the framework to multi-person scenarios.

To promote reproducibility and support future research, we will release our code upon publication.

CRediT authorship contribution statement

Shuo Yang: Writing – original draft, Methodology. **Anh Tuan Luu:** Writing – review & editing. **Xuan Son Nguyen:** Writing – review & editing, Supervision. **Aymeric Histace:** Writing – review & editing, Supervision. **Bart Jansen:** Writing – review & editing, Supervision. **Hichem Sahli:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was funded by the EUTOPIA PhD Co-tutelle Program, France (EUTOPIA is an alliance of ten European universities and six Global partners co-funded by the European Union) grant EUTOPIA-PhD-2021-0000000061.

Data availability

All experiments are conducted with open datasets. We would share the code of our method upon publication.

References

[1] H. Duan, J. Wang, K. Chen, D. Lin, DG-STGCN: dynamic spatial-temporal modeling for skeleton-based action recognition, 2022, arXiv preprint arXiv:2210.05895.
[2] H. Duan, Y. Zhao, K. Chen, D. Lin, B. Dai, Revisiting skeleton-based action recognition, in: CVPR, 2022, pp. 2969–2978.
[3] J. Lee, M. Lee, D. Lee, S. Lee, Hierarchically decomposed graph convolutional networks for skeleton-based action recognition, in: ICCV, 2023, pp. 10444–10453.
[4] H. Xu, Y. Gao, Z. Hui, J. Li, X. Gao, Language knowledge-assisted representation learning for skeleton-based action recognition, 2023, arXiv preprint arXiv:2305.12398.
[5] S. Li, W. Zhang, A.B. Chan, Maximum-margin structured learning with deep networks for 3d human pose estimation, in: ICCV, 2015, pp. 2848–2856.
[6] X. Liu, Z. Zhao, W. Tian, B. Liu, H. He, Capsule network with using shifted windows for 3D human pose estimation, J. Vis. Commun. Image Represent. 108 (2025) 104409.
[7] G. Pavlakos, X. Zhou, K. Daniilidis, Ordinal depth supervision for 3d human pose estimation, in: CVPR, 2018, pp. 7307–7316.
[8] G. Pavlakos, X. Zhou, K.G. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3D human pose, in: CVPR, 2017, pp. 7025–7034.
[9] X. Sun, J. Shang, S. Liang, Y. Wei, Compositional human pose regression, in: ICCV, 2017, pp. 2602–2611.
[10] Z. Islam, A. Ben Hamza, Multi-hop graph transformer network for 3D human pose estimation, J. Vis. Commun. Image Represent. 101 (2024) 104174.
[11] K. Zhai, Q. Nie, B. Ouyang, X. Li, S. Yang, HopFIR: Hop-wise GraphFormer with intragroup joint refinement for 3D human pose estimation, 2023, arXiv preprint arXiv:2302.14581.
[12] L. Zhao, X. Peng, Y. Tian, M. Kapadia, D.N. Metaxas, Semantic graph convolutional networks for 3D human pose regression, in: CVPR, 2019, pp. 3425–3435.
[13] W. Zhao, W. Wang, Y. Tian, Graformer: Graph-oriented transformer for 3d pose estimation, in: CVPR, 2022, pp. 20438–20447.
[14] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: AAAI, 2018, pp. 3538–3545.
[15] U. Alon, E. Yahav, On the bottleneck of graph neural networks and its practical implications, in: ICLR, 2021.
[16] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, P. Tossou, Rethinking graph transformers with spectral attention, in: NeurIPS, 2021, pp. 21618–21629.
[17] Z. Chen, J. Dai, J. Bai, J. Pan, DGFormer: Dynamic graph transformer for 3D human pose estimation, Pattern Recognit. 152 (2024).
[18] S. Mehraban, V. Adeli, B. Taati, Motionagformer: Enhancing 3d human pose estimation with a transformer-gcnformer network, in: WACV, 2024, pp. 6920–6930.
[19] W. Zhu, G. Song, L. Wang, S. Liu, AnchorGT: Efficient and flexible attention architecture for scalable graph transformers, 2024, arXiv preprint arXiv:2405.03481.
[20] M.S. Burtsev, Y. Kuratov, A. Peganov, G.V. Sapunov, Memory transformer, 2020, arXiv preprint arXiv:2006.11527.
[21] T. Darcet, M. Oquab, J. Mairal, P. Bojanowski, Vision transformers need registers, in: ICLR, 2024.
[22] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, C.-J. Hsieh, DynamicViT: Efficient vision transformers with dynamic token sparsification, in: NeurIPS, 2021.
[23] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments, IEEE PAMI 36 (7) (2014) 1325–1339.
[24] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, C. Theobalt, Monocular 3D human pose estimation in the wild using improved CNN supervision, in: 3DV, 2017, pp. 506–516.
[25] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, in: CVPR, 2018, pp. 7103–7112.
[26] H. Ci, C. Wang, X. Ma, Y. Wang, Optimizing network structure for 3d human pose estimation, in: ICCV, 2019, pp. 2262–2271.
[27] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: ECCV, 2016, pp. 483–499.
[28] W. Cao, A. Zhang, Z. He, Y. Zhang, X. Yin, Hierarchical spatial-temporal masked contrast for skeleton action recognition, IEEE Trans. Artif. Intell. 5 (11) (2024) 5801–5814.
[29] A. Dey, S. Biswas, L. Abualigah, Umpire's signal recognition in cricket using an attention based DC-GRU network, Int. J. Eng. 37 (4) (2024) 662–674.
[30] D.-N.L. Arnab Dey, Workout action recognition in video streams using an attention driven residual DC-GRU network, Comput. Mater. Contin. 79 (2) (2024) 3067–3087.

[31] T.N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in: ICLR, 2017.
[32] Z. Zhou, X. Li, Convolution on graph: A high-order and adaptive approach, 2017, arXiv preprint arXiv:1706.09916.
[33] Y. Xie, C. Hong, W. Zhuang, L. Liu, J. Li, HOGFormer: high-order graph convolution transformer for 3D human pose estimation, Int. J. Mach. Learn. Cybern. 16 (1) (2025) 599–610.
[34] C. Hong, L. Chen, Y. Liang, Z. Zeng, Stacked capsule graph autoencoders for geometry-aware 3D head pose estimation, Comput. Vis. Image Un Derstanding 208–209 (2021) 103224.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017, pp. 6000–6010.
[36] Y. Zhu, X. Xu, F. Shen, Y. Ji, L. Gao, H.T. Shen, PoseGTAC: Graph transformer encoder-decoder with atrous convolution for 3D human pose estimation, in: IJCAI, 2021, pp. 1359–1365.
[37] J. Huang, C. Hong, R. Xie, L. Ran, J. Qian, A simple and efficient channel MLP on token for human pose estimation, Int. J. Mach. Learn. Cybern. (2024) 1–9.
[38] Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.-T. Xia, E. Zhou, Tokenpose: Learning keypoint tokens for human pose estimation, in: ICCV, 2021, pp. 11313–11322.
[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: ICLR, 2018.
[40] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., Dinov2: Learning robust visual features without supervision, 2023, arXiv preprint arXiv:2304.07193.
[41] H. Touvron, M. Cord, H. Jégou, Deit iii: Revenge of the vit, in: European Conference on Computer Vision, Springer, 2022, pp. 516–533.
[42] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K.G. Derpanis, K. Daniilidis, Monocap: Monocular human motion capture using a cnn coupled with a geometric prior, IEEE PAMI 41 (4) (2018) 901–914.
[43] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[44] K. Liu, R. Ding, Z. Zou, L. Wang, W. Tang, A comprehensive study of weight sharing in graph networks for 3d human pose estimation, in: ECCV, 2020, pp. 318–334.
[45] J. Martinez, R. Hossain, J. Romero, J.J. Little, A simple yet effective baseline for 3d human pose estimation, in: ICCV, 2017, pp. 2640–2649.
[46] T. Xu, W. Takano, Graph stacked hourglass networks for 3D human pose estimation, in: CVPR, 2021, pp. 16105–16114.
[47] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, X. Wang, 3D human pose estimation in the wild by adversarial learning, in: CVPR, 2018, pp. 5255–5264.
[48] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, W. Zhang, Deep kinematics analysis for monocular 3d human pose estimation, in: CVPR, 2020, pp. 899–908.
[49] B.X. Yu, Z. Zhang, Y. Liu, S.-h. Zhong, Y. Liu, C.W. Chen, Gla-gcn: Global-local adaptive graph convolutional network for 3d human pose estimation from monocular video, in: ICCV, 2023, pp. 8818–8829.
[50] J. Zhang, Z. Tu, J. Yang, Y. Chen, J. Yuan, Mixste: Seq2seq mixed spatio-temporal encoder for 3d human pose estimation in video, in: CVPR, 2022, pp. 13232–13242.
[51] Q. Zhao, C. Zheng, M. Liu, P. Wang, C. Chen, PoseFormerV2: Exploring frequency domain for efficient and robust 3D human pose estimation, in: CVPR, 2023, pp. 8877–8886.
[52] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, Z. Ding, 3D human pose estimation with spatial and temporal transformers, in: ICCV, 2021, pp. 11656–11665.
[53] M. Andriluka, L. Pishchulin, P. Gehler, B. Schiele, 2D human pose estimation: New benchmark and state of the art analysis, in: CVPR, 2014, pp. 3686–3693.
[54] Q. Nie, Z. Liu, Y. Liu, Lifting 2D human pose to 3D with domain adapted 3D body concept, Int. J. Comput. Vis. 131 (5) (2023) 1250–1268.
[55] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, S. Lin, Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach, in: ECCV, 2020, pp. 507–523.
[56] Z. Zou, W. Tang, Modulated graph convolutional network for 3D human pose estimation, in: ICCV, 2021, pp. 11477–11487.
[57] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: NIPS, 2016, pp. 3844–3852.