Pattern Recognition 158 (2025) 110925



Contents lists available at ScienceDirect

Pattern Recognition

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





GraphMLP: A graph MLP-like architecture for 3D human pose estimation

Wenhao Li [a], Mengyuan Liu [a,*], Hong Liu [a], Tianyu Guo [b], Ti Wang [a], Hao Tang [c], Nicu Sebe [d]

[a] State Key Laboratory of General Artificial Intelligence, Peking University, Shenzhen Graduate School, China
[b] Ant Group, China
[c] National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, China
[d] University of Trento, Italy

ARTICLE INFO

Keywords:
3D human pose estimation
Multi-layer perceptron
Graph convolutional network

ABSTRACT

Modern multi-layer perceptron (MLP) models have shown competitive results in learning visual representations without self-attention. However, existing MLP models are not good at capturing local details and lack prior knowledge of human body configurations, which limits their modeling power for skeletal representation learning. To address these issues, we propose a simple yet effective graph-reinforced MLP-Like architecture, named GraphMLP, that combines MLPs and graph convolutional networks (GCNs) in a *global-local-graphical* unified architecture for 3D human pose estimation. GraphMLP incorporates the graph structure of human bodies into an MLP model to meet the domain-specific demand of the 3D human pose, while allowing for both local and global spatial interactions. Furthermore, we propose to flexibly and efficiently extend the GraphMLP to the video domain and show that complex temporal dynamics can be effectively modeled in a simple way with negligible computational cost gains in the sequence length. To the best of our knowledge, this is the first MLP-Like architecture for 3D human pose estimation in a single frame and a video sequence. Extensive experiments show that the proposed GraphMLP achieves state-of-the-art performance on two datasets, *i.e.*, Human3.6M and MPI-INF-3DHP. Code and models are available at https://github.com/Vegetebird/GraphMLP.

1. Introduction

3D human pose estimation from images is important in numerous applications, such as action recognition, motion capture, and augmented/virtual reality. Most existing works solve this task by using a 2D-to-3D pose lifting method, which takes graph-structured 2D joint coordinates detected by a 2D keypoint detector as input [1–4]. This is an inherently ambiguous problem since multiple valid 3D joint locations may correspond to the same 2D projection in the image space. However, it is practically solvable since 3D poses often lie on low-dimensional manifolds, which can provide important structural priors to mitigate the depth ambiguity [5,6].

Early works attempted to employ fully connected networks (FCN) [1] to lift the 2D joints into 3D space. However, the dense connection of FCN often results in overfitting and poor performance [7]. To address this problem, recent works consider that the skeleton of a human body can be naturally represented as the graph structure and utilize graph convolutional networks (GCNs) for this task [8,6,9]. Although GCN-based methods are effective at aggregating neighboring nodes to extract local features, they often suffer from limited receptive fields to obtain stronger representation power [10], as these methods usually rely on learning relationships between human body joints utilizing first-hop neighbors. However, global information between distant body joints is crucial for understanding the overall body posture and movement patterns [11,10]. For instance, the position of the hand relative to the foot can indicate whether the person is standing, sitting, or lying down. One way to capture global information is by stacking multiple GCN layers, which expands the model's receptive fields to cover the entire kinematic chain of a human. However, this leads to the over-smoothing issue where valuable information is lost in deeper layers [12].

Recently, modern multi-layer perceptron (MLP) models (in particular MLP-Mixer [13]) with global receptive fields provide new architectural designs in vision [14–17]. The MLP model stacks only fully-connected layers without self-attention and consists of two types of blocks. The spatial MLP aggregates global information among tokens, and the channel MLP focuses on extracting features for each token. By stacking these two MLP blocks, it can be simply built with less inductive bias and achieves impressive performance in learning visual representations. This motivates us to explore an MLP-Like architecture for skeleton-based representation learning.

* Corresponding author.
E-mail addresses: wenhaoli@pku.edu.cn (W. Li), liumengyuan@pku.edu.com (M. Liu), hongliu@pku.edu.cn (H. Liu), guotianyu.gty@antgroup.com (T. Guo), tiwang@stu.pku.edu.cn (T. Wang), haotang@pku.edu.cn (H. Tang), niculae.sebe@unitn.it (N. Sebe).

https://doi.org/10.1016/j.patcog.2024.110925
Received 8 October 2023; Received in revised form 22 July 2024; Accepted 20 August 2024
Available online 23 August 2024
0031-3203/© 2024 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

However, there remain two critical challenges in adapting the MLP models from vision to skeleton: **(i)** Despite their successes in vision, existing MLPs are less effective in modeling graph-structured data due to their simple connections among all nodes. Different from RGB images represented by highly dense pixels, skeleton inputs are inherently sparse and graph-structured data (see Fig. 2 (left)). Without incorporating the prior knowledge of human body configurations, the model is prone to learning spurious dependencies, which can lead to physically implausible poses [18,7]. **(ii)** While such models are capable of capturing global interactions between distant body joints (such as the hand and foot) via their spatial MLPs, they may not be good at capturing local interactions due to the lack of careful designs for modeling relationships between adjacent joints (such as the head and neck). However, local information is also essential for 3D human pose estimation, as it can help the model to understand fine-grained movement details [19,20]. For example, the movement of the head relative to the neck can indicate changes in gaze direction or subtle changes in posture.

To overcome both limitations, we present GraphMLP, a new graph-reinforced MLP-Like architecture for 3D human pose estimation, as depicted in Fig. 2. Our GraphMLP is conceptually simple yet effective: it builds a strong collaboration between modern MLPs and GCNs to construct a *global-local-graphical* unified architecture for learning better skeletal representations. Specifically, GraphMLP mainly contains a stack of novel GraphMLP layers. Each layer consists of two Graph-MLP blocks where the spatial graph MLP (SG-MLP) and the channel graph MLP (CG-MLP) blocks are built by injecting GCNs into the spatial MLP and channel MLP, respectively. By combining MLPs and GCNs in a unified architecture, our GraphMLP is able to obtain the prior knowledge of human configurations encoded by the graph's connectivity and capture both local and global spatial interactions among body joints, hence yielding better performance.

Furthermore, we extend our GraphMLP from a single frame to a video sequence. Most existing video-based methods [21–23] typically model temporal information by treating each frame as a token or treating the time axis as a separated dimension. However, these methods often suffer from redundant computations that make little contribution to the final performance because nearby poses are similar [24]. Additionally, they are too computationally expensive to process long videos (*e.g.*, 243 frames), thereby limiting their practical utility in real-world scenarios. To tackle these issues, we propose to utilize a simple and efficient video representation of pose sequences. This representation captures complex temporal dynamics by mixing temporal information in the feature channels and treating each joint as a token, offering negligible computational cost gains in the sequence length. It is also a unified and flexible representation that can accommodate arbitrary-length sequences (*i.e.*, a single frame and variable-length videos).

The proposed GraphMLP is evaluated on two challenging datasets, Human3.6M [25] and MPI-INF-3DHP [26]. Extensive experiments show the effectiveness and generalization ability of our approach, which advances the state-of-the-art performance for estimating 3D human poses from a single image. Its performance surpasses MGCN [27] by 1.4 mm in mean per joint position error (MPJPE) on the Human3.6M dataset. Besides, as shown in Fig. 1, it brings clear 6.6 mm and 6.9 mm improvements in MPJPE for the MLP model [13] and the GCN model [18] on the MPI-INF-3DHP dataset, respectively. Surprisingly, compared to video pose Transformer (*e.g.*, PoseFormer [22]), even with 5× fewer computational costs, our MLP-Like architecture achieves better performance.

Overall, our main contributions are summarized as follows:

- We present, to the best of our knowledge, the first MLP-Like architecture called GraphMLP for 3D human pose estimation. It combines the advantages of modern MLPs and GCNs, including globality, locality, and connectivity.
- The novel SG-MLP and CG-MLP blocks are proposed to encode the graph structure of human bodies within MLPs to obtain domain-specific knowledge about the human body while enabling the model to capture both local and global interactions.
- A simple and efficient video representation is proposed to extend our GraphMLP to the video domain flexibly. This representation enables the model to effectively process arbitrary-length sequences with negligible computational cost gains.
- Extensive experiments demonstrate the effectiveness and generalization ability of the proposed GraphMLP and show new state-of-the-art results on two challenging datasets, *i.e.*, Human3.6M [25] and MPI-INF-3DHP [26].

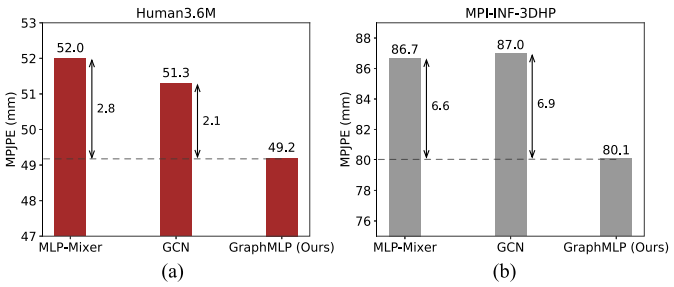


Fig. 1. Performance comparison with MLP-Mixer [13] and GCN [18] on Human3.6M (a) and MPI-INF-3DHP (b) datasets. The proposed GraphMLP absorbs the advantages of modern MLPs and GCNs to effectively learn skeletal representations, consistently outperforming each of them. The evaluation metric is MPJPE (the lower the better).

2. Related work

3D Human Pose Estimation. There are mainly two categories to estimate 3D human poses. The first category of methods directly regresses 3D human joints from RGB images [28–31]. The second category is the 2D-to −3D pose lifting method [1,21,32], which employs an off-the-shelf 2D pose detection as the front end and designs a 2D-to −3D lifting network using detected 2D poses as input. This lifting method can achieve state-of-the-art performance and has become the mainstream method due to its efficiency and effectiveness. For example, FCN [1] shows that 3D poses can be regressed simply and effectively from 2D keypoints with fully-connected networks. TCN [21] extends the FCN to video by utilizing temporal convolutional networks to exploit temporal information from 2D pose sequences. Liu et al. [33] incorporate the attention mechanism into TCN to enhance the modeling of long-range temporal relationships across frames. SRNet [34] proposes a split-and-recombine network that splits the human body joints into multiple local groups and recombines them with a low-dimensional global context. PoSynDA [35] uses domain adaptation through multi-hypothesis pose synthesis for 3D human pose estimation.

Since the physical skeleton topology can form a graph structure, recent progress has focused on employing graph convolutional networks (GCNs) to address the 2D-to −3D lifting problem. LCN [6] introduces a locally connected network to improve the representation capability of GCN. SemGCN [8] allows the model to learn the semantic relationships among the human joints. MGCN [27] improves SemGCN by introducing a weight modulation and an affinity modulation.

Transformers in Vision. Recently, Transformer-based methods achieve excellent results on various computer vision tasks, such as image classification [36–38], object detection [39–41], and pose estimation [22,42,43]. The seminal work of ViT [36] divides an image into 16×16 patches and uses a pure Transformer encoder to extract visual features. PoseFormer [22] utilizes a pure Transformer-based architecture to model spatial and temporal relationships from videos. Strided Transformer [24] incorporates strided convolutions into Transformers to aggregate information from local contexts for video-based 3D human

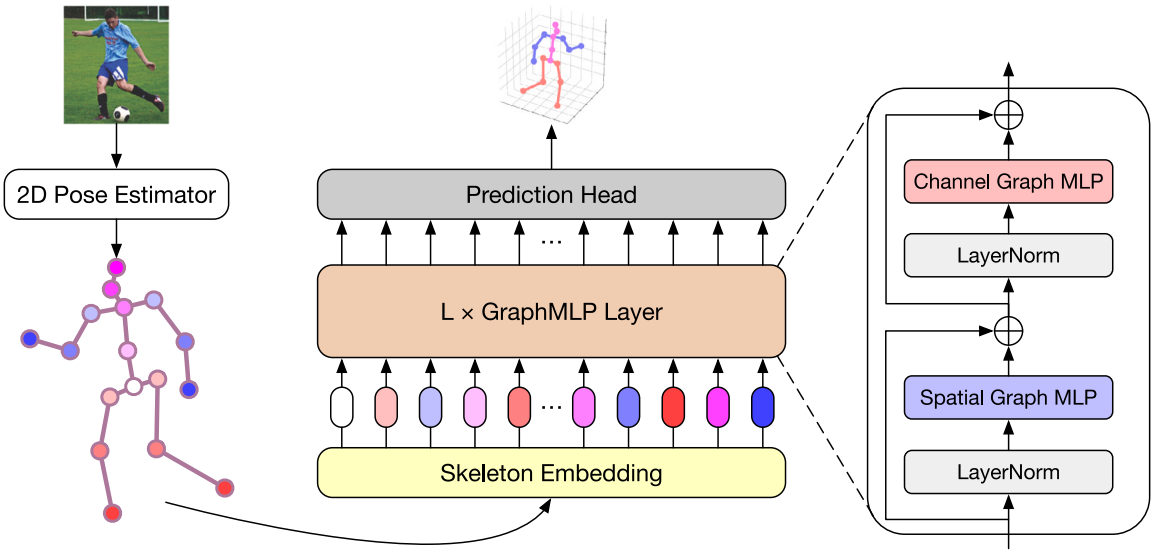


Fig. 2. Overview of the proposed GraphMLP architecture. The left illustrates the skeletal structure of the human body. The 2D joint inputs detected by a 2D pose estimator are sparse and graph-structured data. GraphMLP treats each 2D keypoint as an input token, linearly embeds each of them through the skeleton embedding, feeds the embedded tokens to GraphMLP layers, and finally performs regression on resulting features to predict the 3D pose via the prediction head. Each GraphMLP layer contains one spatial graph MLP (SG-MLP) and one channel graph MLP (CG-MLP). For easy illustration, we show the architecture using a single image as input.

pose estimation. HDFormer [42] proposes a High-order Directed Transformer to utilize high-order information on a directed skeleton graph based on Transformer. RTPCA [44] introduces a temporal pyramidal compression-and-amplification design for enhancing temporal modeling in 3D human pose estimation. Mesh Graphormer [20] combines GCNs and attention layers in a serial order to capture local and global dependencies for human mesh reconstruction.

Unlike [20], we mainly investigate how to combine more efficient architectures (*i.e.*, modern MLPs) and GCNs to construct a stronger architecture for 3D human pose estimation and adopt a parallel manner to make it possible to model local and global information at the same time. Moreover, different from previous video-based methods [22,24,23] that treat each frame as a token for temporal modeling, we mix features across all frames and maintain each joint as a token, which makes the network to be economical and easy to train.

MLPs in Vision. Modern MLP models are proposed to reduce the inductive bias and computational cost by replacing the complex self-attentions of Transformers with spatial-wise linear layers [14,15,45]. MLP-Mixer [13] firstly proposes an MLP-Like model, which is a simple network architecture containing only pure MLP layers. Compared with FCN [1] (*i.e.*, conventional MLPs), this architecture introduces some modern designs, *e.g.*, layer normalization (LN) [46], GELU [47], mixing spatial information. Moreover, ResMLP [48] proposes a purely MLP architecture with the Affine transformation. CycleMLP [17] proposes a cycle fully-connected layer to aggregate spatial context information and deal with variable input image scales. However, these modern MLP models have not yet been applied to 3D human pose estimation.

Inspired by their successes in vision, we first attempt to explore how MLP-Like architectures can be used for 3D human pose estimation in non-Euclidean skeleton data. The difference between our approach and the existing MLPs is that we introduce the inductive bias of the physical skeleton topology by combining MLP models with GCNs, providing more physically plausible and accurate estimations. We further investigate applying MLP-Like architecture in video and design an efficient video representation, which is seldom studied.

3. Proposed GraphMLP

Fig. 2 illustrates the overall architecture of the proposed GraphMLP. Our approach takes 2D joint locations $P \in \mathbb{R}^{N\times 2}$ estimated by an off-the-shelf 2D pose detector as input and outputs predicted 3D poses $\widetilde{X} \in \mathbb{R}^{N\times 3}$, where N is the number of joints. The proposed GraphMLP architecture consists of a skeleton embedding module, a stack of L identical GraphMLP layers, and a prediction head module. The core operation of GraphMLP architecture is the GraphMLP layer, each of which has two parts: a spatial graph MLP (SG-MLP) block and a channel graph MLP (CG-MLP) block. Our GraphMLP has a similar architecture to the original MLP-Mixer [13], but we incorporate graph convolutional networks (GCNs) into the model to meet the domain-specific requirement of the 3D human pose estimation and learn the local and global interactions of human body joints.

3.1. Preliminary

In this subsection, we briefly introduce the preliminaries in GCNs and modern MLPs.

3.1.1. Graph convolutional networks

Let $G = (V, A)$ denotes a graph where V is a set of N nodes and $A \in \{0,1\}^{N\times N}$ is the adjacency matrix encoding the edges between the nodes. Given a $(\ell-1)$th layer feature $X_{\ell-1} \in \mathbb{R}^{N\times C}$ with C dimensions, a generic GCN [49] layer, which is used to aggregate features from neighboring nodes, can be formulated as:

$$X_{\ell} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X_{\ell-1}W, \tag{1}$$

where $W \in \mathbb{R}^{C\times C'}$ is the learnable weight matrix and $\widetilde{A} = A + I$ is the adjacency matrix with added self-connections. I is the identity matrix, $\widetilde{D}$ is a diagonal matrix, and $\widetilde{D}^{ii} = \sum_j \widetilde{A}^{ij}$.

The design principle of the GCN block is similar to [18], but we adopt a simplified single-frame version that contains only one graph layer. For the GCN blocks of our GraphMLP, the nodes in the graph denote the joint locations of the human body in Fig. 3(a), and the adjacency matrix represents the bone connections between every two joints for node information passing in Fig. 3(b), *e.g.*, the rectangle of 1st row and 2nd column denotes the connection between joint 0 and joint 1. In addition, the different types of bone connections use different kernel weights following [18].

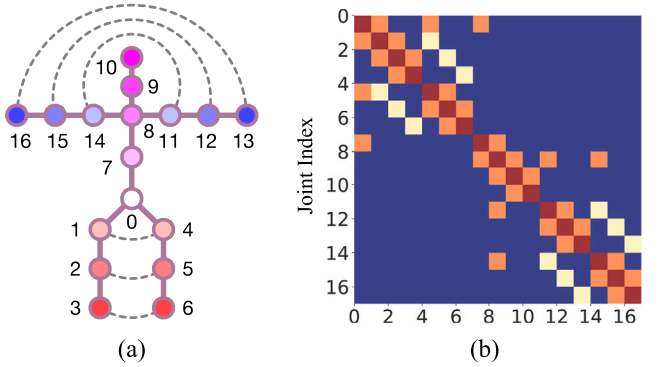


Fig. 3. (a) The human skeleton graph in physical and symmetrical connections. (b) The adjacency matrix used in the GCN blocks of GraphMLP. Different colors denote the different types of bone connections.

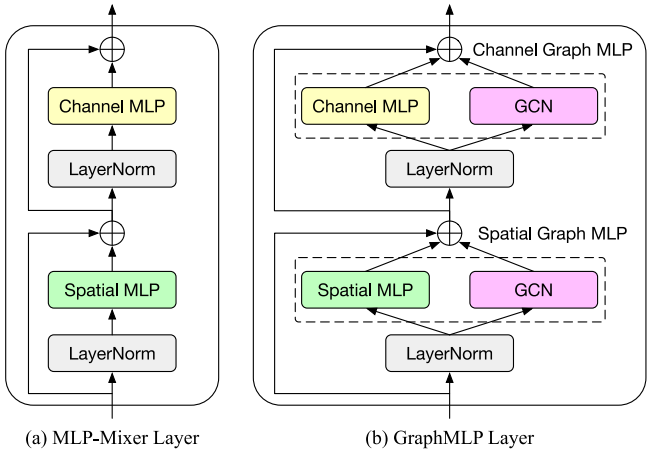


Fig. 4. Comparison of MLP Layers. (a) MLP-Mixer Layer [13]. (b) Our GraphMLP Layer. Compared with MLP-Mixer, our GraphMLP incorporates graph structural priors into the MLP model via GCN blocks. The MLPs and GCNs are in a paralleled design to model both local and global interactions.

3.1.2. Modern MLPs

MLP-Mixer is the first modern MLP model proposed in [13]. It is a simple and attention-free architecture that mainly consists of a spatial MLP and a channel MLP, as illustrated in Fig. 4(a). The spatial MLP aims to transpose the spatial axis and channel axis of tokens to mix spatial information. Then the channel MLP processes tokens in the channel dimension to mix channel information. Let $X_{\ell-1} \in \mathbb{R}^{N\times C}$ be an input feature, an MLP-Mixer layer can be calculated as:

$$X'_{\ell} = X_{\ell-1} + \text{Spatial}-\text{MLP}(\text{LN}(X_{\ell-1}^{\top}))^{\top}, \tag{2}$$

$$X_{\ell} = X'_{\ell} + \text{Channel}-\text{MLP}(\text{LN}(X'_{\ell})), \tag{3}$$

where $\text{LN}(\cdot)$ is layer normalization (LN) [46] and $\top$ is the matrix transposition. Both spatial and channel MLPs contain two linear layers and a GELU [47] non-linearity in between.

We adopt this MLP model as our baseline, which is similar to [13], but we transpose the tokens before LN in the spatial MLP (*i.e.*, normalize tokens along the spatial dimension). However, such a simple MLP model neglects to extract fine-grained local details and lacks prior knowledge about the human configurations, which are perhaps the bottlenecks restricting the representation ability of MLP-Like architectures for learning skeletal representations.

3.2. Network architecture

In this work, we present a novel GraphMLP built upon the MLP-Mixer described in [13] to overcome the aforementioned limitations of existing MLP models. Below we elaborate on each module used in GraphMLP and provide its detailed implementations.

3.2.1. Skeleton embedding

Raw input data are mapped to latent space via the skeleton embedding module. Given the input 2D pose $P \in \mathbb{R}^{N\times 2}$ with N body joints, we treat each joint as an input token. These tokens are projected to the high-dimension token feature $X_0 \in \mathbb{R}^{N\times C}$ by a linear layer, where C is the hidden size.

3.2.2. GraphMLP layer

The existing MLP simply connects all nodes but does not take advantage of graph structures, making it less effective in handling graph-structured data. To tackle this issue, we introduce the GraphMLP layer that unifies globality, locality, and connectivity in a single layer. Compared with the original MLP-Mixer in Fig. 4(a), the main difference of our GraphMLP layer (Fig. 4(b)) is that we utilize GCNs for local feature communication. This modification retains the domain-specific knowledge of human body configurations, which induces an inductive bias enabling the GraphMLP to perform very well in skeletal representation learning.

Specifically, our GraphMLP layer is composed of an SG-MLP and a CG-MLP. The SG-MLP and CG-MLP are built by injecting GCNs into the spatial MLP and channel MLP (mentioned in Section 3.1), respectively. To be more specific, the SG-MLP contains a spatial MLP block and a GCN block. These blocks process token features in parallel, where the spatial MLP extracts features among tokens with a global receptive field, and the GCN block focuses on aggregating local information between neighboring joints. The CG-MLP has a similar architecture to SG-MLP but replaces the spatial MLP with the channel MLP and has no matrix transposition. Based on the above description, the MLP layers in Eqs. (2) and (3) are modified to process tokens as:

$$X'_{\ell} = X_{\ell-1} + \text{Spatial}-\text{MLP}(\text{LN}(X_{\ell-1}^{\top}))^{\top} + \text{GCN}(\text{LN}(X_{\ell-1}^{\top})^{\top}), \tag{4}$$

$$X_{\ell} = X'_{\ell} + \text{Channel}-\text{MLP}(\text{LN}(X'_{\ell})) + \text{GCN}(\text{LN}(X'_{\ell})), \tag{5}$$

where $\text{GCN}(\cdot)$ denotes the GCN block, $\ell \in [1, \ldots, L]$ is the index of GraphMLP layers. Here X'_{ℓ} and X_{ℓ} are the output features of the SG-MLP and the CG-MLP for block ℓ, respectively.

3.2.3. Prediction head

Different from [36,13] that use a classifier head to do classification, our prediction head performs regression with a linear layer. It is applied on the extracted features $X_L \in \mathbb{R}^{N\times C}$ of the last GraphMLP layer to predict the final 3D pose $\widetilde{X} \in \mathbb{R}^{N\times 3}$.

3.2.4. Loss function

To train our GraphMLP, we apply an L_2-norm loss to calculate the difference between prediction and ground truth. The model is trained in an end-to-end fashion, and the L_2-norm loss is defined as follows:

$$\mathcal{L} = \sum_{n=1}^{N} \left\| J_n - \widetilde{X}_n \right\|_2, \tag{6}$$

where $\widetilde{X}_n$ and J_n are the predicted and ground truth 3D coordinates of joint n, respectively.

3.3. Extension in the video domain

To extend our GraphMLP for capturing temporal information, we introduce a simple video representation that changes the skeleton embedding module in the original architecture. Specifically, given a 2D pose sequence $P \in \mathbb{R}^{T\times N\times 2}$ with T frames and N joints, we first concatenate features between the coordinates of x-y axis and all frames for each joint into $P' \in \mathbb{R}^{N\times(2\cdot T)}$, and then fed it into a linear layer

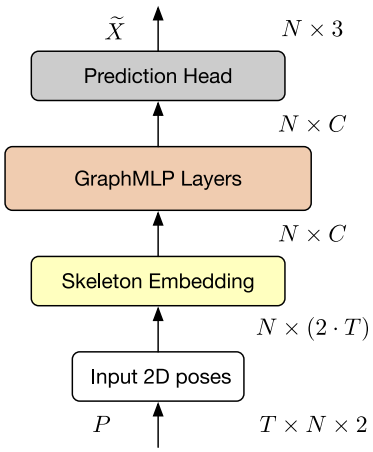


Fig. 5. Illustration of the process of GraphMLP in the video domain.

$W \in \mathbb{R}^{(2\cdot T)\times C}$ to map the tokens and get $X_0 \in \mathbb{R}^{N\times C}$. Subsequently, each joint is treated as an input token and fed into the GraphMLP layers and prediction head module to output the 3D pose of the center frame $\widetilde{X} \in \mathbb{R}^{N\times 3}$. These processes of GraphMLP in the video domain are illustrated in Fig. 5.

This is also a unified and flexible representation strategy that can process arbitrary-length sequences, *e.g.*, a single frame with $T = 1$, and variable-length videos with $T = 27, 81, 243$. Interestingly, we find that using a single linear layer to encode temporal information can achieve competitive performance without explicitly temporal modeling by treating each frame as a token. More importantly, this representation is efficient since the amount of N is small (*e.g.*, 17 joints). The increased computational costs from single frame to video sequence are only in one linear layer that weights are $W \in \mathbb{R}^{(2\cdot T)\times C}$ with linear computational complexity to sequence length, which can be neglected. Meanwhile, the computational overhead and the number of parameters in the GraphMLP layers and the prediction head module are the same for different input sequences.

4. Experiments

In this section, we first introduce experimental settings and implementation details for evaluation. Then, we compare the proposed GraphMLP with state-of-the-art methods. We also conduct detailed ablation studies on the importance of designs in our proposed approach.

4.1. Datasets and evaluation metrics

Human3.6M. Human3.6M [25] is the largest benchmark for 3D human pose estimation. It contains 3.6 million video frames captured by a motion capture system in an indoor environment, where 11 professional actors perform 15 actions such as greeting, phoning, and sitting. Following previous works [1,19,7], our model is trained on five subjects (S1, S5, S6, S7, S8) and tested on two subjects (S9, S11). We report our performance using two evaluation metrics. One is the mean per joint position error (MPJPE), referred to as Protocol #1, which calculates the mean Euclidean distance in millimeters between the predicted and the ground truth joint coordinates. The other is the PA-MPJPE that measures the MPJPE after Procrustes analysis (PA) [50] and is referred to as Protocol #2.

MPI-INF-3DHP. MPI-INF-3DHP [26] is a large-scale 3D human pose dataset containing both indoor and outdoor scenes. Its test set consists of three different scenes: studio with green screen (GS), studio without green screen (noGS), and outdoor scene (Outdoor). Following previous works [6,7,27], we use the Percentage of Correct Keypoint (PCK) with the threshold of 150 mm and Area Under Curve (AUC) for a range of PCK thresholds as evaluation metrics. To verify the generalization ability of our approach, we directly apply the model trained on Human3.6M to the test set of this dataset.

4.2. Implementation details

In our implementation, the proposed GraphMLP is stacked by $L = 3$ GraphMLP layers with hidden size $C = 512$, the MLP dimension of CG-MLP, *i.e.*, $D_C = 1024$, and the MLP dimension of SG-MLP, *i.e.*, $D_S = 256$. The whole framework is trained in an end-to-end fashion from scratch on a single NVIDIA RTX 2080 Ti GPU. The learning rate starts from 0.001 with a decay factor of 0.95 utilized in each epoch and 0.5 utilized per 5 epochs. We follow the generic data augmentation (horizontal flip augmentation) in [21,18,27]. Following [19,27,7], we use 2D joints detected by cascaded pyramid network (CPN) [51] for Human3.6M and 2D joints provided by the dataset for MPI-INF-3DHP.

4.3. Comparison with state-of-the-art methods

Comparison with Single-frame Methods. Table 1 reports the performance comparison between our GraphMLP and previous state-of-the-art methods that take a single frame as input on Human3.6M. It can be seen that our approach reaches 49.2 mm in MPJPE and 38.6 mm in PA-MPJPE, which outperforms previous methods. Note that some works [18,27,53] adopt a pose refinement module to boost the performance further. Compared with them, GraphMLP achieves lower MPJPE (48.0 mm), surpassing MGCN [27] by a large margin of 1.4 mm error reduction (relative 3% improvements).

Due to the uncertainty of 2D detections, we also report results using ground truth 2D keypoints as input to explore the upper bound of the proposed approach. As shown in Table 2, our GraphMLP obtains substantially better performance when given precise 2D joint information and attains state-of-the-art performance, which indicates its effectiveness.

Table 3 further compares our GraphMLP against previous state-of-the-art single-frame methods on cross-dataset scenarios. We only train our model on the Human3.6M dataset and test it on the MPI-INF-3DHP dataset. The results show that our approach obtains the best results in all scenes and all metrics, consistently surpassing other methods. This verifies the strong generalization ability of our approach to unseen scenarios.

Comparison with Video-based Methods. As shown in Table 4, our method achieves outstanding performance against video-based methods in both CPN and GT inputs. The proposed method surpasses our baseline model (*i.e.*, MLP-Mixer [13]) by a large margin of 4.6 mm (10% improvements) with CPN inputs and 4.9 mm (14% improvements) with GT inputs. These results further demonstrate the effectiveness of our GraphMLP, which combines MLPs and GCNs to learn better skeleton representations. Compared with the most related work, Poseformer [22], a self-attention-based architecture, our GraphMLP, such a self-attention free architecture, improves the results from 31.3 mm to 30.3 mm with GT inputs (3% improvements). Besides, our simple video representation, which uses a single linear layer to encode temporal information instead of being designed explicitly for temporal enhancement like previous works [21,54,22], is also capable of performing well. This indicates that our method can alleviate the issue of video redundancy by compressing the video information into a single vector, leading to impressive results.

Table 5 further reports the comparison of parameters, FLOPs, and MPJPE with PoseFormer [22], MixSTE [55], STCFormer [56], and MotionAGFormer [57] in different input frames. Surprisingly, compared with PoseFormer, our method requires only 22% FLOPs (356M vs. 1625M) while achieving better performance. Although MixSTE, STCFormer, and MotionAGFormer achieve better performance with the 243-frame model, they respectively require 779× (356M vs. 277248M), 439× (356M vs. 156215M), and 437× (356M vs. 155634M) more FLOPs than ours, which leads to a time-consuming training process and difficulties in deployment. Note that our method offers negligible FLOPs gains in the sequence length, which allows it easy to deploy in real-time applications and has great potential for better results with longer sequences. These results demonstrate that our GraphMLP in video reaches competitive performance with fewer computational costs and can serve as a strong baseline for video-based 3D human pose estimation.

Table 1
Quantitative comparison with state-of-the-art single-frame methods on Human3.6M under Protocol #1 and Protocol #2. Detected 2D keypoints are used as input.

Protocol #1	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
FCN [1] ICCV'17	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
TCN [21] CVPR'19	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8
ST-GCN [18] ICCV'19§	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	53.5	39.2	41.2	50.6
SRNet [34] ECCV'20	44.5	48.2	47.1	47.8	51.2	56.8	50.1	45.6	59.9	66.4	52.1	45.3	54.2	39.1	40.3	49.9
GraphSH [19] CVPR'21	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
MGCN [27] ICCV'21§	45.4	49.2	45.7	49.4	50.4	58.2	47.9	46.0	57.5	63.0	49.7	46.6	52.2	38.9	40.8	49.4
GraFormer [10] CVPR'22	45.2	50.8	48.0	50.0	54.9	65.0	48.2	47.1	60.2	70.0	51.6	48.7	54.1	39.7	43.1	51.8
UGRN [52] AAAI'23	47.9	50.0	47.1	51.3	51.2	59.5	48.7	46.9	56.0	61.9	51.1	48.9	54.3	40.0	42.9	50.5
RS-Net [53] TIP'23§	44.7	48.4	44.8	49.7	49.6	58.2	47.4	44.8	55.2	59.7	49.3	46.4	51.4	38.6	40.6	48.6
GraphMLP (Ours)	45.4	50.2	45.8	49.2	51.6	57.9	47.3	44.9	56.9	61.0	49.5	46.9	53.2	37.8	39.9	49.2
GraphMLP (Ours)§	43.7	49.3	45.5	47.9	50.5	56.0	46.3	44.1	55.9	59.0	48.4	45.7	51.2	37.1	39.1	**48.0**
Protocol #2	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
FCN [1] ICCV'17	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
TCN [21] CVPR'19	36.0	38.7	38.0	41.7	40.1	45.9	37.1	35.4	46.8	53.4	41.4	36.9	43.1	30.3	34.8	40.0
ST-GCN [18] ICCV'19§	36.8	38.7	38.2	41.7	40.7	46.8	37.9	35.6	47.6	51.7	41.3	36.8	42.7	31.0	34.7	40.2
SRNet [34] ECCV'20	35.8	39.2	36.6	36.9	39.8	45.1	38.4	36.9	47.7	54.4	38.6	36.3	39.4	30.3	35.4	39.4
MGCN [27] ICCV'21§	35.7	38.6	36.3	40.5	39.2	44.5	37.0	35.4	46.4	51.2	40.5	35.6	41.7	30.7	33.9	39.1
SGNN [7] ICCV'21	33.9	37.2	36.8	38.1	38.7	43.5	37.8	35.0	47.2	53.8	40.7	38.3	41.8	30.1	31.4	39.0
RS-Net [53] TIP'23§	35.5	38.3	36.1	40.5	39.2	44.8	37.1	34.9	45.0	49.1	40.2	35.4	41.5	31.0	34.3	38.9
GraphMLP (Ours)	35.0	38.4	36.6	39.7	40.1	43.9	35.9	34.1	45.9	48.6	40.0	35.3	41.6	30.0	33.3	38.6
GraphMLP (Ours)§	35.1	38.2	36.5	39.8	39.8	43.5	35.7	34.0	45.6	47.6	39.8	35.1	41.1	30.0	33.4	**38.4**

§ - adopts the same refinement module as [18,27,53].

Table 2
Quantitative comparison with state-of-the-art single-frame methods on Human3.6M under Protocol #1. Ground truth 2D keypoints are used as input.

Protocol #1	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
FCN [1] ICCV'17	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
SemGCN [8] CVPR'19	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
ST-GCN [18] ICCV'19	33.4	39.0	33.8	37.0	38.1	47.3	39.5	37.3	43.2	46.2	37.7	38.0	38.6	30.4	32.1	38.1
SRNet [34] ECCV'20	35.9	36.7	29.3	34.5	36.0	42.8	37.7	31.7	40.1	44.3	35.8	37.2	36.2	33.7	34.0	36.4
GraphSH [19] CVPR'21	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
GraFormer [10] CVPR'22	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
PHGANet [11] IJCV'23	32.4	36.5	30.1	33.3	36.3	43.5	36.1	30.5	37.5	45.3	33.8	35.1	35.3	27.5	30.2	34.9
GraphMLP (Ours)	32.2	38.2	29.3	33.4	33.5	38.1	38.2	31.7	37.3	38.5	34.2	36.1	35.5	28.0	29.3	**34.2**

Table 3
Performance comparison with state-of-the-art single-frame methods on MPI-INF-3DHP.

Method	GS ↑	noGS ↑	Outdoor ↑	All PCK ↑	All AUC ↑
FCN [1] ICCV'17	49.8	42.5	31.2	42.5	17.0
LCN [6] ICCV'19	74.8	70.8	77.3	74.0	36.7
SGNN [7] ICCV'21	–	–	84.6	82.1	46.2
MGCN [27] ICCV'21	86.4	86.0	85.7	86.1	53.7
GraFormer [10] CVPR'22	80.1	77.9	74.1	79.0	43.8
UGRN [52] AAAI'23	86.2	84.7	81.9	84.1	53.7
RS-Net [53] TIP'23	–	–	–	85.6	53.2
GraphMLP (Ours)	**87.3**	**87.1**	**86.3**	**87.0**	**54.3**

Table 4
Quantitative comparisons with video-based methods on Human3.6M under MPJPE. CPN and GT denote the inputs of 2D poses detected by CPN and ground truth 2D poses, respectively.

Method	CPN	GT
ST-GCN [18] ($T = 9$)	48.8	37.2
TCN [21] ($T = 243$)	46.8	37.8
SRNet [34] ($T = 243$)	44.8	32.0
PoseFormer [22] ($T = 81$)	44.3	31.3
Anatomy3D [54] ($T = 243$)	44.1	32.3
Baseline ($T = 243$)	48.4	35.2
GraphMLP (Ours, $T = 243$)	**43.8**	**30.3**

Table 5
Comparison of parameters, FLOPs, and MPJPE with PoseFormer [22], MixSTE [55], STCFormer [56], and MotionAGFormer [57] in different input frames on Human3.6M. Frame per second (FPS) was computed on a single GeForce RTX 3090 GPU.

Model	T	Param (M)	FLOPs (M)	MPJPE (mm) ↓
PoseFormer [22]	1	9.56	20	51.3
PoseFormer [22]	27	9.57	541	47.0
PoseFormer [22]	81	9.60	1625	44.3
PoseFormer [22]	243	9.69	4874	–
MixSTE [55]	1	33.66	1141	51.1
MixSTE [55]	27	33.67	30805	45.1
MixSTE [55]	81	33.70	92416	42.4
MixSTE [55]	243	33.78	277248	40.9
STCFormer [56]	1	–	–	–
STCFormer [56]	27	4.75	4347	44.1
STCFormer [56]	81	4.75	13041	42.0
STCFormer [56]	243	18.93	156215	40.5
MotionAGFormer [57]	1	–	–	–
MotionAGFormer [57]	27	2.24	2023	45.1
MotionAGFormer [57]	81	4.82	13036	42.5
MotionAGFormer [57]	243	19.01	155634	38.4
GraphMLP (Ours)	1	9.49	348	49.2
GraphMLP (Ours)	27	9.51	349	45.5
GraphMLP (Ours)	81	9.57	351	44.5
GraphMLP (Ours)	243	9.73	356	43.8

4.4. Ablation study

The large-scale ablation studies with 2D detected inputs on the Human3.6M dataset are conducted to investigate the effectiveness of our model (using the single-frame model).

Model Configurations. We start our ablation studies by exploring the GraphMLP on different hyper-parameters. The results are shown in Table 6. It can be observed that using the expanding ratio of 2 ($C = 512$, $D_C = 2C = 1024$) works better than the ratio value of 4 which is common in vision Transformers and MLPs. Increasing or

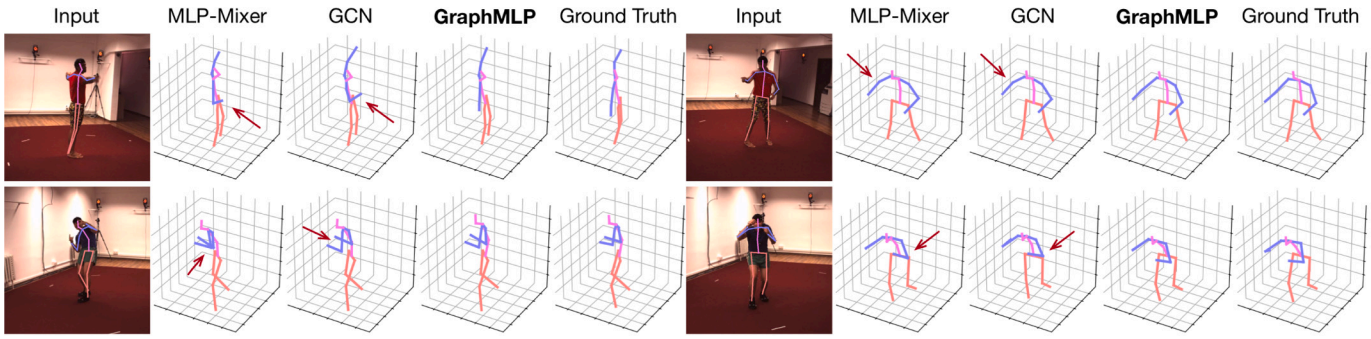


Fig. 6. Qualitative comparison with MLP-Mixer [13] and GCN [18] for reconstructing 3D human poses on Human3.6M dataset. Red arrows highlight wrong estimations.

Table 6
Ablation study on various configurations of our approach. L is the number of GraphMLP layers, C is the hidden size, D_C is the MLP dimension of CG-MLP, and D_S is the MLP dimension of SG-MLP.

L	C	D_C	D_S	Params (M)	MPJPE (mm) ↓
3	512	1024	256	9.49	**49.2**
3	512	1024	128	9.47	49.7
3	512	2048	128	12.62	49.7
3	512	2048	256	12.64	49.6
2	512	1024	256	6.33	50.2
4	512	1024	256	12.65	50.0
5	512	1024	256	15.81	50.3
6	512	1024	256	18.97	50.4
3	128	256	64	0.60	50.2
3	256	512	128	2.38	50.1
3	384	768	192	5.35	49.7
3	768	1536	384	21.31	49.4

Table 7
Ablation study on different 2D detectors.

Method	2D Mean Error	MPJPE (mm) ↓
SH [58]	9.03	56.9
Detectron [21]	7.77	54.5
CPN [51]	6.67	49.2
2D Ground Truth	0	**34.2**

Table 8
Ablation study on different network architectures.

Method	MPJPE (mm) ↓
FCN [1]	53.5
GCN [18]	51.3
Transformer [60]	52.7
MLP-Mixer [13]	52.0
Mesh Graphormer [20]	51.3
Graph-Mixer	50.4
Transformer-GCN	51.5
GraphMLP	**49.2**

Table 9
Ablation study on different design options of combining MLPs and GCNs.

Method	MPJPE (mm) ↓
Baseline	52.0
GCN Before Spatial MLP	51.6
GCN After Spatial MLP	50.6
GCN After Channel MLP	50.9
GCN and MLP in Parallel*	50.2
GCN and MLP in Parallel	**49.2**

* means using spatial GCN block in SG-MLP.

Table 10
Ablation study on different components in GraphMLP.

Method	SG-MLP	CG-MLP	MPJPE (mm) ↓
Baseline	✗	✗	52.0
GraphMLP (SG-MLP)	✓	✗	50.6
GraphMLP (CG-MLP)	✗	✓	50.5
GraphMLP	✓	✓	**49.2**

reducing the number of GraphMLP layers L hurts performance while using $L = 3$ performs best. The hidden size C is important to determine the modeling ability. While increasing the C from 128 to 512 (keeping the same MLP ratios), the MPJPE decreases from 50.2 mm to 49.2 mm. Meanwhile, the number of parameters increases from 0.60M to 9.49M. The performance saturates when C surpasses 512. Therefore, the optimal hyper-parameters for our model are $L = 3$, C=512, $D_C = 1024$, and $D_S = 256$, which are different from the original setting of MLP-Mixer [13]. The reason for this may be the gap between vision and skeleton data, where the Human3.6M dataset is not diverse enough to train a large GraphMLP model.

Input 2D Detectors. A high-performance 2D detector is vital in achieving accurate 3D pose estimation. Table 7 reports the performance of our model with ground truth 2D joints, and detected 2D joints from Stack Hourglass (SH) [58], Detectron [21], CPN [51], and HRNet [59]. We can observe that our approach can produce more accurate results with a better 2D detector and is effective on different 2D estimators.

Network Architectures. To clearly demonstrate the advantage of the proposed GraphMLP, we compare our approach with various baseline architectures. The 'Graph-Mixer' is constructed by replacing the linear layers in MLP-Mixer with GCN layers, and the 'Transformer-GCN' is built by replacing the spatial MLPs in GraphMLP with multi-head self-attention blocks and adding a position embedding module. To ensure a consistent and fair evaluation, the parameters of these architectures (*e.g.*, the number of layers, hidden size) remain the same. As shown in Table 8, our proposed approach consistently surpasses all other architectures. For example, our approach can improve the GCN-based model [18] from 51.3 mm to 49.2 mm, resulting in relative 4.1% improvements. It is worth noting that the MLP-Like model ('MLP-Mixer', 'GraphMLP') outperforms the Transformer-based model ('Transformer', 'Transformer-GCN'). This may be because a small number of tokens (*e.g.*, 17 joints) are less effective for self-attention in learning long-range dependencies. Overall, these results confirm that GraphMLP can serve as a new and strong baseline for 3D human pose estimation. The implementation details of these network architectures can be found in the supplementary.

Network Design Options. Our approach allows the model to learn strong structural priors of human joints by injecting GCNs into the baseline model (*i.e.*, MLP-Mixer [13]). We also explore the different combinations of MLPs and GCNs to find an optimal architecture. As shown in Table 9, the location matters of GCN are studied by five different designs: (i) The GCN is placed before spatial MLP. (ii) The GCN is placed after spatial MLP. (iii) The GCN is placed after channel MLP. (iv) The GCN and MLP are in parallel but using a spatial GCN

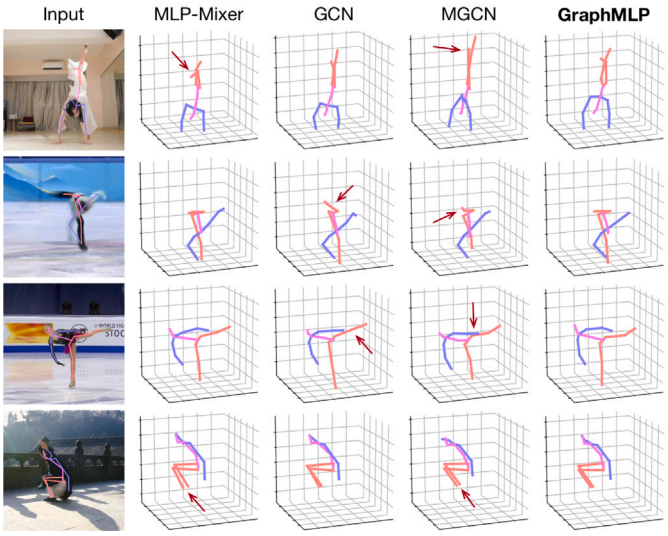


Fig. 7. Qualitative comparison with MLP-Mixer [13], GCN [18], and MGCN [27] for reconstructing 3D human poses on challenging in-the-wild images. Red arrows highlight wrong estimations.

block (process tokens in spatial dimension) in SG-MLP. (v) The GCN and MLP are in parallel. The results show that all these designs can help the model produce more accurate 3D poses, and using GCN and MLP in parallel achieves the best estimation accuracy. The illustrations of these design options can be found in the supplementary.

Model Components. We also investigate the effectiveness of each component in our design. In Table 10, the first row corresponds to the baseline model (*i.e.*, MLP-Mixer [13]) that does not use any GCNs in the model. The MPJPE is 52.0 mm. The rest of the rows show the results of replacing its spatial MLP with SG-MLP or channel MLP with CG-MLP by adding a GCN block into the baseline. It can be found that using our SG-MLP or CG-MLP can improve performance (50.6 mm and 50.5 mm respectively). When enabling both SG-MLP and CG-MLP, GraphMLP improves the performance over the baseline by a clear margin of 2.8 mm (relatively 5.4% improvements), indicating that the proposed components are mutually reinforced to produce more accurate 3D poses. These results validate the effectiveness of our motivation: combining modern MLPs and GCNs in a unified architecture for better 3D human pose estimation.

5. Qualitative results

Fig. 6 presents the qualitative comparison among the proposed GraphMLP, MLP-Mixer [13], and GCN [18] on the Human3.6M dataset. Furthermore, Fig. 7 presents the qualitative comparison among our GraphMLP and three other methods, namely MLP-Mixer [13], GCN [18], and MGCN [27], on more challenging in-the-wild images. Note that these actions from in-the-wild images are rare or absent from the training set of Human3.6M. GraphMLP, benefiting from its globality, locality, and connectivity, performs better and is able to predict accurate and plausible 3D poses. It indicates both the effectiveness and generalization ability of our approach. However, there are still some failure cases, where our approach fails to produce accurate 3D human poses due to large 2D detection error, half body, rare poses, and heavy occlusion, as shown in Fig. 8. More qualitative results can be found in the supplementary.

6. Conclusion and future works

In this paper, we propose a new graph-reinforced MLP-Like architecture, termed GraphMLP, which represents the first use of modern MLP

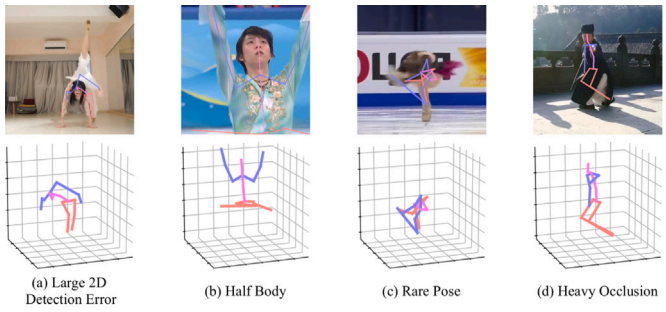


Fig. 8. Challenging scenarios where GraphMLP fails to produce accurate 3D human poses.

dedicated to 3D human pose estimation. Our GraphMLP inherits the advantages of both MLPs and GCNs, making it a *global-local-graphical* unified architecture without self-attention. Extensive experiments show that the proposed GraphMLP achieves state-of-the-art performance and can serve as a simple yet effective modern baseline for 3D human pose estimation in both single frame and video sequence. We also show that complex temporal dynamics can be effectively modeled in a simple way without explicitly temporal modeling, and the proposed GraphMLP in video reaches competitive results with fewer computational costs.

Although the proposed approach has shown promising results, its performance is still limited by the quality of available datasets. High-quality 3D annotations rely on motion capture systems and are challenging and expensive to obtain. The most commonly used dataset for 3D human pose estimation, Human3.6M, lacks sufficient variations in terms of human poses, environments, and activities, which leads to models that do not generalize well to real-world scenarios. Domain adaptation and synthetic data are potential directions to improve the generalization ability of our approach. Moreover, since the MLPs and GCNs in our GraphMLP are straightforward, we look forward to incorporating more powerful MLPs or GCNs to further improve performance.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62373009), Natural Science Foundation of Guangdong Province (No. 2024A1515012089), Natural Science Foundation of Shenzhen (No. JCYJ20230807120801002), Shenzhen Innovation in Science and Technology Foundation for The Excellent Youth Scholars (No. RCYX20231211090248064), the Fundamental Research Funds for the Central Universities, Peking University, the MUR PNRR project FAIR (PE00000013) funded by the NextGenerationEU, and the EU Horizon project ELIAS (No. 101120237).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patcog.2024.110925.

References

[1] 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.

[2] G. Wei, C. Lan, W. Zeng, Z. Chen, View invariant 3D human pose estimation, IEEE Trans. Circuits Syst. Video Technol. 30 (12) (2019) 4601–4610.

[3] H.-W. Kim, G.-H. Lee, W.-J. Nam, K.-M. Jin, T.-K. Kang, G.-J. Yang, S.-W. Lee, MHCanonNet: Multi-hypothesis canonical lifting network for self-supervised 3D human pose estimation in the wild video, Pattern Recognit. (2023) 109908.

[4] G. Hua, H. Liu, W. Li, Q. Zhang, R. Ding, X. Xu, Weakly-supervised 3D human pose estimation with cross-view U-shaped graph convolutional network, IEEE Trans. Multimed. 25 (2022) 1832–1843.

[5] C. Wang, Y. Wang, Z. Lin, A.L. Yuille, W. Gao, Robust estimation of 3D human poses from a single image, in: CVPR, 2014, pp. 2361–2368.

[6] H. Ci, C. Wang, X. Ma, Y. Wang, Optimizing network structure for 3D human pose estimation, in: ICCV, 2019, pp. 2262–2271.

[7] A. Zeng, X. Sun, L. Yang, N. Zhao, M. Liu, Q. Xu, Learning skeletal graph neural networks for hard 3D pose estimation, in: ICCV, 2021, pp. 11436–11445.

[8] 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.

[9] 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.

[10] W. Zhao, W. Wang, Y. Tian, GraFormer: Graph-oriented transformer for 3D pose estimation, in: CVPR, 2022, pp. 20438–20447.

[11] S. Zhang, C. Wang, L. Nie, H. Yao, Q. Huang, Q. Tian, Learning enriched hop-aware correlation for robust 3D human pose estimation, Int. J. Comput. Vis. 131 (6) (2023) 1566–1583.

[12] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: AAAI, 2018.

[13] I.O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, et al., MLP-Mixer: An all-MLP architecture for vision, in: NeurIPS, vol. 34, 2021, pp. 24261–24272.

[14] D. Lian, Z. Yu, X. Sun, S. Gao, AS-MLP: An axial shifted MLP architecture for vision, in: ICLR, 2022.

[15] H. Liu, Z. Dai, D. So, Q.V. Le, Pay attention to MLPs, in: NeurIPS, vol. 34, 2021, pp. 9204–9215.

[16] B. Ren, H. Tang, N. Sebe, Cascaded cross MLP-mixer GANs for cross-view image translation, in: BMVC, 2021.

[17] S. Chen, E. Xie, C. GE, R. Chen, D. Liang, P. Luo, CycleMLP: A MLP-like architecture for dense prediction, in: ICLR, 2022.

[18] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, N.M. Thalmann, Exploiting spatial-temporal relationships for 3D pose estimation via graph convolutional networks, in: ICCV, 2019, pp. 2272–2281.

[19] T. Xu, W. Takano, Graph stacked hourglass networks for 3D human pose estimation, in: CVPR, 2021, pp. 16105–16114.

[20] K. Lin, L. Wang, Z. Liu, Mesh graphormer, in: ICCV, 2021, pp. 12939–12948.

[21] D. Pavllo, C. Feichtenhofer, D. Grangier, M. Auli, 3D human pose estimation in video with temporal convolutions and semi-supervised training, in: CVPR, 2019, pp. 7753–7762.

[22] 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.

[23] W. Li, H. Liu, H. Tang, P. Wang, L. Van Gool, MHFormer: Multi-hypothesis transformer for 3D human pose estimation, in: CVPR, 2022, pp. 13147–13156.

[24] W. Li, H. Liu, R. Ding, M. Liu, P. Wang, W. Yang, Exploiting temporal contexts with strided transformer for 3D human pose estimation, IEEE Trans. Multimed. 25 (2022) 1282–1293.

[25] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2013) 1325–1339.

[26] 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.

[27] Z. Zou, W. Tang, Modulated graph convolutional network for 3D human pose estimation, in: ICCV, 2021, pp. 11477–11487.

[28] S. Li, A.B. Chan, 3D human pose estimation from monocular images with deep convolutional neural network, in: ACCV, 2014, pp. 332–347.

[29] 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.

[30] Z. Qiu, K. Qiu, J. Fu, D. Fu, Weakly-supervised pre-training for 3D human pose estimation via perspective knowledge, Pattern Recognit. 139 (2023) 109497.

[31] C. Han, X. Yu, C. Gao, N. Sang, Y. Yang, Single image based 3D human pose estimation via uncertainty learning, Pattern Recognit. 132 (2022) 108934.

[32] W. Li, M. Liu, H. Liu, P. Wang, J. Cai, N. Sebe, Hourglass tokenizer for efficient transformer-based 3D human pose estimation, in: CVPR, 2024, pp. 604–613.

[33] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, V. Asari, Attention mechanism exploits temporal contexts: Real-time 3D human pose reconstruction, in: CVPR, 2020, pp. 5064–5073.

[34] 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.

[35] H. Liu, J.-Y. He, Z.-Q. Cheng, W. Xiang, Q. Yang, W. Chai, G. Wang, X. Bao, B. Luo, Y. Geng, et al., PosynDA: Multi-hypothesis pose synthesis domain adaptation for robust 3D human pose estimation, in: ACM MM, 2023, pp. 5542–5551.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: ICLR, 2021.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: ICCV, 2021, pp. 10012–10022.

[38] Q. Wang, J. Wang, H. Deng, X. Wu, Y. Wang, G. Hao, AA-trans: Core attention aggregating transformer with information entropy selector for fine-grained visual classification, Pattern Recognit. 140 (2023) 109547.

[39] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: ECCV, 2020, pp. 213–229.

[40] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable DETR: Deformable transformers for end-to-end object detection, in: ICLR, 2021.

[41] Z. Zhao, Z. Wu, Y. Zhuang, B. Li, J. Jia, Tracking objects as pixel-wise distributions, in: ECCV, 2022.

[42] H. Chen, J.-Y. He, W. Xiang, Z.-Q. Cheng, W. Liu, H. Liu, B. Luo, Y. Geng, X. Xie, HDFormer: High-order directed transformer for 3D human pose estimation, in: IJCAI, 2023, pp. 581–589.

[43] W. Li, H. Liu, H. Tang, P. Wang, Multi-hypothesis representation learning for transformer-based 3D human pose estimation, Pattern Recognit. 141 (2023) 109631.

[44] H. Li, W. Xiang, J.-Y. He, Z.-Q. Cheng, B. Luo, Y. Geng, X. Xie, Refined temporal pyramidal compression-and-amplification transformer for 3D human pose estimation, 2023, arXiv preprint arXiv:2309.01365.

[45] J.-H. Shi, Q. Zhang, Y.-H. Tang, Z.-Q. Zhang, Polyp-mixer: An efficient context-aware MLP-based paradigm for polyp segmentation, IEEE Trans. Circuits Syst. Video Technol. 33 (1) (2022) 30–42.

[46] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, 2016, arXiv preprint arXiv:1607.06450.

[47] D. Hendrycks, K. Gimpel, Gaussian error linear units (GELUs), 2016, arXiv preprint arXiv:1606.08415.

[48] H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, et al., ResMLP: Feedforward networks for image classification with data-efficient training, 2021, arXiv preprint arXiv:2105.03404.

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

[50] J.C. Gower, Generalized procrustes analysis, Psychometrika 40 (1) (1975) 33–51.

[51] 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.

[52] H. Li, B. Shi, W. Dai, H. Zheng, B. Wang, Y. Sun, M. Guo, C. Li, J. Zou, H. Xiong, Pose-oriented transformer with uncertainty-guided refinement for 2D-to-3D human pose estimation, in: AAAI, 2023.

[53] M.T. Hassan, A.B. Hamza, Regular splitting graph network for 3D human pose estimation, IEEE Trans. Image Process. (2023).

[54] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, J. Luo, Anatomy-aware 3D human pose estimation with bone-based pose decomposition, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2021) 198–209.

[55] 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.

[56] Z. Tang, Z. Qiu, Y. Hao, R. Hong, T. Yao, 3D human pose estimation with spatio-temporal criss-cross attention, in: CVPR, 2023, pp. 4790–4799.

[57] S. Mehraban, V. Adeli, B. Taati, MotionAGFormer: Enhancing 3D human pose estimation with a transformer-gcnformer network, in: WACV, 2024, pp. 6920–6930.

[58] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: ECCV, 2016, pp. 483–499.

[59] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: CVPR, 2019, pp. 5693–5703.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.



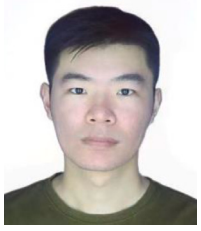
Wenhao Li received the B.E. degree in automation in 2019. He is currently working toward the Ph. D. degree in the School of Computer Science, Peking University (PKU), China, under the supervision of Prof. Hong Liu. His research interests lie in deep learning, machine learning, and their applications to computer vision.



Mengyuan Liu is an Assistant Professor at Peking University. He received a Ph.D. degree in 2017 under the supervision of Prof. H. Liu from the School of EE&CS, Peking University (PKU), China. His research interests include human action recognition, human motion prediction, and human motion generation using RGB, depth, and skeleton data. Related methods have been published in T-IP, T-CSVT, T- MM, PR, CVPR, ECCV, ACM MM, AAAI, and IJCAI. He has been invited to be a Technical Program Committee (TPC) member for ACPR, ACM MM, and AAAI.



Hong Liu received the Ph.D. degree in Mechanical Electronics and Automation in 1996, and serves as a Full Professor in the School of EE&CS, Peking University (PKU), China. Prof. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Pla" since 2013. His research interests include computer vision, pattern recognition and robot motion planning. He is also reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing, and IEEE Trans. on PAMI.



Tianyu Guo received the B.S. degree in electronic and information engineering in 2020. He is currently a graduate student in the School of Electronic and Computer Engineering, Peking University (PKU), China, under the supervision of Prof. H. Liu. His research interests include computer vision, machine learning, and action recognition.



Ti Wang is a master student in the School of Electronic and Computer Engineering, Peking University (PKU) , China, under the supervision of Prof. H. Liu. He received the B.E. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology in 2021. His current research mainly focuses on 3D human pose and shape estimation.



Hao Tang is an Assistant Professor at Peking University, China. Previously, he held postdoctoral positions at both CMU, USA, and ETH Zürich, Switzerland. He earned a master's degree from Peking University, China, and a Ph.D. from the University of Trento, Italy. He was a visiting Ph.D. student at the University of Oxford, UK, and an intern at IIAI, UAE. His research interests include AIGC, AI4Science, machine learning, computer vision, embodied AI, and their applications to scientific domains.



Nicu Sebe is a Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013 and 2022. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a fellow of IAPR and of ELLIS.