Skeleton-Aware Graph-Based Adversarial Networks for Human Pose Estimation from Sparse IMUs

KAIXIN CHEN, LIN ZHANG, ZHONG WANG, and SHENGJIE ZHAO, School of Software Engineering, Tongji University, Shanghai, China
YICONG ZHOU, Department of Computer and Information Science, University of Macau, Macau, China

Recently, sparse-inertial human pose estimation (SI-HPE) with only a few IMUs has shown great potential in various fields. The most advanced work in this area achieved fairish results using only six IMUs. However, there are still two major issues that remain to be addressed. First, existing methods typically treat SI-HPE as a temporal sequential learning problem and often ignore the important spatial prior of skeletal topology. Second, there are far more synthetic data in their training data than real data, and the data distribution of synthetic data and real data is quite different, which makes it difficult for the model to be applied to more diverse real data. To address these issues, we propose "Graph-based Adversarial Inertial Poser (GAIP)," which tracks body movements using sparse data from six IMUs. To make full use of the spatial prior, we design a multi-stage pose regressor with graph convolution to explicitly learn the skeletal topology. A joint position loss is also introduced to implicitly mine spatial information. To enhance the generalization ability, we propose supervising the pose regression with an adversarial loss from a discriminator, bringing the ability of adversarial networks to learn implicit constraints into full play. Additionally, we construct a real dataset that includes hip support movements and a synthetic dataset containing various motion categories to enrich the diversity of inertial data for SI-HPE. Extensive experiments demonstrate that GAIP produces results with more precise limb movement amplitudes and relative joint positions, accompanied by smaller joint angle and position errors compared to state-of-the-art counterparts. The datasets and codes are publicly available at https://cslinzhang.github.io/GAIP/.

CCS Concepts: • **Computing methodologies** → **Motion capture**;

Additional Key Words and Phrases: IMU, human pose estimation, graph convolutional network, GRU, adversarial training

This work was supported in part by the National Natural Science Foundation of China under Grant 62272343, Grant 61973235, and Grant 61936014; in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400; in part by the Shuguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 21SG23; and in part by the Fundamental Research Funds for the Central Universities.
Authors' Contact Information: Kaixin Chen, School of Software Engineering, Tongji University, Shanghai, China; e-mail: 2131506@tongji.edu.cn; Lin Zhang (Corresponding author), School of Software Engineering, Tongji University, Shanghai, China; e-mail: cslinzhang@tongji.edu.cn; Zhong Wang, School of Software Engineering, Tongji University, Shanghai, China; e-mail: 2010194@tongji.edu.cn; Shengjie Zhao, School of Software Engineering, Tongji University, Shanghai, China; e-mail: shengjiezhao@tongji.edu.cn; Yicong Zhou, Department of Computer and Information Science, University of Macau, Macau, China; e-mail: yicongzhou@um.edu.mo.

ACM 1551-6865/2025/3-ART106
https://doi.org/10.1145/3669904

ACM Reference format:
Kaixin Chen, Lin Zhang, Zhong Wang, Shengjie Zhao, and Yicong Zhou. 2025. Skeleton-Aware Graph-Based Adversarial Networks for Human Pose Estimation from Sparse IMUs. *ACM Trans. Multimedia Comput. Commun. Appl.* 21, 4, Article 106 (March 2025), 22 pages.
https://doi.org/10.1145/3669904

1 Introduction

Three dimensional (3D) human pose estimation (HPE) has been a fundamental and active research area due to its great potential in various fields, such as autonomous driving [10, 18], video surveillance [44], motion recognition or localization [12, 22], and even sign language translation [13, 14, 38]. Most existing methods for 3D HPE are built from vision data. However, those vision-based approaches require expensive infrastructure and invasive devices, hindering their application and promotion. Moreover, they are easily subject to environmental factors such as poor lighting conditions and occlusions, making it difficult to stably capture human movements. They are also prone to flickering artifacts or motion blur and distortion in the captured pictures or videos due to the photon noise introduced by photographic devices [37]. These introduced factors further constraining vision-based methods' efficacy. Compared with optical cameras, **inertial measurement units (IMUs)** as active sensors can capture high-frequency carrier motion states (orientations and accelerations) and are gradually becoming promising sensors for HPE. As a result, recent years have witnessed a growing effort to explore HPE using only a few IMUs [16, 43], opening a new field, **sparse-inertial HPE (SI-HPE)**, with great potential and challenges.

As a recently emerging research field, to our knowledge, the studies on SI-HPE are still quite sporadic. The concept of SI-HPE can be traced back to **Sparse Inertial Poser (SIP)** [35], which shows that it is feasible to learn and estimate human pose with only six IMUs' measurements of the five leaf joints (the left/right lower legs, the left/right forearms, and the head) and the root joint (the pelvis). Subsequently, Huang et al. proposed **Deep Inertial Poser (DIP)** [16], which employs a bidirectional **Recurrent Neural Network (RNN)** to learn the mapping from acceleration and orientation readings to joint rotations. As a pioneer, DIP achieves fair results by data-driven learning of the temporal prior of motion sequences. Based on DIP, recent works [17, 43] optimize its network structure and loss function to improve the accuracy of pose estimation while focusing on improving real-time performance. However, both DIP and its variants, such as Transpose [43], still have the following two common problems remaining to be solved.

Firstly, there is a lack of topological modeling for the skeleton in SI-HPE. Due to the sparseness of the inputs, accurate pose estimation relies heavily on prior information. Previous methods focus on temporal priors of human motions, treating HPE as a sequence-to-sequence learning problem and simply forming feature vectors with joint coordinates and joint rotations at each time step. However, using only temporal priors for pose estimation is undoubtedly limited, as it does not explicitly mine the topological relationships among joints in the **human kinematic tree (HKT)**, the effectiveness of which has been demonstrated in many vision-based methods. Since the low-level inputs of SI-HPE are very sparse and has less effective information than the vision-based approaches, we believe that this topological prior can be beneficial for understanding human behavior. Unfortunately, none of the existing SI-HPE methods have explored the incorporation of spatial priors of the joint topology.

Secondly, existing methods are difficult to generalize to other datasets. Both Transpose and DIP follow the same training procedure of first pre-training on a simulated dataset and subsequently fine-tuning on a small-scale real dataset. Their results indicate two common characteristics: (1) the outputs of the pre-trained model on the simulated dataset are significantly better than those on real

datasets and (2) the performance of the fine-tuned model on different test sets is different. These phenomena indicate that the data distribution between the synthetic dataset used for pre-training and the real dataset used for fine-tuning is quite different, and the generalization ability of the model relies on fine-tuning on small datasets. Once the model needs to process more diverse human poses, it becomes difficult to perform good estimations.

Based on the above analysis, we propose a **graph-based adversarial inertial poser (GAIP)** for short, which takes the following two measures to compensate for the previously identified deficiencies to some extent.

To make full use of skeleton topology priors, we design a **graph convolutional network (GCN)**-based regressor to explicitly learn the spatial information of HKTs. Specifically, our regressor is mainly composed of three structurally alike sub-networks, the inputs of which mainly consist of two types of data: inertial measurements and joint positions. Both types of data are related to joint connectivity and contain important spatial priors. To model human skeletal sequences on a large-scale dataset and explicitly learn the topological information of HKT, **spatial GC (s-GC)** modules are embedded into each sub-network. Furthermore, we introduce a joint position cost term to the regressor's loss function, which distinguishes our method from similar ones that only regard rotations as feature vectors. This cost term involves forward kinematics of the body and can implicitly learn the spatial information of human joints.

Regarding the enhancement and evaluation of generalization ability, we conduct studies that encompass training strategies and dataset diversity. For one aspect, given that **generative adversarial networks (GANs)** have the ability to explore the implicit features of input data [8], we attempt to improve the network framework by incorporating the adversarial loss to learn implicit constraints of human poses and suppress the model's tendency to overfit to specific data. Specifically, in GAIP, we construct a discriminator to judge the difference between the regressed poses and the ground truth. During training, the discriminator receives the predicted and real pose sequences and outputs an adversarial loss, which is input into the aforementioned regressor to further constrain the estimated results. With this strategy, the discriminator achieves synchronized training with the regressor and encourages it to learn the implicit features of the human poses.

For another aspect, we construct new datasets to comprehensively evaluate the generalization ability. At present, the widely used real inertial dataset for SI-HPE is DIP-IMU [16]. Although DIP-IMU contains multiple subjects' inertial measurements, it mainly includes basic standing motions and interactions. Therefore, we propose two additional datasets including more diverse motion sequences for SI-HPE. One is SingleOne-IMU, which contains about 60 minutes of real inertial data. Unlike DIP-IMU, which regards the pose parameters estimated by SIP [35] as ground truth, ours instead obtains joint rotations using Noitom's[1] commercial solution of inertial **motion capture (MoCap)** as shown in Figure 1. Along with common foot support actions, we also include a series of hip support actions to provide more diverse motion categories. Another self-collected dataset is a synthesized dataset created from Mixamo,[2] named Mixamo-IMU. It includes various dances, climbs, and diverse poses and can serve as another benchmark for evaluating the generalization abilities of SI-HPE approaches.

We evaluate GAIP on TotalCapture [32], DIP-IMU, as well as our self-collected SingleOne-IMU and Mixamo-IMU. In qualitative evaluation, GAIP yields results with more precise limb movement amplitudes and relative joint positions. In quantitative experiments, the poses estimated by GAIP are of smaller joint **angle error (Ang Err)** and **position error (Pos Err)** compared with the state-of-the-art competitors.

[1] https://www.noitom.com/
[2] https://www.mixamo.com/

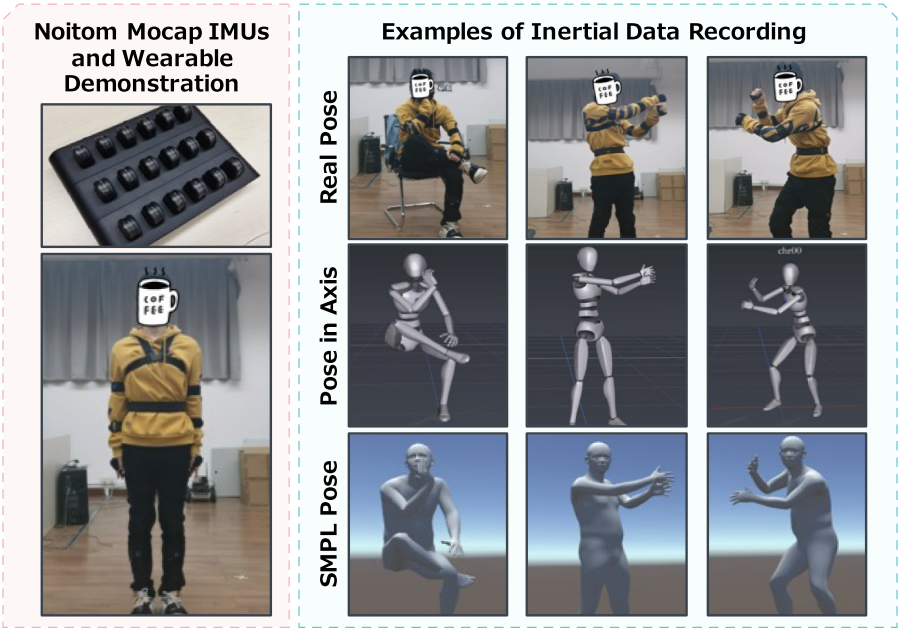

Fig. 1. Inertial data collection equipment and examples. Left: the wearable MoCap IMUs. Right: three groups of inertial data recording examples. In each group, from top to bottom are the participants' poses, the measured poses in Noitom Axis software, and the poses displayed with the Skinned Multi-person Linear model (SMPL) model in Unity3D, respectively.

In summary, our contributions are threefold:

- To address the issue of the lack of spatial prior in SI-HPE, we propose an SI-HPE framework that explicitly integrates human skeleton topology. Our framework has an s-GC module to learn the spatial prior of joint connectivity and contains a joint position loss to implicitly learn spatial information with forward kinematics. This approach attains state-of-the-art results on mainstream 3D pose estimation benchmarks.
- To enhance the generalization ability of SI-HPE methods, we propose an adversarial training strategy to learn implicit features. By synchronously training the discriminator and the regressor, an adversarial loss is introduced to further guide the regression in GAIP. Such a strategy encourages the regressor to produce more realistic and accurate movements.
- To expand the diversity of the inertial datasets, we construct both a real dataset and a synthetic dataset. The real dataset includes rare hip support actions, while the synthetic dataset includes complex and diverse full-body movements. These datasets enrich the current inertial data and can be used for additional fine-tuning and evaluation for SI-HPE.

To ensure the full reproducibility of our results, all our data and codes are made online available at https://cslinzhang.github.io/GAIP/.

2 Related Work

In this section, we review the studies on 3D HPE involving IMUs, including visual-inertial and sparse-inertial ones. The graph convolution and adversarial learning on HPE are also discussed.

2.1 Inertial HPE Approaches

Visual-inertial HPE. In general, an IMU consists of three main components: an accelerometer, a gyroscope, and a magnetometer. These components allow sensor fusion algorithms to extract useful inertial information from the measured raw signals [1, 2]. There are many multi-modal approaches that employ IMU signals as additional constraints to improve the motions predicted from vision. For example, Von Marcard et al. [34] developed a seminal approach to combine multi-view video and inertial data for pose estimation. They used silhouettes from multi-view images extracted with background subtraction and limb orientation from inertial data to minimize a hybrid energy term. Their results demonstrated that information from videos and IMUs can complement each other for HPE. Later, Trumble et al. [32] proposed the first large-scale MoCap dataset that consists of inertial measurements and fused multi-view data for 3D HPE. Going beyond multi-view inputs, Von Marcard et al. attempted to estimate human pose using inertial and monocular data [33]. They paired each **two dimensional (2D)** skeleton with a 3D pose and shape via graph optimization together with a 2D multi-person pose estimation algorithm [4] and an **Skinned Multi-person Linear model (SMPL)** fitting module [24]. In [15], to reduce reliance on feature engineering, Huang et al. even developed an end-to-end trainable 3D convolutional network containing a refinement module using visual-inertial data. Moreover, those prospective results obtained by the aforementioned visual-inertial works also encouraged researchers to fuse IMU with some other sensors, such as red-green-blue depth cameras [11, 31], lidars [9], and so on.

Sparse-inertial HPE. With the price reduction and the improved accuracy of IMU, it is now possible to use IMUs alone for HPE. Such a rapid development also catches the attention of industry and academia. Today, there are already mature commercial solutions that can independently perform high-quality real-time dense-inertial HPE using 17 IMUs on the whole body. However, dense-inertial HPE suffers from high invasiveness and low portability. Therefore, sparse-inertial HPE with just a few IMUs becomes highly desired. To achieve this goal, Von Marcard et al. [35] proposed SIP to estimate the 3D human pose offline using only six IMUs. They represented the human pose by finding SMPL pose parameters that best matches the inertial measurements. In an end-to-end manner, Huang et al. learned a deep neural network model called DIP [16] from a large amount of MoCap data to map IMU signals directly to the pose parameters in real-time. They constructed a bidirectional LSTM network that took the data from IMUs at the five leaf joints as input, normalized it by the root joint's inertial data, and eventually mapped it to the target pose parameters of the SMPL model. Based on DIP, Nagaraj et al. [28] resorted to a bidirectional RNN ensemble to further optimize the network structure. Instead of using deep networks as [28], Frank et al. [39] explored the performance of sparse-inertial HPE using shallow networks. Unlike the aforementioned sparse-inertial HPE approaches that focus on reconstructing the local human pose, Yi et al. proposed a neural model called Transpose [43], which can estimate global translations by merging the velocities of the support foot and the root joint while regressing the pose parameters through multiple sub-networks. Their follow-up work, **Physical Inertial Poser (PIP)** [42], further optimized the estimation of pose and translation with physical constraints. Additionally, the most recent work Transformer Inertial Poser [17] achieves a comparable optimization effect by introducing the Transformer structure. However, none of the aforementioned works included topology information of the human skeleton, which means they did not learn all of the prior information, leaving room for improvement.

2.2 Graph-Based Approaches

It is widely known that GCNs are well-suited for modeling non-Euclidean spatial data. To date, in human pose related fields, GCN is mainly used for vision-based tasks. For example, Yan et al. [41]

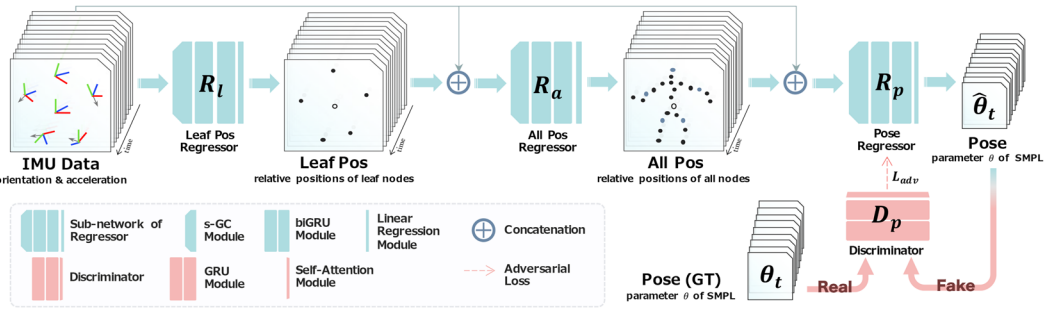

Fig. 2. Overview of GAIP. GAIP consists of two parts: a regressor and a discriminator. The regressor is composed of three sub-networks: (1) leaf joint position regressor $\mathcal{R}_l$, which estimates the relative position of the left and right lower leg joints, the left and right forearm joints, and the head joints to the pelvis root joint from inertial data; (2) full joint position regressor $\mathcal{R}_a$, which estimates the relative position of all joints and the leaf joint position; and (3) human pose regressor $\mathcal{R}_p$, which regresses joint rotations from inertial data and all joint positions. The discriminator $\mathcal{D}_p$ takes predicted and ground truth sequences as inputs, calculates adversarial loss, and inputs it to constrain predicted results in $\mathcal{R}_p$.

took the joints and bones of HKT's topology in each frame as a spatial graph, connected the same joint among different frames as a temporal graph, and performed convolution on spatial-temporal graphs to detect human pose. They also proposed three methods to construct the adjacency matrix of human skeleton spatial graphs according to different standards. Without including all joints in the spatial graph, Ci et al. [7] proposed the **local connection network (LCN)** with higher generalization ability according to the different distances of human joints. Compared with Yan et al.'s approach and LCN, apart from the modifications on modeling formulas, Li et al. [23] further adopted an adaptive adjacency matrix and proposed a graph-based GRU unit for human motion prediction.

As a natural undirected graph, HKT's topology is a type of non-Euclidean spatial data that can be efficiently learned by GCN. Inspired by the aforementioned studies, we believe that GCN has the potential to significantly contribute to the field of HPE.

2.3 Adversarial Networks Related Approaches

GANs are widely used in generative image modeling, as well as in recurrent architectures for simulating sequence-to-sequence data [21, 27]. Interestingly, HPE can be viewed as a similar sequence-to-sequence task. Research on motion modeling has also verified that adversarial training can facilitate HPE [5, 6, 30] or the human pose prediction [3] with previous motion sequences. Following this finding, many works [29, 36, 40] introduced adversarial training to optimize the results of HPE, especially in the vision-based ones. As a typical example, Video Inference for Body pose and shape Estimation [20] is one of the most influential video inference methods for human pose and shape estimation. In [20], Kocabas et al. conducted adversarial training on SMPL pose parameters estimated by image sequences and the large-scale dataset AMASS [26] to improve the authenticity and accuracy of pose estimation. Motivated by this idea of adversarial training, we hope to learn some implicit priors to improve the accuracy of HPE results and our model's generalization ability.

3 Methodology

3.1 Framework Overview

The overall framework of our GAIP is shown in Figure 2 which consists of two parts: a regressor and a discriminator. The regressor, which is composed of three tandem sub-networks ($\mathcal{R}_l$, $\mathcal{R}_a$, and $\mathcal{R}_p$),

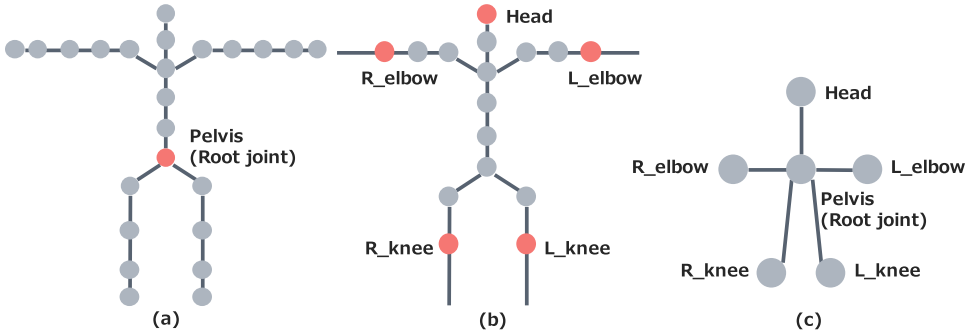

Fig. 3. HKT. (a) The complete 24-joint HKT of the SMPL model, with the root (pelvis) joint marked in red. (b) The 16-joint HKT of the main trunk involved in rotation estimation, with the five leaf joints (the left/right lower legs, the left/right forearms, and the head) marked in red. (c) The simplified six-joint HKT.

learns the relative positions ($\mathcal{R}_l, \mathcal{R}_a$) and pose parameters ($\mathcal{R}_p$) of the joints in different stages from the orientation and acceleration measurements of sparse IMUs. When measurements arrive, an s-GC module is firstly used to fuse the topological prior of the human skeleton. Subsequently, the fused output is passed to a **Bidirectional Gated Recurrent Unit (biGRU)** module, where the latent variables containing information incorporated from the past and future frames are encoded. Finally, a linear regression module takes over to estimate the resulting human pose. In the discriminator ($\mathcal{D}_p$), a GRU is utilized to extract temporal features, and a learned attention mechanism is designed to amplify the contribution of key frames. Eventually, GAIP is jointly supervised by an adversarial loss and a regression loss to minimize the error between the predicted results and the ground truth.

3.2 Background: SMPL

The full name of the SMPL model is "Skinned Multi-Person Linear Model." It uses $\boldsymbol{\Theta}$ to represent the pose and shape of the body and returns a 3D mesh with $N = 6890$ vertices via a differentiable function $M(\boldsymbol{\Theta})$. $\boldsymbol{\Theta}$ consists of pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{72}$ and shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$. The pose parameters $\boldsymbol{\theta}$ includes the global rotation of the root joint and the relative rotations of 23 joints in axis-angle format (the tree structure of 24 joints and their parent–child relationships are shown in Figure 3(a)). $\boldsymbol{\beta}$ is a vector consisting of 10 scalar values. Each value represents the expansion or shrinkage amount of a human subject along a specific direction.

In the setting of sparse-inertial HPE, the IMUs are only attached to the five leaf joints, meaning that they cannot describe the movement of more terminal joints like hands or feet. It is therefore almost impossible to accurately estimate the motion of these joints based solely on inertial measurements. To address this issue, we adopt a skeleton topology with only 16 joints, consistent with previous work as shown in Figure 3(b). As a result, our estimated results only include the rotations of 15 joints, as the global rotation of the root joint is directly measured.

3.3 s-GC

In SMPL models, skeletons are often represented as tree-structured undirected graphs, known as the HKT. This makes it natural to use GC modules for processing the topology of HKT. In this case, the connectivity of the joints can be represented by an adjacency matrix $\mathbf{A}$ representing the connectivity between joints and an identity matrix $\mathbf{I}$ representing the self-connectivity. The corresponding GC can be implemented using a formula similar to the one suggested in [19]

$$\mathbf{X}_{out}(t) = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}\mathbf{X}_{in}(t)\mathbf{W}, \tag{1}$$

where $\mathbf{D}$ represents the diagonal matrix used for normalization, with diagonal elements $\mathbf{D}^{(i,i)} = \sum_j (\mathbf{A}^{(i,j)} + \mathbf{I}^{(i,j)})$; $\mathbf{W}$ is a weight matrix formed by concatenating weight vectors of multiple channels; and $\mathbf{X}_{in}(t)$ and $\mathbf{X}_{out}(t)$ are the module's input and output data of the tth frame, respectively. Both $\mathbf{X}_{in}(t)$ and $\mathbf{X}_{out}(t)$ have a size of $v \times c$, where v is the number of nodes (also known as joints in HKT) corresponding to the adjacency matrix, and c represents the feature dimension.

Such a convolution, however, does not fully exploit the spatial relationship of the HKT in the time domain. The main reason is that the range of motion for various joints is not uniform. In other words, for any joint in the HKT's topology, the joints that are farther from the root joint often have a larger range of motion. Therefore, unlike the traditional adjacency matrix, in GC for HKT, different adjacent joints should have varying weights within a single frame. Consequently, inspired by [41], when constructing the adjacency matrix of the HKT's topology, we divide the neighbor set of each joint node into three subsets: (1) the node itself; (2) a centripetal group closer to the skeleton's center of gravity, where we approximate the center of gravity with the root node; and (3) a centrifugal group otherwise. We construct the adjacency matrix respectively for each of the above three subsets using the following formula:

$$\mathbf{A}_k^{(i,j)} = 1,\ k = \begin{cases} 0, & r_j = r_i \\ 1, & r_j < r_i \\ 2, & r_j > r_i \end{cases}, \tag{2}$$

where $\mathbf{A}_k^{(i,j)}$ represents the element in the ith row and jth column of the kth adjacency matrix, and r_i/r_j represents the topological distance of the ith/jth node from the root node, which corresponds to the root joint.

We set up GC layers in the s-GC module and use the above method to extract the spatial features of HKT-related data. Specifically, when the data are input into a GC layer of the s-GC module, we construct the adjacency matrices $\mathbf{A_0}$, $\mathbf{A_1}$, and $\mathbf{A_2}$, as the initial values of convolution kernels based on corresponding HKT, and perform the GC operation via

$$\mathbf{X}_{out}(t) = \sum_k \mathbf{D}_k^{-\frac{1}{2}} (\hat{\mathbf{A}}_k^{(v)}) \mathbf{D}_k^{-\frac{1}{2}} \mathbf{X}_{in}(t), \tag{3}$$

$$\hat{\mathbf{A}}_k^{(v)} = \mathbf{W}_k^{(v)} \mathbf{A}_k^{(v)} + \mathbf{b}_k^{(v)},\ k = 0, 1, 2, \tag{4}$$

where v represents the number of joints in the HKT of input data, and for each adjacency matrix $\mathbf{A}_k^{(v)} \in \mathbb{R}^{v \times v}$, we accompany it with a learnable weight matrix $\mathbf{W}_k^{(v)}$ and a learnable offsets matrix $\mathbf{b}_k^{(v)}$ to update the convolution kernel.

3.4 Regressor

Our regressor consists of three main sub-networks: the leaf joint position regressor $\mathcal{R}_l$, the full position regressor $\mathcal{R}_a$, and the human pose regressor $\mathcal{R}_p$. As shown in Figure 2, the three connected sub-networks are similar in structure and are all composed of a s-GC module, a biGRU module, and a linear regression module.

3.4.1 Leaf Joint Position Regressors. Firstly, in the leaf joint position regressors $\mathcal{R}_l$, we estimate the relative positions of the five leaf joints to the root joint. The detailed structure and data flow of $\mathcal{R}_l$ is shown in Figure 4.

Specifically, before entering $\mathcal{R}_l$, the acceleration and orientation measurements are normalized with respect to the root joint. Assuming that the inertial data measured at tth frame are $\mathbf{a}(t)_{raw} \in \mathbb{R}^{c \times 3}$, denoting the three-dimensional acceleration, and $\mathbf{R}(t)_{raw} \in \mathbb{R}^{c \times 3 \times 3}$, representing the rotation matrices of the skeletal orientations, where $c = 6$ refers to the number of IMUs.

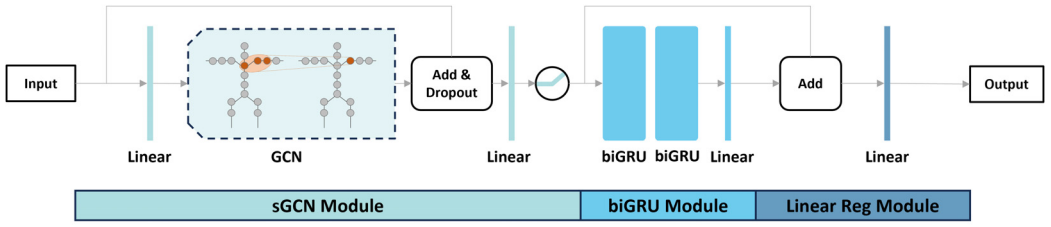

Fig. 4. Structure of the joint position regressors R_l and R_a. The joint position regressor contains three modules, a s-GC module, a biGRU module, and a linear regression module. Specifically, the s-GC module includes two linear layers, a GC layer and an activation function layer; the biGRU module consists of a linear layer and two layers of bidirectional GRU; the linear regression module is composed of linear layers. When the 2-d data is input, it will be flattened into a 1-d vector before entering the biGRU module, and finally reshaped back to a 2-d matrix after the linear regression module as the output data.

Similar to DIP [16], we normalize the orientation and acceleration measured by each IMU via

$$\mathbf{R}(t)_i = \mathbf{R}_{root}^{-1}(t) \cdot \mathbf{R}(t)_{raw,i}, i = 1, ..., c, \tag{5}$$

$$\mathbf{a}(t)_i = \mathbf{R}_{root}^{-1}(t) \cdot (\mathbf{a}(t)_{raw,i} - \mathbf{a}(t)_{root}), i = 1, ..., c, \tag{6}$$

where $\mathbf{a}_{root}(t)$ and $\mathbf{R}_{root}(t)$ represent the acceleration and orientation of the root joint at tth frame, and $(\cdot)^{-1}$ denotes the inverse of the involved matrix. As a result, the measurements are normalized to $\mathbf{R}(t) \in \mathbb{R}^{c\times 3}$ and $\mathbf{a}(t) \in \mathbb{R}^{c\times 3\times 3}$. We then flatten the rotation matrices $\mathbf{R}(t)$ into $\mathbf{r}(t) \in \mathbb{R}^{v_1\times 9}$ and concatenate it with $\mathbf{a}(t)$ to obtain the input inertial data $\mathbf{X}_l(t) = [\mathbf{a}(t), \mathbf{r}(t)] \in \mathbb{R}^{v_1\times 12}$, where $v_1 = c = 6$ represents the number of joints whose inertial data are measured directly by IMUs, that is, the five leaf joints and the root joint.

In $\mathcal{R}_l$, firstly, we input $\mathbf{X}_l(t)$ into the s-GC module. In the s-GC module, the data are fed into a linear layer and an s-GC layer, where the convolution kernel is $\hat{\mathbf{A}}_k^{(v_1)} \in \mathbb{R}^{v_1\times v_1}$ representing the simplified six-joint topology shown in Figure 3(c), since the five leaf joints are relatively scattered. The output result of the s-GC layer is then added by the input data, and afterwards fed into a linear layer and a activation function layer to obtain the latent features $\hat{\mathbf{X}}_l(t) \in \mathbb{R}^{v_1\times 12}$ that incorporates the topology prior.

Secondly, we flatten the data $\hat{\mathbf{X}}_l(t)$ into 1-d vectors and feed it into the biGRU module, which contains two layers of bidirectional GRU and a linear layer, to learn temporal features. After that, the output of the biGRU module is added with $\hat{\mathbf{X}}_l(t)$ and input into the regression module, which is constructed by a linear layer, to estimate a 1-d vector $\mathbf{P}_l(t) \in \mathbb{R}^{((v_1-1)\times 3)}$ representing the root-relative positions of the five leaf joints. We finally reshape it to a 2-d matrix $\hat{\mathbf{P}}_l(t) \in \mathbb{R}^{(v_1-1)\times 3}$ as the output of the leaf joint position regressor $\mathcal{R}_l$. The network is trained with a l-2 loss defined as

$$L_l = \sum_t ||\hat{\mathbf{P}}_l(t) - \mathbf{P}_l^{GT}(t)||_2^2, \tag{7}$$

where $\mathbf{P}_l^{GT}(t)$ represents the ground-truth value.

3.4.2 Full Joint Position Regressors. The full joint position regressors $\mathcal{R}_a$ is similar to $\mathcal{R}_l$ in terms of structure as shown in Figure 4, but with different data sizes. The input data for $\mathcal{R}_a$ are the concatenation of two parts: (1) the normalized inertial measurement data, $\mathbf{a}(t)$ and $\mathbf{r}(t)$, and (2) the root-relative positions of six joints $\hat{\mathbf{P}}'_l(t) = [\hat{\mathbf{P}}_l(t), \mathbf{0}] \in \mathbb{R}^{v_1\times 3}$, which is formed by concatenating

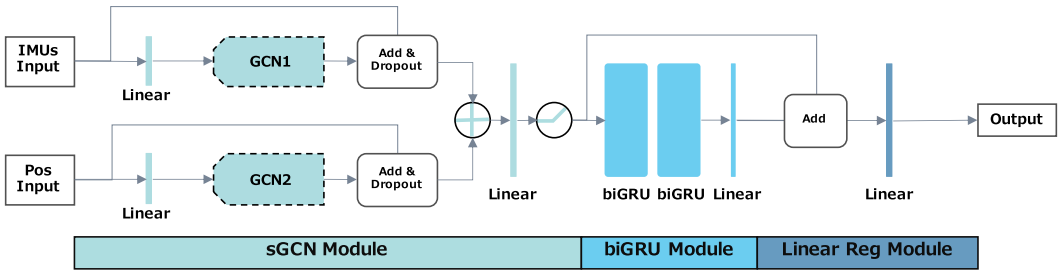

Fig. 5. Structure of the human pose regressors R_p. R_p contains three modules, a s-GC module, a biGRU module, and a linear regression module. Among them, the biGRU module and the linear regression module have the same structure as the joint position regressors. Instead, the difference is that the s-GC module of R_p includes two parts with similar structures, which perform GC operations with different convolution kernel sizes on the inertial data and the joint position data, respectively. The convolution results of these two parts will be concatenated and flattened into a 1-d vector before being fed into the biGRU module.

the $\mathcal{R}_l$'s output with the root joint's position initialized with zeros. As a result, the $\mathcal{R}_a$'s input is formally denoted by $\mathbf{X}_a(t) = [\mathbf{a}(t), \mathbf{r}(t), \hat{\mathbf{P}}'_l(t)] \in \mathbb{R}^{v_1 \times 15}$.

In $\mathcal{R}_a$, $\mathbf{X}_a(t)$ undergoes the same processing as $\mathbf{X}_l(t)$ in $\mathcal{R}_l$. Firstly, it is fed into the s-GC module to obtain the latent features $\hat{\mathbf{X}}_a(t) \in \mathbb{R}^{v_1 \times 15}$. Secondly, $\hat{\mathbf{X}}_a(t)$ is flattened into a 1-d vector and input into the biGRU module. Thirdly, the biGRU module's output is added with $\hat{\mathbf{X}}_a(t)$ and fed into the regression module to obtain a 1-d vector $\mathbf{P}_a(t) \in \mathbb{R}^{((v_2-1)\times 3)}$ denoting all non-root joints' root-relative positions. Finally, the 1-d vector is reshaped to a 2-d matrix $\hat{\mathbf{P}}_a(t) \in \mathbb{R}^{(v_2-1)\times 3}$, where $v_2 = 24$.

In addition, $\mathcal{R}_a$ also uses a l-2 loss for training, which is defined by

$$L_a = \sum_t ||\hat{\mathbf{P}}_a(t) - \mathbf{P}_a^{GT}(t)||_2^2, \tag{8}$$

where $\mathbf{P}_a^{GT}(t)$ represents the ground-truth value.

3.4.3 Human Pose Regressor. In pose regressor $\mathcal{R}_p$, we estimate global joint rotations, that is, root-relative joint rotations, from joint positions. Specifically, $\mathcal{R}_p$ receives two parts of data as input, which are the inertial measurements $\mathbf{X}_p(t)$ and the joint coordinates $\hat{\mathbf{P}}'_a(t)$. The former $\mathbf{X}_p(t) = [\mathbf{a}(t)', \mathbf{r}(t)'] \in \mathbb{R}^{v_3 \times 12}$ is constructed based on $\mathbf{X}_l(t)$, where $v_3 = 16$ refers to the number of joints of the main trunk of the body as shown in Figure 3(b). In $\mathbf{X}_p(t)$, the joints containing no inertial measurements are initialized with zeros. The latter $\hat{\mathbf{P}}'_a(t) = [\hat{\mathbf{P}}_a(t), \mathbf{0}] \in \mathbb{R}^{v_2 \times 3}$ is formed by initializing the root joint's position with zeros on the basis of $\hat{\mathbf{P}}_a(t)$.

After being input to $\mathcal{R}_p$, whose structure is shown in Figure 5, $\mathbf{X}_p(t)$ and $\hat{\mathbf{P}}_a(t)$ will be sent to different s-GC modules firstly, since the inertial data $\mathbf{X}_p(t)$ and the joint positions $\hat{\mathbf{P}}'_a(t)$ do not correspond to the same topology. Specifically, the convolution kernels of the two s-GC modules are $\hat{\mathbf{A}}_k^{(v_3)} \in \mathbb{R}^{v_3 \times v_3}$ and $\hat{\mathbf{A}}_k^{(v_2)} \in \mathbb{R}^{v_2 \times v_2}$, representing the 16-joints topology as shown in Figure 3(b) and 24-joints topology as shown in Figure 3(a), respectively. As a result, we get $\hat{\mathbf{X}}_{pi}(t) \in \mathbb{R}^{v_3 \times 12}$ and $\hat{\mathbf{X}}_{pp}(t) \in \mathbb{R}^{v_2 \times 3}$. Then, we flatten the above two results into $\hat{\mathbf{X}}'_{pi}(t) \in \mathbb{R}^{(v_3 \times 12)}$ and $\hat{\mathbf{X}}'_{pp}(t) \in \mathbb{R}^{(v_2 \times 3)}$, and then concatenate them to get a 1-d vector $\hat{\mathbf{X}}_p(t) = [\hat{\mathbf{X}}'_{pi}(t), \hat{\mathbf{X}}'_{pp}(t)] \in \mathbb{R}^{(v_3 \times 12 + v_2 \times 3)}$. Afterwards, $\hat{\mathbf{X}}_p(t)$ is fed into the biGRU module to obtain the latent features with temporal priors.

Eventually, the latent features are input into the regression module to estimate a 1-d vector $\boldsymbol{\theta}_p^{6D}(t) \in \mathbb{R}^{((v_3-1)\times 6)}$ in the six dimensional representation [45], representing the root-relative

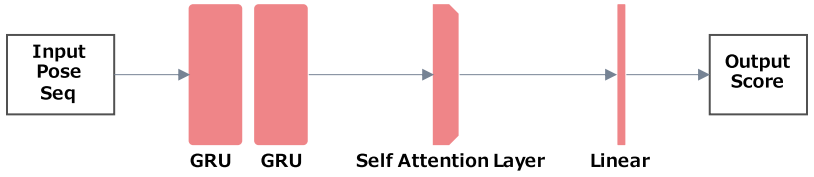

Fig. 6. Structure of the discriminator D_p. The discriminator consists of two layers of GRU, a self-attention layer and a linear layer. D_p receives a pose sequence, captures sequence features with GRU and self-attention layers, and finally outputs a score representing the authenticity of the pose sequence by a linear layer.

rotations of 15 non-root joints as shown in Figure 3(b). We reshape it to a 2-d matrix $\hat{\boldsymbol{\theta}}_p^{6D}(t) \in \mathbb{R}^{(v_3-1)\times 6}$. Given that the rotation of the root joint $\mathbf{R}_{root}(t)$ can be directly measured, we combine all these rotations and convert them to the rotation matrices and obtain body's pose as $\hat{\boldsymbol{\theta}}_p(t) \in \mathbb{R}^{v_3\times 9}$.

The overall loss (L_p) of the pose regressor $\mathcal{R}_p$ is composed of three parts, the joint rotation loss (L_{rot}), the joint position loss (L_{pos}), and the adversarial loss (L_{adv}), i.e.,

$$L_p = L_{rot} + L_{pos} + L_{adv}. \tag{9}$$

We leave the adversarial loss being introduced in Section 3.5.

Here, L_{rot} is a l-2 rotation loss aiming to make the estimated rotation numerically close to the ground truth, which is formally defined by,

$$L_{rot} = \sum_t ||\hat{\boldsymbol{\theta}}_p^{6D}(t) - \boldsymbol{\theta}_p^{6D,GT}(t)||_2^2, \tag{10}$$

where $\boldsymbol{\theta}_p^{6D,GT}(t)$ represents the ground truth.

L_{pos} is a l-2 position loss of the joint positions calculated by forward kinematics, which gives the loss function a certain physical meaning, rather than simply comparing numbers. To obtain L_{pos}, we need to calculate the joint positions by $\hat{\boldsymbol{\theta}}_p(t)$. This requires knowledge of the bones' length in HKT. However, some of the current inertial datasets do not include SMPL shape parameters, which means that we cannot determine the length of bones. Without this information, calculating the position of joints via forward kinematics is impossible. Therefore, for data that provide SMPL shape parameters $\boldsymbol{\beta}$, we use these parameters to calculate the bones' lengths and then derive the joint positions with forward kinematics. For data without SMPL shape parameters, we directly use the default male model for calculation. The joint position loss L_{pos} is accordingly given by

$$L_{pos} = \sum_t ||FK(\hat{\boldsymbol{\theta}}_p(t), \boldsymbol{\beta}) - FK(\boldsymbol{\theta}_p^{GT}(t), \boldsymbol{\beta})||_2^2, \tag{11}$$

where FK represents the forward kinematics function, which takes joint rotations and SMPL shape parameters as input and returns the relative positions of all joints to the root joint; $\hat{\boldsymbol{\theta}}_p(t)$ indicates the estimated body posture; and $\boldsymbol{\theta}_p^{GT}(t)$ is the ground-truth value.

3.5 Pose Discriminator

Simply using l-2 losses is not sufficient to constrain the sequence of human poses. Since only the inertial data of some joints are available, inaccurate and ambiguous poses may be recognized as valid. To address this issue, we use a pose discriminator D_p to determine whether the generated pose sequence corresponds to the ground truth sequences. As shown in Figure 6, D_p includes two GRU layers, a self-attention layer and a linear layer.

Given a inertial measurements sequence of length T, we firstly take the output pose sequence $\hat{\boldsymbol{\theta}}_p = (\hat{\boldsymbol{\theta}}_p(1), \hat{\boldsymbol{\theta}}_p(2), ..., \hat{\boldsymbol{\theta}}_p(T))$ of the regressor as input. Then, we feed the sequence into GRU layers, where the latent codes is estimated. Subsequently, the latent codes aggregated by the self-attention layer.

Finally, the linear layer outputs a prediction $D_p(\hat{\boldsymbol{\theta}}_p) \in [0, 1]$, representing the probability that $\hat{\boldsymbol{\theta}}_p$ belongs to the plausible human motion manifold. In summary, the adversarial loss term backpropagated to $\mathcal{R}_p$ is

$$L_{adv} = \mathbb{E}[||D_p(\hat{\boldsymbol{\theta}}_p) - 1||_2^2], \tag{12}$$

where $\mathbb{E}$ represents the mean value.

Accordingly, the objective function of the discriminator D_p is

$$L_{D_p} = \mathbb{E}[||D_p(\boldsymbol{\theta}_p^{GT}) - 1||_2^2] + \mathbb{E}[||D_p(\hat{\boldsymbol{\theta}}_p)||_2^2], \tag{13}$$

where $\boldsymbol{\theta}_p^{GT}$ represents the actual pose sequences, while $\hat{\boldsymbol{\theta}}_p$ refers to our estimated values.

During training, the discriminator is trained synchronously with the regressor with the loss functions described above. In each training step, firstly, the regressor and the discriminator propagate forward and calculate their respective results according to the loss functions. After that, for the regressor, we freeze the discriminator's backpropagation by set its gradient to 0, and then perform the backpropagation process of the regressor to update the network. For the discriminator, we unfreeze the backpropagation of the discriminator and update it individually.

However, due to the simpler task of the discriminator, its training speed is usually faster than that of the regressor, which can easily lead to the collapse of the adversarial training. To alleviate this problem, we set a step interval k for the backpropagation of the discriminator. In each step of training, we will only perform the backpropagation of the discriminator to update its parameters when the current training step is a multiple of k. Here, k is a hyperparameter, which we set it to 4 empirically. In this way, we balance the training speed of the regressor and the discriminator so that the network can train better results.

3.6 Self-Collected Datasets

3.6.1 Real Dataset. To enrich the diversity of real data for SI-HPE, we constructed a real IMU dataset called SingleOne-IMU. In SingleOne-IMU, we recorded data from 5 persons wearing 17 Noitom IMUs. The participants included four males and one female. They were required to repeat two types of movements, foot support movements and hip support movements, each of which included controlled limb movements and natural whole body movements (as shown in Table 1). We collected about 60 minutes long of data (about 12 minutes per person) using Noitom Perception Neuron 3, a commercial MoCap solution. The Noitom Axis software was used to post-process the recorded data so as to obtain the ground truth of the pose parameters.

In SingleOne-IMU dataset, we decided to make the joint length constant. The reason of this decision was that we found that in the SMPL model, pose parameters and shape parameters do not interfere with each other. In other words, SMPL model assumes that joints' orientation exists independently of bones' length. We followed this idea and ignored the influence of body template on joint orientation when recording the inertial data.

3.6.2 Synthetic Dataset. Additionally, we constructed a synthetic dataset called Mixamo-IMU to accommodate more challenging poses. We collected 949 pose sequences from four categories (Combat, Adventure, Sport, and Dancing) from Mixamo and generated their SMPL pose parameters. The corresponding orientation and acceleration data of the default SMPL model were calculated from those pose parameters.

The Mixamo-IMU dataset includes a total of 238,468 frames of motions, with the longest motion sequence being 2,801 frames and the shortest being only 60 frames. Compared to existing synthetic

Table 1. Capture Protocol used to Record the SingleOne-IMU Dataset

Categories	Detailed Categories	Motions (Repetitions)	Frames	Minutes
Foot support movements	Limb	Arm raises (5), legs raises (5), body stretches (5), squats (5), leg stretches (3).	67,802	18.83
	Locomotion	Walks (3), goose steps (3), runs (3), basket shoots, cross steps (3), jumps (3), high jumps (3), drags, carries.	80,604	22.39
	Freestyle	The subject can perform two sets of free activities while keep standing.	45,437	12.62
Hip support movements	Limb	Leg raises (5), leg curls (5), belly crunches (5), legs crosses (5).	17,777	4.94
	Locomotion	Sit-downs and stand-ups (5), circles (3).	13,269	3.68
	Freestyle	The subject can perform two sets of free activities while keep sitting.	17,132	4.76

Table 2. Comparison of Different Datasets

Dataset Name	Sources of Inertial Data	Sources of Human Pose	Length	Real Data
DIP-IMU	Measurements of 17 Xsens IMUs	Calculated offline by SIP [35]	About 90 minutes	T
TC-IMU	Calculated on the measurements of 13 IMUs by OpenDR [25]	Marker-based MoCap system	About 50 minutes	T
SingleOne-IMU	Measurements of 17 Noitom IMUs	Noitom's Axis software	About 60 minutes	T
Mixamo-IMU	Export from the Mixamo website	Export from the Mixamo website	About 60 minutes	F

datasets, most sequences in Mixamo-IMU are generally shorter. However, there are more unique and diverse types of motions in it. As a result, we use Mixamo-IMU as a test set to evaluate the model's generalization ability.

4 Experiment

4.1 Setup

4.1.1 Datasets. Our training set consisted of AMASS, the train split of DIP-IMU, and the train split of SingleOne-IMU, with an overall data ratio of nearly 93:4:3. All models involved in the experiments were retrained using this training set. The pre-processing for training data was consistent with Transpose [43].

Regarding the evaluation datasets, aside from the two publicly available real datasets, DIP-IMU [16] and **Total Capture (TC)**-IMU [32], used in previous studies, we also performed experiments on self-collected datasets, SingleOne-IMU and Mixamo-IMU. The specific information about these four datasets used for evaluations is shown in Table 2. In addition, the usage of these four datasets is as follows:

—DIP-IMU: Eight out of its 10 sets were utilized for training, while the remaining two were designated for testing.

—TC-IMU: All its data were used for testing. Note that, similar to the previous methods, as part of the AMASS dataset, TotalCapture's pose parameters and simulated inertial data were employed for training, whereas for testing, the IMU measurements were sourced from the real data within the TotalCapture dataset [32].
—SingleOne-IMU: Four out of its five sets were used for training, with one set reserved for testing.
—Mixamo-IMU: Its data were only used for testing.

4.1.2 Evaluation Settings and Metrics. We compared our GAIP with the state-of-the-art methods Transpose [43] and DIP [16]. We adopted both offline and online settings. In offline setting, the full sequence was available at the test time. In the online (real-time) setting, GAIP and competitors' models accessed 20 past frames, 1 current frame, and 5 future frames in a window sliding manner.

Regardless of whether the evaluation was conducted in an offline manner or an online manner, the quality of pose estimation was assessed using the following metrics:

—**SIP error (SIP Err)**: measures the average global angle difference between the reconstructed and ground-truth poses over four joints (the up arms and the up legs), represented in axis-angle, in degrees.
—Ang Err: measures the average global angle difference between the reconstructed and ground-truth poses over all 16 joints shown in Figure 3(b), represented in axis-angle, in degrees.
—Pos Err: measures the average Euclidean distance error between the reconstructed and ground-truth poses of all estimated joints, in centimeters, aligned with the root joint.
—**Mesh error (Mesh Err)**: measures the average vertex distance error between the reconstructed and ground-truth meshes, in centimeters, aligned with the root position and orientation.

4.1.3 Implementation Details. Our model was implemented using PyTorch 1.10.2 and CUDA 11.4. All training and evaluation processes were conducted on a computer equipped with an Intel® Core™ i9-10920X CPU and an NVIDIA GeForce RTX 3060 Ti/PCIe/SSE2 graphics card.

During training, we first train three sub-networks R_l, R_a, and R_p, respectively, with the aforementioned L_l, L_a, and L_{rot} on the AMASS dataset, so as to reduce the difficulty of the subsequent unified training. Afterwards, as described in Section 3.5, we use L_p and L_{D_p} to train the regressor and the discriminator of GAIP simultaneously on the training sets, including the AMASS dataset, the DIP-IMU dataset and the SingleOne-IMU dataset. During testing, only the regressor of GAIP is used to obtain the estimation results.

4.2 Quantitative Results

Quantitative results of offline and online settings are shown in Tables 3 and 4, respectively. We present the mean and standard deviations of each metric compared with PIP, Transpose, and DIP.

As shown in the tables, it is clear that our GAIP achieves smaller joint rotation and position errors across all datasets, whether in offline or online evaluations, demonstrating that our model is able to utilize the spatial prior and the implicit distribution of the data for more accurate estimation. In addition, in Tables 3 and 4, we can find that compared with the DIP-IMU dataset, our method has a more prominent optimization effect on the SingleOne-IMU dataset. This implies that GAIP can more accurately estimate the rotation of the thigh joint to identify ambiguous motions and determine sitting poses, which should be attributed to the learning of joint connectivity by s-GC modules. We believe that the spatial prior of joint connectivity has the ability to enhance the advantages of incorporating joint positions in joint rotation estimation, particularly when a multi-stage regressor has already computed joint positions as intermediate results.

Table 3. Results of Offline Evaluations

		PIP [42]	Transpose [43]	DIP [16]	GAIP (Ours)
DIP-IMU	SIP Err	15.02 (± 7.90)	15.17 (± 7.99)	16.61 (± 8.63)	**14.60 (± 6.99)**
	Ang Err	8.13 (± 4.21)	8.22 (± 4.18)	13.83 (± 7.19)	**7.76 (± 3.97)**
	Pos Err	5.19 (± 2.95)	5.26 (± 2.91)	6.72 (± 3.67)	**5.01 (± 2.81)**
	Mesh Err	6.91 (± 3.81)	7.10 (± 4.31)	7.65 (± 4.45)	**5.99 (± 3.28)**
TC-IMU	SIP Err	13.63 (± 8.18)	14.00 (± 8.79)	16.92 (± 8.71)	**12.59 (± 6.86)**
	Ang Err	12.23 (± 6.03)	12.39 (± 6.08)	15.82 (± 8.00)	**12.02 (± 5.71)**
	Pos Err	5.91 (± 3.63)	6.11 (± 3.79)	8.87 (± 4.81)	**5.56 (± 3.16)**
	Mesh Err	6.69 (± 4.06)	6.89 (± 4.15)	9.23 (± 5.23)	**6.44 (± 3.60)**
SingleOne-IMU	SIP Err	20.78 (± 9.37)	22.36 (± 9.67)	24.94 (± 11.34)	**18.32 (± 8.67)**
	Ang Err	9.42 (± 4.64)	9.86 (± 4.42)	17.27 (± 8.58)	**8.73 (± 3.92)**
	Pos Err	6.51 (± 3.47)	6.92 (± 3.63)	9.18 (± 4.31)	**6.15 (± 3.14)**
	Mesh Err	7.18 (± 3.84)	7.78 (± 3.98)	9.88 (± 5.07)	**6.84 (± 3.37)**
Mixamo-IMU	SIP Err	26.72 (± 9.15)	26.02 (± 8.26)	27.91 (± 8.95)	**24.78 (± 7.24)**
	Ang Err	12.90 (± 4.77)	11.51 (± 4.54)	19.88 (± 5.61)	**10.89 (± 3.62)**
	Pos Err	9.72 (± 3.83)	9.33 (± 3.42)	12.06 (± 5.75)	**8.77 (± 2.88)**
	Mesh Err	11.10 (± 4.25)	10.55 (± 3.71)	12.97 (± 4.17)	**9.92 (± 3.19)**

For each performance metric, the value achieved by the method with the best performance is bold.

Table 4. Results of Online Evaluations

		PIP [42]	Transpose [43]	DIP [16]	GAIP (Ours)
DIP-IMU	SIP Err	20.21 (± 9.43)	20.88 (± 9.97)	22.78 (± 11.60)	**19.80 (± 9.36)**
	Ang Err	9.41 (± 5.43)	9.67 (± 5.37)	15.18 (± 7.05)	**9.08 (± 4.69)**
	Pos Err	6.16 (± 4.15)	6.30 (± 3.92)	7.79 (± 4.61)	**5.97 (± 3.65)**
	Mesh Err	7.42 (± 4.77)	7.82 (± 4.62)	8.82 (± 4.72)	**7.11 (± 4.22)**
TC-IMU	SIP Err	14.62 (± 8.47)	14.74 (± 8.69)	18.86 (± 9.01)	**13.87 (± 8.27)**
	Ang Err	12.28 (± 6.23)	12.35 (± 7.21)	15.01 (± 8.03)	**12.05 (± 5.93)**
	Pos Err	6.31 (± 4.15)	6.80 (± 3.87)	8.82 (± 4.57)	**5.93 (± 3.62)**
	Mesh Err	7.51 (± 4.54)	7.81 (± 4.87)	9.76 (± 5.10)	**6.70 (± 4.00)**
SingleOne-IMU	SIP Err	26.83 (± 12.27)	27.11 (± 13.35)	28.56 (± 14.21)	**23.54 (± 11.82)**
	Ang Err	10.96 (± 4.82)	11.34 (± 5.97)	19.21 (± 9.88)	**9.84 (± 4.70)**
	Pos Err	8.45 (± 4.55)	8.72 (± 4.98)	13.38 (± 7.36)	**7.91 (± 4.49)**
	Mesh Err	9.23 (± 4.88)	9.77 (± 5.49)	14.36 (± 7.95)	**8.89 (± 4.95)**
Mixamo-IMU	SIP Err	27.69 (± 8.99)	26.32 (± 7.82)	28.22 (± 9.38)	**25.21 (± 7.67)**
	Ang Err	12.79 (± 4.73)	12.22 (± 4.31)	15.91 (± 4.94)	**10.82 (± 3.72)**
	Pos Err	9.63 (± 3.79)	9.43 (± 3.42)	10.46 (± 4.13)	**9.17 (± 3.36)**
	Mesh Err	10.79 (± 3.86)	10.45 (± 3.72)	12.17 (± 4.66)	**10.38 (± 3.72)**

For each performance metric, the value achieved by the method with the best performance is bold.

In addition, we can find from the tables that GAIP achieves the best estimation on the Mixamo-IMU dataset, which is not involved in training. This exhibits that our model can be applied to more diverse human pose data and is with stronger generalization ability, which can be attributed to the adversarial training strategy allows GAIP to learn the implicit features of the data, thereby establishing a more comprehensive mapping relationship between inertial data and the pose parameters, and further improves the model's expression capability and generalization abilities.

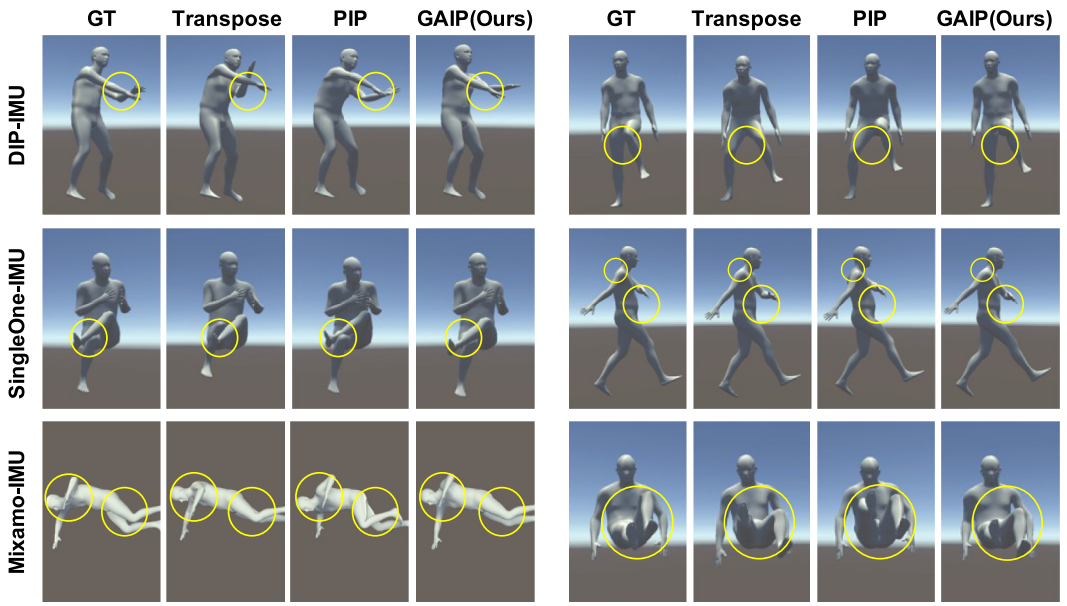

Fig. 7. Qualitative results. From top to bottom are the online evaluation results on DIP-IMU, SingleOne-IMU, and Mixamo-IMU, respectively. The parts with large differences are circled in yellow.

4.3 Qualitative Performance

To conduct qualitative evaluations, we resorted to the SMPL model to visualize poses in Unity3D. Since we only estimated pose parameters, during qualitative evaluation, we fixed the position of the root joint and set all nodes except for the 16 joints shown in Figure 3(b) to identity. As a result, Figure 7 presents the estimated results of our GAIP and Transpose on selected frames from various datasets. More comparison results can be found in our supplementary video on the project's website https://cslinzhang.github.io/GAIP/.

From the examples shown in the first row of Figure 7 selected from the DIP-IMU dataset, we can see that our method has the ability to estimate the human poses more accurately, especially the parts circled in yellow. We believe that this improvement can be mainly attributed to GAIP's more accurate estimation of the upper arms compared to Transpose's and PIP's. Including the upper arm, such limb joints without direct inertial measurements play an important role in the representation of human poses. With better estimations of these joints, the ambiguities (e.g., there are similar measurements of arm lift when sitting and standing) in estimation can be effectively reduced.

In addition, in the second row selected from the SingleOne-IMU dataset, our GAIP performs better in estimating the pose of the legs in a sitting pose sequence. In the results of the Mixamo-IMU dataset presented in the last row, our method produces more satisfactory estimations of challenging poses. This superiority can be attributed to the sufficient utilization of spatial priors with the help of s-GC modules and the joint position loss, as well as the implicit constraints learned with adversarial training.

4.4 Ablation Study

4.4.1 s-GC Module, Joint Position Loss, and Adversarial Loss. Here we show the effectiveness of the key components of our GAIP. We evaluated the following three variants of GAIP: (1) GAIP-V1: without the s-GC module; (2) GAIP-V2: without the adversarial loss (L_{adv}); and (3) GAIP-V3: without

Table 5. Results of the Offline Ablation Study of the s-GC Module, the Joint Position Loss, and the Adversarial Loss

		GAIP-V1 (w/o s-GC)	GAIP-V2 (w/o L_{adv})	GAIP-V3 (w/o L_{pos})	GAIP
DIP-IMU	SIP Err	15.26 (± 7.69)	14.79 (± 6.75)	14.79 (± 7.03)	**14.60 (± 6.99)**
	Ang Err	8.22 (± 4.45)	8.14 (± 4.05)	7.85 (± 3.98)	**7.76 (± 3.97)**
	Pos Err	5.09 (± 3.01)	5.04 (± 2.66)	5.09 (± 2.82)	**5.01 (± 2.81)**
	Mesh Err	6.15 (± 3.76)	6.07 (± 3.26)	6.11 (± 3.31)	**5.99 (± 3.28)**
TC-IMU	SIP Err	13.43 (±7.17)	13.15 (± 6.91)	12.64 (± 6.80)	**12.59 (± 6.86)**
	Ang Err	12.47 (± 5.73)	12.21 (± 5.72)	12.04 (± 5.70)	**12.02 (± 5.71)**
	Pos Err	5.90 (± 3.19)	5.79 (± 3.17)	5.62 (± 3.16)	**5.56 (± 3.16)**
	Mesh Err	6.83 (± 3.74)	6.72 (± 3.63)	6.53 (± 3.60)	**6.44 (± 3.60)**
SingleOne-IMU	SIP Err	21.12 (± 10.26)	19.56 (± 10.33)	18.49 (± 8.79)	**18.32 (± 8.67)**
	Ang Err	9.93 (± 4.74)	9.16 (± 4.30)	8.83 (± 4.04)	**8.73 (± 3.92)**
	Pos Err	7.35 (± 3.79)	6.51 (± 3.66)	6.28 (± 3.34)	**6.15 (± 3.14)**
	Mesh Err	8.33 (± 4.14)	7.11 (± 3.76)	6.97 (± 3.57)	**6.84 (± 3.37)**
Mixamo-IMU	SIP Err	25.70 (± 7.72)	25.46 (± 7.16)	24.93 (± 7.20)	**24.78 (± 7.24)**
	Ang Err	11.76 (± 4.18)	11.36 (± 3.66)	**10.88 (± 3.59)**	10.89 (± 3.62)
	Pos Err	8.82 (± 3.06)	8.89 (± 2.82)	8.91 (± 2.92)	**8.77 (± 2.88)**
	Mesh Err	10.04 (± 3.15)	10.06 (± 3.16)	10.11 (± 3.19)	**9.92 (± 3.19)**

For each performance metric, the value achieved by the method with the best performance is bold.

the joint position loss (L_{pos}). We compared these three variants for offline pose estimations with GAIP on the datasets used for quantitative experiments.

Firstly, we assess the role of the s-GC module in GAIP by comparing the results of GAIP and GAIP-V1. From Table 5, it can be seen that without the s-GC module, the model GAIP-V1 has significantly higher joint rotation errors and joint position errors than the results of GAIP in each dataset. These phenomena indicate that the s-GC module has the ability to improve the accuracy of HPE and plays the most important optimization role in our GAIP. We attribute this optimization to the efficient introduction of spatial priors.

Secondly, we analyze the contribution of the adversarial loss by comparing the results of GAIP and GAIP-V2. From Table 5, we can find that without the adversarial loss, the results of GAIP-V2 deteriorate on all datasets, but to a lesser extent on the DIP-IMU dataset and the SingleOne-IMU datasets, while to a greater extent on the TC-IMU dataset. We believe that adversarial training encourages the model to learn the implicit distribution of the data, allowing the model to achieve better performance on the TC-IMU dataset, which has a similar distribution to the AMASS dataset. Therefore, after discarding the adversarial loss, the performance of GAIP-V2 deteriorates the most on the TC-IMU dataset. The above analysis indicates that in addition to improving the accuracy of HPE, the adversarial loss can also balance the performance of the model on different datasets, prevent possible overfitting, and enable the model to use more diverse datasets for training.

Thirdly, we evaluate the role of the position loss by comparing the results of GAIP-V3 and GAIP. From Table 5, it can be seen that without the position loss, the performances of GAIP-V3 on the DIP-IMU dataset, the TC-IMU dataset, and the SingleOne-IMU dataset are slightly deteriorated. However, on the Mixamo-IMU dataset, the estimation of GAIP-V3 exhibits the largest joint Pos Err and Mes Err. Therefore, we believe that in addition to moderately improving the accuracy of HPE, the position loss can also explicitly constrain joint positions to reduce joint position errors.

4.4.2 Hyperparameter k. The effect of hyperparameter k was also evaluated. We conducted training under k = 2, 4, 6, and 8, respectively, while ensuring the same training datasets, framework, and learning rate. The loss changes of the regressor and the discriminator during the training

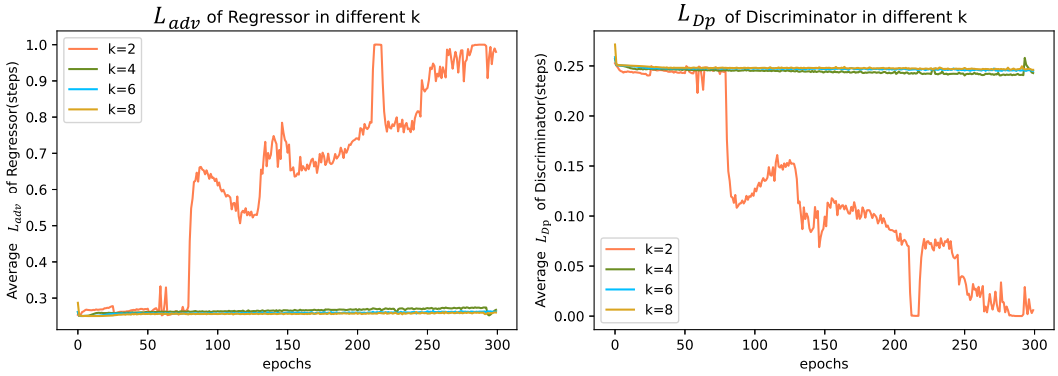

Fig. 8. Loss terms in ablation experiments of the hyperparameter k. The left subfigure represents L_{adv} of Regressors during the training process, while the right one represents L_{D_p} of Discriminator during the training process. During adversarial training, unstable L_{adv} and L_{D_p} will cause training to collapse. It can be seen that training collapse occurs when $k = 2$.

Table 6. Results of the Offline Ablation Study of the Hyperparameter k

		$k = 2$	$k = 4$	$k = 6$	$k = 8$
DIP-IMU	SIP Err	18.11 (± 8.37)	**14.60 (± 6.99)**	15.10 (± 6.74)	15.16 (± 7.00)
	Ang Err	9.64 (± 4.86)	**7.76 (± 3.97)**	8.08 (± 3.95)	8.11 (± 4.00)
	Pos Err	5.89 (± 3.26)	**5.01 (± 2.81)**	5.05 (± 2.67)	5.12 (± 2.73)
	Mesh Err	6.88 (± 3.26)	**5.99 (± 3.28)**	6.06 (± 3.35)	6.08 (± 3.40)
TC-IMU	SIP Err	13.63 (± 8.18)	**12.59 (± 6.86)**	12.77 (± 6.65)	13.01 (± 6.76)
	Ang Err	12.20 (± 5.84)	**12.02 (± 5.71)**	12.25 (± 5.66)	12.32 (± 5.70)
	Pos Err	5.91 (± 3.63)	**5.56 (± 3.16)**	5.78 (± 3.08)	5.84 (± 3.14)
	Mesh Err	6.84 (± 3.71)	**6.44 (± 3.60)**	6.69 (± 4.06)	6.89 (± 4.15)
SingleOne-IMU	SIP Err	20.51 (± 9.19)	**18.32 (± 8.67)**	19.61 (± 10.42)	19.67 (± 10.20)
	Ang Err	8.99 (± 4.11)	**8.73 (± 3.92)**	9.15 (± 4.32)	9.28 (± 4.37)
	Pos Err	6.86 (± 3.44)	**6.15 (± 3.14)**	6.54 (± 3.68)	6.56 (± 3.64)
	Mesh Err	7.63 (± 3.51)	**6.84 (± 3.37)**	7.18 (± 3.54)	7.38 (± 3.98)
Mixamo-IMU	SIP Err	27.72 (± 9.15)	**24.78 (± 7.24)**	24.80 (± 7.16)	24.85 (± 7.17)
	Ang Err	12.90 (± 4.77)	**10.89 (± 3.62)**	10.92 (± 3.63)	11.01 (± 3.67)
	Pos Err	9.72 (± 3.83)	**8.77 (± 2.88)**	8.78 (± 2.79)	8.81 (± 2.80)
	Mesh Err	10.68 (± 4.26)	**9.92 (± 3.19)**	10.13 (± 3.25)	10.07 (± 3.26)

For each performance metric, the value achieved by the method with the best performance is bold.

process are shown in Figure 8, and the results of different models on all datasets are shown in Table 6.

From Figure 8, we can find that when $k < 4$ ($k = 2$ as example), due to the different difficulty levels of the training tasks, the discriminator network will converge too quickly, causing the adversarial network training to collapse. When $k \geq 4$, the network training can maintain stability for a long time. Besides, from Table 6, it can be seen that when $k = 4$, the model has the best results on each dataset. Therefore, we believe that $k = 4$ is a suitable choice.

4.4.3 Convolutional Kernel Schemes in s-GC Modules. In GC, the initialization scheme of the convolution kernel is not unique. Considering that the range of motion of different joints varies greatly, in s-GC modules of GAIP, we do not simply use the adjacency matrix to initialize the

Table 7. Results of the Offline Ablation Study of the Convolutional Kernel Schemes in s-GC Modules

		GAIP-Uni	GAIP
DIP-IMU	SIP Err	14.80 (± 7.09)	**14.60 (± 6.99)**
	Ang Err	8.02 (± 4.04)	**7.76 (± 3.97)**
	Pos Err	5.06 (± 2.87)	**5.01 (± 2.81)**
	Mesh Err	6.13 (± 3.36)	**5.99 (± 3.28)**
TC-IMU	SIP Err	13.17 (± 6.96)	**12.59 (± 6.86)**
	Ang Err	12.30 (± 5.76)	**12.38 (± 5.75)**
	Pos Err	5.82 (± 3.20)	**5.56 (± 3.16)**
	Mesh Err	6.54 (± 3.75)	**6.44 (± 3.60)**
SingleOne-IMU	SIP Err	19.52 (± 9.98)	**18.32 (± 8.67)**
	Ang Err	9.31 (± 4.27)	**8.73 (± 3.92)**
	Pos Err	6.54 (± 3.56)	**6.15 (± 3.14)**
	Mesh Err	7.16 (± 3.52)	**6.84 (± 3.37)**
Mixamo-IMU	SIP Err	24.94 (± 7.15)	**24.78 (± 7.24)**
	Ang Err	11.01 (± 3.65)	**10.89 (± 3.62)**
	Pos Err	8.83 (± 2.91)	**8.77 (± 2.88)**
	Mesh Err	10.01 (± 3.37)	**9.92 (± 3.19)**

For each performance metric, the value achieved by the method with the best performance is bold.

convolution kernel. Instead, in order to fully utilize the spatial priors, we split the adjacency matrix and construct three matrices representing the centripetal set, the node itself, and the centrifugal set, respectively, to initialize three convolution kernels.

Here, we attempt to demonstrate the effectiveness of this strategy by comparing GAIP with a variant GAIP-uni. In GAIP-uni, the adjacency matrix is used to initialize the convolution kernel in the s-GC modules. GAIP and GAIP-uni were evaluated offline on DIP-IMU [16], TC-IMU (real data from TotalCapture Dataset [32]), SingleOne-IMU, and Mixamo-IMU with the same metrics mentioned in the main body. The relevant quantitative experimental results are summarized in Table 7.

From Table 7, it can be seen that GAIP basically achieves smaller joint rotation loss and joint position loss. This phenomenon confirms the effectiveness of the convolution kernel initialization strategy in our s-GC modules.

4.4.4 Training Strategies. In the comparative experiments of training strategies, we tested two training strategies. (1) "pre-training + fine-tuning": this strategy performs pre-training on the AMASS dataset, and then fine-tunes on the training set merged from the DIP-IMU dataset and the SingleOne-IMU dataset. (2) "unified training": this strategy performs training on the unified data that combine the training sets of the AMASS dataset, the DIP-IMU dataset, and the SingleOne-IMU dataset. The results of models with different training strategies on different datasets are provided in Table 8.

From Table 8, it can be seen that the performance difference between the model obtained by "pre-training + fine-tuning" and the model obtained by "unified training" on datasets is limited. The model trained by "pre-training + fine-tuning" performs slightly better on the SingleOne-IMU dataset, while the model trained by "unified training" performs better on other datasets. From these results, it can be concluded that the model achieved by unified training that performs better on more test datasets has a stronger generalization ability. However, given that fine-tuning strategies

Table 8. Results of the Ablation Study of the Training Strategies

		GAIP (Pre-Train + Fine-Tune)	GAIP (Unified Train)
DIP-IMU	SIP Err	15.44 (± 6.49)	**14.60 (± 6.99)**
	Ang Err	8.04 (± 3.93)	**7.76 (± 3.97)**
	Pos Err	5.23 (± 2.53)	**5.01 (± 2.81)**
	Mesh Err	6.12 (± 3.25)	**5.99 (± 3.28)**
TC-IMU	SIP Err	14.93 (± 6.32)	**12.59 (± 6.86)**
	Ang Err	12.48 (± 5.56)	**12.02 (± 5.71)**
	Pos Err	5.61 (± 2.95)	**5.56 (± 3.16)**
	Mesh Err	6.57 (± 3.55)	**6.44 (± 3.60)**
SingleOne-IMU	SIP Err	**18.29 (± 8.75)**	18.32 (± 8.67)
	Ang Err	8.78 (± 3.95)	**8.73 (± 3.92)**
	Pos Err	**6.14 (± 3.14)**	6.15 (± 3.14)
	Mesh Err	**6.84 (± 3.37)**	**6.84 (± 3.37)**
Mixamo-IMU	SIP Err	24.80 (± 7.23)	**24.78 (± 7.24)**
	Ang Err	10.91 (± 3.63)	**10.89 (± 3.62)**
	Pos Err	8.83 (± 2.89)	**8.77 (± 2.88)**
	Mesh Err	10.00 (± 3.20)	**9.92 (± 3.19)**

For each performance metric, the value achieved by the method with the best performance is bold.

are very diverse and have different effects, a more appropriate fine-tuning strategy may further optimize the performance of the model.

5 Conclusion

In this article, we propose GAIP, a framework for reconstructing complete human pose using six inertial sensors. GAIP designs a multi-stage regressor comprising s-GC modules to fuse topological information from HKT and a joint position loss to implicitly learn spatial information with forward kinematics. GAIP also conducts adversarial learning to optimize the regressor's parameters. Extensive comparative and ablation experiments demonstrate that our GAIP achieves higher pose estimation accuracy than the state-of-the-art counterparts. Additionally, we release a real dataset SingleOne-IMU, including hip-supported and foot-supported movements, and a synthetic dataset Mixmo-IMU, with diverse movements for SI-HPE training and evaluation. However, our estimated limb poses are still not accurate enough for poses with wide range motions, and thus we will continue to devote efforts in this area. In addition, the sparse IMU HPE method is extremely dependent on a powerful human body model, such as the SMPL model, and how to get rid of this dependence is also a direction worth studying in the future.

References

[1] Norhafizan Ahmad, Raja Ariffin Raja Ghazilla, Nazirah M. Khairi, and Vijayabaskar Kasi. 2013. Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications. *International Journal of Signal Processing Systems* 1, 256–262.

[2] Eric R. Bachmann, Robert B. McGhee, Xiaoping Yun, and Michael J. Zyda. 2001. Inertial and Magnetic Posture Tracking for Inserting Humans into Networked Virtual Environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. 9–16.

[3] Emad Barsoum, John Kender, and Zicheng Liu. 2018. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1418–1427.

[4] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 7291–7299.

[5] Zhongzheng Cao, Rui Wang, Xiangyang Wang, Zhi Liu, and Xiaoqiang Zhu. 2019. Improving Human Pose Estimation with Self-attention Generative Adversarial Networks. In *IEEE International Conference on Multimedia & Expo Workshops*. 567–572.
[6] Chia-Jung Chou, Jui-Ting Chien, and Hwann-Tzong Chen. 2018. Self Adversarial Training for Human Pose Estimation. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. 17–30.
[7] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. 2019. Optimizing Network Structure for 3D Human Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2262–2271.
[8] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine* 35, 53–65.
[9] Yudi Dai, Yitai Lin, Chenglu Wen, Siqi Shen, Lan Xu, Jingyi Yu, Yuexin Ma, and Cheng Wang. 2022. HSC4D: Human-Centered 4D Scene Capture in Large-scale Indoor-Outdoor Space Using Wearable IMUs and LiDAR. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 6792–6802.
[10] Xiaoxiao Du, Ram Vasudevan, and Matthew Johnson-Roberson. 2019. Bio-LSTM: A Biomechanically Inspired Recurrent Neural Network for 3-D Pedestrian Pose and Gait Prediction. *IEEE Robotics and Automation Letters* 4, 1501–1508.
[11] Mattia Guidolin, Emanuele Menegatti, and Monica Reggiani. 2022. UNIPD-BPE: Synchronized RGB-D and Inertial Data for Multimodal Body Pose Estimation and Tracking. *Data* 7, 79 (2022), 1–14.
[12] Dan Guo, Kun Li, Bin Hu, Yan Zhang, and Meng Wang. 2024. Benchmarking Micro-Action Recognition: Dataset, Method, and Application. *IEEE Transactions on Circuits and Systems for Video Technology* (2024).
[13] Dan Guo, Shengeng Tang, and Meng Wang. 2019. Connectionist Temporal Modeling of Video and Language: A Joint Model for Translation and Sign Labeling. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 751–757.
[14] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2018. Hierarchical LSTM for Sign Language Translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 6845–6852.
[15] Fuyang Huang, Ailing Zeng, Minhao Liu, Qiuxia Lai, and Qiang Xu. 2020. Deepfuse: An IMU-aware Network for Real-time 3D Human Pose Estimation from Multi-View Image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 429–438.
[16] Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J. Black, Otmar Hilliges, and Gerard Pons-Moll. 2018. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *ACM Transactions on Graphics* 37, 185:1–185:15.
[17] Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W. Winkler, and C. Karen Liu. 2022. Transformer Inertial Poser: Real-Time Human Motion Reconstruction from Sparse IMUs with Simultaneous Terrain Generation. In *Special Interest Group for Computer Graphics and Interactive Techniques Asia 2022 Conference Papers*. 3 1–9.
[18] Wonhui Kim, Manikandasriram S. Ramanagopal, Charles Barto, Ming-Yuan Yu, Karl Rosaen, Nick Goumas, Ram Vasudevan, and Matthew Johnson-Roberson. 2019. Pedx: Benchmark Dataset for Metric 3-D Pose Estimation of Pedestrians in Complex Urban Intersections. *IEEE Robotics and Automation Letters* 4, 1940–1947.
[19] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907.
[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. 2020. VIBE: Video Inference for Human Body Pose and Shape Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5253–5263.
[21] Jogendra N. Kundu, Maharshi Gor, and R. Venkatesh Babu. 2019. BiHMP-GAN: Bidirectional 3D Human Motion Prediction GAN. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 8553–8560.
[22] Kun Li, Dan Guo, and Meng Wang. 2021. Proposal-Free Video Grounding with Contextual Pyramid Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1902–1910.
[23] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. 2020. Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 214–223.
[24] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: A Skinned Multi-person Linear Model. *ACM Transactions on Graphics* 34, 248:1–248:16.
[25] Matthew M. Loper and Michael J. Black. 2014. OpenDR: An Approximate Differentiable Renderer. In *Proceedings of the Computer Vision–ECCV 2014: 13th European Conference*, Part VII 13. 154–169.
[26] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture as Surface Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5442–5451.
[27] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. arXiv:1611.09904.
[28] Deepak Nagaraj, Erik Schake, Patrick Leiner, and Dirk Werth. 2020. An RNN-Ensemble Approach for Real Time Human Pose Estimation from Sparse IMUs. In *Proceedings of the 3rd International Conference on Applications of Intelligent Systems* Vol. 32. 1–6.

[29] Pourya Shamsolmoali, Masoumeh Zareapoor, Huiyu Zhou, and Jie Yang. 2020. AMIL: Adversarial Multi-instance Learning for Human Pose Estimation. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 1s (2020), Article 23, 1–23.
[30] Lei Tian, Peng Wang, Guoqiang Liang, and Chunhua Shen. 2021. An Adversarial Human Pose Estimation Network Injected with Graph Structure. *Pattern Recognition* 115, 1–9.
[31] Yushuang Tian, Xiaoli Meng, Dapeng Tao, Dongquan Liu, and Chen Feng. 2015. Upper Limb Motion Tracking with the Integration of IMU and Kinect. *Neurocomputing* 159, 207–218.
[32] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *Proceedings of 28th British Machine Vision Conference*. 1–13.
[33] Timo Von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. 2018. Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera. In *Proceedings of the European Conference on Computer Vision*. 601–617.
[34] Timo Von Marcard, Gerard Pons-Moll, and Bodo Rosenhahn. 2016. Human Pose Estimation from Video and IMUs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 1533–1547.
[35] Timo Von Marcard, Bodo Rosenhahn, Michael J. Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. In *Proceedings of Computer Graphics Forum*. 349–360.
[36] Bastian Wandt and Bodo Rosenhahn. 2019. RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 7782–7791.
[37] Fei Wang, Dan Guo, Kun Li, and Meng Wang. 2024. Eulermormer: Robust Eulerian Motion Magnification via Dynamic Filtering within Transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 5345–5353.
[38] Shuo Wang, Dan Guo, Wen-gang Zhou, Zheng-Jun Zha, and Meng Wang. 2018. Connectionist Temporal Fusion for Sign Language Translation. In *Proceedings of the ACM international conference on Multimedia*. 1483–1491.
[39] Frank J. Wouda, Matteo Giuberti, Nina Rudigkeit, Bert-Jan F. van Beijnum, Mannes Poel, and Peter H. Veltink. 2019. Time Coherent Full-body Poses Estimated Using only Five Inertial Sensors: Deep Versus Shallow Learning. *Sensors* 19, 17 (2019), 3716.
[40] Hailun Xia and Meng Xiao. 2020. 3D Human Pose Estimation With Generative Adversarial Networks. *IEEE Access* 8, 206198–206206.
[41] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
[42] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. 2022. Physical Inertial Poser (PIP): Physics-Aware Real-time Human Motion Tracking from Sparse Inertial Sensors. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 13167–13178.
[43] Xinyu Yi, Yuxiao Zhou, and Feng Xu. 2021. Transpose: Real-Time 3D Human Translation and Pose Estimation with Six Inertial Sensors. *ACM Transactions on Graphics* 40, 86.
[44] Liang Zheng, Yujia Huang, Huchuan Lu, and Yi Yang. 2019. Pose-Invariant Embedding for Deep Person Re-Identification. *IEEE Transactions on Image Processing* 28, 4500–4509.
[45] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. 2019. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. 5745–5753.

Received 3 September 2023; revised 19 May 2024; accepted 25 May 2024