IMUCoCo: Enabling Flexible On-Body IMU Placement for Human Pose Estimation and Activity Recognition

Haozhe Zhou
Carnegie Mellon University
Pittsburgh, PA, USA
haozhezh@cs.cmu.edu

Riku Arakawa
Carnegie Mellon University
Pittsburgh, PA, USA
rarakawa@cs.cmu.edu

Yuvraj Agarwal
Carnegie Mellon University
Pittsburgh, PA, USA
yuvraj@cs.cmu.edu

Mayank Goel
Carnegie Mellon University
Pittsburgh, PA, USA
mayankgoel@cmu.edu

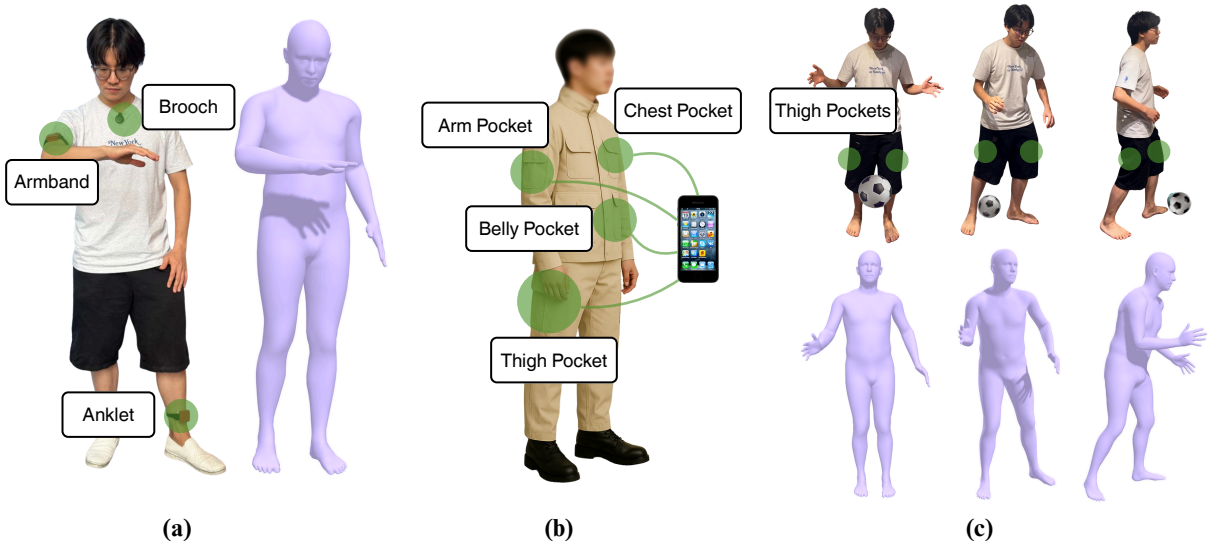

Figure 1: (a) IMUCoCo enables pose estimation from atypical locations like the upper arm, upper chest, and ankle. (b) IMUCoCo allows users to put their IMU sensing devices in different pockets of their clothing for convenience. (c) With IMUCoCo, users can move the sensors to appropriate locations according to different application requirements (*e.g.*, IMUs in the thigh pockets to track leg movements during soccer).

Abstract

IMUs are regularly used to sense human motion, recognize activities, and estimate full-body pose. Users are typically required to place sensors in predefined locations that are often dictated by common wearable form factors and the machine learning model's training process. Consequently, despite the increasing number of everyday devices equipped with IMUs, the limited adaptability has significantly constrained the user experience to only using a few well-explored device placements (*e.g.*, wrist and ears). In this paper, we rethink IMU-based motion sensing by acknowledging that signals can be captured from any point on the human body. We introduce **IMU** over **Co**ntinuous **Co**ordinates (IMUCoCo), a novel framework that maps signals from a variable number of IMUs placed on the body surface into a unified feature space based on their spatial coordinates. These features can be plugged into downstream models for pose estimation and activity recognition. Our evaluations demonstrate that IMUCoCo supports accurate pose estimation in a wide range of typical and atypical sensor placements. Overall, IMUCoCo supports significantly more flexible use of IMUs for motion sensing than the state-of-the-art, allowing users to place their sensors-laden devices according to their needs and preferences. The framework also supports the ability to change device locations depending on the context and suggests placement depending on the use case.

UIST '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2037-6/2025/09
https://doi.org/10.1145/3746059.3747695

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; *Interactive systems and tools.*

Keywords

pose estimation, activity recognition, on-body IMU

ACM Reference Format:

Haozhe Zhou, Riku Arakawa, Yuvraj Agarwal, and Mayank Goel. 2025. IMUCoCo: Enabling Flexible On-Body IMU Placement for Human Pose Estimation and Activity Recognition. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28-October 1, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3746059.3747695

1 Introduction

The ubiquity offered by IMUs has made them an attractive sensor for sensing human motion and pose [2, 18]. Consumer IMUs are used for many applications, such as gait analysis using foot-mounted devices [49], fall risk estimation using a smartphone [16], hyperactivity monitoring using smartwatches [3], and human pose estimation using different consumer devices [46, 48].

However, the inherent assumption in these approaches is that each IMU, while part of a consumer device, is still a specialized device, worn at specific locations. Under this assumption, machine learning models have to be trained for specific devices and placements, which limits their practicality. For example, IMUPoser [28] inferred 3D body pose using a few common device placements, such as pants pocket, ear, and wrist. The authors built a model that adapted to varying availability of devices and worked on consumer products. However, a user might want to move the same device to different locations over the course of a day. Current models fail to adapt when users choose to carry their phone in a jacket pocket, wear it on an armband while exercising, or mount it on their back as a posture tracker while seated. For each new location, the ML models need to be retrained, and that severely limits the flexibility of any ML model aimed at ubiquitous human motion sensing.

To bridge this gap, we need to reconsider the potential source of IMU signals for training the models and not remain limited to a set of key locations, but to any point on the human body's surface. Figure 2 shows the full-body acceleration of a golf swing captured by a camera-based motion capture system. Each arrow in the figure represents the acceleration vector synthesized at that point. It also plots the temporal changes in synthesized acceleration for 6 key points on the user's arm. Signals from devices placed on regions that move together appear to be kinematically related, with some cases observed in previous studies [14, 43]. Although sometimes certain parts of the body can move entirely independently, in many other situations, these correlations suggest the existence of an underlying mechanism that connects such signals. Inspired by this observation, the question arises: **Can we create a unified model that can take the IMU data captured from any point on the human body surface for human motion sensing?**

To achieve this vision, there are several technical challenges that need to be addressed. First, no existing machine learning architecture for IMU data is designed to process input from all possible points on the human body surface. Existing approaches rely on creating separate feature encoders for each location (*e.g.*, [17]) or training together with multiple inputs (*e.g.*, [28]). Thus, learning a model for a large number of inputs remains unscalable, especially when the number of possible input locations is infinite. Second, there is no existing dataset that includes data from nearly infinite possible IMU locations. To our knowledge, the most comprehensive IMU-based dataset yet uses 17 real IMUs [15], which is still far from our goal. Third, a training framework to learn the relationship between the large number of IMU signals on the body surface and the underlying human motion has not yet been built.

In this paper, we present *IMUCoCo*, a novel approach enabling IMU placement over "Continuous Coordinates" that maps IMU signals obtained from different locations on the body into a unified feature space defined by the spatial coordinates of the sensor. IMUCoCo models the kinematic relationship between body locations using a self-supervised learning strategy trained on extensive synthetic IMU signals generated from existing motion capture datasets [11, 13, 15, 24, 26, 29, 31]. To address the scalability challenge, IMUCoCo learns to align IMU signals placed flexibly on the body's surface with a constant number of human joint movements. The unified mechanism for modeling body motion is driven by the insight that signals across the body surface are influenced by the shared kinematic structure connected by the joints of the human body, thus leading to a model of the body motion regardless of the number of input signals and does not need to be retrained when placements change.

To demonstrate the performance of our system, we conducted a series of evaluations. In Evaluation #1, we collected a custom dataset for IMU data at atypical locations on a user's body and ground truth recorded using camera-based motion capture. We demonstrated that our approach generalizes well to all atypical locations that are not covered by prior approaches. In Evaluation #2, we showed that for typical locations, our approach achieves competitive performance compared to the existing work [46, 51–53]. Finally, we demonstrate IMUCoCo's utility in supporting different usage scenarios ranging from allowing users to use atypical device placements, the ability to change device locations depending on the context, and suggesting placement based on the use case. Thus, IMUCoCo provides a human motion sensing model that allows users to wear their devices as they need or prefer. It also supports users in tracking their bodies for specific applications by allowing them to move devices as needed.

2 Related Work

Our research builds on existing studies that utilize on-body IMUs for human motion sensing. In this section, we provide a brief overview of these efforts. We also surveyed recent advances in other on-body sensors that might augment IMUs for motion sensing in the future.

2.1 Body Pose Tracking from IMUs

Inside-out full-body pose tracking from sparse sensor sets has been intensively studied using different sensing modalities, such as vision [2, 18, 37], RFID [19], pressure [10], acoustic [25], and electromagnetic-field [5, 45]. Among these, IMU-based solutions are often more deployable, as they are present on many consumer devices. These solutions learn the mapping between the IMU signals from designated locations on the body to the body pose parameters, which is often represented as the SMPL model [23]. For instance, Sparse Inertial Poser [42] and Deep Inertial Poser [15] are pose tracking systems with 6-17 on-body IMUs. Several other works

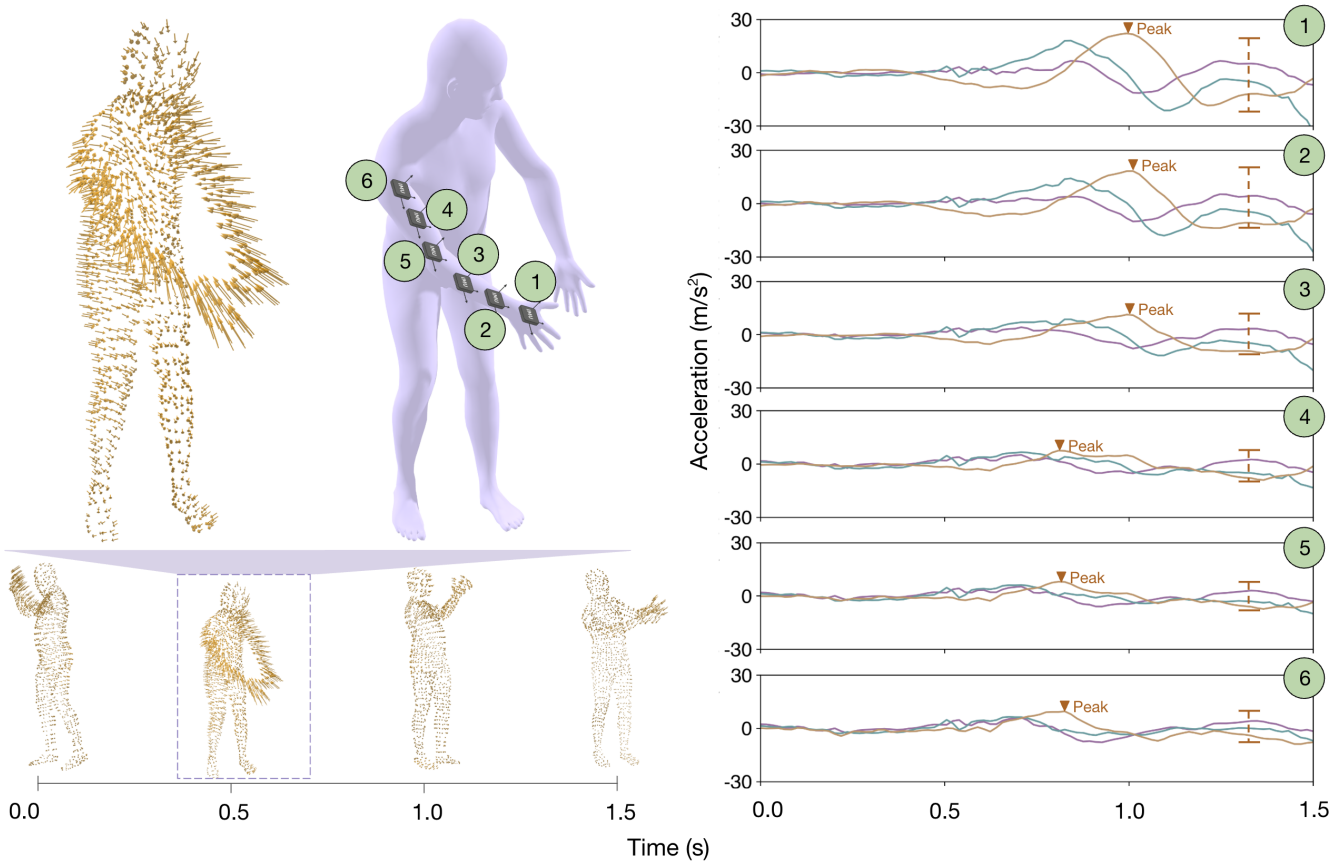

Figure 2: Synthesized acceleration signals across the body surface from a person swinging a golf club from T=0.0s to T=1.5s. The direction and magnitude of acceleration are visualized as arrows (Left Figure). The acceleration selected from 6 "devices" on the right arm from hand to shoulder is plotted (Right Figure), corresponding to the six devices on the avatar's right arm. The magnitude and timing of the peak in the acceleration signal at each location are different. IMUCoCo models such differences and supports placement and movement of the IMU to different points on the body to give the user flexibility.

have been proposed to improve the accuracy of these pose tracking systems [51–53, 57]. Researchers have also explored ways for more practical device placements; for example, IMUPoser [28] and MobilePoser [48] support different combinations of commercial devices, including earbuds, phones, and watches.

Although current systems achieve high accuracy in pose estimation, they rely on placing IMUs on predefined body parts. This rigidity restricts adaptability, particularly when encountering placements not represented in the training data. For example, what if a user prefers to wear an anklet instead of a watch? While recent approaches such as DiffusionPoser [46] attempt to generalize by using diffusion models to infer missing sensor data, they are still confined to 13 predetermined sensor locations. Consequently, sensor placements outside these predefined points remain unsupported. Additionally, the computationally intensive multi-step diffusion process significantly hampers real-time deployment on resource-limited devices. Crucially, existing research overlooks opportunities to explore the vast and continuous space of the human body for sensor placement, potentially missing configurations that could yield superior accuracy and usability. In this paper, we explore this opportunity and demonstrate its benefits.

2.2 Physical Activity Recognition from IMUs

Body motion data can also enable human activity recognition [4, 6] and exercise tracking [40, 47]. Several datasets have been developed that collect sensor data from diverse devices such as smartphones and smartwatches during various daily physical activities, including walking and climbing stairs [8, 21, 35]. Using these datasets, numerous machine learning-based approaches have emerged [7, 34, 56]. For example, Müller *et al.* [30] developed a method specifically for exercise tracking using IMUs placed on the wrists and ankles. Additionally, Kwon *et al.* [22] introduced a technique for synthesizing IMU data from videos, allowing more flexible activity recognition.

However, these methods continue to follow the fixed-placement approach previously discussed in body pose tracking research, relying on a predetermined set of IMU locations. To overcome this limitation, some studies have investigated location-invariant methods. Rey *et al.* [36] proposed a method to transfer sensor data across

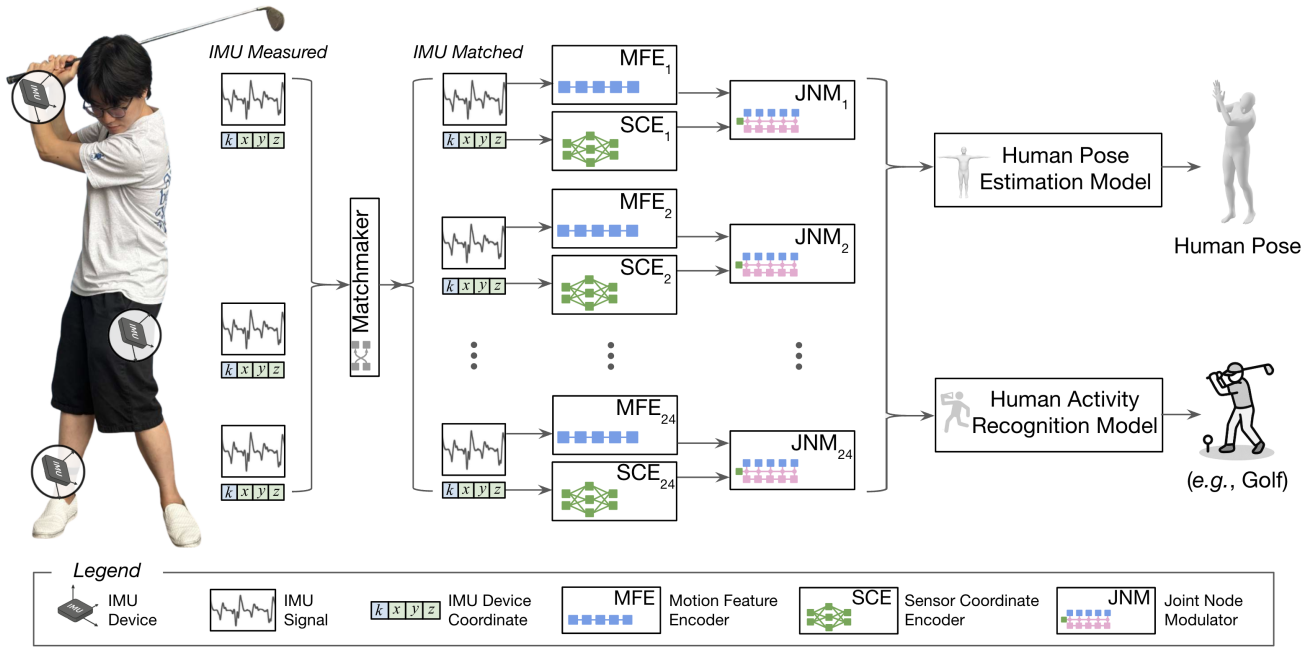

Figure 3: Overview of the IMUCoCo system during test time. IMU data is mapped into a placement-adaptive representation, which can be easily tuned for downstream tasks, such as pose estimation and activity recognition. Modules and synthetic IMUs that are only used during training time are omitted from the figure for brevity of presentation.

typical placements (*e.g.*, from wrist to ankle). SenseHAR [17] combined multiple sensor types from wearables and phones into a shared latent feature space, enabling downstream activity recognition models to train on unified features. Adaimi *et al.* [1] developed a location-invariant model using a large-scale dataset suitable for diverse activities such as driving. Still, these location-invariant approaches are limited to the sensor placements explicitly captured in their datasets. In contrast, we demonstrate that IMUCoCo can seamlessly integrate into activity recognition pipelines, accommodating a continuous range of sensor placements beyond those predefined locations.

2.3 Advancements in On-Body Devices

Existing on-body sensing research predominantly utilizes common wearable locations, such as glasses, wrists, and thigh pockets. Beyond these typical placements, researchers have also explored other locations and form factors, such as embedding sensors within everyday accessories [12], magnetic sensing through rings [32], multimodal eating detection via necklaces [55], and touch input through belts [9]. Recent advances in fabrication technologies and robotics have further expanded feasible sensor placements across continuous body surfaces. For instance, Yu *et al.* [54] introduced a scarf-shaped device integrated with electrical impedance tomography for activity recognition. Similarly, SkinMarks [44] used stretchable electronic tattoos strategically placed on epidermal landmarks, transforming natural skin features and limb movements into interactive touch inputs. We anticipate that these innovations will facilitate the integration of IMU sensors to capture complete body motion data in the future. Our proposed system, IMUCoCo, can potentially benefit from these advances to enable rapid prototyping of motion sensing systems without requiring additional data collection or extensive model training.

3 Design of IMUCoCo

3.1 System Overview

3.1.1 Insights. IMUCoCo's architecture aims to enable scalable learning of mapping countless potential placements of IMUs into a tractable space. Unlike conventional approaches, IMUCoCo enumerates a massive number of possible placements at training time. For this reason, we must restrict the growth of the architecture size with respect to the number of IMU placements.

We draw insights from several related fields. Articulated human pose models [23] render realistic human body meshes from compact parameters, such as joint rotations and body shapes. Techniques such as blend skinning formulate mesh surfaces as weighted bone transformations. Inspired by this, IMUCoCo is designed to align IMU signals to joint movement representations, limiting the target space to a constant size. In addition, coordinate-based or implicit neural representation approaches model signals as a neurally parameterized function of coordinates, demonstrating a powerful ability to perform tasks such as view synthesis [27] and compression [38]. Leveraging this idea, IMUCoCo modulates the IMU signals by their coordinates, effectively encompassing the spatial relationship among various placements without repeatedly increasing the size of the input layers.

3.1.2 Architecture. The IMUCoCo maps a variable number of IMU devices placed on the body to a unified space, so that downstream models, such as a pose estimation model, can function without retraining for different IMU placements. The placements of the devices are represented as spatial coordinates in the 3-dimensional space, formally as $\mathbf{r} = (x, y, z)$. This spatial coordinate is measured relative to the root of the human body during a standard T-pose and remains unchanged until the IMU device is repositioned or removed from the human body. Such location information can be easily obtained for users and does not need to be error-free. For example, one approach is simply to tap on the corresponding location on a rendered avatar or simply state one of the predefined locations, such as the neck, ankle, or elbow. We discuss other potential approaches to obtain coordinates in real life in Section 6.

Inspired by the kinematic trees used for articulated human body models [23], IMUCoCo breaks the whole body feature space into 24 joint nodes, represented as $\mathbf{z} = \{\mathbf{z}_1, .., \mathbf{z}_{24}\}$, which corresponds to the motion of each joint. Each of these joint nodes is then processed through a separate pathway that maps the IMU device's signal into the corresponding joint node features.

As shown in Figure 3, IMUCoCo comprises several modules. First, each IMU's signal is passed through a Motion Feature Encoder (MFE) to encode the raw IMU data. At the same time, device placement is encoded using a Sensor Coordinate Encoder (SCE) to derive placement codes that inform the transfer function for the corresponding features into the target joint node feature. After this step, the Joint Node Modulator (JNM) takes both the extracted IMU features and the placement codes. It modulates the features to produce a representation that describes the target joint's movement. To train these modules, the features transferred to the joint node are passed to auxiliary regression tasks using Kinematics Regressors (KRs) to regress to kinematics attributes, including velocity, position, and orientation, as well as a full-body Pose Regressor (PR) to infer the full-body pose. The KRs and PR are dropped once the training is completed. A detailed training architecture and procedure are illustrated in Appendix A.2. In the next subsections, we describe each module of IMUCoCo, placement adaptation, and applying IMUCoCo for downstream tasks in further detail.

3.2 IMUCoCo Modules

3.2.1 Motion Feature Encoder (MFE). The Motion Feature Encoder (MFE) module encodes the raw IMU input into the IMU feature representations. This module is analogous to the conventional method of feature extraction for IMU signals. The MFE module first projects the data from a single IMU's 9 channels into higher dimensions using a linear layer with ReLU activations. We implement MFE using an LSTM model following previous studies [15, 28, 53]. We chose a single-directional LSTM over a bi-directional one to preserve historical information without the limitation of fixed window sizes [51]. Formally, the MFE module for a joint node is represented by $\mathbf{h} = \text{MFE}(\mathbf{s})$, where $\mathbf{s}$ is one input IMU signal from one point on the body, and $\mathbf{h}$ is the extracted feature.

3.2.2 Sensor Coordinate Encoder (SCE). The Sensor Coordinate Encoder (SCE) module encodes the sensor coordinate $r = (x, y, z)$ into placement codes q that instruct the subsequent modules of IMUCoCo to adapt to the sensor's placement. The detailed structure

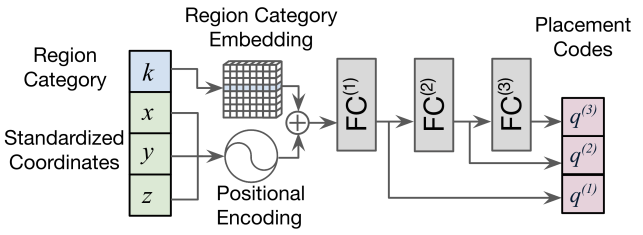

Figure 4: The detailed architecture of the Sensor Coordinate Encoder (SCE) module. The standardized sensor coordinates (x, y, z) are encoded using periodic functions. The sensor region category (k) is encoded using a learnable embedding layer. The concatenated features are passed through fully connected (FC) layers, producing multi-layers of placement codes $\mathbf{q}^{(l)}$ for each layer l.

of SCE is shown in Figure 4. First, we standardize the raw spatial coordinates to have the origin at the target joint location r_j, divided by the range of all vertex spatial coordinates. Formally, this is represented by $r_{st} = (r - r_j)/(r_{max} - r_{min})$, where r_{max} and r_{min} are obtained by taking the maximum and minimum values of each of the three axes on the coordinates of all vertices. Previous research has shown that applying positional encoding helps extract high-frequency changes along coordinates [27]. Following this, the standardized coordinate $\mathbf{r}_{st}$ is passed through a positional encoding using periodic functions. This mapping is defined as:

$$\phi_f(\mathbf{r}_{st}) = [\sin(2\pi f \cdot \mathbf{r}_{st}), \cos(2\pi f \cdot \mathbf{r}_{st})] \tag{1}$$

, where f represents the frequency bands in increasing powers of 2, *i.e.*, $f = 2^p$ for $p = 0, 1, \ldots, n_{freq} - 1$. This encoding transforms r_{st} into a high-dimensional feature space that provides multi-scale spatial information.

Additionally, we partition the human body surface into 24 regions based on joint positions. Our intuition is that the region where the sensor is placed provides a semantic meaning that is useful to determine the transference as well. A detailed illustration of the partition is provided in Appendix A.2. Thus, for each IMU sensor, we categorize its placement region into one of the 24 regions, denoted as k, based on the provided coordinate r. Subsequently, we encode the category of regions k corresponding to the spatial coordinate using a learnable embedding layer, denoted as *Emb*. The positional encodings and the region embedding are concatenated to form the input feature vector, denoted as $\mathbf{q}^{(0)} = [\phi(r_{st}), \text{Emb}(k)]$, to the subsequent L fully-connected (FC) layers.

The first FC layer takes input from $\mathbf{q}_0$, while subsequent fully connected layers produce placement codes. Formally, for layer l:

$$\mathbf{q}^{(l)} = \text{FC}^{(l)}\left(\mathbf{q}^{(l-1)}\right) \tag{2}$$

, where $\mathbf{q}^{(l)} = (\gamma^{(l)}, \beta^{(l)})$ are the placement codes that subsequently inform the modulation of the motion features based on placement. Finally, the output of the SCE module is a list of placement codes $\mathbf{q} = \{\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(L)}\}$, one for each MLP layer.

3.2.3 Joint Node Modulator (JNM). The Joint Node Modulator (JNM) modulates the features obtained from the MFE module to

the features representing the corresponding joint kinematics on the body based on q, the placement codes generated from the SCE module. Similar to MFE, we adopt an LSTM-based structure to encode the temporal information of the motion feature corresponding to each joint node. For each layer, we applied Feature-wise Linear Modulation (FiLM) [33] to the LSTM outputs to allow modulating the motion features based on the inferred placement codes. The input to the first layer is just the output of the corresponding MFE module $\mathbf{z}^{(0)} = \mathbf{h}$. Formally, for layer l in the JNM module,

$$\mathbf{z}^{(l)} = \text{LSTM}^{(l)}\left(\gamma^{(l)} \odot \mathbf{z}^{(l-1)} + \beta^{(l)}\right) \tag{3}$$

Finally, the output of the JNM module is the last layer's output $z^{(L)}$.

3.2.4 Kinematics Regressor (KR) and Pose Regressor (PR). We used regressions to kinematic attributes as an auxiliary task to learn useful representations of cross-placement IMU signals. To achieve this, we used two linear layers with ReLU activation as a Kinematics Regressor (KR). We used this straightforward architecture for KR primarily to enforce the quality of representations learned only from the previous modules instead of letting a complex model excessively compensate for the performance of the regression task. For each joint node, we used five joint-level KR modules to predict the joint's velocity, position, local orientation, global orientation, and the body root's velocity. In addition, we used one body-level Pose Regressor(PR) module, with the same architecture as KR, that takes the features from all 24 joints and together regresses to the full-body pose. Note that the KRs and PR are used as auxiliary regressors only and will be removed from the model once it is trained.

3.2.5 Matchmaker. When IMUCoCo is supplied with a variable number of IMUs, the Matchmaker module dynamically allocates joint nodes to the optimal IMU device based on a loss map. Specifically, after training, for each joint node, we iterate through all vertices on the body, assuming that an IMU is placed at each vertex, and compute the loss values for transferring information to the target joint node. This process is repeated for all 24 joints, producing a loss table $\mathbf{M} \in \mathbb{R}^{24\times V}$, where V is the total number of vertices in the articulated pose model, and $\mathbf{M}(j, v)$ represents the loss value of a virtual IMU placed at the vertex v when transferred to the target joint node j. With the constructed loss table, each of the 24 joint nodes will be assigned to one IMU device that gives the lowest loss. Formally, given a set $\mathcal{D}$ of IMU devices attached to the body, where the IMU d is located at the coordinate $\mathbf{r}_d$, the optimally assigned IMU for the joint node j is computed as $d_j^* = \text{argmin}_{d\in\mathcal{D}} \mathbf{M}(j, v_{\mathbf{r}_d})$, where $v_{\mathbf{r}_d}$ denotes the nearest vertex in the loss table to the provided coordinate $\mathbf{r}_d$. For a newly attached or moved device, we use its coordinates to query the table and retrieve 24 loss values, and update the assignment accordingly. Note that this module is only used during test time.

3.3 Training Process

As mentioned in the introduction, the absence of a dedicated dataset for IMU data on continuous body coordinates presents a technical challenge. We introduce an approach to synthesizing virtual IMU data across the entire body mesh, expanding beyond the joint-only focus of previous research. This section explains the approach, followed by details of our training approach.

3.3.1 Virtual IMU Synthesis. Training IMUCoCo requires IMU data sampled from all over the body surface, for which relying on real IMU data is not feasible. We synthesized our IMU data based on human pose datasets, including AMASS [24], DIP-IMU [15], and other XSens-based datasets [11, 13, 26, 29, 31]. From these datasets, we obtained full body pose and used the SMPL model [23] for forward kinematics to determine the positions of joints and mesh.

To calculate the acceleration, we applied the second derivative of the positions. Existing work typically simplifies bone orientation to synthesize IMU orientation (*e.g.*, [28]). While this approach may function well when the IMU is attached to areas that do not deform, such as the middle of the upper arm, it is inaccurate for areas that deform, such as abdominal regions and areas close to each joint (*e.g.*, the elbow). Thus, we calculate the orientation based on the faces of the mesh. These virtual IMUs are also virtually calibrated using a T-pose [52] We include a more detailed illustration in Appendix A.1.

3.3.2 Loss Functions. We used a combination of loss functions to establish learning feature representations suitable for the downstream tasks. First, we used kinematic loss $\mathcal{L}_{\text{kinematic}}$ for multiple kinematic attributes, including velocity, root velocity, position, global orientation, and local orientation, to facilitate the model in extracting kinematic quantities. For velocity, position, global orientation, and local orientation, we used Mean Squared Error (MSE) loss. For root velocity, we used multi-frame losses at consecutive 1, 3, 9, and 27 frames [52]. Second, we used the full-body pose loss $\mathcal{L}_{\text{pose}}$ in the global frame, also implemented as an MSE loss, to encourage the model to coordinate organically with the representation from different joint nodes. Note that this fully-body pose loss can only be calculated when all the joint nodes have completed their forward passes, which, if done at the same time, can add an excessive memory burden during training. We used a buffered approach to resolve this, as described in Appendix A.2. Third, we used an alignment loss $\mathcal{L}_{\text{align}}$ based on cosine similarity to encourage IMUCoCo to produce a feature representation for the sampled mesh synthetic IMUs similar to the representation for the joint synthetic IMU. Formally, the training loss can be represented as:

$$\begin{aligned}\mathcal{L}_j(\mathbf{z}_j) = {} & \lambda_{\text{kinematic}}\mathcal{L}_{\text{kinematic}}(KR(\mathbf{z}_j), \mathbf{K}_{j\text{GT}}) \\ & + \lambda_{\text{pose}}\mathcal{L}_{\text{pose}}(PR(\mathbf{z}_j, \mathbf{z}_{\text{buffer}}), \mathbf{P}_{\text{GT}}) \\ & + \lambda_{\text{align}}\mathcal{L}_{\text{align}}(\mathbf{z}_j, \mathbf{z}_{\text{ref}})\end{aligned} \tag{4}$$

, where $\mathbf{K}_{j\text{GT}}$ are the true kinematics of the joint, $\mathbf{P}_{\text{GT}}$ is the ground truth pose, $\mathbf{z}_j$ is the representation obtained from the mesh IMU, $\mathbf{z}_{\text{ref}}$ is the reference representation obtained from joint virtual IMU, and $\mathbf{z}_{\text{buffer}}$ is the buffered representations initially filled with the joint virtual IMU representations while gradually replaced by the mesh IMU representations, and $\lambda_{\text{kinematic}}, \lambda_{\text{pose}}, \lambda_{\text{align}}$ are hyperparameters for weighing the losses.

3.3.3 Training Setup. Training such a model consumes a large dataset, and we designed a training scheme to minimize the computing resources needed. Our training process is split into two phases. In phase one, we train the IMUCoCo model only with the

24 joint virtual IMUs using kinematic loss and pose loss until convergence. In phase two, with the warmed-up model, we sampled mesh IMUs and trained them with kinematic and pose loss while aligning them to joint IMUs. A more detailed procedure is illustrated in Appendix A.2. Overall, we trained our model on one L40S GPU, which has 48GB CUDA VRAM, for 200 hours, including 90 hours of training in the joint-only phase and 110 hours of training in the end-to-end phase. While training IMUCoCo is relatively prolonged, mainly due to the need to consume a large number of samples on the human body surface, this model is still very lightweight with only 23M parameters and feasible for inference without excessive computing.

3.4 Applying IMUCoCo to Downstream Tasks

The features extracted from IMUCoCo enable the downstream model, such as a pose estimation model, to process without worrying about the number or locations of IMU devices. To train a downstream task, one can freeze the IMUCoCo model and feed the IMU data from the devices existing in this dataset for the task. During training, each provided IMU went through the placement adaptation process and was matched to the joint node to produce the extracted representations. For our paper, we explored pose estimation and activity recognition as our downstream tasks.

For pose estimation, we mainly adopt the DynaIP [58] architecture, as a state-of-the-art pose estimation model using 6-IMU inputs. As DynaIP itself does not give translation estimation, we adopt the translation estimation module from TransPose [52]. We use the abbreviation DTP (DynaIP with TransPose) to denote this pose estimation model. For convenience of presentation, in the evaluation section, we use IMUCoCo to refer to IMUCoCo + DTP in the context of pose estimation. We then trained a pose estimator using the conventional 6 IMUs (pelvis, head, left lower arm, right lower arm, left lower leg, right lower leg) from AMASS, DIP-IMU, and XSens dataset [58].

For activity recognition, we used a Spatial-Temporal Graph Convolutional Neural Network (ST-GCN) targeted for skeleton-based activity recognition [50]. We then trained our activity recognition model using the 3 IMUs (wrist, pocket, ear) using custom datasets that we collected ourselves. For convenience of presentation, in the evaluation section, we use IMUCoCo to refer to IMUCoCo + ST-GCN in the context of activity recognition. Both models are trained when freezing the IMUCoCo model and only using the IMU data at the different locations provided in the dataset. We conducted separate ablation experiments that verified that the improved performance is attributable to IMUCoCo rather than the downstream models.

4 Evaluation

4.1 Evaluation Overview

We conducted two studies to evaluate the following hypotheses:

(1) IMUCoCo achieves consistent body motion sensing performance when IMUs are placed at atypical locations.
(2) IMUCoCo delivers comparable body motion sensing performance to existing systems when IMUs are placed at typical locations.

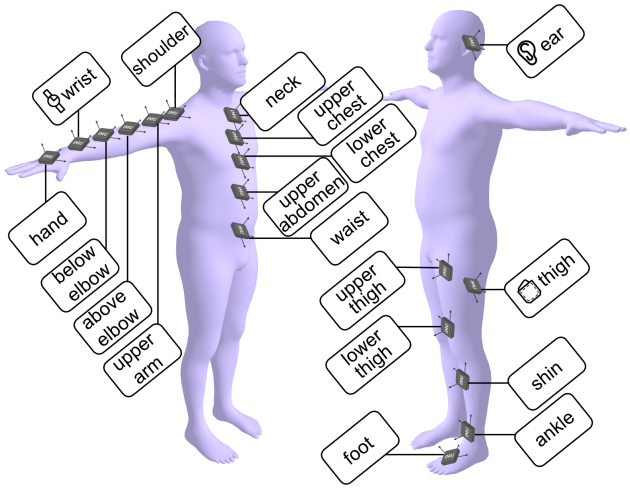

Figure 5: Illustration of the dense IMU placement configurations used in Evaluation #1 shown on a right-handed person. We started with the standard 3 IMU placement (wrist, thigh, ear) to mimic the most popular consumer devices [28]. Additionally, we placed a dense set of IMUs on one of the three body sections (arm, leg, or torso). On the arm, we chose the shoulder, the middle of the upper arm, the upper arm just above the elbow, the forearm just below the elbow, and the hand. On the leg, we chose the upper thigh, lower thigh, shin, ankle, and foot. On the torso, we chose the neck, upper chest, lower chest, upper abdomen, and waist. For a left-handed person, the placements were mirrored accordingly.

For Evaluation #1, we used our newly collected custom dataset to assess performance in body pose tracking and activity recognition across sensor placements. For Evaluation #2, we utilized multiple existing datasets to evaluate body pose tracking capabilities.

4.2 Evaluation #1: Motion Sensing at Atypical Locations

4.2.1 Data Collection. Due to the absence of datasets with dense IMU placement, we collected our own custom dataset to understand IMUCoCo's performance at fine-grained placement variations. We utilized 8 Apple Watches (Series 7 or newer) with a custom data collection application that we implemented to record IMU data at 50 Hz. We collected motion capture data using the OptiTrack system.

We recruited 12 participants from our institution (8 males, 4 females; age range 23-33; 1 left-handed, 11 right-handed) and collected their motion capture data. Participants wore a mocap suit with optical markers, followed by skeleton calibration in the OptiTrack system. The participants also wore watches at three typical locations (ear, wrist, and pocket). Additionally, we placed five more devices on one of the three dense placement configurations: arm, leg, and torso, as illustrated in Figure 5. Participants then performed jumps to align timestamps across all IMU devices. For each placement configuration, participants performed a set of predefined activities inspired by previous research [17], designed to encompass

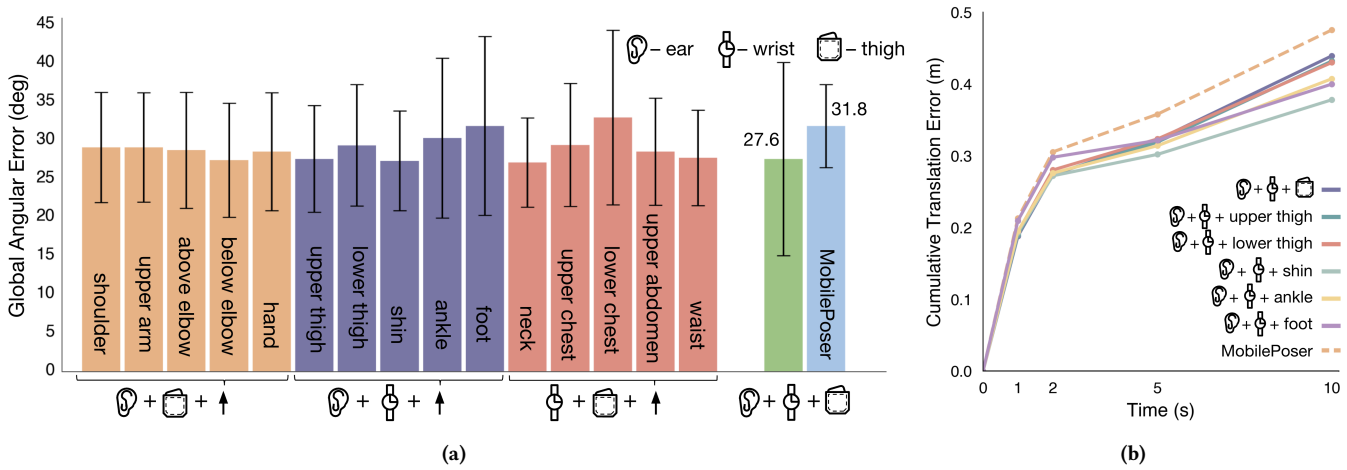

Figure 6: (a) Pose tracking Global Angular Error (GAE) is measured at different sensor placements, averaged over all activities. IMUCoCo achieves lower error than the state-of-the-art [48] when IMUs are placed at typical positions (bar graph on the right in (a)). A similar effect is seen at all atypical positions (*i.e.*, bars on the left). Error bars show standard deviation. (b) The same effect was seen for cumulative translation error using different sensor placements for the lower body. IMUCoCo achieves consistently lower translation error than MobilePoser over time. The effect remains the same as the participant changes the device location along their leg.

diverse body movements: walking, running, vacuuming, watching television, drinking water, table cleaning, golf swing, shot put, and squats. Before each activity, the participants performed a T-pose for calibration purposes [52], with each activity lasting 30 seconds. Participants rested between placement configurations as needed, and the entire data collection process required approximately one hour per participant. All activities were video-recorded with timestamps for subsequent annotation. Following data collection, we extracted the OptiTrack body pose data, synchronized it with all IMU sensor readings, and annotated the dataset with activity labels. Overall, our collected dataset consists of 3 hours of body motion data in total. All studies are approved by our institution's IRB.

4.2.2 Body Pose Tracking. We first evaluated the body pose tracking performance of IMUCoCo by comparing different sensor placements, specifically examining how error metrics vary as the IMU sensor is progressively moved along a body part (*i.e.*, arm, leg, or torso). Consistent with prior research [46], we employ Global Angular Error (GAE) [1] that serves as our primary metric for pose estimation quality, which quantifies the rotational discrepancy between the ground truth and the reconstructed global segment orientations. It is important to note that state-of-the-art systems, such as MobilePoser [48], do not support IMUs positioned at unconventional locations. Therefore, to keep the comparison fair, we compare IMUCoCo and previous work for sensors placed in standard locations. For atypical positions, we compare the performance within IMUCoCo's output.

These comparisons are summarized in Figure 6a. Overall, IMUCoCo showed consistency in performance as the IMU device moves along the arm, leg, and torso regions. In particular, we observed only slight variation as the IMU is repositioned along the arm. Slightly higher errors appear when the ear sensor is moved to the lower chest. We attribute this mainly to the flexibility of clothing, where the physical IMU often folds or shifts together with the fabric, introducing additional motion artifacts that are not representative of true body movement. In addition, IMUCoCo demonstrated superior performance (GAE = 27.6°) in pose estimation than MobilePoser [48] (GAE = 31.8°) ($p < .001$). Similarly, IMUCoCo showed resistance in translation estimation as the user moves the leg IMU from thigh pockets to other areas across the leg, as shown in Figure 6b. It is important to note here that, unlike MobilePoser, IMUCoCo or DTP are not specifically trained or fine-tuned with the IMU at these locations (wrist, thigh pocket, ear), and thus even the standard placement relies on IMUCoCo's capability to transfer input IMU signals to the DTP model that were trained using the 6-IMU configuration.

We then measured the pose estimation performance from IMUCoCo using standard placement (wrist, thigh pocket, ear) to the optimal placement for each type of activity. Figure 7 summarizes the result. We see that selecting the best placement for IMUCoCo enables greater tracking performance (GAE = 24.8°) compared to the standard location (GAE = 27.6°) ($p < .001$). For some specific activities, changing the placement appears to have more impact on the pose estimation accuracy. For instance, golf swings are better captured by moving the IMU from the wrist to below the elbow (or the forearm). For computer work, moving the IMU from the thigh pocket to the ankle produces less error (GAE=22.8°) than the standard location of keeping the sensor in the pocket (GAE=30.9°). We examined the inferred pose difference and observed that the primary difference is the angle of the hip with respect to the ground when sitting. We consider that this difference arises mainly due to the more rigid nature of the lower leg clothing and bones, compared

[1]We followed the procedure with DiffusionPoser [46], where we ignored root, wrists, fingers, and toes joints and excluded them from averaging the final error calculations. For this, our reported error values may appear greater than their originally reported ones.

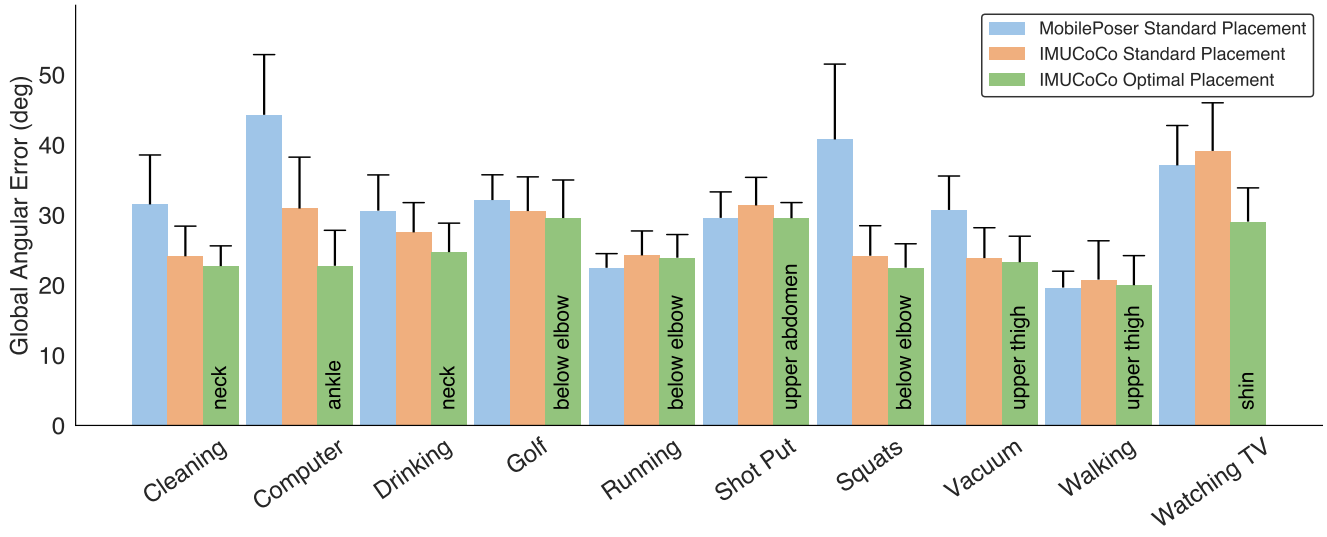

Figure 7: Comparison of Global Angular Error (GAE) when the sensor is placed at Standard Placements (wrist, pocket, ear) and when one of the three sensors is moved to an optimal location. For example, for the Golf activity, the model generates a better pose (*i.e.*, lower error) when the IMU on the wrist is moved below the elbow. The Figure also shows MobilePoser's GAE for the standard placement. Error bars show standard deviation.

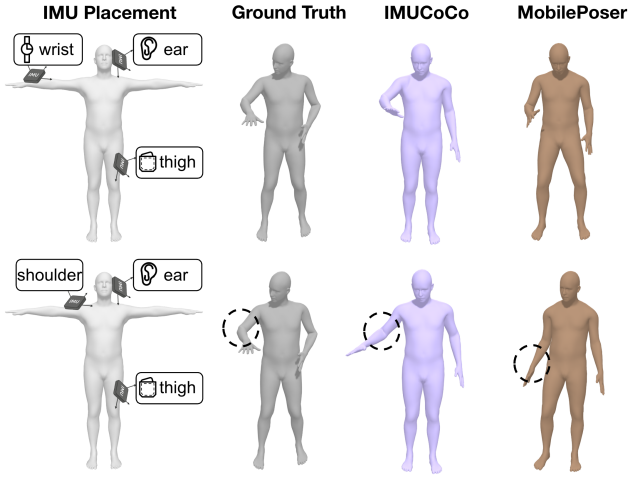

Figure 8: A demonstration of pose estimation when the sensor on the arm is moved, *i.e.*, from the wrist (top row) to just below the shoulder (bottom row). When the user performs a sweeping motion (ground truth), IMUCoCo is able to adapt to the new IMU placement and infer the upper arm motion accurately (bottom row, third column). MobilePoser still assumes the input was at the wrist and incorrectly only slightly raises the forearm (bottom row, fourth column).

to the IMU in the upper thigh pocket, which introduces additional flexibility when sitting. Thus, if a user wants to better capture a specific type of posture, IMUCoCo can recommend and support optimal sensor placement.

We also evaluated the pose estimation results with different sensor placements in more detail to understand the sources of errors. Consistent with our hypothesis, IMUCoCo effectively adapts the sensor based on its placement coordinate, and renders the pose reasonably based on the sensor placement. Figure 8 shows an example motion of sweeping to clean a table, which primarily involves arm movements while leaning forward. As the IMU is repositioned from the wrist to the upper arm (below the shoulder), IMUCoCo can adapt the signal and accurately infer the upper arm motion. We attempted to retrieve the pose estimation from MobilePoser [48], which is not designed to take IMU input from the upper arm. As expected, the state-of-the-art that is not designed to adapt to new sensor placements did not render the pose accurately. It is important to note that IMUCoCo still lacks sensor input from the lower arm and can only provide its best prediction by naturally extending the lower arm. In this case, some error in lower arm tracking is anticipated. Based on these observations, we conclude that understanding tracking confidence for all body parts, given specific sensor placements, and controlling the level of model hallucination is essential for critical applications.

4.2.3 Body Activity Recognition. We divided our 12 participants into six groups of 2 and performed a 6-fold cross-validation to examine the activity recognition performance of our approach. Figure 9 summarizes the results of the averaged macro F1-score from 6-fold cross validation. Overall, using the three standard placements, IMUCoCo achieves 73.7 in macro F1 score. We then tested performance by repositioning the wrist IMU at other locations along the arm. As shown in Figure 9, relocation of the IMU to the hand or lower arm (below the elbow) has minimal impact on overall performance, while relocation to the upper arm decreases activity recognition

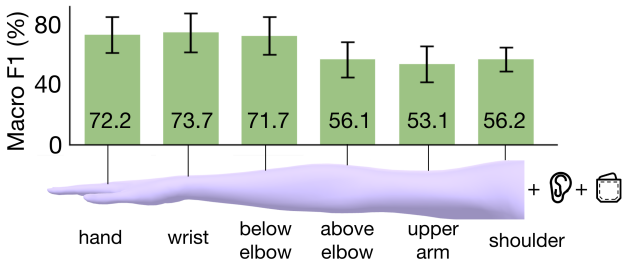

Figure 9: Activity recognition results in macro F1-score across 10 activities with 6-fold cross validation on our custom dataset when placing the arm IMU at different locations. Higher is better. Error bars show standard deviation.

Table 1: Comparison of pose estimation approaches using the conventional 6 IMUs (pelvis, head, left wrist, right wrist, left shank, right shank) on TotalCapture dataset. Lower Global Angular Error (GAE) is better.

Method	Compatible Placement	GAE (deg)
Transpose [52]	Fixed 6	16.1
PIP [51]	Fixed 6	14.4
PNP [53]	Fixed 6	10.4
DiffusionPoser [46]	Select From 13	14.4
IMUCoCo	**Anywhere on Body**	**14.0**

accuracy. This decrease is reasonable, as fine-grained hand or lower arm motions cannot be directly captured from an upper arm placement, but must instead be inferred by the model, naturally creating additional challenges for activity recognition.

4.3 Evaluation #2: Motion Sensing at Typical Locations Using Existing Datasets

4.3.1 Body Pose Tracking from 6 IMUs. We systematically analyzed the performance of IMUCoCo in tracking body poses with state-of-the-art models using the TotalCapture dataset [41] for ease of comparison with previous work. Following Wouwe *et al.* [46], we evaluated using all six real IMU sensors placed in the pelvis, head, wrists, and shanks, and also subsets of them. In this section, we use different terminology for sensor locations to remain consistent with the datasets used. In consistency with previous work [46, 51–53], we used TotalCapture only for testing and excluded it from our training data set at all stages.

The comparison using the conventional 6 IMUs (pelvis, head, left wrist, right wrist, left shank, right shank) on the TotalCapture datasets is summarized in Table 1. Overall, IMUCoCo with DTP still achieves reasonable performance, even when compared with pose estimation models specifically designed for 6-IMU setups. We believe that the 6-IMU configuration offers a unique advantage for estimating pose, as it converts all the leaf IMU to be relative to the pelvis IMU, which eliminates the signal variation caused by different facing directions. This normalization approach is used in several prior works, such as PNP [53], DynaIP [58], PIP [51], and Transpose [52]. For Diffusion Poser [46], IMUPoser [28], MobilePoser [48], and our work, we can only use the global frame IMU measurements as these use cases do not always have a pelvis IMU.

Table 2: Comparison of Global Angular Error (GAE) between IMUCoCo + DTP and DiffusionPoser under different test placements using the TotalCapture dataset. Lower is better. (P = pelvis; H = head; RLA = right lower arm; LLA = left lower arm; RLL = right lower leg; LLL = left lower leg)

Test Placement	IMUCoCo	DiffusionPoser [46]
All 6 IMUs	**14.0**	14.4
P+H+RLA+LLA+RLL	**15.7**	19.4
P+H+RLL+LLL	**21.8**	24.9
P+RLL+LLL	**23.8**	36.4
RLL+LLL	**26.6**	39.2
H	**32.0**	39.2

4.3.2 Body Pose Tracking from Flexible IMUs. Next, we examined the performance of IMUCoCo using a flexible combination from a set of sensor locations. DiffusionPoser [46] employs a generative diffusion process to infer the missing sensor and pose features and supports IMU placement from at most 13 possible locations. To the best of our knowledge, DiffusionPoser is currently the most flexible IMU-based full-body pose method. Table 2 summarizes the comparison result using 6, 5, 3, 2, and 1 IMU sensors on the TotalCapture dataset. IMUCoCo performs better on all the listed sensor combinations than DiffusionPoser [46].

4.4 Ablation Analysis

In this section, we provide a more detailed analysis of the individual components of the IMUCoCo system and its application performance. The evaluation is conducted on both the benchmark dataset and our custom datasets. We first implemented DTP as our pose estimation model using the same configuration in the experiment (Section 3.4), but without using IMUCoCo at all, denoted as **w/o IMUCoCo**. To ensure the model has the same dimension and depth, we mapped the original IMU inputs using 24 different linear projections so that it shared the same dimensions as the extracted features from IMUCoCo. We hypothesize that while these models should still provide reasonable performance on the benchmark datasets with the same 6-IMU configurations, they will generalize poorly to new sensor locations in our custom dataset.

Then, we specifically tested the detailed designs of the IMUCoCo model by evaluating the impact of the sensor coordinate information. For this testing, we drop the sensor coordinate information by always setting it to zero; we name this approach IMUCoCo w/o Sensor Coordinates or **w/o SC**. In this test, each joint node will transfer the matched IMU signal, but without knowing where exactly the sensor comes from.

Regardless of the models used in the ablation analysis, the input dimensions were still allocated the same using the IMUCoCo's transfer loss map based on the spatial location of the provided real IMU sensors, as otherwise, the downstream model will have no

Table 3: Ablation analysis of IMUCoCo for pose estimation measured in Global Angular Error (GAE) on TotalCapture datasets. Lower is better. (P = pelvis; H = head; RLA = right lower arm; LLA = left lower arm; RLL = right lower leg; LLL = left lower leg)

Test Placement	IMUCoCo	w/o IMUCoCo	w/o SC
All 6 IMUs	14.0	16.6	26.8
P+H+RLA+LLA+RLL	15.7	18.1	26.5
P+H+RLL+LLL	21.9	33.8	27.2
P+RLL+LLL	23.8	35.7	27.9
RLL+LLL	26.7	42.9	35.0
H	32.0	43.5	37.2

Table 4: Ablation analysis of IMUCoCo for pose estimation measured in Global Angular Error (GAE) on our custom datasets. Lower is better. The arrow indicates moving one standard placement (wrist, thigh, and ear) to another placement.

Test Placement	IMUCoCo	w/o IMUCoCo	w/o SC
wrist, pocket, ear	27.6	43.4	35.0
wrist → arm	28.6	43.5	34.9
pocket → leg	29.3	42.0	36.2
ear → torso	29.1	42.9	35.2

information on where to take the input of various IMU sensors to produce a meaningful comparison.

Table 3 shows the GAE on the approaches above for pose estimation on the TotalCapture dataset. In comparison with DTP w/o IMUCoCo, which is specifically trained only using the 6-IMU configuration as used for this dataset, IMUCoCo still achieves better performance. We attribute this improvement to the known IMUs, primarily to the architecture of IMUCoCo, which not only enables flexibility in testing, but its scalable architecture during the pretraining phase allows utilizing the information from rich augmented mesh IMU signals, which helps to learn a more robust motion feature representation. Comparing w/o SC to w/o IMUCoCo, we found a significant degradation of performance if using IMUCoCo without providing the correct IMU device coordinate. In this case, the joint node module in IMUCoCo confuses how to transfer the provided signal correctly. This also verifies that IMUCoCo has learned to adapt based on the sensor coordinate of IMU devices.

Table 4 presents GAE for different IMU placement configurations on our custom dataset, where each row corresponds to a variant of the standard 3-IMU setting (wrist, thigh pocket, ear) with one IMU repositioned at each time. The model trained without IMUCoCo fails immediately across all configurations, as it cannot generalize to these unseen or altered IMU placements from its training. Note that this model is trained using the 6-IMU combinations, rather than the wrist, thigh pocket, and ear. The version without sensor coordinate (w/o SC) still shows significant degradation in comparison with IMUCoCo, confirming that IMUCoCo utilizes the sensor coordinate information to adapt accurately to the input IMU signals.

5 Application Scenarios

IMUCoCo allows users to put their devices with IMUs anywhere they prefer. This capability opens up a wide variety of novel applications (*e.g.*, Figure 1).

IMUCoCo allows end-users to utilize the myriad of smart devices they may have that are equipped with an IMU. Prior work has focused on enabling motion and pose sensing from the most common devices. We support the user to use the device they have or need. For example, runners typically do not like to keep their phones in their pants pockets. We enable them to track their running form, if they like, from an armband placement. Similarly, if a user has a necklace or smart innerwear, which sometimes has an IMU built into the waistband, we do not need to retrain a model that is tuned to estimate torso movements from the ear.

IMUCoCo adapts to shifts in sensor placement that may occur throughout the day. For example, a user might prefer to keep their phone in different pockets throughout the day or in various activities. As this user moves their phone from their pant pocket to their sweatshirt's kangaroo pocket to their pants' back pocket to the forearm pocket in their ski jacket, IMUCoCo adapts to these changing placements and provides accurate pose and activity estimates for those specific activities. As long as IMUCoCo is aware of the active sensor's location, it can adjust to these changes without the need for switching models. This flexibility in changing device locations dynamically supports adaptive sensing, aligning with the variable contexts of daily activities.

Moreover, IMUCoCo also supports changing or suggesting placements to meet the user's specific needs. Khurana *et al.* [20]proposed a detachable smartwatch that can be placed on different parts of the body depending on the end need. IMUCoCo can now support such scenarios with accurate motion sensing. For example, a person with two IMU devices can record their footwork when playing soccer by attaching them to two thigh pockets (See Figure 1C). If the user's focus then shifts to upper-body posture, the user can move the sensor to their chest or arms. Or, if the user wants to track their squat form more accurately, we can recommend an optimal placement for the watch (perhaps wear it as an anklet) and help the user ensure that their knees do not go beyond their toes. Throughout these adjustments, IMUCoCo consistently provides tracking and analysis without the need to switch or retrain models. No doubt, the placement recommendations will need to take into account the user's device form factor and available adapters/straps.

6 Limitations and Future Work

The current implementation of IMUCoCo is not without limitations. First, the current synthetic IMU data generation method is based on the assumption of a rigid human body. However, in practical scenarios, IMU devices are often attached to clothing, which introduces variability due to the flexibility and movement of the fabric. This discrepancy between the synthetic data and real-world data results in inaccuracies in the signals, as the synthetic model does not account for the flux caused by clothing. Future work should continue refining the synthesis process to include models that simulate the impact of clothing [59] and other factors such as contacting objects. Enhancing the realism of synthetic datasets will improve

the robustness and applicability of the system across more realistic human activity scenarios.

Secondly, it is challenging to accurately estimate full-body movements from sparse sensors placed at any location, especially when the point of interest is far from the sensor. When the placement is not ideal for an application, IMUCoCo will still produce its estimate but with higher errors. The model outputs can involve hallucinations, relying on learned correlations within the training data rather than on direct observations. This approach may suffice for certain applications, but it falls short when precise and reliable full-body tracking is required. Future improvements should aim to increase the transparency of model predictions by visualizing which body parts IMUCoCo estimates with confidence and which parts are less accurately represented. This enhancement will provide users with a clearer understanding of the model's capabilities and limitations, facilitating a more informed application of the technology in complex scenarios. Moreover, it is crucial to understand what potential placement might be preferable by users and by applications. Such consideration extends beyond optimizing performance to include factors such as comfort and the stability of device attachment.

Finally, future work shall consider further simplifying the specification of the on-body IMU placement by exploring an intuitive interface or calibration step. For instance, Sztyler and Stuckenschmidt [39] have proposed a method to recognize the on-body location of each device from IMU signals. In addition, using a dedicated network or sensing technology to identify the location of the device might also be possible. These developments would lower the barriers for configuring IMUCoCo and support the application scenarios we discussed above.

7 Conclusion

The integration of IMUs into consumer devices has significantly advanced the field of human motion and pose estimation. Our evaluations demonstrated that IMUCoCo can effectively utilize IMUs placed at flexible locations on the body by projecting their signals onto a placement-adaptive representation, which can be adjusted for downstream tasks such as pose tracking and activity recognition. IMUCoCo offers numerous advantages that enhance the practicality of on-body motion sensing: it accommodates devices at unconventional locations, allows for the seamless movement of devices throughout the day, and optimizes sensor placement based on specific activities, all while using a single model without the need for retraining. We believe our approach enables a wide range of practical applications, from personal fitness tracking to rehabilitation. More broadly, we hope our work encourages further research in scaling up the development of IMU-based sensing models for human motion understanding.

Acknowledgments

This work was supported in part by the Center for Machine Learning and Health at CMU, Samsung, and Masason Foundation. We also thank Vasco Xu, Karan Ahuja, and Vimal Mollyn for their valuable advice and for sharing their implementations, and David Lindlbauer and Yi Fei Cheng for their assistance with motion capture equipment.

References

[1] Rebecca Adaimi, Abdelkareem Bedri, Jun Gong, Richard Kang, Joanna Arreaza-Taylor, Gerri-Michelle Pascual, Michael Ralph, and Gierad Laput. 2024. Advancing Location-Invariant and Device-Agnostic Motion Activity Recognition on Wearable Devices. *CoRR* abs/2402.03714 (2024). https://doi.org/10.48550/ARXIV.2402.03714

[2] Karan Ahuja, Sven Mayer, Mayank Goel, and Chris Harrison. 2021. Pose-on-the-Go: Approximating User Pose with Smartphone Sensor Fusion and Inverse Kinematics. In *CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama, Japan, 9:1–9:12. https://doi.org/10.1145/3411764.3445582

[3] Riku Arakawa, Karan Ahuja, Kristie Mak, Gwendolyn Thompson, Sam Shaaban, Oliver Lindhiem, and Mayank Goel. 2023. LemurDx: Using Unconstrained Passive Sensing for an Objective Measurement of Hyperactivity in Children with no Parent Input. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 2 (2023), 46:1–46:23. https://doi.org/10.1145/3596244

[4] Riku Arakawa, Hiromu Yakura, Vimal Mollyn, Suzanne Nie, Emma Russell, Dustin P. DeMeo, Haarika A. Reddy, Alexander K. Maytin, Bryan T. Carroll, Jill Fain Lehman, and Mayank Goel. 2022. PrISM-Tracker: A Framework for Multimodal Procedure Tracking Using Wearable Sensors and State Transition Information with User-Driven Handling of Errors and Uncertainty. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4 (2022), 156:1–156:27. https://doi.org/10.1145/3569504

[5] Riku Arakawa, Bing Zhou, Gurunandan Krishnan, Mayank Goel, and Shree K. Nayar. 2023. MI-Poser: Human Body Pose Tracking Using Magnetic and Inertial Sensor Fusion with Metal Interference Mitigation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3 (2023), 85:1–85:24. https://doi.org/10.1145/3610891

[6] Sara Ashry, Tetsuji Ogawa, and Walid Gomaa. 2020. CHARM-Deep: Continuous Human Activity Recognition Model Based on Deep Neural Network Using IMU Sensors of Smartwatch. *IEEE Sensors Journal* 20, 15 (Aug. 2020), 8757–8770. https://doi.org/10.1109/jsen.2020.2985374

[7] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* 46, 3 (2014), 33:1–33:33. https://doi.org/10.1145/2499621

[8] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R. Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognit. Lett.* 34, 15 (2013), 2033–2042. https://doi.org/10.1016/J.PATREC.2012.12.014

[9] David Dobbelstein, Philipp Hock, and Enrico Rukzio. 2015. Belt: An Unobtrusive Touch Input Device for Head-worn Displays. In *CHI Conference on Human Factors in Computing Systems*. ACM, Seoul, South Korea, 2135–2138. https://doi.org/10.1145/2702123.2702450

[10] Yang Gao, Wenbo Zhang, Junbin Ren, Ruihao Zheng, Yingcheng Jin, Di Wu, Lin Shu, Xiangmin Xu, and Zhanpeng Jin. 2024. PressInPose: Integrating Pressure and Inertial Sensors for Full-Body Pose Estimation in Activities. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 8, 4 (2024), 197:1–197:28. https://doi.org/10.1145/3699773

[11] Jack H Geissinger and Alan T Asbeck. 2020. Motion inference using sparse inertial sensors, self-supervised learning, and a new dataset of unscripted human motion. *Sensors* 20, 21 (2020), 6330.

[12] Francine Gemperle, Chris Kasabach, John Stivoric, Malcolm Bauer, and Richard Martin. 1998. Design for Wearability. In *Second International Symposium on Wearable Computers*. IEEE Computer Society, Pittsburgh, PA, USA, 116–122. https://doi.org/10.1109/ISWC.1998.729537

[13] 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, 6 (2022), 79.

[14] Halim Hicheur, Alexander V Terekhov, and Alain Berthoz. 2006. Intersegmental coordination during human locomotion: does planar covariation of elevation angles reflect central constraints? *Journal of Neurophysiology* 96, 3 (2006), 1406–1419.

[15] 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 Trans. Graph.* 37, 6 (2018), 185. https://doi.org/10.1145/3272127.3275108

[16] Takuya Isho, Hideyuki Tashiro, and Shigeru Usuda. 2015. Accelerometry-Based Gait Characteristics Evaluated Using a Smartphone and Their Association with Fall Risk in People with Chronic Stroke. *Journal of Stroke and Cerebrovascular Diseases* 24, 6 (June 2015), 1305–1311. https://doi.org/10.1016/j.jstrokecerebrovasdis.2015.02.004

[17] Jeya Vikranth Jeyakumar, Liangzhen Lai, Naveen Suda, and Mani B. Srivastava. 2019. SenseHAR: a robust virtual activity sensor for smartphones and wearables. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*. ACM, New York, NY, USA, 15–28. https://doi.org/10.1145/3356250.3360032

[18] Jiaxi Jiang, Paul Streli, Huajian Qiu, Andreas Fender, Larissa Laich, Patrick Snape, and Christian Holz. 2022. AvatarPoser: Articulated Full-Body Pose Tracking from

Sparse Motion Sensing. In *The European Conference on Computer Vision*, Vol. 13665. Springer, Tel Aviv, Israel, 443–460. https://doi.org/10.1007/978-3-031-20065-6_26

[19] Haojian Jin, Jingxian Wang, Zhijian Yang, Swarun Kumar, and Jason I. Hong. 2018. RF-Wear: Towards Wearable Everyday Skeleton Tracking Using Passive RFIDs. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. ACM, Singapore, 369–372. https://doi.org/10.1145/3267305.3267567

[20] Rushil Khurana, Mayank Goel, and Kent Lyons. 2019. Detachable smartwatch: More than a wearable. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 1–14.

[21] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel Moore. 2010. Activity recognition using cell phone accelerometers. *SIGKDD Explor.* 12, 2 (2010), 74–82. https://doi.org/10.1145/1964897.1964918

[22] HyeokHyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. 2020. IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3 (2020), 87:1–87:29. https://doi.org/10.1145/3411841

[23] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. 2015. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.* 34, 6 (2015), 248:1–248:16. https://doi.org/10.1145/2816795.2818013

[24] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. 2019. AMASS: Archive of Motion Capture As Surface Shapes. In *2019 IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, South Korea, 5441–5450. https://doi.org/10.1109/ICCV.2019.00554

[25] Saif Mahmud, Ke Li, Guilin Hu, Hao Chen, Richard Jin, Ruidong Zhang, François Guimbretière, and Cheng Zhang. 2023. PoseSonic: 3D Upper Body Pose Estimation Through Egocentric Acoustic Sensing on Smartglasses. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3 (2023), 111:1–111:28. https://doi.org/10.1145/3610895

[26] Pauline Maurice, Adrien Malaisé, Clélie Amiot, Nicolas Paris, Guy-Junior Richard, Olivier Rochel, and Serena Ivaldi. 2019. Human movement and ergonomics: An industry-oriented dataset for collaborative robotics. *The International Journal of Robotics Research* 38, 14 (2019), 1529–1537.

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[28] Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2023. IMUPoser: Full-Body Pose Estimation using IMUs in Phones, Watches, and Earbuds. In *CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg, Germany, 529:1–529:12. https://doi.org/10.1145/3544548.3581392

[29] Aurélie Mourot, Silke Girard, Frédéric Bevilacqua, Nicolas Szezepanski, Sophie Dubuisson, Patrik Vuilleumier, and Donald Glowinski. 2024. Emokine: A kinematic dataset and computational framework for scaling up the creation of highly controlled emotional full-body movement datasets. *Behavior Research Methods* 56 (2024), 7498–7542. https://doi.org/10.3758/s13428-024-02433-0

[30] Philipp Niklas Müller, Alexander Josef Müller, Philipp Achenbach, and Stefan Göbel. 2024. IMU-Based Fitness Activity Recognition Using CNNs for Time Series Classification. *Sensors* 24, 3 (2024), 742. https://doi.org/10.3390/S24030742

[31] Manuel Palermo, Sara M Cerqueira, João André, António Pereira, and Cristina P Santos. 2022. From raw measurements to human pose-a dataset with low-cost and high-end inertial-magnetic sensor data. *Scientific Data* 9, 1 (2022), 591.

[32] Farshid Salemi Parizi, Eric Whitmire, and Shwetak N. Patel. 2022. AuraRing: precise electromagnetic finger tracking. *Commun. ACM* 65, 10 (2022), 85–92. https://doi.org/10.1145/3556639

[33] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32. AAAI, New Orleans, Louisiana, USA, 3942–3951.

[34] E. Ramanujam, Thinagaran Perumal, and S. Padmavathi. 2021. Human Activity Recognition With Smartphone and Wearable Sensors Using Deep Learning Techniques: A Review. *IEEE Sensors Journal* 21, 12 (June 2021), 13029–13040. https://doi.org/10.1109/jsen.2021.3069927

[35] Attila Reiss and Didier Stricker. 2012. Creating and benchmarking a new dataset for physical activity monitoring. In *The 5th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2012*, Fillia Makedon (Ed.). ACM, Heraklion, Crete, Greece, 40. https://doi.org/10.1145/2413097.2413148

[36] Vítor Fortes Rey, Sungho Suh, and Paul Lukowicz. 2022. Learning from the Best: Contrastive Representations Learning Across Sensor Locations for Wearable Activity Recognition. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. ACM, Cambridge, United Kingdom, 28–32. https://doi.org/10.1145/3544794.3558464

[37] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. 2011. Motion capture from body-mounted cameras. *ACM Trans. Graph.* 30, 4 (2011), 31. https://doi.org/10.1145/2010324.1964926

[38] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. 2022. Implicit neural representations for image compression. In *European Conference on Computer Vision*. Springer, Tel Aviv, Israel, 74–91.

[39] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications*. IEEE Computer Society, Sydney, Australia, 1–9. https://doi.org/10.1109/PERCOM.2016.7456521

[40] Jiacheng Tian, Pan Zhou, Fangmin Sun, Tao Wang, and Hexiang Zhang. 2021. Wearable IMU-based Gym Exercise Recognition Using Data Fusion Methods. In *The Fifth International Conference on Biological Information and Biomedical Engineering*. ACM, Hangzhou, China, 27:1–27:7. https://doi.org/10.1145/3469678.3469705

[41] 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*. BMVA Press, Cardiff, UK, 1–13.

[42] Timo von Marcard, Bodo Rosenhahn, Michael J. Black, and Gerard Pons-Moll. 2017. Sparse Inertial Poser: Automatic 3D Human Pose Estimation from Sparse IMUs. *Comput. Graph. Forum* 36, 2 (2017), 349–360. https://doi.org/10.1111/CGF.13131

[43] Osamu Wada, Hiroshige Tateuchi, and Noriaki Ichihashi. 2014. The correlation between movement of the center of mass and the kinematics of the spine, pelvis, and hip joints during body rotation. *Gait & posture* 39, 1 (2014), 60–64.

[44] Martin Weigel, Aditya Shekhar Nittala, Alex Olwal, and Jürgen Steimle. 2017. SkinMarks: Enabling Interactions on Body Landmarks Using Conformal Skin Electronics. In *CHI Conference on Human Factors in Computing Systems*. ACM, Denver, CO, USA, 3095–3105. https://doi.org/10.1145/3025453.3025704

[45] Eric Whitmire, Farshid Salemi Parizi, and Shwetak N. Patel. 2019. Aura: Inside-out Electromagnetic Controller Tracking. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, Seoul, South Korea, 300–312. https://doi.org/10.1145/3307334.3326090

[46] Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott L. Delp, and C. Karen Liu. 2024. DiffusionPoser: Real-Time Human Motion Reconstruction From Arbitrary Sparse Sensors Using Autoregressive Diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, WA, USA, 2513–2523. https://doi.org/10.1109/CVPR52733.2024.00243

[47] Chengshuo Xia, Xinrui Fang, Riku Arakawa, and Yuta Sugiura. 2022. VoLearn: A Cross-Modal Operable Motion-Learning System Combined with Virtual Avatar and Auditory Feedback. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2 (2022), 81:1–81:26. https://doi.org/10.1145/3534576

[48] Vasco Xu, Chenfeng Gao, Henry Hoffmann, and Karan Ahuja. 2024. MobilePoser: Real-Time Full-Body Pose Estimation and 3D Human Translation from IMUs in Mobile Consumer Devices. In *Proceedings of the 37th Annual ACM Symposium on User Interface Softwareand Technology*. ACM, Pittsburgh, PA, USA, 70:1–70:11. https://doi.org/10.1145/3654777.3676461

[49] Masataka Yamamoto, Koji Shimatani, Yuto Ishige, and Hiroshi Takemura. 2022. Verification of gait analysis method fusing camera-based pose estimation and an IMU sensor in various gait conditions. *Scientific Reports* 12, 1 (Oct. 2022), 17719. https://doi.org/10.1038/s41598-022-22246-5

[50] 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*, Vol. 32. AAAI, New Orleans, LA, USA, 7444–7452.

[51] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, New Orleans, LA, USA, 13157–13168. https://doi.org/10.1109/CVPR52688.2022.01282

[52] Xinyu Yi, Yuxiao Zhou, and Feng Xu. 2021. TransPose: real-time 3D human translation and pose estimation with six inertial sensors. *ACM Trans. Graph.* 40, 4 (2021), 86:1–86:13. https://doi.org/10.1145/3450626.3459786

[53] Xinyu Yi, Yuxiao Zhou, and Feng Xu. 2024. Physical Non-inertial Poser (PNP): Modeling Non-inertial Effects in Sparse-inertial Human Motion Capture. In *ACM SIGGRAPH 2024 Conference Papers*. ACM, Denver, CO, USA, 50. https://doi.org/10.1145/3641519.3657436

[54] Tianhong Catherine Yu, Riku Arakawa, James McCann, and Mayank Goel. 2023. uKnit: A Position-Aware Reconfigurable Machine-Knitted Wearable for Gestural Interaction and Passive Sensing using Electrical Impedance Tomography. In *CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg, Germany, 628:1–628:17. https://doi.org/10.1145/3544548.3580692

[55] Shibo Zhang, Yuqi Zhao, Dzung Tri Nguyen, Runsheng Xu, Sougata Sen, Josiah D. Hester, and Nabil Alshurafa. 2020. NeckSense: A Multi-Sensor Necklace for Detecting Eating Activities in Free-Living Conditions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2 (2020), 72:1–72:26. https://doi.org/10.1145/3397313

[56] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A Framework for Human Activity Recognition Using IMU Fusion and ConvTransformer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2 (2022), 88:1–88:26. https://doi.org/10.1145/3534584

[57] Yu Zhang, Songpengcheng Xia, Lei Chu, Jiarui Yang, Qi Wu, and Ling Pei. 2024. Dynamic Inertial Poser (DynaIP): Part-Based Motion Dynamics Learning for Enhanced Human Pose Estimation with Sparse Inertial Sensors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, WA, USA, 1889–1899. https://doi.org/10.1109/CVPR52733.2024.00185

[58] Yu Zhang, Songpengcheng Xia, Lei Chu, Jiarui Yang, Qi Wu, and Ling Pei. 2024. Dynamic Inertial Poser (DynaIP): Part-Based Motion Dynamics Learning for Enhanced Human Pose Estimation with Sparse Inertial Sensors. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, WA, USA, 1889–1899. https://doi.org/10.1109/CVPR52733.2024.00185

[59] Chengxu Zuo, Yiming Wang, Lishuang Zhan, Shihui Guo, Xinyu Yi, Feng Xu, and Yipeng Qin. 2024. Loose Inertial Poser: Motion Capture with IMU-attached Loose-Wear Jacket. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, Seattle, WA, USA, 2209–2219. https://doi.org/10.1109/CVPR52733.2024.00215

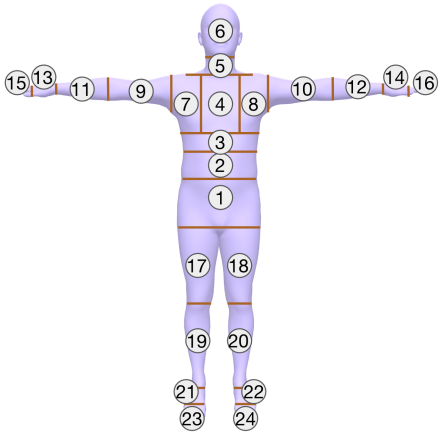

Figure 10: Illustration of categorizing body regions based on horizontal and vertical planes passing through the joints at T pose.

A System Details

A.1 Virtual IMU Synthesis

For virtual mesh IMU, we synthesize the orientation based on faces. Specifically, for each vertex on the human body mesh, we took the face norm as all the faces connecting to this vertex and took the average norm vector of the faces as the y-axis. The y-axis then always points outside the body surface and is perpendicular to the tangent plane of the body surface. z-axis is then determined by the direction orthogonal to the plane formed by the bone direction and the face norm, and the x-axis is orthogonal to both y and z-axis.

We also synthesize IMU from the joints in addition to the vertices on the body surface to create a reference for IMUCoCo during training. To synthesize IMU and kinematics attributes for joints, we did not use the original joint positions. This is because the original joint's acceleration and orientation do not fall into the same distribution as an IMU attached to the surface. For example, the orientation at the right elbow joint is similar to the orientation of an IMU attached to the lower arm, but its acceleration is more similar to the acceleration of an IMU attached to the upper arm. Thus, we modified the locations to sample the acceleration, velocity, and position of the joint from its child joint while keeping the joint orientation from its own. For example, the joint node at the right elbow will synthesize its acceleration from the right wrist, while keeping its own orientation at the right elbow. The coordinates of this joint node are defined by the root-centered position of the right wrist instead of the right elbow. For the leaf joint in the head, hands, and feet, we created five additional vertices on the body on the top of the head, fingertips, and toes to calculate acceleration, velocity, and position.

A.2 Training Architecture and Procedure

Figure 10 shows the categorization k by sensor coordinates. The regions are partitioned by horizontal and vertical planes passing through the joints at T-pose. Vertices in each of the categorized regions often exhibit more similarities than those in distant region categories. Together with frequency-based positional encoding of the coordinates, the concatenated features are fed into the MLP layers to derive placement codes.

Figure 11 shows the full architecture, including the auxiliary components and mesh IMU sampling, during the training process.

We used a two-phase training procedure. In the first phase, we warm up the model using joint IMUs. The advantage of this phase lies in the small size of the input (24 joint signals), but at the same time, this input contains enough high-quality information that can lead to an accurate description of the full-body motion. Therefore, this step allows the model to provide a warm start to all the modules, especially to learn accurate KRs and PR that are used for supervision in the next phase.

In phase two, we extensively sample the full-body dense mesh virtual IMUs from the body surface. We freeze the KRs, as each sample is trained repeatedly on the 24 joint nodes; otherwise, the KRs will overfit to this batch quickly. To focus on the regions that are physically plausible for transference, we apply a weighted stratified sampling scheme. We calculate a weight decay based on the number of hops from the sampled point to each joint node, as well as the initial vertex density of the articulated pose model, so that fewer hops lead to higher chances of being sampled. For each motion sequence, we first perform a forward pass using the 24 virtual joint IMUs, and save this representation as a buffer. Next, for each joint node, we sample a large amount (384 points per motion sequence in our training) of mesh virtual IMUs for each joint node, and perform a forward pass. The inferred feature from this forward pass will replace the corresponding joint feature from the buffer, and together, pass through the full-body pose regression. The advantage of this buffered approach is that gradient updates are applied to each joint node for each forward pass using the mesh virtual IMU, without the need to save the gradient and wait until all 24 joint nodes finish their forward pass (for which our GPU does not have enough memory to execute). We then applied the kinematic loss to both the output using the joint virtual IMU and the mesh virtual IMU, as well as the alignment loss between the two.

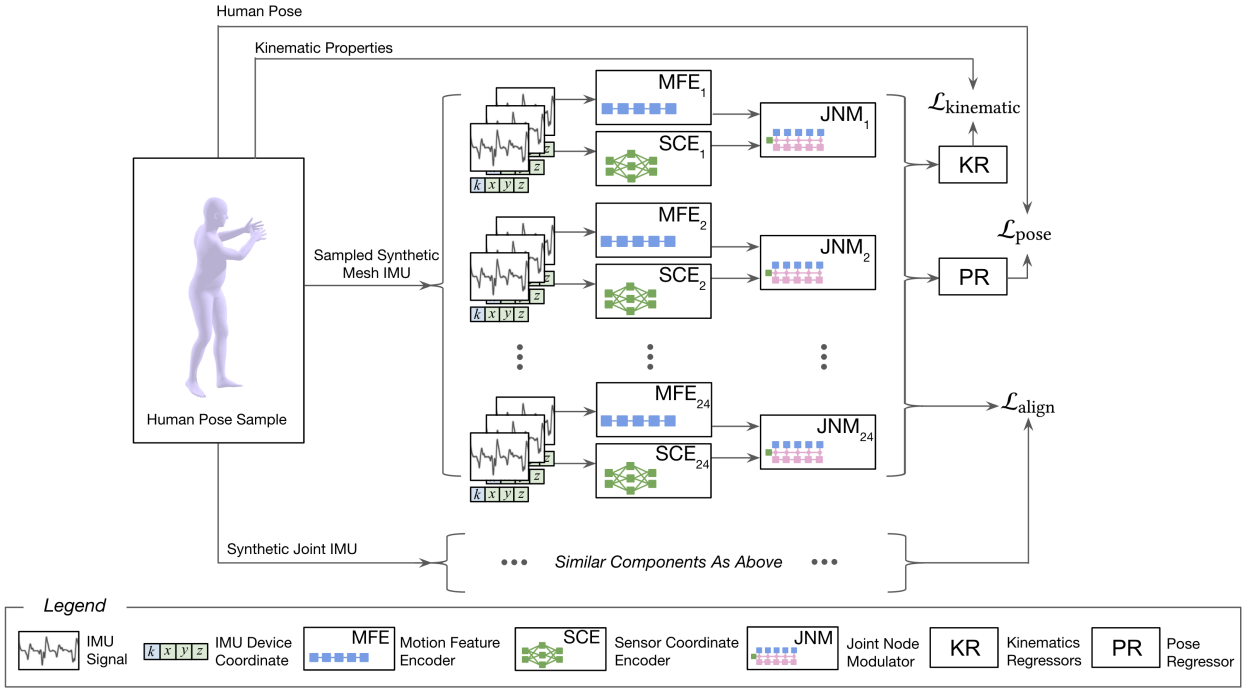

Figure 11: Overview of the IMUCoCo system during training time.