PROXIHG: LATENT PROXIMAL-JOINT HYPERGRAPH ENCODING FOR SPARSE-IMU 3D HUMAN POSE ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse-IMU-based 3D human pose estimation aims to recover full-body motion from a small number of inertial sensors. Under the standard six-IMU setting, key proximal joints, such as the shoulders and hips, are not directly observed. As a result, similar distal IMU signals may correspond to substantially different full-body poses, leading to inherent pose ambiguity. To address this issue, we propose ProxiHG, an encoder-side structural representation framework that explicitly introduces latent proximal joints before pose decoding. Specifically, ProxiHG employs a two-level hypergraph to first model relations among observable IMU nodes and then capture high-order dependencies among sensors, latent proximal joints, and body parts. The resulting structured representation is temporally fused and progressively decoded into full-body motion. Experiments on DIP-IMU and TotalCapture demonstrate consistent improvements in pose accuracy, spatial reconstruction, motion stability, and cross-dataset generalization. These results show that explicitly modeling unobserved proximal joints provides an effective structural inductive bias for sparse-IMU-based full-body motion estimation.

1 INTRODUCTION

3D human pose estimation aims to recover complete human pose and global motion from partial observations. Compared with camera-based approaches, wearable inertial measurement unit (IMU)-based methods are less sensitive to occlusion, illumination variations, and restricted fields of view, making them an attractive sensing modality for human motion capture. However, using only a small number of sensors inevitably leaves the articulated human body only partially observed. In common sparse-IMU settings, only a few body joints can be directly observed, while structurally important joints such as the shoulders and hips remain unconstrained by sensors. Consequently, sparse-IMU-based human motion estimation requires inferring a complete and coherent human pose from limited observations, making structural ambiguity a fundamental challenge of this task.

Existing sparse-IMU methods have made substantial progress through temporal modeling and prediction strategies. DIP trains temporal networks with synthetic IMU data for real-time pose estimation using six sensors Huang et al. (2018). TransPose decomposes the task into staged predictions of leaf-joint positions, full-body joint positions, and body rotations, reducing the difficulty of direct pose regression Yi et al. (2021). More recently, HiPoser models motion coordination among the torso, lower body, upper body, and global translation through part-level hierarchical shared learning Liao et al. (2025). Despite these advances, existing methods primarily improve the prediction stage through stronger temporal modeling, staged regression, physical constraints, or body-part decomposition. The structural information missing from sparse observations is still largely inferred implicitly by the predictor. In particular, proximal joints that connect the observed distal limbs to the torso are not explicitly represented before pose decoding, leaving a critical structural gap in the intermediate representation.

Sparse-IMU systems adopt different sensor counts and placements depending on the target application. In this work, we study the six-IMU configuration commonly used in sparse human motion capture, where sensors are attached to the head, pelvis, wrists, and lower legs Huang et al. (2018); Yi et al. (2021). However, these sensors provide only the local orientations and accelerations of their

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

attached body segments, while the shoulders and hips that connect the torso to the limbs remain unobserved. Consequently, distal observations alone cannot uniquely determine the corresponding shoulder, hip, and torso configurations that connect the limbs to the torso. As shown in Fig. 1, although the six-IMU orientation inputs of Pose A and Pose B are highly similar across most channels, their corresponding full-body poses differ substantially. Such incomplete structural observations introduce significant ambiguity into the mapping from sparse IMU measurements to full-body poses. A model that directly regresses full-body poses from observable sensor features may therefore confuse candidate poses with similar distal measurements, leading to errors in proximal-joint rotations, body configuration, and temporal consistency.

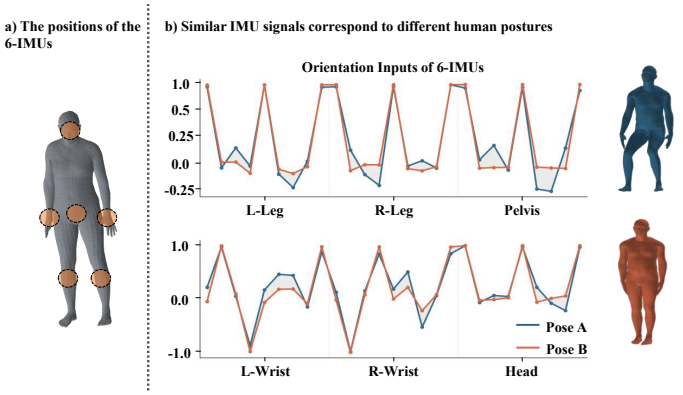


Figure 1: Sparse-IMU pose ambiguity caused by unobserved proximal joints. (a) The standard six-IMU placement. (b) Pose A and Pose B exhibit highly similar orientation inputs across the six IMUs but correspond to markedly different full-body poses.

Although existing methods incorporate skeletal priors by propagating information along pairwise bone connections in predefined joint-level skeletons and infer unobserved joints through generic graph propagation or subsequent staged regression Puchert & Ropinski (2023); Chen et al. (2025), they do not explicitly represent key proximal joints such as the shoulders and hips, nor do they model high-order structural dependencies among observed sensors, proximal joints, and body parts. Consequently, when sparse observations provide insufficient constraints on proximal body configurations, existing predictors must resolve this uncertainty without an explicit representation of the missing intermediate structure.

To address this issue, we propose ProxiHG, an encoder-side structural representation framework for full-body motion estimation from sparse inertial observations. ProxiHG first employs an Observed HGraph to model pelvis-centered and body-part-level relations among six observable IMU nodes. It then introduces latent shoulder and hip nodes through a Latent Proximal-joint HyperGraph, jointly representing structural dependencies among observable sensors, latent proximal joints, and body parts. The resulting structured features are further processed by State-aware Temporal Fusion and a staged decoder to recover full-body motion. Further related work is discussed in Appendix A.

Overall, our contributions are summarized as follows:

- We identify incomplete proximal structural observation as an important source of ambiguity in sparse-IMU motion estimation and explicitly represent the unobserved shoulder and hip states as latent proximal-joint representations before pose decoding.
- We design a two-level hypergraph encoder that first models pelvis-centered and body-part relations among observable IMUs and then incorporates latent proximal joints to model high-order structural dependencies among sensors, proximal joints, and body parts.
- Experiments on DIP-IMU and TotalCapture demonstrate consistent improvements in pose reconstruction, temporal consistency, and cross-dataset generalization. Controlled comparisons with pairwise graph structures further validate the benefits of latent proximal-joint modeling and high-order structural encoding.

2 Preliminaries

2.1 Problem Formulation

Given a dataset $\mathcal{D} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1}^{N}$ containing N motion samples, each sample consists of a sparse IMU sequence $\mathbf{X}^{(i)}$ and its corresponding full-body motion annotation $\mathbf{Y}^{(i)}$. In this work, we focus on a six-IMU configuration with sensors attached to the left wrist, right wrist, left lower

leg, right lower leg, pelvis, and head Huang et al. (2018); Yi et al. (2021). This configuration has been widely adopted in sparse-IMU motion estimation and provides a standard evaluation setting that facilitates direct comparison with prior work while maintaining a sparse wearable setup. Each motion sequence contains T frames. For the s-th IMU at frame t, its input feature is represented as $\mathbf{x}_t^s = [\mathbf{a}_t^s; \mathrm{vec}(\mathbf{R}_t^s)] \in \mathbb{R}^{12}$ where $\mathbf{a}_t^s \in \mathbb{R}^3$ denotes the 3-axis acceleration measurement of the s-th IMU at frame t, $\mathbf{R}_t^s \in \mathbb{R}^{3\times3}$ denotes the orientation matrix, and $\mathrm{vec}(\mathbf{R}_t^s) \in \mathbb{R}^9$ denotes the 9-dimensional vector obtained by vectorizing the orientation matrix. Therefore, the IMU observation at frame t can be represented as $\mathbf{X}_t = [\mathbf{x}_t^1, \mathbf{x}_t^2, \ldots, \mathbf{x}_t^S] \in \mathbb{R}^{S\times12}$. The goal of this work is to recover full-body human motion from sparse IMU sequences. The target output at frame t is denoted as $\mathbf{Y}_t = \{\mathbf{Y}_t^{\mathrm{pose}}, \mathbf{Y}_t^{\mathrm{trans}}\}$, where $\mathbf{Y}_t^{\mathrm{pose}} \in \mathbb{R}^{24\times6}$ represents the 6D rotation representation of 24 SMPL joints Zhou et al. (2019), and $\mathbf{Y}_t^{\mathrm{trans}} \in \mathbb{R}^3$ denotes the global translation of the human body.

2.2 Hypergraph Representation

To explicitly model the structural dependencies between observed IMU nodes and latent human joint nodes, we use a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{H})$ to represent human structural relations Zhou et al. (2006); Feng et al. (2019). Here, $\mathcal{V}$ denotes the node set, $\mathcal{E}$ denotes the hyperedge set, and $\mathbf{H} \in \{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ denotes the incidence matrix. Each element $\mathbf{H}_{ij}$ indicates whether node v_i belongs to hyperedge e_j: if $v_i \in e_j$, then $\mathbf{H}_{ij} = 1$; otherwise, $\mathbf{H}_{ij} = 0$.

ProxiHG represents human structure through a pelvis-centered star relation, an observed body-part hypergraph, and a latent proximal-joint hypergraph. These structures capture sensor-level dependencies among observable IMUs and high-order body relations introduced by latent shoulder and hip nodes. Their detailed construction is presented in the following method section.

3 Methodology

3.1 Overall Framework

The overall architecture of ProxiHG is shown in Fig. 2. Given orientation and acceleration sequences from six IMUs, the model first obtains observable node features through an IMU embedding layer. The Observed HGraph then models pelvis-centered and body-part relations, while the Latent Proximal-joint HyperGraph further introduces latent shoulder and hip nodes to complement missing proximal structures and reduce pose ambiguity. State-aware Temporal Fusion subsequently integrates the structured features with initial pose and translation states to generate condition tokens. Finally, the Staged Direct Pose Decoder progressively predicts leaf-joint positions, full-body joint positions, joint rotations, and global translation, which are converted into the final human motion through SMPL.

3.2 Why Hypergraph

Under sparse-IMU sensing, the unobserved shoulder and hip states are jointly constrained by multiple observations rather than by any single sensor pair. For example, a plausible arm configuration should simultaneously agree with the wrist observation, torso orientation, pelvis-centered body state, and temporal context. Conventional pairwise graph propagation represents these dependencies through separate pairwise connections, but does not explicitly instantiate the missing proximal states. In contrast, ProxiHG represents each unobserved proximal joint as a latent structural variable and couples it with the corresponding distal sensor and central-body observations through semantic hyperedges. Such aggregation introduces coherent part-level structural constraints before pose decoding, thereby reducing the ambiguity of full-body poses consistent with the sparse measurements.

3.3 Structural Hypergraph Encoder

IMU Embedding. Given the input sequence $\mathbf{X} \in \mathbb{R}^{T\times S\times12}$, where $S = 6$, the model first maps each IMU into a d-dimensional node feature space:

$$\mathbf{z}_t^s = \mathrm{LN}\left(\mathbf{W}_e \mathbf{x}_t^s\right) + \mathbf{p}_s, \tag{1}$$

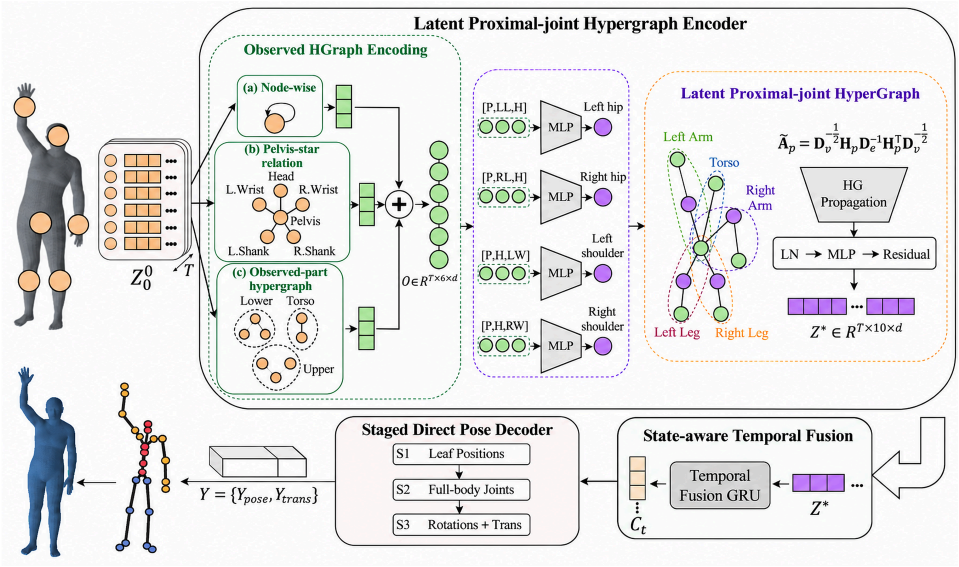


Figure 2: Overview of ProxiHG. The Observed HGraph encodes node-wise, pelvis-centered, and body-part relations among the six observable IMUs. Target-specific mappings generate latent shoulder and hip nodes, which serve as structural bridges between distal sensors and the torso, complementing the missing proximal information and reducing pose ambiguity. The observable and latent nodes are then organized into a Latent Proximal-joint HyperGraph to capture high-order dependencies among sensors, proximal joints, and body parts. The resulting structured representation $\mathbf{Z}^*$ is processed by State-aware Temporal Fusion, followed by staged pose decoding and SMPL-based motion reconstruction.

where $\mathrm{LN}(\cdot)$ denotes Layer Normalization Ba et al. (2016). The temporal representation of the six observable IMU nodes is then denoted as $\mathbf{Z}_o^0 \in \mathbb{R}^{T\times 6\times d}$.

Observed HGraph Encoding. To model structural relations among observable IMU nodes, we construct an Observed HGraph. It contains both a pelvis-centered star relation and an observed part hypergraph based on body parts. Let $\mathbf{A}_{star}$ denote the adjacency matrix of the pelvis-star relation. Its normalized propagation matrix is

$$\widetilde{\mathbf{A}}_{\mathrm{star}} = \mathbf{D}_{\mathrm{star}}^{-\frac{1}{2}}\mathbf{A}_{star}\mathbf{D}_{\mathrm{star}}^{-\frac{1}{2}}. \tag{2}$$

Let $\mathbf{H}_{obs} \in \{0,1\}^{6\times 3}$ denote the incidence matrix of the observed body-part hypergraph. Its initial degree-normalized propagation matrix is

$$\mathbf{G}_{\mathrm{obs}}^{(0)} = \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}_{obs}\mathbf{D}_e^{-1}\mathbf{H}_{obs}^{\top}\mathbf{D}_v^{-\frac{1}{2}}. \tag{3}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ denote the vertex-degree and hyperedge-degree matrices, respectively; the resulting propagation matrix initializes a learnable relation $\mathbf{G}_{\mathrm{obs}}$, whose column-normalized form $\widehat{\mathbf{G}}_{\mathrm{obs}}$ is used to model high-order dependencies among sensor nodes, while $\mathbf{W}_0^l$, $\mathbf{W}_s^l$, and $\mathbf{W}_p^l$ in the l-th observed HGraph block denote the learnable transformations for the self, star, and part-hypergraph branches, with α_s and α_p as learnable scaling coefficients.The node features are updated as:

$$\mathbf{Z}_o^{l+1} = \mathbf{Z}_o^l + \mathrm{MLP}\Big(\mathrm{LN}\Big(\mathbf{Z}_o^l\mathbf{W}_0^l + \alpha_s\widetilde{\mathbf{A}}_{star}\mathbf{Z}_o^l\mathbf{W}_s^l + \alpha_p\widehat{\mathbf{G}}_{\mathrm{obs}}\mathbf{Z}_o^l\mathbf{W}_p^l\Big)\Big), \tag{4}$$

The three terms model self-node information, pelvis-centered star relations, and high-order body-part dependencies, respectively. After several propagation layers, we obtain the structure-enhanced observable node representation $\mathbf{O} \in \mathbb{R}^{T\times 6\times d}$.

Latent Proximal-joint HyperGraph Encoding. To explicitly represent the unobserved shoulders and hips, ProxiHG instantiates four latent node representations associated with the left/right

shoulders and hips from target-specific observable neighborhoods. For latent joint $j \in \{\mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}\}$, let $\mathcal{N}_j$ denote its target-specific observable neighborhood, $\phi_j(\cdot)$ a joint-specific MLP, and $\mathbf{e}_j$ a learnable latent-node embedding. Its feature at frame t is defined as:

$$\mathbf{u}_t^j = \phi_j\left(\mathrm{Concat}\{\mathbf{o}_t^i \mid i \in \mathcal{N}_j\}\right) + \mathbf{e}_j, \tag{5}$$

The observable neighborhoods are $\mathcal{N}_{\mathrm{LH}} = \{\mathrm{P}, \mathrm{LL}, \mathrm{H}\}$, $\mathcal{N}_{\mathrm{RH}} = \{\mathrm{P}, \mathrm{RL}, \mathrm{H}\}$, $\mathcal{N}_{\mathrm{LS}} = \{\mathrm{P}, \mathrm{H}, \mathrm{LW}\}$, and $\mathcal{N}_{\mathrm{RS}} = \{\mathrm{P}, \mathrm{H}, \mathrm{RW}\}$. The four latent features are stacked as $\mathbf{U} \in \mathbb{R}^{T\times 4\times d}$ and concatenated with the observable features to form $\mathbf{Z}_p^0 = [\mathbf{O}; \mathbf{U}] \in \mathbb{R}^{T\times 10\times d}$.

In the Latent Proximal-joint HyperGraph, nodes are grouped into body-part hyperedges, including the torso, left arm, right arm, left leg, and right leg. Let $\mathbf{H}_p$ denote the incidence matrix of this hypergraph, and let $\widetilde{\mathbf{A}}_p$ denote the corresponding normalized propagation matrix. The update rule of the k-th latent proximal-joint hypergraph layer is defined as:

$$\bar{\mathbf{Z}}_p^k = \mathbf{Z}_p^k + \alpha_{\mathrm{msg}} \widetilde{\mathbf{A}}_p \,\mathrm{Linear}\left(\mathrm{LN}(\mathbf{Z}_p^k)\right), \tag{6}$$

$$\mathbf{Z}_p^{k+1} = \bar{\mathbf{Z}}_p^k + \alpha_{\mathrm{ffn}} \mathrm{MLP}\left(\mathrm{LN}(\bar{\mathbf{Z}}_p^k)\right). \tag{7}$$

where α_{msg} and α_{ffn} are learnable residual scaling coefficients. Eq. (6) performs hypergraph message propagation, while Eq. (7) refines the node features through a feed-forward network. After L_p propagation layers, the final structured representation is obtained as $\mathbf{Z}^* = \mathbf{Z}_p^{L_p}$.

3.4 State-aware Temporal Fusion and Staged Motion Decoding

State-aware Temporal Fusion. Given the structurally augmented representation $\mathbf{Z}^*$, we use a part-aware recurrent fusion module to aggregate temporal context before motion decoding. As shown in Fig. 3, $\mathbf{Z}^*$ is divided into torso, lower-body, upper-body, and all-node feature groups, denoted as $\mathbf{Z}_T$, $\mathbf{Z}_L$, $\mathbf{Z}_U$, and $\mathbf{Z}_{all}$, respectively. These four groups are fed into four gated recurrent unit (GRU) branches, namely Torso GRU, Lower GRU, Upper GRU, and Trans GRU Cho et al. (2014) to generate the condition token $\mathbf{c}_t$ through hierarchical state propagation.

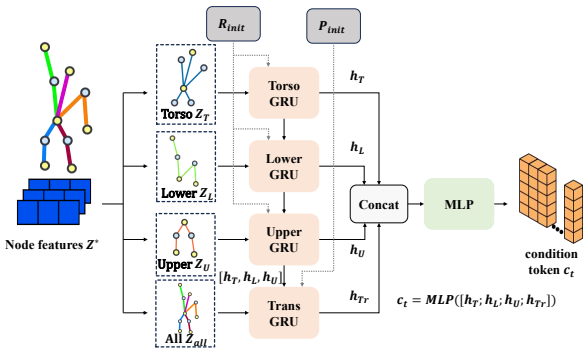


Figure 3: State-aware Temporal Fusion. Part-level GRUs hierarchically fuse structured node features and initial motion states to produce the condition token $\mathbf{c}_t$.

First, the initial pose state $\mathbf{R}_{init}$ and the translation state $\mathbf{P}_{init}$ are mapped to the GRU hidden states as $\mathbf{h}_0^{pose} = \psi_r(\mathrm{vec}(\mathbf{R}_{init}))$ and $\mathbf{h}_0^{trans} = \psi_p(\mathbf{P}_{init})$, where $\psi_r(\cdot)$ and $\psi_p(\cdot)$ are learnable mappings. During training and standard evaluation, $\mathbf{R}_{\mathrm{init}}$ and $\mathbf{p}_{\mathrm{init}}$ are obtained from the first frame of the complete motion sequence and are shared by all windows sampled from that sequence. Identity rotations and zero translation are used only in the initialization ablation.

The four GRU branches then model temporal dependencies in a part-level hierarchy:

$$\mathbf{H}_T = \mathrm{GRU}_T(\mathbf{Z}_T, \mathbf{h}_0^{pose}), \tag{8}$$

$$\mathbf{H}_L = \mathrm{GRU}_L([\mathbf{Z}_L; \mathbf{H}_T], \mathbf{h}_0^{pose}), \tag{9}$$

$$\mathbf{H}_U = \mathrm{GRU}_U([\mathbf{Z}_U; \mathbf{H}_L], \mathbf{h}_0^{pose}), \tag{10}$$

$$\mathbf{H}_{Tr} = \mathrm{GRU}_{Tr}([\mathbf{Z}_{all}; \mathbf{H}_T; \mathbf{H}_L; \mathbf{H}_U], \mathbf{h}_0^{trans}). \tag{11}$$

This design first establishes torso motion states and then progressively incorporates lower-body, upper-body, and global translation information to model part-level motion coordination. For the t-th frame, the hidden states from all branches are concatenated and mapped into the condition token $\mathbf{c}_t = \psi_c([\mathbf{h}_t^T; \mathbf{h}_t^L; \mathbf{h}_t^U; \mathbf{h}_t^{Tr}])$,

Table 1: Comparison with state-of-the-art methods on DIP-Test after AMASS pre-training and DIP-Train fine-tuning, and on TotalCapture with AMASS-only training. PNP is evaluated on DIP-Test using its official checkpoint and excluded from TotalCapture because its complete training pipeline is unavailable. Lower is better.

Methods	DIP-Test					TotalCapture				
	Ang (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)	*Ang* (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)
DIP (TOG'18)	9.02	14.65	5.25	6.27	32.42	13.11	16.07	6.71	7.49	23.71
TransPose (TOG'21)	8.05	13.84	4.98	5.73	28.84	12.60	15.81	6.41	7.22	21.22
TIP (SA'22)	8.24	13.88	4.67	5.64	33.36	12.61	15.01	6.17	7.20	22.34
PIP (CVPR'22)	8.11	13.71	4.44	5.25	32.42	12.31	14.43	6.02	6.84	24.99
PNP (SIGGRAPH'24)	8.29	12.73	4.52	5.25	28.28	–	–	–	–	–
DynaIP (CVPR'24)	6.86	14.01	4.66	5.67	31.15	12.17	15.61	6.06	6.79	26.80
GAIP (TOMM'25)	7.49	13.21	4.62	5.41	30.33	12.61	14.62	6.00	6.75	21.47
ProxiHG (Ours)	**6.71**	**12.28**	**4.14**	**4.86**	**27.78**	**11.97**	**14.16**	**5.90**	**6.63**	**18.29**

Staged Direct Pose Decoder. The condition token $\mathbf{c}_t$ is concatenated with the current-frame IMU feature to form the input of the staged pose decoder $\mathbf{q}_t = [\mathbf{c}_t; \mathrm{vec}(\mathbf{X}_t)]$.

Inspired by the staged prediction strategy in TransPose Yi et al. (2021), the decoder sequentially predicts leaf-joint positions, full-body joint positions, joint rotations, and global translation:

$$\hat{\mathbf{P}}_{leaf} = g_1(\mathbf{q}_t), \tag{12}$$

$$\hat{\mathbf{P}}_{full} = g_2([\mathbf{q}_t; \hat{\mathbf{P}}_{leaf}]), \tag{13}$$

$$\hat{\mathbf{R}}_{red} = g_3([\mathbf{q}_t; \hat{\mathbf{P}}_{full}]), \tag{14}$$

$$\hat{\boldsymbol{\tau}}_t = g_{tr}([\mathbf{q}_t; \hat{\mathbf{P}}_{full}]). \tag{15}$$

where $\hat{\mathbf{P}}_{leaf}$ denotes leaf-joint positions, $\hat{\mathbf{P}}_{full}$ denotes full-body joint positions, $\hat{\mathbf{R}}_{red}$ denotes the 6D rotation representation of reduced SMPL joints, and $\hat{\boldsymbol{\tau}}_t$ denotes global translation. The reduced rotations are then converted into a full 24-joint SMPL pose according to the SMPL kinematic hierarchy Loper et al. (2015), and the pelvis IMU orientation is used as the root orientation. The final output is $\hat{\mathbf{Y}}_t = \{\hat{\mathbf{Y}}_t^{pose}, \hat{\mathbf{Y}}_t^{trans}\}$.

Training Objective. During training, the model supervises both the final pose output and intermediate predictions. The overall loss is defined as:

$$\mathcal{L} = \lambda_{pose}\mathcal{L}_{pose} + \lambda_{leaf}\mathcal{L}_{leaf} + \lambda_{full}\mathcal{L}_{full} + \lambda_{tran}\mathcal{L}_{tran} + \lambda_{vel}\mathcal{L}_{vel}, \tag{16}$$

where $\mathcal{L}_{pose}$ supervises joint rotations, $\mathcal{L}_{leaf}$ and $\mathcal{L}_{full}$ supervise leaf-joint and full-body joint positions, respectively, and $\mathcal{L}_{tran}$ supervises global translation. To improve temporal consistency, joint-velocity loss is further introduced:

$$\mathcal{L}_{vel} = \|\Delta\hat{\mathbf{J}} - \Delta\mathbf{J}\|_2^2. \tag{17}$$

where $\hat{\mathbf{J}}$ and $\mathbf{J}$ denote the predicted and ground-truth root-relative SMPL joint positions, respectively, and Δ denotes the first-order temporal difference.

4 Experiments

4.1 Experimental Setup

Datasets. To comprehensively evaluate the proposed method, we first generate synthetic IMU measurements from the large-scale AMASS dataset and use them to pre-train ProxiHG Mahmood et al. (2019). We then fine-tune the model on DIP-Train and evaluate it on the held-out DIP-Test set Huang et al. (2018). For cross-dataset generalization, we train the model using AMASS only and directly evaluate it on TotalCapture Trumble et al. (2017). No TotalCapture sequence is used during training or model selection in this setting.

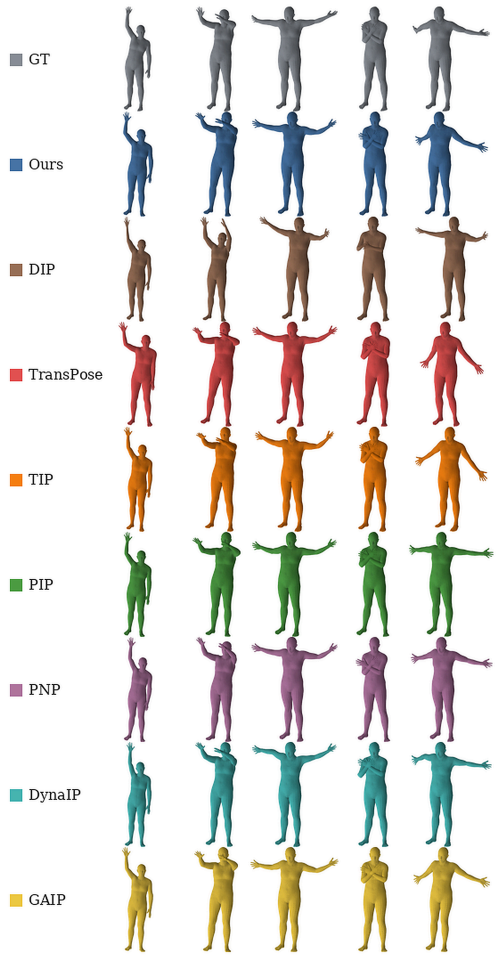


Figure 4: Qualitative comparison of ProxiHG with SOTAs.

Baselines. We compare ProxiHG with representative sparse-IMU methods, including DIP Huang et al. (2018), TransPose Yi et al. (2021), TIP Jiang et al. (2022), PIP Yi et al. (2022), PNP Yi et al. (2024), DynaIP Zhang et al. (2024), and GAIP Chen et al. (2025).

We discuss HiPoser Liao et al. (2025) as a closely related part-based approach, but do not report its numerical results in Table 1 because we were unable to identify an official implementation or checkpoint that would allow reproduction under the same training and evaluation protocol. To ensure a fair and consistent comparison, we therefore avoid combining results obtained under non-aligned experimental settings.

Metrics. Following existing sparse-IMU human pose estimation protocols Yi et al. (2021); Jiang et al. (2022); Yi et al. (2022); Zhang et al. (2024), we evaluate pose and body reconstruction performance using the following metrics. We additionally report joint velocity error to measure temporal motion stability. All metrics are lower-is-better: 1) *SIP error* (°), the mean global rotation error of the upper arms and upper legs; 2) *Ang* (°), the mean global rotation error over all body joints; 3) *Pos* (cm), the mean Euclidean distance between predicted and ground-truth 3D joint positions; 4) *Mesh* (cm), the mean Euclidean distance between predicted and ground-truth SMPL mesh vertices; and 5) *JVE* (cm/s), the mean error between predicted and ground-truth joint velocities.

Implementation Details. All motion sequences are sampled at 60 Hz and divided into windows of 300 frames. The model is first pre-trained on synthetic AMASS data for 30 epochs and then fine-tuned on DIP-Train for 40 epochs, with a batch size of 36. We use the AdamW optimizer Loshchilov & Hutter (2017) together with a cosine learning rate schedule Loshchilov & Hutter (2016). During pre-training, the learning rates of the hypergraph encoder and pose decoder are set to 1×10^{-4} and 3×10^{-4}, respectively. During fine-tuning, they are reduced to 1×10^{-5} and 8×10^{-5}, respectively, with a weight decay of 1×10^{-4}. We reserve 25% of DIP-Train at the sequence level for model selection and evaluate on DIP-Test only after training is completed. Additional implementation, preprocessing, and evaluation details are provided in the appendix.

4.2 Quantitative Results

We evaluate ProxiHG on DIP-Test and TotalCapture in Table 1. On DIP-IMU, all reproducible methods are pre-trained on AMASS, fine-tuned on DIP-Train, and evaluated on DIP-Test. Given the limited size of TotalCapture, the reproducible methods are trained only on AMASS and directly evaluated on TotalCapture to assess cross-dataset generalization.

ProxiHG outperforms the compared baselines across all reported metrics on DIP-Test and also demonstrates stronger cross-dataset performance on TotalCapture. These results support our hypothesis that explicitly representing latent proximal structures before pose decoding provides effective structural constraints. The performance gains on TotalCapture further suggest improved robustness to dataset-specific motion patterns, while the lower JVE indicates better temporal consistency.

Figure 4 compares ProxiHG with representative sparse-IMU baselines across several motion frames. While most methods recover the overall body configuration, differences remain in upper-body articulation, especially shoulder–arm configuration and torso alignment. ProxiHG more closely matches the ground truth and preserves more consistent limb–torso relations, supporting the benefit of explicit latent proximal-joint modeling. The supplementary video provides sequence-level comparisons with ground truth, ProxiHG without latent proximal nodes, and DynaIP.

Table 2: Ablation of the structural hypergraph encoder.

Setting			*Ang* (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)
Star	Part	Latent					
–	–	–	7.00	12.78	4.39	5.17	28.10
✓	–	–	6.79	12.53	4.27	5.03	27.91
✓	✓	–	6.83	12.68	4.45	5.23	27.95
✓	–	✓	6.83	12.53	4.35	5.11	27.87
–	✓	✓	6.85	12.65	4.37	5.15	**27.78**
✓	✓	✓	**6.71**	**12.28**	**4.14**	**4.86**	**27.78**

4.3 Ablation Study

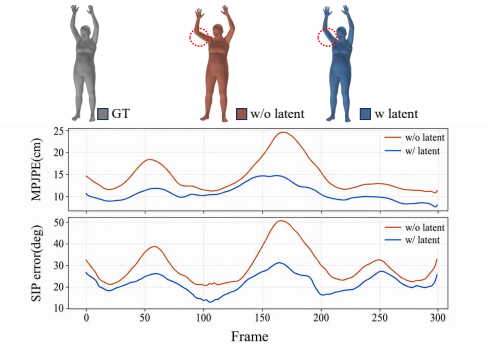


Figure 5: Latent nodes improve shoulder–arm reconstruction and reduce sequence errors.

Structural Hypergraph Components. To investigate the contributions of different components in the structural hypergraph encoder, we evaluate multiple combinations of the pelvis-star relation, observed part hypergraph, and latent proximal-joint hypergraph. The results are reported in Table 2.

Using only the pelvis-star relation already provides a stable performance improvement, indicating that modeling the connection between the pelvis root and distal sensors provides useful global structural constraints.

Partial combinations do not consistently achieve the best performance across all metrics, suggesting that the three structural relations are not simply additive but play distinct and complementary roles. The full model achieves the strongest overall performance, validating the complementarity of the pelvis-star relation, observed part hypergraph, and latent proximal-joint hypergraph.

In particular, the latent proximal nodes establish explicit structural links between distal IMUs and the torso. This enables the shoulder and hip representations to be jointly inferred from observations of the wrists, lower legs, head, and pelvis, providing additional structural constraints for subsequent pose decoding and alleviating ambiguity caused by missing proximal structures.

Figure 5 compares predictions with and without latent proximal nodes together with their frame-wise error curves. The results show that latent proximal nodes improve both single-frame structural reconstruction and the consistency of body topology over continuous motion.

Table 3: Effect of hypergraph modeling.

Encoder	Latent Proximal	*Ang* (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)
Observed Graph	–	7.04	13.21	4.75	5.56	28.19
Observed Hypergraph	–	6.83	12.68	4.45	5.23	27.95
Latent Graph	✓	6.93	12.93	4.53	5.32	28.03
ProxiHG	✓	**6.71**	**12.28**	**4.14**	**4.86**	**27.78**

Effect of Hypergraph Modeling.
To isolate the effect of high-order structural modeling, we construct pairwise graph structures using the same node configurations and structural groupings as ProxiHG, replacing each hyperedge with pairwise connections while keeping the remaining components unchanged. As shown in Table 3, high-order structural modeling consistently improves all reported metrics under both the observable-only and latent-node configurations. Pairwise graph structures represent a body-part relation through multiple separate pairwise connections, whereas ProxiHG directly organizes multiple structurally related nodes within a shared semantic hyperedge. This allows body-part-level dependencies to be represented within a unified structural relation rather than only through a collection of pairwise connections. Moreover, introducing latent proximal-joint nodes improves both relation settings, further

supporting the benefit of explicitly representing the missing shoulder and hip structures before pose decoding.

Number of Latent Joints. To examine the effect of latent-joint scale, we vary the number of latent joints while keeping the observable IMUs, hypergraph depth, temporal fusion module, and training protocol unchanged. The 4-joint configuration includes the bilateral shoulders and hips; the 8-joint configuration further adds the elbows and knees; and the 12-joint configuration additionally incorporates spine- and neck-related joints.

Table 4: Effect of the number of latent joints under identical model and training settings.

# Latent Joints	Params (M)	*Ang* (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)
4	30.07	**6.71**	**12.28**	**4.14**	**4.86**	**27.78**
8	31.12	6.80	12.61	4.35	5.09	27.79
12	32.47	**6.71**	12.46	4.33	5.09	27.82

As shown in Table 4, increasing the number of latent joints leads to higher model complexity without consistent performance gains. Among the evaluated configurations, the four-joint shoulder-and-hip setting achieves the best accuracy–complexity trade-off, suggesting that these proximal joints provide the most informative missing structural cues under the current sensor configuration. We therefore adopt four latent joints as the default setting.

State-aware Temporal Fusion. To verify the effectiveness of hierarchical temporal fusion, we replace the part-level temporal module with a single GRU that directly fuses all node features, while keeping the hypergraph encoder and initialization setting. As shown in Table 5, the complete model consistently performs better across all metrics, demonstrating that organizing temporal features into torso, lower-body, upper-body, and global translation branches is beneficial for capturing region-specific motion patterns and cross-part coordination.

Table 5: Ablation study of the State-aware Temporal Fusion module.

Setting	*Ang* (°)	*SIP* (°)	*Pos* (cm)	*Mesh* (cm)	*JVE* (cm/s)
Single GRU	6.85	12.61	4.38	5.13	27.92
w/o init state	6.79	12.44	4.22	4.97	27.80
Full Temporal Fusion	**6.71**	**12.28**	**4.14**	**4.86**	**27.78**

We investigate the role of the initial states by replacing the sequence-level initial rotation and translation at test time with an identity rotation and a zero vector. This simplified initialization results in only a minor performance degradation, suggesting that the initial pose and translation states mainly provide a continuous starting state for each motion window rather than being essential for accurate pose estimation. The model is therefore relatively insensitive to the choice of initialization.

Limitations. ProxiHG currently assumes a fixed six-IMU configuration and predefined latent proximal structures. While this design is effective under the current setting, it does not explicitly adapt to arbitrary sensor numbers or placements. Moreover, ProxiHG produces deterministic predictions and does not model multiple plausible poses under highly ambiguous observations.

5 Conclusion

We present ProxiHG, an encoder-side structural representation framework for full-body pose estimation from sparse IMUs. Unlike conventional approaches that recover missing structural information primarily within the predictor, ProxiHG explicitly introduces latent shoulder and hip representations before pose decoding and models their high-order interactions with observable sensors. Experiments on DIP-IMU and TotalCapture demonstrate consistent improvements in reconstruction accuracy, temporal consistency, and cross-dataset generalization. More broadly, our results show that explicitly representing structurally critical latent joints can directly incorporate human structural priors into the model, providing more effective structural constraints for full-body pose recovery under sparse observations. In future work, we will investigate adaptive latent structures, flexible sensor configurations, and uncertainty-aware motion reconstruction to further improve the generality and robustness of ProxiHG.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and polishing, literature discovery, research ideation and experimental planning, and the drafting or revision of portions of the manuscript. AI-generated outputs were treated as supporting material rather than authoritative sources. All technical claims, experimental procedures and results, citations, implementations, and research decisions were independently reviewed and verified by the authors. In particular, literature and citations identified through AI-assisted discovery were checked against the original sources. The authors take full responsibility for the final content of this work, including all text and research materials developed with the assistance of generative AI.

ETHICS STATEMENT

This work uses existing human-motion benchmark datasets and does not involve new human-subject recruitment or data collection. The datasets are used in accordance with their stated licenses and usage conditions. No new human-subject data are collected or released as part of this work.

REPRODUCIBILITY STATEMENT

The main paper describes the proposed architecture, structural modeling strategy, datasets, training objectives, and evaluation metrics. The appendix provides detailed sensor configurations, data preprocessing procedures, hypergraph construction, temporal fusion and decoder settings, training hyperparameters, baseline reproduction details, and complete evaluation protocols. These materials are provided to facilitate reproduction of the reported results. An anonymized implementation, including source code and configuration files, is provided in the supplementary material.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

Adithya Balasubramanyam, Suchir Murali Velpanur, Sushma Edhala Jeevarathnam, Tejasree Chekuri Jayachandra, Prasad Honnavalli, and Gowri Srinivasa. Sparsetrack: A physics-informed transformer framework for real-time human motion reconstruction from sparse imus. *Sensors*, 26(10):3262, 2026.

Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2272–2281, 2019.

Kaixin Chen, Lin Zhang, Zhong Wang, Shengjie Zhao, and Yicong Zhou. Skeleton-aware graph-based adversarial networks for human pose estimation from sparse imus. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(4):1–22, 2025.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gulçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1724–1734, 2014.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019.

Dominik Hollidt, Tommaso Bendinelli, and Christian Holz. Ultra diffusion poser: Diffusion-based human motion tracking from sparse inertial sensors and ranging-based between-sensor distances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7046, 2026.

Ryosuke Hori, Jyun-Ting Song, Zhengyi Luo, Jinkun Cao, Soyong Shin, Hideo Saito, and Kris Kitani. Ground reaction inertial poser: Physics-based human motion capture from sparse imus and insole pressure sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28435–28445, 2026.

Yinghao Huang, Manuel Kaufmann, Emre Aksan, Michael J Black, Otmar Hilliges, and Gerard Pons-Moll. Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)*, 37(6):1–15, 2018.

Yifeng Jiang, Yuting Ye, Deepak Gopinath, Jungdam Won, Alexander W Winkler, and C Karen Liu. Transformer inertial poser: Real-time human motion reconstruction from sparse imus with simultaneous terrain generation. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–9, 2022.

Guorui Liao, Chunyuan Zheng, Li Cheng, Haoyu Xie, Shanshan Huang, Jun Liao, Haoxuan Li, and Li Liu. Hiposer: 3d human pose estimation with hierarchical shared learning at parts-level using inertial measurement units. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14309–14317, 2025.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6), 2015.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5442–5451, 2019.

Zhaorui Meng, Lu Yin, Yangqing Hou, Anjun Chen, Shihui Guo, and Yipeng Qin. Improving sparse imu-based motion capture with motion label smoothing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8034–8042, 2026.

Patrik Puchert and Timo Ropinski. A3gc-ip: Attention-oriented adjacency adaptive recurrent graph convolutions for human pose estimation from sparse inertial measurements. *Computers & Graphics*, 117:96–104, 2023.

Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of 28th British Machine Vision Conference*, pp. 1–13, 2017.

Timo Von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer graphics forum*, volume 36, pp. 349–360. Wiley Online Library, 2017.

Xixia Xu, Qi Zou, and Xue Lin. Adaptive hypergraph neural network for multi-person pose estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 2955–2963, 2022.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: Real-time 3d human translation and pose estimation with six inertial sensors. *ACM Transactions On Graphics (TOG)*, 40(4):1–13, 2021.

Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13167–13178, 2022.

Xinyu Yi, Yuxiao Zhou, and Feng Xu. Physical non-inertial poser (pnp): Modeling non-inertial effects in sparse-inertial human motion capture. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Yu Zhang, Songpengcheng Xia, Lei Chu, Jiarui Yang, Qi Wu, and Ling Pei. Dynamic inertial poser (dynaip): Part-based motion dynamics learning for enhanced human pose estimation with sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1889–1899, 2024.

Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3425–3435, 2019.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019.

Zhiming Zou and Wei Tang. Modulated graph convolutional network for 3d human pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11477–11487, 2021.

A Additional Related Work

A.1 Sparse-IMU Human Motion Estimation

Early sparse-inertial motion reconstruction methods primarily relied on human-body constraints and optimization-based inference. Sparse Inertial Poser (SIP) fits orientation and acceleration measurements from six IMUs to the SMPL body model through multi-frame optimization, demonstrating the feasibility of recovering full-body poses from sparse inertial observations Von Marcard et al. (2017); Loper et al. (2015). Deep Inertial Poser (DIP) subsequently introduces temporal neural networks trained with synthetic inertial data for real-time full-body pose regression Huang et al. (2018). TransPose further decomposes the reconstruction task into intermediate predictions of leaf-joint positions, full-body joint positions, and joint rotations, thereby reducing the difficulty of direct pose regression Yi et al. (2021).

Subsequent methods further improve physical plausibility, temporal consistency, and long-term motion tracking. PIP combines neural kinematics estimation with physics-based optimization Yi et al. (2022), while TIP adopts Transformer-based temporal modeling together with stationary body-point estimation to reduce joint and global-motion drift Jiang et al. (2022). PNP further models non-inertial effects in sparse-inertial motion capture Yi et al. (2024). DynaIP improves local motion modeling through body-part partitioning and pseudo-velocity regression Zhang et al. (2024), whereas HiPoser captures inter-part motion dependencies through hierarchical shared learning at the body-part level Liao et al. (2025).

Recent studies explore several complementary directions. Meng et al. introduce motion label smoothing as a plug-and-play regularization strategy for sparse-IMU motion capture Meng et al. (2026). SparseTrack investigates real-time motion reconstruction using only five wearable IMUs with physics-informed and biomechanical constraints Balasubramanyam et al. (2026). GRIP combines sparse IMUs with insole pressure measurements and physics-based simulation to incorporate ground-interaction information into motion reconstruction Hori et al. (2026). Ultra Diffusion Poser further augments sparse inertial sensing with UWB-based inter-sensor distances and explicitly exploits the resulting geometric constraints during motion reconstruction Hollidt et al. (2026). These recent works either target complementary training objectives or adopt sensor configurations and sensing modalities that differ from the standard six-IMU-only setting considered in this work. We therefore discuss them as closely related directions rather than treating their reported numbers as directly comparable under our evaluation protocol.

Table 6: Six-IMU configuration and sensor ordering used in our experiments.

Index	Body location	Symbol
1	Left wrist	LW
2	Right wrist	RW
3	Left lower leg	LL
4	Right lower leg	RL
5	Head	H
6	Pelvis	P

Across much of the existing literature, unobserved body structures are either inferred implicitly from sparse inertial observations or further constrained through physical priors and additional sensing modalities. In contrast, ProxiHG retains the standard six-IMU setting and explicitly represents the unobserved shoulder and hip structures as latent encoder-side representations before pose decoding.

A.2 Graph-based Structural Modeling

Graph neural networks have been widely used to model human skeletal topology. SemGCN represents 2D and 3D poses as skeletal graphs and learns semantic relations among joints Zhao et al. (2019). Spatio-temporal graph approaches further combine skeletal connectivity with temporal dependencies across frames Cai et al. (2019), while modulated graph convolution improves the adaptive modeling of node and edge relations Zou & Tang (2021).

Graph-based structural modeling has also been introduced into sparse-IMU pose estimation. A3GC-IP integrates adjacency-adaptive graph convolution and spatial attention into recurrent motion modeling Puchert & Ropinski (2023), while GAIP combines graph convolution with adversarial supervision in a multi-stage pose estimation framework Chen et al. (2025). These methods demonstrate that explicit skeletal topology can provide useful structural priors for inertial motion reconstruction.

A.3 Hypergraph Representation Learning

Conventional graphs primarily model pairwise relations between nodes. However, coordinated human motion often involves higher-order dependencies shared by multiple joints within the same body part. Hypergraphs generalize ordinary graphs by allowing a single hyperedge to connect multiple nodes, making them well suited for representing such higher-order interactions Zhou et al. (2006); Feng et al. (2019); Yadati et al. (2019); Bai et al. (2021); Xu et al. (2022). ProxiHG exploits this property to jointly model observable sensors, latent proximal joints, and body-part structures, rather than relying solely on pairwise skeletal connections.

B Data Processing and Sensor Configuration

B.1 Sensor Configuration

We adopt the commonly used six-IMU configuration, with sensors attached to the left wrist, right wrist, left lower leg, right lower leg, head, and pelvis. The sensor ordering used throughout our experiments is summarized in Table 6. Each IMU provides a three-dimensional acceleration vector and a 3×3 orientation matrix. For sensor s at frame t, the input feature is defined as

$$\mathbf{x}_t^s = \left[\widetilde{\mathbf{a}}_t^s; \mathrm{vec}(\widetilde{\mathbf{R}}_t^s)\right] \in \mathbb{R}^{12}, \tag{18}$$

where $\widetilde{\mathbf{a}}_t^s$ and $\widetilde{\mathbf{R}}_t^s$ denote the acceleration and orientation after the pelvis-referenced transformation described below. The six sensor features are stacked to form the frame-level observation $\mathbf{X}_t \in \mathbb{R}^{6\times 12}$.

For the five non-pelvis sensors, acceleration and orientation are expressed relative to the pelvis:

$$\widetilde{\mathbf{a}}_t^s = \frac{(\mathbf{R}_t^P)^\top(\mathbf{a}_t^s - \mathbf{a}_t^P)}{30}, \qquad \widetilde{\mathbf{R}}_t^s = (\mathbf{R}_t^P)^\top \mathbf{R}_t^s, \quad s \neq P. \tag{19}$$

Table 7: SMPL indices used for synthetic IMU generation. Joint indices determine sensor orientations, while vertex indices determine sensor positions.

Sensor site	Orientation joint	Position vertex
Left wrist / lower arm	18	1961
Right wrist / lower arm	19	5424
Left lower leg	4	1176
Right lower leg	5	4662
Head	15	411
Pelvis	0	3021

For the pelvis sensor, its acceleration is retained without pelvis-acceleration subtraction and expressed in the pelvis coordinate frame, while its global orientation is preserved as a root-orientation cue:

$$\widetilde{\mathbf{a}}_t^P = \frac{(\mathbf{R}_t^P)^\top \mathbf{a}_t^P}{30}, \qquad \widetilde{\mathbf{R}}_t^P = \mathbf{R}_t^P. \tag{20}$$

B.2 DIP-IMU Processing

Following the official DIP-IMU split, subjects `s_01`–`s_08` are used for training, while `s_09` and `s_10` are used for testing. The six sensors are selected from the original recordings using the zero-based indices $[7, 8, 11, 12, 0, 2]$, yielding the ordering shown in Table 6. We retain the native sampling rate of 60 Hz and apply neither additional temporal filtering nor ground-truth-based orientation calibration.

Invalid acceleration and orientation entries are filled using four rounds of forward and backward propagation from temporally adjacent valid frames. The first and last six frames of each sequence are then removed. Sequences that still contain invalid values or contain fewer than 300 frames are discarded.

The ground-truth axis-angle poses are converted to rotation matrices for all 24 SMPL joints. Since the adopted DIP-IMU protocol does not use global trajectory supervision, the translation target is set to zero. The processed acceleration and orientation measurements are stored in the world coordinate frame and converted to the pelvis-referenced representation only before being fed into ProxiHG.

B.3 Synthetic IMU Generation from AMASS

We generate synthetic inertial observations from AMASS for pre-training. Sequences recorded at 59 or 60 Hz are retained directly and treated as nominal 60 Hz sequences, whereas sequences recorded at 120 Hz are downsampled by a factor of two. Sequences with unsupported frame rates, invalid pose or translation values, or fewer than 300 frames are discarded.

For SMPL-H sequences containing more than 37 joints, joint 23 is replaced by joint 37 before retaining the first 24 body joints, following the body-joint convention adopted in our preprocessing pipeline. We use the first ten AMASS shape coefficients for SMPL forward kinematics. A fixed coordinate transformation, $(x, y, z) \mapsto (x, z, -y)$, is applied to the root orientation and global translation to align AMASS motions with the DIP-IMU coordinate convention. The transformed global trajectory is further shifted such that its first frame is located at the origin.

Virtual sensor orientations are obtained from the global rotations of the corresponding SMPL joints, while virtual sensor positions are determined by fixed SMPL mesh vertices. The exact indices are listed in Table 7.

Given the virtual sensor position $\mathbf{v}_t^s$, the synthetic acceleration is computed using a second-order central difference:

$$\mathbf{a}_t^s = f_s^2 \left(\mathbf{v}_{t-1}^s + \mathbf{v}_{t+1}^s - 2\mathbf{v}_t^s \right), \qquad f_s = 60 \text{ Hz}. \tag{21}$$

The acceleration values of the first and last frames are set to zero. A sequence is discarded if the absolute value of any synthesized acceleration component exceeds $500\,\mathrm{m/s^2}$. The resulting

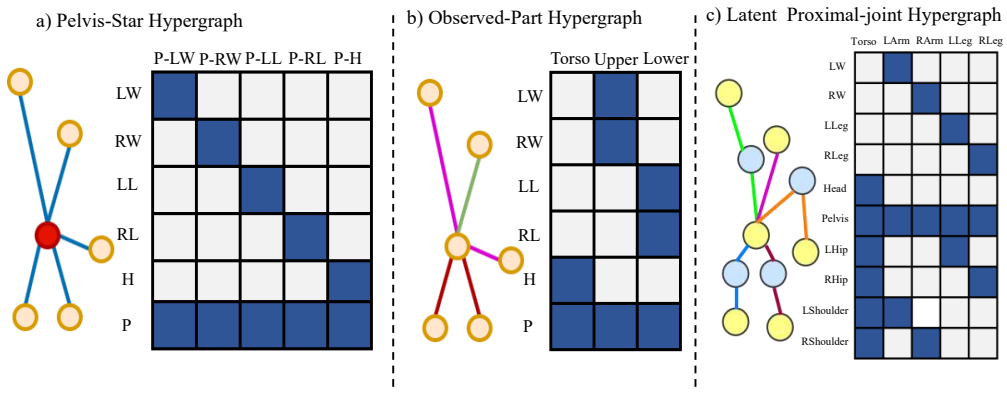


Figure 6: Detailed construction of the structural representation in ProxiHG. (a) The pelvis-star relation connects the pelvis to all non-pelvis observable IMU nodes. (b) The observed-part hypergraph models torso-, upper-body-, and lower-body-level relations among the observable sensors. (c) The latent proximal-joint hypergraph augments the observable nodes with latent left/right hip and shoulder representations and organizes the resulting nodes into torso and limb hyperedges.

synthetic measurements follow the same sensor ordering and pelvis-referenced input transformation as the real DIP-IMU data.

B.4 Input Normalization and Temporal Windowing

Acceleration normalization is incorporated into the pelvis-referenced transformation defined above. Specifically, the transformation subtracts the pelvis acceleration from the five non-pelvis sensors, rotates the measurements into the pelvis coordinate frame, and scales the acceleration by a fixed factor of $1/30$. The pelvis orientation remains in the global frame to preserve root-orientation information. No additional statistical normalization is applied to the effective input of ProxiHG, and no validation- or test-set statistics are used to construct the input representation.

All processed sequences are represented at a nominal sampling rate of 60 Hz and divided into 300-frame windows, corresponding to approximately five seconds of motion. During AMASS pretraining, training windows are extracted with a stride of 150 frames. During DIP-Train fine-tuning, a stride of 30 frames is used to provide denser supervision from real inertial measurements. Validation and test windows use a stride of 300 frames and are therefore non-overlapping. Only complete 300-frame windows are retained.

C Detailed Hypergraph Construction

Figure 6 illustrates the detailed structural construction used in ProxiHG. The structural encoder comprises three components: a pelvis-centered star relation, an observed body-part hypergraph, and a latent proximal-joint hypergraph.

C.1 Observable Nodes

Following the sensor ordering defined in Appendix B, the ordered set of observable nodes is

$$\mathcal{V}_o = (\mathrm{LW}, \mathrm{RW}, \mathrm{LL}, \mathrm{RL}, \mathrm{H}, \mathrm{P}). \tag{22}$$

At each frame, the 12-dimensional feature of each sensor is projected into a d-dimensional feature space:

$$\mathbf{Z}_o^0 = \mathrm{LN}(\mathbf{X}\mathbf{W}_e) + \mathbf{P}_o, \tag{23}$$

where $\mathbf{W}_e$ denotes a learnable linear projection, $\mathbf{P}_o$ is a sensor-specific positional embedding, and $\mathrm{LN}(\cdot)$ denotes Layer Normalization. We set $d = 128$ in all experiments.

C.2 Pelvis-Star Relation

The pelvis serves as the common root reference for all observed body segments. We therefore construct a bidirectional star relation centered at the pelvis:

$$\mathcal{E}_{\mathrm{star}} = \{(\mathrm{P}, \mathrm{LW}), (\mathrm{P}, \mathrm{RW}), (\mathrm{P}, \mathrm{LL}), (\mathrm{P}, \mathrm{RL}), (\mathrm{P}, \mathrm{H})\}. \tag{24}$$

Let $\mathbf{A}_{star} \in \{0,1\}^{6\times 6}$ denote the corresponding symmetric adjacency matrix. Its normalized propagation operator is defined as

$$\widetilde{\mathbf{A}}_{\mathrm{star}} = \mathbf{D}_{\mathrm{star}}^{-\frac{1}{2}} \mathbf{A}_{star} \mathbf{D}_{\mathrm{star}}^{-\frac{1}{2}}, \tag{25}$$

where $\mathbf{D}_{\mathrm{star}}$ denotes the node-degree matrix. This star relation enables direct information exchange between the pelvis and each non-pelvis sensor.

C.3 Observed-Part Hypergraph

In addition to the pelvis-star relation, the observable nodes are organized into three body-part hyperedges:

$$e^o_{\mathrm{torso}} = \{\mathrm{H}, \mathrm{P}\}, \tag{26}$$

$$e^o_{\mathrm{upper}} = \{\mathrm{P}, \mathrm{LW}, \mathrm{RW}\}, \tag{27}$$

$$e^o_{\mathrm{lower}} = \{\mathrm{P}, \mathrm{LL}, \mathrm{RL}\}. \tag{28}$$

The corresponding incidence matrix is $\mathbf{H}_o \in \{0,1\}^{6\times 3}$. The initial degree-normalized hypergraph propagation matrix is defined as

$$\mathbf{G}^{(0)}_{\mathrm{obs}} = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H}_o \mathbf{D}_e^{-1} \mathbf{H}_o^\top \mathbf{D}_v^{-\frac{1}{2}}, \tag{29}$$

where $\mathbf{D}_v$ and $\mathbf{D}_e$ denote the vertex- and hyperedge-degree matrices, respectively.

The observed-part propagation matrix is initialized with $\mathbf{G}^{(0)}_{\mathrm{obs}}$ and subsequently optimized as a learnable parameter. Before message propagation, it is column-normalized as

$$\widehat{\mathbf{G}}_{\mathrm{obs}} = \mathbf{G}_{\mathrm{obs}} \,\mathrm{Diag}\left[\left(\mathbf{1}^\top \mathbf{G}_{\mathrm{obs}} + \epsilon\right)^{-1}\right], \qquad \epsilon = 10^{-3}. \tag{30}$$

Each observed hypergraph block combines three information paths: a self-node projection, pelvis-star propagation, and observed-part hypergraph propagation. The star and part messages are independently modulated by learnable scalar weights, followed by residual feature refinement. ProxiHG employs four observed hypergraph blocks. Their output is further projected and normalized to obtain the structure-enhanced observable representation

$$\mathbf{O} \in \mathbb{R}^{T\times 6\times d}. \tag{31}$$

C.4 Latent Proximal-Joint Nodes

We introduce four latent proximal-joint nodes corresponding to the left hip, right hip, left shoulder, and right shoulder. Their ordered node set is

$$\mathcal{V}_l = (\mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}). \tag{32}$$

Each latent node is inferred from a joint-specific neighborhood of observable nodes:

$$\mathbf{u}_t^j = \phi_j\left(\mathrm{Concat}\left\{\mathbf{o}_t^i \mid i \in \mathcal{N}_j\right\}\right) + \mathbf{e}_j, \tag{33}$$

where ϕ_j is a joint-specific two-layer MLP and $\mathbf{e}_j$ denotes a learnable latent-type embedding. The observable neighborhoods and their concatenation orders are defined as

$$\mathcal{N}_{\mathrm{LH}} = (\mathrm{P}, \mathrm{LL}, \mathrm{H}), \tag{34}$$
$$\mathcal{N}_{\mathrm{RH}} = (\mathrm{P}, \mathrm{RL}, \mathrm{H}), \tag{35}$$
$$\mathcal{N}_{\mathrm{LS}} = (\mathrm{P}, \mathrm{H}, \mathrm{LW}), \tag{36}$$
$$\mathcal{N}_{\mathrm{RS}} = (\mathrm{P}, \mathrm{H}, \mathrm{RW}). \tag{37}$$

Each MLP maps the concatenated $3d$-dimensional feature to a d-dimensional latent representation using a linear layer, Layer Normalization, GELU activation, and a second linear layer.

The latent representations are appended to the observable nodes in the following order:

$$\mathcal{V}_p = (\mathrm{LW}, \mathrm{RW}, \mathrm{LL}, \mathrm{RL}, \mathrm{H}, \mathrm{P}, \mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}). \tag{38}$$

This yields the initial ten-node representation

$$\mathbf{Z}_p^0 = [\mathbf{O}; \mathbf{U}] \in \mathbb{R}^{T\times 10\times d}. \tag{39}$$

C.5 Latent Proximal-Joint Hypergraph

The augmented node set is organized into five semantic hyperedges:

$$e^p_{\text{torso}} = \{\mathrm{H}, \mathrm{P}, \mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}\}, \tag{40}$$
$$e^p_{\text{left-arm}} = \{\mathrm{LW}, \mathrm{P}, \mathrm{LS}\}, \tag{41}$$
$$e^p_{\text{right-arm}} = \{\mathrm{RW}, \mathrm{P}, \mathrm{RS}\}, \tag{42}$$
$$e^p_{\text{left-leg}} = \{\mathrm{LL}, \mathrm{P}, \mathrm{LH}\}, \tag{43}$$
$$e^p_{\text{right-leg}} = \{\mathrm{RL}, \mathrm{P}, \mathrm{RH}\}. \tag{44}$$

These hyperedges explicitly connect each distal observation to its corresponding latent proximal joint and the pelvis, while the torso hyperedge jointly models the head, pelvis, hips, and shoulders.

Let $\mathbf{H}_p \in \{0,1\}^{10\times 5}$ denote the corresponding incidence matrix. The fixed degree-normalized propagation matrix is defined as

$$\widetilde{\mathbf{A}}_p = \mathbf{D}_{v,p}^{-\frac{1}{2}} \mathbf{H}_p \mathbf{D}_{e,p}^{-1} \mathbf{H}_p^\top \mathbf{D}_{v,p}^{-\frac{1}{2}}. \tag{45}$$

For the k-th latent hypergraph layer, message propagation and feature refinement are performed as

$$\overline{\mathbf{Z}}_p^k = \mathbf{Z}_p^k + \alpha_{\text{msg}} \widetilde{\mathbf{A}}_p \,\mathrm{Linear}\left(\mathrm{LN}(\mathbf{Z}_p^k)\right), \tag{46}$$
$$\mathbf{Z}_p^{k+1} = \overline{\mathbf{Z}}_p^k + \alpha_{\text{ffn}} \,\mathrm{MLP}\left(\mathrm{LN}(\overline{\mathbf{Z}}_p^k)\right). \tag{47}$$

The residual scaling coefficients α_{msg} and α_{ffn} are learnable and initialized to 0.1. ProxiHG employs two latent hypergraph layers. The resulting structured representation is

$$\mathbf{Z}^* = \mathbf{Z}_p^2 \in \mathbb{R}^{T\times 10\times d}, \tag{48}$$

which is subsequently passed to the State-aware Temporal Fusion module.

D Temporal Fusion and Decoder Details

D.1 Hierarchical Temporal Fusion

After structural encoding, ProxiHG produces six observable-node features and four latent proximal-joint features, with each node represented by a 128-dimensional feature vector. Following the node ordering

$$\{\mathrm{LW}, \mathrm{RW}, \mathrm{LL}, \mathrm{RL}, \mathrm{H}, \mathrm{P}, \mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}\},$$

the ten nodes are organized into four partially overlapping groups:

$$\mathcal{V}_T = \{\mathrm{H}, \mathrm{P}, \mathrm{LH}, \mathrm{RH}, \mathrm{LS}, \mathrm{RS}\}, \tag{49}$$
$$\mathcal{V}_L = \{\mathrm{LL}, \mathrm{RL}, \mathrm{P}, \mathrm{LH}, \mathrm{RH}\}, \tag{50}$$
$$\mathcal{V}_U = \{\mathrm{LW}, \mathrm{RW}, \mathrm{P}, \mathrm{LS}, \mathrm{RS}\}, \tag{51}$$
$$\mathcal{V}_A = \mathcal{V}_o \cup \mathcal{V}_l. \tag{52}$$

Here, $\mathcal{V}_T$, $\mathcal{V}_L$, $\mathcal{V}_U$, and $\mathcal{V}_A$ denote the torso, lower-body, upper-body, and all-node groups, respectively.

The node features within each group are flattened along the node dimension, yielding

$$\mathbf{Z}_T \in \mathbb{R}^{T\times 768}, \qquad \mathbf{Z}_L \in \mathbb{R}^{T\times 640}, \tag{53}$$

$$\mathbf{Z}_U \in \mathbb{R}^{T\times 640}, \qquad \mathbf{Z}_A \in \mathbb{R}^{T\times 1280}. \tag{54}$$

Temporal dependencies are modeled using four single-layer unidirectional GRU branches, each with a hidden dimension of 256:

$$\mathbf{H}_T = \mathrm{GRU}_T\left(\mathbf{Z}_T, \mathbf{h}_0^{\mathrm{pose}}\right), \tag{55}$$

$$\mathbf{H}_L = \mathrm{GRU}_L\left([\mathbf{Z}_L; \mathbf{H}_T], \mathbf{h}_0^{\mathrm{pose}}\right), \tag{56}$$

$$\mathbf{H}_U = \mathrm{GRU}_U\left([\mathbf{Z}_U; \mathbf{H}_L], \mathbf{h}_0^{\mathrm{pose}}\right), \tag{57}$$

$$\mathbf{H}_{\mathrm{Tr}} = \mathrm{GRU}_{\mathrm{Tr}}\left([\mathbf{Z}_A; \mathbf{H}_T; \mathbf{H}_L; \mathbf{H}_U], \mathbf{h}_0^{\mathrm{trans}}\right). \tag{58}$$

The corresponding input dimensions of the four GRU branches are 768, 896, 896, and 2048, respectively. This hierarchical design first extracts a torso motion representation, which conditions the lower-body branch. The resulting lower-body state then conditions the upper-body branch, while the translation branch integrates all node features and the three pose-related temporal representations.

The outputs of the four branches are concatenated and projected into a compact temporal condition:

$$\mathbf{C} = \psi_c\left([\mathbf{H}_T; \mathbf{H}_L; \mathbf{H}_U; \mathbf{H}_{\mathrm{Tr}}]\right) \in \mathbb{R}^{T\times 170}, \tag{59}$$

where ψ_c consists of a linear layer, Layer Normalization, a GELU activation, and a second linear layer. The resulting 170-dimensional condition at frame t is denoted by $\mathbf{c}_t$.

D.2 Sequence-level State Initialization

The recurrent branches are initialized using the pose and translation states from the first frame of each motion sequence. Let $\mathbf{R}_{\mathrm{init}} \in \mathbb{R}^{6\times 3\times 3}$ denote the six SMPL rotation matrices corresponding to the sensor-bearing body segments. Following the sensor ordering $\{\mathrm{LW}, \mathrm{RW}, \mathrm{LL}, \mathrm{RL}, \mathrm{H}, \mathrm{P}\}$, their SMPL joint indices are

$$\mathcal{J}_{\mathrm{IMU}} = \{18, 19, 4, 5, 15, 0\}. \tag{60}$$

Let $\mathbf{p}_{\mathrm{init}} \in \mathbb{R}^3$ denote the corresponding global translation. The recurrent initial states are computed as

$$\mathbf{h}_0^{\mathrm{pose}} = \psi_r\left(\mathrm{vec}(\mathbf{R}_{\mathrm{init}})\right) \in \mathbb{R}^{256}, \tag{61}$$

$$\mathbf{h}_0^{\mathrm{trans}} = \psi_p\left(\mathbf{p}_{\mathrm{init}}\right) \in \mathbb{R}^{256}, \tag{62}$$

where both ψ_r and ψ_p consist of a linear layer, Layer Normalization, and a GELU activation. Accordingly, the pose projection maps the flattened $6 \times 3 \times 3 = 54$ dimensional rotation input to 256 dimensions, while the translation projection maps the three-dimensional translation vector to 256 dimensions.

During both training and standard evaluation, the ground-truth pose and translation from the first frame of each complete motion sequence are used as sequence-level initialization and shared across all windows sampled from that sequence. Identity rotations and zero translation are used only in the corresponding initialization ablation.

D.3 Staged Motion Decoder

At each frame, the temporal condition is combined with the processed six-IMU observation. Since each IMU contributes 12 input features, the flattened inertial observation has 72 dimensions. We first construct

$$\mathbf{q}_t = [\mathbf{c}_t; \mathrm{vec}(\mathbf{X}_t)] \in \mathbb{R}^{242}, \tag{63}$$

which is then projected into a 384-dimensional decoder feature:

$$\mathbf{f}_t = \mathrm{Dropout}\left(\mathrm{GELU}\left(\mathbf{W}_q\,\mathrm{LN}(\mathbf{q}_t) + \mathbf{b}_q\right)\right) \in \mathbb{R}^{384}, \tag{64}$$

where the dropout rate is set to 0.2.

The decoder follows a three-stage pose estimation procedure together with a dedicated translation branch:

$$\widehat{\mathbf{P}}^{\mathrm{leaf}} = g_1(\mathbf{F}) \in \mathbb{R}^{T\times 5\times 3}, \tag{65}$$

$$\widehat{\mathbf{P}}^{\mathrm{full}} = g_2\left([\mathbf{F}; \mathrm{vec}(\widehat{\mathbf{P}}^{\mathrm{leaf}})]\right) \in \mathbb{R}^{T\times 23\times 3}, \tag{66}$$

$$\widehat{\mathbf{R}}^{\mathrm{red}} = g_3\left([\mathbf{F}; \mathrm{vec}(\widehat{\mathbf{P}}^{\mathrm{full}})]\right) \in \mathbb{R}^{T\times 15\times 6}, \tag{67}$$

$$\widehat{\boldsymbol{\tau}} = g_{\mathrm{tr}}\left([\mathbf{F}; \mathrm{vec}(\widehat{\mathbf{P}}^{\mathrm{full}})]\right) \in \mathbb{R}^{T\times 3}, \tag{68}$$

where $\mathbf{F} = \{\mathbf{f}_t\}_{t=1}^{T} \in \mathbb{R}^{T\times 384}$. Each of g_1, g_2, g_3, and g_{tr} is implemented as a two-layer bidirectional LSTM with a hidden dimension of 384 in each direction and a dropout rate of 0.2, followed by a linear output layer. Their corresponding input dimensions are 384, 399, 453, and 453, respectively.

The first stage estimates the root-relative positions of five endpoint joints with SMPL indices

$$\mathcal{J}_{\mathrm{leaf}} = \{7, 8, 12, 20, 21\}. \tag{69}$$

The second stage estimates the root-relative positions of all 23 non-root SMPL joints:

$$\mathcal{J}_{\mathrm{full}} = \{1, \ldots, 23\}. \tag{70}$$

Conditioned on the predicted full-body joint positions, the third stage estimates the pelvis-relative global rotations of 15 reduced joints using the 6D rotation representation:

$$\mathcal{J}_{\mathrm{red}} = \{1, 2, 3, 4, 5, 6, 9, 12, 13, 14, 15, 16, 17, 18, 19\}. \tag{71}$$

The predicted reduced global rotations are converted to local joint rotations according to the SMPL kinematic hierarchy. Joints not included in the reduced pose representation are assigned identity local rotations. The global orientation measured by the pelvis IMU is used as the SMPL root orientation. The final decoder output is

$$\widehat{\mathbf{Y}}_t = [\mathrm{vec}(\widehat{\mathbf{R}}_t); \widehat{\boldsymbol{\tau}}_t] \in \mathbb{R}^{147}, \tag{72}$$

where $\widehat{\mathbf{R}}_t \in \mathbb{R}^{24\times 6}$ denotes the complete 24-joint SMPL pose representation and $\widehat{\boldsymbol{\tau}}_t \in \mathbb{R}^3$ denotes the global translation.

E Training Objective and Optimization Details

E.1 Training Objective

Let $\mathrm{MSE}(\mathbf{A}, \mathbf{B})$ denote the mean squared error computed over all elements of $\mathbf{A}$ and $\mathbf{B}$. ProxiHG is trained with a weighted combination of rotation, intermediate joint-position, translation, and temporal velocity supervision:

$$\mathcal{L} = \lambda_{\mathrm{pose}}\mathcal{L}_{\mathrm{pose}} + \lambda_{\mathrm{leaf}}\mathcal{L}_{\mathrm{leaf}} + \lambda_{\mathrm{full}}\mathcal{L}_{\mathrm{full}} + \lambda_{\mathrm{tran}}\mathcal{L}_{\mathrm{tran}} + \lambda_{\mathrm{vel}}\mathcal{L}_{\mathrm{vel}}. \tag{73}$$

The pose loss supervises the pelvis-relative global rotations of the 15 reduced joints using the 6D rotation representation:

$$\mathcal{L}_{\mathrm{pose}} = \mathrm{MSE}\left(\widehat{\mathbf{R}}^{\mathrm{red}}, \mathbf{R}^{\mathrm{red}}\right). \tag{74}$$

Here, $\mathbf{R}^{\mathrm{red}}$ is obtained by converting the ground-truth local SMPL rotations to global rotations and subsequently expressing them relative to the pelvis orientation.

The two intermediate position losses supervise root-relative SMPL joint positions:

$$\mathcal{L}_{\mathrm{leaf}} = \mathrm{MSE}\left(\widehat{\mathbf{P}}^{\mathrm{leaf}}, \mathbf{P}^{\mathrm{leaf}}\right), \tag{75}$$

$$\mathcal{L}_{\mathrm{full}} = \mathrm{MSE}\left(\widehat{\mathbf{P}}^{\mathrm{full}}, \mathbf{P}^{\mathrm{full}}\right), \tag{76}$$

Table 8: Loss weights used during synthetic pre-training and DIP-Train fine-tuning. A zero weight indicates that the corresponding loss is not used in that stage.

Loss weight	Pre-training	Fine-tuning
λ_{pose}	1.0	1.5
λ_{leaf}	1.0	1.0
λ_{full}	1.0	1.0
λ_{tran}	0.05	0.05
λ_{vel}	0	0.02

where $\mathbf{P}^{\text{leaf}} \in \mathbb{R}^{T\times 5\times 3}$ contains the positions of the five endpoint joints, and $\mathbf{P}^{\text{full}} \in \mathbb{R}^{T\times 23\times 3}$ contains the positions of all non-root SMPL joints.

Global translation is supervised by

$$\mathcal{L}_{\text{tran}} = \text{MSE}\left(\widehat{\boldsymbol{\tau}}, \boldsymbol{\tau}\right). \tag{77}$$

During fine-tuning, temporal consistency is further encouraged using a joint-velocity loss. The final predicted pose and the ground-truth pose are first converted to root-relative SMPL joint positions, denoted by $\widehat{\mathbf{J}}_t$ and $\mathbf{J}_t$, respectively. Their joint velocities are computed as

$$\widehat{\mathbf{V}}_t = f_s\left(\widehat{\mathbf{J}}_{t+1} - \widehat{\mathbf{J}}_t\right), \qquad \mathbf{V}_t = f_s\left(\mathbf{J}_{t+1} - \mathbf{J}_t\right), \tag{78}$$

where $f_s = 60$ Hz denotes the sampling frequency. The corresponding velocity loss is

$$\mathcal{L}_{\text{vel}} = \text{MSE}\left(\widehat{\mathbf{V}}, \mathbf{V}\right). \tag{79}$$

No separate acceleration loss is used in the final training configuration.

The loss weights used during synthetic pre-training and DIP-Train fine-tuning are summarized in Table 8.

E.2 Optimization

ProxiHG is first pre-trained for 30 epochs using synthetic IMU data generated from AMASS and subsequently fine-tuned for 40 epochs on DIP-Train. Both stages use a batch size of 36 and the AdamW optimizer with a weight decay of 1×10^{-4}. The random seed is fixed to 3407 for all experiments.

During pre-training, the hypergraph encoder and staged motion decoder are jointly optimized with learning rates of 1×10^{-4} and 3×10^{-4}, respectively. During fine-tuning, the decoder is optimized with a learning rate of 8×10^{-5}. The hypergraph encoder is frozen during the first fine-tuning epoch and is unfrozen from the second epoch onward, after which it is jointly optimized with the decoder using a learning rate of 1×10^{-5}. No learning-rate warm-up is employed.

A cosine annealing schedule is used with a minimum learning rate of 1×10^{-6}. During fine-tuning, the optimizer and learning-rate scheduler are reinitialized when the hypergraph encoder is unfrozen. Gradients of both the encoder and decoder are clipped to a maximum norm of 1.0.

For synthetic pre-training, the AMASS motion sequences are divided into training and validation subsets using a 97%/3% sequence-level split. The pre-trained checkpoint with the lowest validation angular error is used to initialize the fine-tuning stage. For fine-tuning, 25% of the DIP-Train sequences are held out as a sequence-level validation set. The fine-tuned checkpoint is selected according to

$$\mathcal{S}_{\text{val}} = \text{Ang} + 0.05\ \text{Jitter}, \tag{80}$$

where lower values indicate better validation performance. DIP-Test is used exclusively for final evaluation after training and model selection are completed.

Table 9: Optimization settings used for ProxiHG.

Setting	Pre-training	Fine-tuning
Dataset	Synthetic AMASS	DIP-Train
Epochs	30	40
Batch size	36	36
Encoder learning rate	1×10^{-4}	1×10^{-5}
Decoder learning rate	3×10^{-4}	8×10^{-5}
Encoder update	All epochs	Epochs 2–40
Weight decay	1×10^{-4}	
Optimizer	AdamW	
Schedule	Cosine annealing	
Minimum learning rate	1×10^{-6}	
Gradient clipping	1.0	
Random seed	3407	

F Evaluation Protocol and Metric Definitions

F.1 DIP-IMU Protocol

For models trained under our experimental protocol, pre-training is first performed using synthetic IMU data generated from AMASS, followed by fine-tuning on DIP-Train. Model selection is conducted exclusively on the sequence-level validation subset held out from DIP-Train, as described in Appendix E. DIP-Test is used only for final evaluation.

All sequences are represented at 60 Hz and divided into 300-frame windows, corresponding to five seconds of motion. During evaluation, a stride of 300 frames is used, such that adjacent evaluation windows do not overlap. Consequently, no averaging, blending, or selection between overlapping predictions is required. Only complete 300-frame windows are evaluated, and any remaining frames that do not constitute a complete window are excluded.

Each window is processed independently. The sequence-level initial pose and translation described above are shared across all windows sampled from the same motion sequence, while predictions from one window are not propagated to subsequent windows.

Although the temporal fusion module employs unidirectional GRUs, the staged motion decoder contains bidirectional LSTMs and therefore has access to both past and future frames within the current window. Accordingly, the results reported in the main comparison are obtained under an offline, non-causal evaluation protocol.

F.2 TotalCapture Cross-dataset Protocol

For cross-dataset evaluation on TotalCapture, the TotalCapture subset is excluded from the AMASS data used to generate the synthetic pre-training set. The resulting AMASS-noTC model is directly evaluated on real TotalCapture IMU measurements without fine-tuning on TotalCapture and without using TotalCapture for model selection.

TotalCapture measurements are processed using the same deterministic input transformation adopted during AMASS-noTC pre-training. For the five non-pelvis sensors, acceleration and orientation are expressed relative to the pelvis reference. The pelvis orientation is retained in the global frame as the root-orientation cue. Accelerations are rotated into the pelvis coordinate frame and scaled by the fixed factor $1/30$. No dataset-dependent mean or standard deviation normalization is applied to the effective model input.

TotalCapture sequences are represented at 60 Hz and evaluated using the same 300-frame window length and 300-frame stride as for DIP-Test. Adjacent evaluation windows therefore do not overlap, and no averaging or blending of predictions is performed. This setting evaluates direct cross-dataset generalization from synthetic AMASS-noTC inertial measurements to real IMU measurements from an unseen dataset.

F.3 Root Orientation and Translation

When constructing the final SMPL pose, ProxiHG uses the global orientation measured by the pelvis IMU as the root orientation. Global translation is independently estimated by the translation branch of the staged motion decoder.

For metric computation, we follow the root-aligned evaluation protocol adopted by prior sparse-IMU methods. The predicted root rotation is first replaced by the corresponding ground-truth root rotation. In addition, the local rotations of the following joints are set to the identity rotation in both the prediction and ground truth:

$$\mathcal{J}_{\text{ignore}} = \{0, 7, 8, 10, 11, 20, 21, 22, 23\}, \tag{81}$$

where zero-based SMPL joint indices are used. This set includes the root, ankles, feet, wrists, and hands. Global joint rotations and SMPL geometry are then recomputed from the resulting local rotations. This procedure removes the local rotational discrepancies of the ignored joints before metric computation.

For position-based metrics, the predicted and ground-truth joints and vertices are independently centered at their corresponding SMPL root joints. Therefore, the reported joint position, mesh, and joint velocity errors measure root-relative body motion and do not include global translation error. No predicted frame is replaced by its ground-truth pose or translation during the reported evaluation.

F.4 Metric Definitions

Let $\bar{\mathbf{R}}_{t,j}$ and $\mathbf{R}_{t,j}$ denote the predicted and ground-truth global rotations, respectively, after applying the evaluation protocol described above. Their geodesic distance on $\mathrm{SO}(3)$ is defined as

$$d_{\text{geo}}\left(\bar{\mathbf{R}}_{t,j}, \mathbf{R}_{t,j}\right) = \frac{180}{\pi} \arccos\left(\text{clip}\left(\frac{\text{tr}\left(\bar{\mathbf{R}}_{t,j}\mathbf{R}_{t,j}^{\top}\right) - 1}{2}, -1, 1\right)\right). \tag{82}$$

SIP Error. SIP error measures the mean global rotation error over four proximal limb joints:

$$\mathcal{J}_{\text{SIP}} = \{1, 2, 16, 17\}, \tag{83}$$

corresponding to the left hip, right hip, left shoulder, and right shoulder under the zero-based SMPL joint convention. It is computed as

$$\text{SIP} = \frac{1}{4T} \sum_{t=1}^{T} \sum_{j \in \mathcal{J}_{\text{SIP}}} d_{\text{geo}}\left(\bar{\mathbf{R}}_{t,j}, \mathbf{R}_{t,j}\right). \tag{84}$$

Angular Error (Ang). Angular error measures the mean global rotation error over all 24 SMPL joints after applying the root alignment and ignored-joint processing described above:

$$\text{Ang} = \frac{1}{24T} \sum_{t=1}^{T} \sum_{j=0}^{23} d_{\text{geo}}\left(\bar{\mathbf{R}}_{t,j}, \mathbf{R}_{t,j}\right). \tag{85}$$

Joint Position Error (Pos). Let $\bar{\mathbf{J}}_{t,j}$ and $\mathbf{J}_{t,j}$ denote the predicted and ground-truth SMPL joint positions, respectively. Root-relative joint positions are obtained as

$$\widetilde{\mathbf{J}}_{t,j} = \mathbf{J}_{t,j} - \mathbf{J}_{t,0}, \qquad \widetilde{\bar{\mathbf{J}}}_{t,j} = \bar{\mathbf{J}}_{t,j} - \bar{\mathbf{J}}_{t,0}. \tag{86}$$

The joint position error is defined as

$$\text{Pos} = \frac{100}{24T} \sum_{t=1}^{T} \sum_{j=0}^{23} \left\| \widetilde{\bar{\mathbf{J}}}_{t,j} - \widetilde{\mathbf{J}}_{t,j} \right\|_2. \tag{87}$$

The factor of 100 converts distances from meters to centimeters.

Mesh Error. Let $\bar{\mathbf{V}}_{t,v}$ and $\mathbf{V}_{t,v}$ denote the predicted and ground-truth SMPL vertices, respectively. Each mesh is centered using its corresponding SMPL root joint:

$$\widetilde{\mathbf{V}}_{t,v} = \mathbf{V}_{t,v} - \mathbf{J}_{t,0}, \qquad \widetilde{\bar{\mathbf{V}}}_{t,v} = \bar{\mathbf{V}}_{t,v} - \bar{\mathbf{J}}_{t,0}. \tag{88}$$

The mesh error is defined as

$$\text{Mesh} = \frac{100}{TV} \sum_{t=1}^{T} \sum_{v=1}^{V} \left\| \widetilde{\bar{\mathbf{V}}}_{t,v} - \widetilde{\mathbf{V}}_{t,v} \right\|_2, \tag{89}$$

where V denotes the number of SMPL mesh vertices.

Joint Velocity Error (JVE). Joint velocities are computed from consecutive root-relative joint positions:

$$\bar{\mathbf{v}}_{t,j} = f_s \left(\widetilde{\bar{\mathbf{J}}}_{t+1,j} - \widetilde{\bar{\mathbf{J}}}_{t,j} \right), \tag{90}$$

$$\mathbf{v}_{t,j} = f_s \left(\widetilde{\mathbf{J}}_{t+1,j} - \widetilde{\mathbf{J}}_{t,j} \right), \tag{91}$$

where $f_s = 60$ Hz denotes the sampling frequency. JVE is defined as

$$\text{JVE} = \frac{100}{24(T-1)} \sum_{t=1}^{T-1} \sum_{j=0}^{23} \left\| \bar{\mathbf{v}}_{t,j} - \mathbf{v}_{t,j} \right\|_2. \tag{92}$$

SIP and Ang are reported in degrees, Pos and Mesh in centimeters, and JVE in centimeters per second. Each metric is computed independently for every 300-frame evaluation window, and the final reported result is the arithmetic mean over all evaluated windows.

F.5 Baseline Evaluation

For all reproducible baselines, we follow the same dataset-level training and evaluation protocols as ProxiHG whenever feasible. On DIP-IMU, baseline models are pre-trained on AMASS, fine-tuned on DIP-Train, and evaluated on DIP-Test. For TotalCapture cross-dataset evaluation, reproducible baselines are trained on AMASS only and evaluated directly on TotalCapture without dataset-specific adaptation.

PNP is evaluated on DIP-Test using its publicly released official checkpoint. We do not report a TotalCapture result for PNP because the complete training pipeline required to reproduce the corresponding AMASS-only setting is not available.

HiPoser is included in the related-work discussion as a closely related method, but is omitted from the numerical comparison because we could not identify an official implementation or checkpoint that enables evaluation under the same training and testing protocol.