# IMUDINO: SELF-DISTILLED FOUNDATION MODEL FOR INERTIAL SENSING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We present IMUDINO, the first foundation model (FM) for inertial sensing to achieve competitive performance across diverse sensing tasks including human activity recognition (HAR), health, motion capture, and robot odometry. Prior inertial FMs often focus on one subfield of inertial sensing and sometimes underperform state-of-the-art supervised methods (e.g., in HAR), making their scope considerably narrower than FMs for vision and natural language. Moreover, they often require fixed input length and sampling rate, and generalize poorly to unseen sensor locations. To build a general-purpose foundation model for inertial sensing, we adapt the DINO self-supervised learning (SSL) paradigm to inertial data through a set of semantics-preserving motion transformations that simulate variations in motion pattern and sensor setup. For model architecture, we use a plain transformer encoder with frequency-aware positional encoding that natively supports varying input length and sampling rate. This simple setup proved remarkably effective for learning global and dense features generalizable across users, sensor setups, tasks, and, in the case of robot odometry, embodiments. We hope IMUDINO can serve as a backbone for inertial sensing and kickstart the development of general-purpose inertial FMs. Code and checkpoints will be released.

## 1 INTRODUCTION

Self-supervised foundational models (FMs) are core building blocks of modern natural language processing (NLP) and computer vision (CV). By removing the need for labeled data and not being tailored to a specific task, these models can leverage large-scale, easy-to-acquire unlabeled data and learn general-purpose features that can be readily used in downstream tasks. Similar to NLP and CV, inertial sensing covers diverse downstream tasks ranging from HAR to robot odometry, and would benefit immensely from a general-purpose foundation model that learns generalizable motion features from unlabeled IMU data. However, to truly fulfill this vision, an inertial FM must (1) perform competitively among task-specific supervised models on a variety of downstream tasks to justify their adoption, and (2) be versatile enough to support the wide range of input lengths, sampling rates, and sensor locations encountered in real-world scenarios. Existing work falls short in these regards, and a *general-purpose foundation model* for inertial sensing remains an open problem.

To this end, we introduce IMUDINO, the first FM with competitive performance on diverse inertial sensing tasks spanning HAR, health and gait analysis, motion capture, and robot odometry. IMUDINO also provides native support for variable input length and sampling rate, and generalize to diverse sensor placements, which are often overlooked by existing inertial FMs. A key difference between IMUDINO and previous work is the self-supervised learning (SSL) technique: instead of contrastive learning, masked reconstruction, or augmentation prediction, we adapt DINO self-distillation (Caron et al., 2021) to inertial SSL. The core idea of DINO is encouraging different transformed views of the same sample to share a similar global feature representation. The transformation should therefore preserve the global semantics of the sample. For inertial sensing, we propose a set of *semantics-preserving motion transformations*, including cropping, scaling along the magnitude and temporal axes, $SO(3)$ rotation, resampling, and input channel masking. This allows the model to learn a global representation that generalizes across different sensor orientation, user-specific motion patterns, and input channel availability (e.g., acc-only or full 6-axis IMU). We



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.

further adopt the iBOT++ patch-level training objective (Cao et al., 2026) to learn fine-grained dense features essential for tasks like motion capture and robot odometry.

The effectiveness of our training objective allows IMUDINO to outperform previous methods while using a simple model architecture and orders of magnitudes less training data ($\sim$500 vs $\sim$700000 days used by Yuan et al. (2024); Xu et al. (2025)). Specifically, we employ a minimalist transformer architecture augmented only by *frequency-aware positional encoding*, which scales the frequency base with the input's sampling rate and enable the model to learn relational patterns rooted in physical time instead of token index. For training, we use a filtered subset curated from three public datasets, namely Capture24 (Chan et al., 2024), SHL (Gjoreski et al., 2017), and PAAWS (Potter et al., 2025). These datasets are collected from various positions (phone, wrist, thigh, ankle, waist, etc.), contributing to IMUDINO's generalization capabilities with regard to sensor location.

We evaluate IMUDINO on a diverse set of downstream tasks using its frozen features. In HAR, IMUDINO outperformed supervised HAR models, HAR FMs, and general-purpose time series FM baselines on 7 datasets covering diverse activities and heterogeneous sensor setups. In health and gait analysis, IMUDINO features can be readily used for activity intensity estimation, fall detection, and gait recognition. In inertial motion capture, IMUDINO achieves competitive performance on 4 different human pose estimation datasets with a simple MLP probe and no physical priors. Finally, in robot odometry, IMUDINO outperforms Tartan IMU, a SOTA robot odometry model. Together, these results prove that IMUDINO can extract both global and dense features that can generalize across users, sensor setups, tasks, and, in the case of robot odometry, embodiments, making IMUDINO the first general-purpose FM for the broader inertial sensing community.

## 2 Related Work

### 2.1 Foundation Models for Inertial Sensing

The quest for a foundation model for inertial sensing started with self-supervised HAR. Early examples include Saeed et al. (2019), an augmentation prediction method, and LIMU-BERT (Xu et al., 2021), which mirrors the masked reconstruction objective of BERT. In addition, Tang et al. (2020) brought contrastive learning into the field by adapting SimCLR (Chen et al., 2020) to IMU HAR. Work on inertial FMs have since followed the paths set by these three early attempts. For instance, Yuan et al. (2024) expanded augmentation prediction to large scale pretraining data. REBAR (Xu et al., 2024) introduced a two-stage solution, training the model first to reconstruct retrieved motifs and then with contrastive loss. RelCon (Xu et al., 2025) brought motion similarity into contrastive learning. HAR-FM (Qiu et al., 2025) combines the contrastive and masked prediction objectives. However, the results here are mixed: progress in this line of work is mainly measured by improved HAR metrics, yet SSL methods like RelCon struggle to outperform early fully-supervised models like DeepConvLSTM (Ordóñez & Roggen, 2016) and simple ConvNets, let alone more modern supervised HAR methods like TSF (Zhang et al., 2026). Apart from the narrow scope and uncompetitive performance, many of these models (e.g., LIMU-BERT, Yuan et al. (2024), RelCon, etc.) requires fixed input length and sampling rate, limiting their potential in diverse scenarios. Entirely unrelated to this family of work, there is also TartanIMU (Zhao et al., 2025), an FM for inertial odometry in robots. A key difference between the HAR FMs and TartanIMU is that robot odometry relies on *dense features*, while HAR and health tasks use *global features* only.

In summary, the field of inertial FMs remains fragmented as most models only focus on one subfield of inertial sensing. There is also a disconnect between FMs for global features and dense features. The result is a myriad of narrowly-scoped models akin to a vision "FM" useful only for image classification. We argue that a more general-purpose inertial FM supporting global and dense prediction is both desirable and possible. This is the motivation behind IMUDINO, the first inertial FM with broad support for both global tasks (HAR, health) and dense prediction (MoCap, robot odometry).

### 2.2 Why DINO for Inertial Sensing?

DINO (Caron et al., 2021) is a highly successful SSL paradigm in CV. The core objective of DINO is aligning the global representation (i.e., the class token) of transformed views of the same image. Zhou et al. (2022) proposed the iBOT patch-level masked token reconstruction objective to complement DINO's global objective. On top of iBOT, DINOv2 (Oquab et al., 2024) added the KoLeo

regularization loss to encourage uniform spread of the learned features, and massively scaled up the training data. DINOv3 (Siméoni et al., 2025) introduced the Gram anchoring technique for improving dense feature quality during very long training runs to enable further scaling. DINOv2 and DINOv3 have since been used as the backbone in various downstream tasks. Influential examples include DepthAnything (Yang et al., 2024) for depth estimation and AnyLoc for localization (Keetha et al., 2023). DINOv3 further demonstrated SOTA capabilities when swapped into existing architectures for object detection, semantic segmentation, and 3D understanding (Siméoni et al., 2025). Finally, the DINOv2 and DINOv3 SSL recipes have also been successfully adapted to satellite imagery (Tolan et al., 2024; Siméoni et al., 2025) and medical imaging (Vorontsov et al., 2024).

DINOv2's training objective combines the strengths of contrastive methods and masked autoencoder (MAE) (He et al., 2022), which are common approaches to inertial SSL. It provides both global representations with semantic view invariance and fine-grained dense features. Moreover, DINOv2's anti-collapse mechanisms (e.g. EMA teacher, centering and sharpening, KoLeo, etc.) apply to model weights and features, not the input, making them modality-agnostic and easily transferable to inertial SSL. In this work, we adapt the DINOv2 training objective to inertial sensing by generating views through semantics-preserving motion transformations. We demonstrate through experiments that IMUDINO, like its CV counterparts, can be integrated into a wide variety of inertial sensing tasks and achieved competitive performance, significantly broadening the scope of inertial FMs.

# 3 IMUDINO

## 3.1 Training Objective

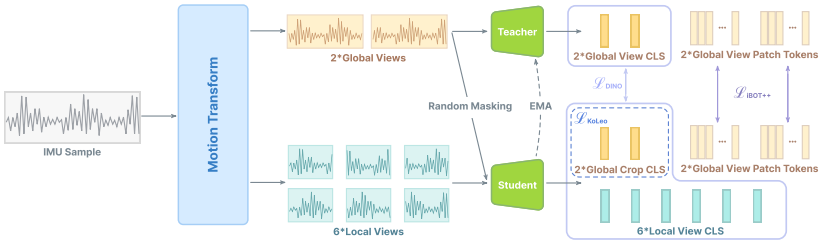


Figure 1: An overview of the IMUDINO training objective.

The training objective of IMUDINO has three components: the global objective ($\mathcal{L}_{\text{DINO}}$), the patch-level objective ($\mathcal{L}_{\text{iBOT++}}$), and a KoLeo regularization loss ($\mathcal{L}_{\text{KoLeo}}$). The student and its DINO and iBOT heads are optimized with backpropagation, whereas the teacher is the EMA of the student.

**Global objective.** This objective concerns the global features (i.e., the class token) extracted by a student and a teacher network, which are given transformed views of the same IMU sample. The student features are passed through a DINO head, which is an MLP that outputs *prototype scores*. Softmax is then applied to the prototype score vector to obtain $p_s$. Similarly, we obtain the teacher's prototype scores using its own DINO head. We then apply sharpening and Sinkhorn-Knopp centering in DINOv2 fashion to obtain $p_t$. The global loss term is therefore:

$$\mathcal{L}_{\text{DINO}} = -\sum p_t \log p_s$$

**Patch-level objective.** As in DINOv2, we apply the iBOT patch-level objective. We mask some of the input patches given to the student, while the same patches remain visible to the teacher. The masked student patch tokens are passed through the student iBOT head (identical in structure to the DINO head). The teacher's corresponding visible patch tokens are processed by the teacher iBOT head. As in $\mathcal{L}_{\text{DINO}}$, we apply centering and softmax before calculating the patch-level loss term:

$$\mathcal{L}_{\text{iBOT}} = -\sum_i p_{ti} \log p_{si}$$

where $i$ are patch indices of masked tokens. TIPSv2 further expanded this loss term to patch token pairs that are visible to *both* the student and the teacher, leading to iBOT++. This anchors the student's patch representation to that of the teacher and encourages the preservation of local semantics. The only difference between $\mathcal{L}_{\text{iBOT}}$ and $\mathcal{L}_{\text{iBOT++}}$ is that $i$ loops over the indices of *all* tokens in iBOT++, regardless of whether a token is masked for the student.

**KoLeo regularizer.** Following DINOv2 and DINOv3, we use the KoLeo regularizer to encourage the features in a batch to spread uniformly in the feature space. The features used here are the $\ell_2$-normalized class tokens. Given $n$ feature vectors $(x_1, ..., x_n)$, the loss is defined as

$$\mathcal{L}_{\text{KoLeo}} = -\frac{1}{n}\sum_{i=1}^{n}\log\left(\min_{j\neq i}||x_i - x_j||\right)$$

Combining all three objectives, we arrive at the full IMUDINO training objective:

$$\mathcal{L} = \lambda_d\mathcal{L}_{\text{DINO}} + \lambda_i\mathcal{L}_{\text{iBOT++}} + \lambda_k\mathcal{L}_{\text{KoLeo}}$$

### 3.2 Semantics-preserving Motion Transformation

The transformations applied to the views reflect what the *global representation* should be robust against. Therefore, we choose the following set of *semantics-preserving motion transformations* from the perspective of desirable global invariance. See Appendix A.1 for detailed implementation.

**Cropping.** Similar to the vision modality, we apply random cropping to generate views. For IMU data, this is grounded in the temporal continuity of human motion: actions happening at around the same time are usually related and semantically similar, and should therefore share similar feature representations. Following DINO, we include 2 types of cropped views: *global views* that cover $0.5\times$ to $0.8\times$ the length of a sample, and *local views* that cover $0.05\times$ to $0.4\times$ the length of a sample. The idea here is to encourage local-global correspondence in the learned representation.

**Scaling.** Scaling is a commonly used data augmentation technique for inertial data (Ye et al., 2026). In IMUDINO, we randomly apply scaling along two axes: the magnitude axes controlling the motion's intensity, and the temporal axes controlling how fast the forces and movements change. This is grounded in the fact that semantically similar motions are performed at different intensity and pace by different people or under different contexts.

**Rotation.** We apply random $SO(3)$ rotations to the views to simulate changes in sensor orientation. This encourages the learned global representation to be less about which direction the sensor is facing and more about the underlying motion itself.

**Resampling.** Whether a motion is recorded at 50Hz or 200Hz does not change its global semantics. Therefore, we apply random resampling to make IMUDINO be robust and consistent when faced with the wide range of sampling rates encountered in real-world applications.

**Channel masking.** We apply random channel masking so that the learned representation is not over-reliant on a certain input channel. This is also useful for adapting IMUDINO to downstream tasks where only triaxial acceleration input is available.

### 3.3 IMUFormer: A Minimalist Transformer for Inertial Data

The transformer architecture has seen increasing adoption in inertial sensing. Beginning with a vanilla transformer encoder (Shavit & Klein, 2021), IMU transformer architectures have since become increasingly complex (Zhang et al., 2022; 2026). To demonstrate that the strong performance of IMUDINO is driven by our effective SSL setup and not complicated model architecture, we return to the basics in this work. Starting from the standard ViT model, we make only two adaptations. First, we replace the 2D image "patchify" layer with an 1D non-overlapping convolution that project the original IMU frames into patch tokens. Second, we replace the learned absolute positional encoding (PE) with a frequency-aware sinusoidal PE that scales sequence positions according to each sample's sampling frequency. For a sample with frequency $f_b > 0$, position $p$ is mapped to $\tilde{p}_b = pf_{\text{ref}}/f_b$, where $f_{\text{ref}}$ is a fixed reference frequency. The encoding is defined as

$$\mathrm{PE}_{b,p,2i} = \sin\left(\tilde{p}_b\, \beta^{-2i/D}\right), \qquad \mathrm{PE}_{b,p,2i+1} = \cos\left(\tilde{p}_b\, \beta^{-2i/D}\right),$$

We empirically set $\beta = 100$ and $f_{\mathrm{ref}} = 100\mathrm{Hz}$. This parameter-free formulation assigns identical positional encodings to equal elapsed times across sampling frequencies. This, in conjunction with the resampling transformation during training, enables IMUDINO to generalize to any reasonable IMU sampling rate. This minimalist architecture is referred to as *IMUFormer* in this paper.

### 3.4 Training Data

We train IMUDINO on a dataset curated from three public wearable inertial datasets covering various sensor locations - Capture24 (Chan et al., 2024), SHL (Gjoreski et al., 2017) and PAAWS-Lab (Potter et al., 2025). The entire train set contains 11.6k hours of 6-axis IMU data. This is orders of magnitude smaller than the 700k - 800k person-days used in RelCon and Yuan et al. (2024).

**Capture24** consists of 3883 hours of wrist-worn accelerometer data collected from 151 free-living participants. We filtered out data with null, incomplete or inconsistent annotation, resulting in 2410 hours of triaxial acceleration data. We then zero-fill the non-existent gyroscope channels.

**SHL** is a smartphone IMU dataset covering 4 phone locations (hip, torso, hand, bag). The train and validation sets from the SHL2026 challenge, totaling 1138 hours of 6-axis IMU data, is used.

**PAAWS-Lab** is collected from a controlled lab environment. The 252 participants contributed 4 hours of data each. 20 IMUs were placed on the participants' body. Specific locations include the waist, wrists, thighs and ankles. We filtered out static segments of the data (See Appendix B for details) to improve training efficiency, resulting in 8069 hours of 6-axis IMU data.

### 3.5 Implementation

We divide the training set into samples using non-overlapping 1024-frame sliding windows. Each sample is then transformed into 2 global views and 6 local views per Sec. 3.2. The teacher receives the 2 global views as input, while the student receives all 8 views. The global objective $\mathcal{L}_{\mathrm{DINO}}$ is computed between the student views and teacher views in the exact same manner as DINOv2. For the patch-level objective $\mathcal{L}_{\mathrm{iBOT++}}$, we randomly select 50% of student global views in a minibatch and replace a random ratio (sampled from $U(0.1, 0.5)$) of patch tokens with a learned mask token. Note that $\mathcal{L}_{\mathrm{iBOT++}}$ applies to both masked and visible tokens. The $\mathcal{L}_{\mathrm{KoLeo}}$ regularization loss is computed over the 2 global view features generated by the student. We empirically set the loss weights to $\lambda_d = \lambda_i = 1.0$ and $\lambda_k = 0.2$. IMUFormer models of two sizes are used in the main experiments. A 6M parameter version comparable in size and FLOPs to most supervised and inertial FM baselines, and a larger 85M parameter version to demonstrate the scaling potential of our approach. Details about the models and training recipe are in Appendix A.2.

## 4 Experiments

We evaluate IMUDINO's performance as a general-purpose inertial FM in four steps. First, we evaluate HAR performance using the global features, a quintessential inertial FM task. Second, we explore using IMUDINO's dense features for inertial motion capture, an inherently challenging task. Third, we test IMUDINO's emergent cross-embodiment transfer capability with robot inertial odometry. Finally, we dissect how our training objective, data augmentation, and the IMUFormer model contribute to performance in an ablation study.

### 4.1 Baselines

For HAR, commonly used to evaluate inertial FMs, we compare IMUDINO to SimCLR (Chen et al., 2020), Yuan et al. (2024)'s method, RelCon (Xu et al., 2025), and HAR-FM (Qiu et al., 2025). To facilitate fair comparison and minimize the influence of training data and model architecture, we train all inertial FM baselines using IMUDINO's training set and the IMUFormer-6M backbone. For baselines that were originally implemented for acceleration-only data, we add 3 input channels for gyroscope. We further include two general-purpose time-series foundation models (TSFMs) as

baselines: TimesFM 3.0 (Das et al., 2024) and Chronos-2 (Ansari et al., 2025) (330.7M & 120M parameters, respectively), and use the official weights off-the-shelf. We also include task-specific supervised baselines in our comparison, namely DeepConvLSTM (Ordóñez & Roggen, 2016), Attend & Discriminate (Abedin et al., 2021), IF-ConvTransformer (Zhang et al., 2022) and TSF (Zhang et al., 2026), which represent the model-driven SOTA for IMU HAR. For inertial motion capture, DIP (Huang et al., 2018), TransPose (Yi et al., 2021), TIP (Jiang et al., 2022), PIP (Yi et al., 2022) and DynaIP (Zhang et al., 2024b) are used as the supervised baselines. For robot inertial odometry, we compare against Tartan IMU (Zhao et al., 2025), the state-of-the-art model for that task.

### 4.2 HAR: IMUDINO HAS EXCEPTIONAL GLOBAL MOTION DESCRIPTORS

We first evaluate IMUDINO on seven HAR datasets: MotionSense (Malekzadeh et al., 2018), HAPT (Reyes-Ortiz et al., 2016), HHAR (Stisen et al., 2015), DSADS (Barshan & Yüksek, 2014), OPPORTUNITY (Chavarriaga et al., 2013), Shoaib et al. (2014), and WEAR (Bock et al., 2024). For all datasets except WEAR, we follow the input window length and overlap settings in TSF (Zhang et al., 2026). For WEAR, an acceleration-only dataset incompatible with the gyroscope-dependent IF-ConvTransformer and TSF, we use the 2-second window and 1-second overlap proposed in the original dataset paper. Dataset and preprocessing details are provided in Appendix D. For the FM baselines and IMUDINO, we report the result of 2-layer MLP probe. In multi-sensor datasets, we extract feature from each sensor separately using the FM, and concatenate the features before the MLP. For the supervised baselines, we follow the protocols in TSF. We adopt leave-one-subject-out cross validation (LOSO CV) for all evaluation and report the macro F1 scores (mF1) in Table 1.

| Type | Method | MoSense | HAPT | HHAR | DSADS | WEAR | OPPO. | Shoaib | Average |
|---|---|---|---|---|---|---|---|---|---|
| Supervised Baselines | DeepConvLSTM | 93.39 | 82.00 | 91.90 | 83.25 | 75.43 | 40.88 | 97.13 | 80.57 |
| | A&D | 95.79 | 81.95 | 94.17 | 86.12 | 77.10 | 46.29 | 97.79 | 82.74 |
| | IF-ConvT. | 95.81 | 85.35 | 95.01 | 88.82 | N/A | 50.29 | 98.28 | 85.59 |
| | TSF | 97.34 | 85.89 | <u>95.81</u> | 89.70 | N/A | 52.44 | 99.35 | 86.76 |
| Self-supervised HAR FMs | SimCLR* | 96.08 | 80.56 | 87.56 | 91.56 | 88.48 | 43.50 | 99.29 | 83.86 |
| | Yuan et al. | 93.41 | 64.66 | 80.41 | 83.37 | 83.10 | 18.28 | 97.30 | 74.36 |
| | RelCon | 81.93 | 55.75 | 70.57 | 81.04 | 81.55 | 19.80 | 97.32 | 69.64 |
| | HAR-FM | 84.93 | 71.41 | 76.47 | 84.93 | 80.01 | 20.24 | 97.67 | 73.20 |
| General-purpose TSFMs | TimesFM 3.0 | 85.64 | 63.30 | 82.97 | 75.09 | 78.92 | 33.50 | 93.61 | 73.29 |
| | Chronos-2 | 83.95 | 57.04 | 72.60 | 73.04 | 62.93 | 24.65 | 90.71 | 66.36 |
| IMUDINO (Ours) | IMUDINO-6M | <u>97.91</u> | <u>87.56</u> | 95.33 | <u>93.24</u> | <u>91.28</u> | <u>53.54</u> | <u>99.58</u> | <u>88.35</u> |
| | IMUDINO-85M | **98.54** | **89.02** | **96.00** | **94.13** | **92.34** | **58.40** | **99.68** | **89.73** |

Table 1: Results on seven HAR datasets. IMUDINO consistently outperforms both supervised and SSL baselines. *Implemented with IMUDINO data augmentation.

The results show that IMUDINO can be used off-the-shelf for HAR and achieve state-of-the-art performance with lightweight classification heads. Since these datasets cover a wide range of sampling rates, input lengths and sensor locations, this also demonstrates IMUDINO's ability to adapt to diverse scenarios without finetuning. Visualization of IMUDINO global features is in Appendix C. We further explore the potential of IMUDINO's global features in complex HAR systems through an online open-set HAR experiment in Appendix E. Finally, we evaluate IMUDINO on four health and gait analysis tasks (Appendix F), where it also achieved competitive performance.

### 4.3 INERTIAL MOCAP: IMUDINO PROVIDES FINE-GRAINED DENSE FEATURES

Motion capture (MoCap) is an important application of inertial sensing. Inertial human pose estimation (HPE) is a challenging task, and existing work often include physical priors in the model and training objective. We explore whether IMUDINO can achieve competitive HPE performance *without* these priors. Since MoCap datasets often omit raw gyroscope readings, we synthesize the gyro channels as the finite difference of the $SO(3)$ sequence (Appendix I) used by DynaIP. In our experiments, we simply concatenate the patch tokens extracted from each sensor along the channel dimension without strong priors about the human body. Note that the input includes a head-mounted IMU, a location unseen in our training set. We then train the decoder, either an MLP or a transformer, to decode the root-relative 6D pose of 16 joints from the patch tokens using plain MSE loss. We use the AnDy (Maurice et al., 2019), UNIPD (Guidolin et al., 2022), CIP (Palermo et al., 2022),

and Virginia Natural Motion (Geissinger & Asbeck, 2020) datasets and follow DynaIP's Xsens-only experiment protocols (Zhang et al., 2024b). Detailed experiment settings are in Appendix I. We report the results in Table 2 and a visualized example in Fig. 2.

| Model | AnDy | | | UNIPD | | | CIP | | | Natural Motion | | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SIP↓ | Ang↓ | Pos↓ | SIP↓ | Ang↓ | Pos↓ | SIP↓ | Ang↓ | Pos↓ | SIP↓ | Ang↓ | Pos↓ | SIP↓ | Ang↓ | Pos↓ |
| DIP (Huang et al., 2018) | 11.39 | 5.73 | 4.34 | 12.02 | 5.47 | 4.22 | 19.13 | 8.61 | 6.86 | 33.43 | 11.91 | 13.33 | 18.99 | 7.93 | 7.19 |
| TransPose (Yi et al., 2021) | 12.15 | 6.29 | 4.91 | 15.11 | 6.05 | 4.82 | 20.06 | 8.75 | 6.86 | 30.62 | 11.26 | 11.99 | 19.49 | 8.09 | 7.15 |
| TIP (Jiang et al., 2022) | 10.11 | 4.55 | 3.56 | 9.85 | 4.06 | 2.78 | 13.05 | 5.67 | 4.30 | 22.06 | 7.90 | 7.92 | 13.77 | 5.55 | 4.64 |
| PIP (Yi et al., 2022) | 9.49 | 4.09 | **3.29** | 8.90 | 3.59 | 2.66 | 12.68 | 5.52 | 4.12 | 19.62 | 7.49 | 6.93 | 12.67 | 5.17 | 4.25 |
| DynaIP (Zhang et al., 2024b) | **8.93** | **3.45** | 3.41 | **7.29** | **2.77** | **2.21** | **11.42** | **4.54** | **3.69** | **15.78** | 7.18 | **5.83** | **10.86** | **4.49** | **3.79** |
| IMUDINO-85M+MLP | 11.64 | 4.24 | 3.99 | 11.66 | 3.95 | 3.29 | 16.25 | 5.87 | 5.13 | 19.03 | 7.30 | 6.44 | 14.65 | 5.34 | 4.71 |
| IMUDINO-85M+Transformer | 10.83 | 3.88 | 3.50 | 10.82 | 3.69 | 2.92 | 14.04 | 5.25 | 4.39 | 19.54 | 7.28 | 6.61 | 13.81 | 5.02 | 4.36 |
| IMUDINO-85M+MLP† | 10.15 | 3.72 | 3.43 | 9.45 | 3.38 | 2.50 | 12.19 | 4.76 | 3.95 | 17.35 | **6.65** | 5.92 | 12.28 | 4.63 | 3.95 |

Table 2: Inertial MoCap results. We report SIP error (°), global angular error (°), and joint position error (cm). Average denotes the macro-average over AnDy, UNIPD, CIP, and Natural Motion. †Backbone initialized from the pretrained IMUDINO checkpoint and finetuned.

The frozen MLP probe had TIP-level performance, while the MLP probe with a finetuned backbone performed better than PIP, trailing only DynaIP. Considering that IMUDINO is not aided by physical priors and does not have access to the raw gyroscope readings (only reconstructed ones), this result, despite not beating the current SOTA (DynaIP), still shows that IMUDINO's dense features contain rich information about human motion that can be used for pose estimation. With carefully crafted, physics-informed decoder architectures and training objectives, IMUDINO can plausibly become a serious contender in inertial HPE. More visualized HPE results are in Appendix K.

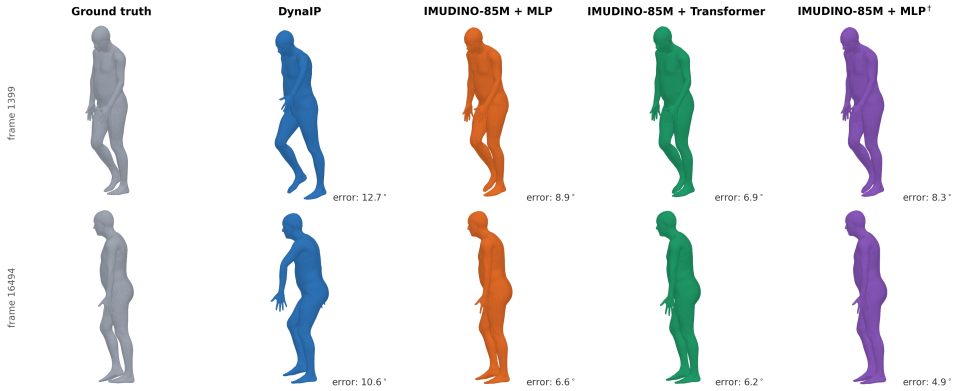


Figure 2: Qualitative comparison of inertial HPE on a natural motion sequence from Virginia Natural Motion. From left to right: ground truth, DynaIP, and the three IMUDINO variants of Table 2 (MLP probe, transformer probe, and MLP probe with finetuned backbone). Numbers below each mesh denote the per-frame mean root-aligned global rotation error (°). See Appendix K for more cases.

### 4.4 Robot Odometry: IMUDINO Generalizes Across Embodiments

Here, we compare IMUDINO to TartanIMU (Zhao et al., 2025), a robot odometry FM. This task is entirely OOD for IMUDINO since it is trained exclusively on wearable IMU data and has no exposure to robot IMUs, and it is kept frozen in this experiment. We train MLP and transformer probes on the 2026 TartanIMU Challenge[1] train split, and report the results on the official validation split (the test split labels are yet to be released at the time of writing) in Table 3. The evaluation covers various robot platforms, including car, dog, drone, and human (Zhao et al., 2025). TartanIMU has an expert model for each platform, and we report both platform-agnostic performance (prediction averaged over 4 experts) and the results of the specialized experts. IMUDINO is evaluated under the *platform-agnostic* setting. We report the standard AVE and ATE20 metrics, and the TartanIMU Challenge score. Visualized examples are in Fig. 3. Experiment settings are in Appendix J.

[1]https://superodometry.com/imuchallenge/

| Model | Car | | Dog | | Drone | | Human | | Average | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | AVE↓ | ATE20↓ | AVE↓ | ATE20↓ | AVE↓ | ATE20↓ | AVE↓ | ATE20↓ | AVE↓ | ATE20↓ | Score↓ |
| TartanIMU* | 0.852 | 2.273 | 0.425 | 1.901 | 1.374 | 2.772 | 0.235 | 1.877 | 0.721 | 2.206 | 0.872 |
| TartanIMU† | 0.199 | 0.999 | 0.129 | 0.644 | 0.966 | 2.006 | 0.107 | **0.814** | 0.350 | 1.116 | 0.429 |
| IMUDINO-85M+MLP* | 0.200 | 0.998 | 0.123 | 0.707 | 0.543 | 1.304 | 0.132 | 1.039 | 0.249 | 1.012 | 0.333 |
| IMUDINO-85M+Transformer* | **0.141** | **0.760** | **0.092** | **0.501** | **0.499** | **1.285** | **0.105** | 0.872 | **0.209** | **0.854** | **0.280** |

Table 3: Results on the TartanIMU Challenge. *Platform-agnostic. †Platform expert.

The results show that IMUDINO outperforms the existing SOTA, TartanIMU, on robot inertial odometry despite the task being OOD. This frozen-backbone cross-embodiment transfer indicates that IMUDINO dense features can readily transfer to unseen dynamics, in this case robot motion. To our knowledge, IMUDINO is the first inertial FM to demonstrate such capabilities, making progress towards *general-purpose foundation models* for the broader inertial sensing community.

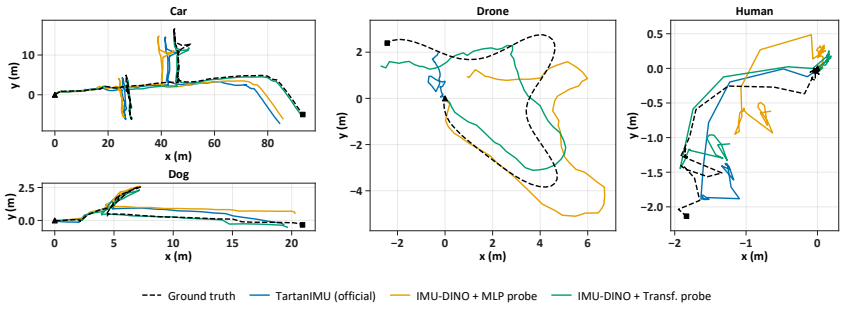


Figure 3: Visualized results on the TartanIMU Challenge. ▲ denotes the starting point of each trajectory. IMUDINO+Transformer (green line) traces the ground truth (black dashed line) most closely. The TartanIMU results (blue line) are from the corresponding expert model for each platform.

### 4.5 Ablation Study

In our ablation study, we dissect how each component of IMUDINO's training recipe contribute to overall performance. For training objective, we consider the impact of the patch-level objective and the KoLeo regularizer. For data augmentation, we analyze the effect of scaling, rotation, resampling and channel masking. We also explore the contribution of frequency-aware PE. Finally, we include results of randomly initialized backbones to quantify the gain from SSL pretraining. IMUFormer-85M is used in all experiments. We report HAR, MoCap and robot odometry results in Table 4.

| Type | Setting | HAR | MoCap (MLP Probe) | | | Robot Odometry (MLP Probe) | | |
|---|---|---|---|---|---|---|---|---|
| | | mF1 | SIP↓ | Ang↓ | Pos↓ | AVE↓ | ATE20↓ | Score↓ |
| Training Objective | w/o patch-level objective | 89.41 | 15.21 | 5.77 | 4.85 | 0.293 | 1.045 | 0.373 |
| | w/o KoLeo | 89.40 | 14.50 | 5.31 | 4.58 | 0.252 | 1.004 | 0.335 |
| Data Augmentation | w/o scaling | 89.63 | 14.87 | 5.35 | 4.74 | 0.257 | 1.016 | 0.340 |
| | w/o rotation | 88.66 | 15.67 | 5.52 | 4.93 | 0.253 | 1.036 | 0.339 |
| | w/o resampling | 88.86 | 14.64 | 5.25 | 4.59 | 0.255 | 1.012 | 0.338 |
| | w/o channel masking | 89.04 | 14.02 | 5.19 | 4.41 | 0.258 | 1.023 | 0.342 |
| Model | w/o freq-aware PE | 89.60 | 14.51 | 5.26 | 4.52 | 0.260 | 1.022 | 0.343 |
| Self-supervised Pretraining | random init, frozen backbone | 65.32 | 17.34 | 5.90 | 5.62 | 0.854 | 6.976 | 1.592 |
| | random init, trainable backbone | 83.70 | 14.89 | 5.32 | 4.77 | 0.534 | 2.852 | 0.802 |
| IMUDINO-85M | | 89.73 | 14.65 | 5.34 | 4.71 | 0.249 | 1.012 | 0.333 |

Table 4: Result of the ablation study. The results are averaged over multiple datasets and correspond to the *Average* columns in Tables 1, 2 and 3.

The ablation of training objectives show that the inclusion of a patch-level objective effectively improved IMUDINO's dense features. While the KoLeo regularizer lead to a slight drop in MoCap performance, it improved HAR performance and made training more stable. For data augmentation,

applying scaling and rotation led to improvements across the board. Resampling led to better HAR performance but virtually no change in MoCap and robot odometry performance. Considering resampling is key to IMUDINO's support of variable sampling rate, we kept resampling in the final recipe. Channel masking led to improved HAR performance but reduced MoCap metrics. However, we observed improved training stability with channel masking, so it is kept as well. Frequency-aware PE is positive overall, improving HAR and odometry while slightly hindering MoCap. Finally, IMUDINO outperformed a trainable, randomly-initialized backbone in all 3 tasks, indicating that our SSL pretraining yielded gains beyond the reach of conventional supervised learning.

**More ablation results.** An HAR stress test for variable input length and sampling rate is in Appendix G. A scaling experiment is in Appendix H.

### 4.6 Balancing Global and Dense Representations

How does IMUDINO balance global and patch-level representation? A general-purpose inertial FM needs to produce high-quality global and patch-level representations simultaneously. This creates a natural tension: global features demand semantic invariance, while dense features need to capture local variations. IMUDINO's training objective circumvents this conflict by applying the invariance-inducing view transformations only to the global objective $\mathcal{L}_{\text{DINO}}$, while the patch-level representation remains sensitive to these transformations. In Fig. 4, we compare the cosine similarity between tokens after different rotations around the x axis and resampling.

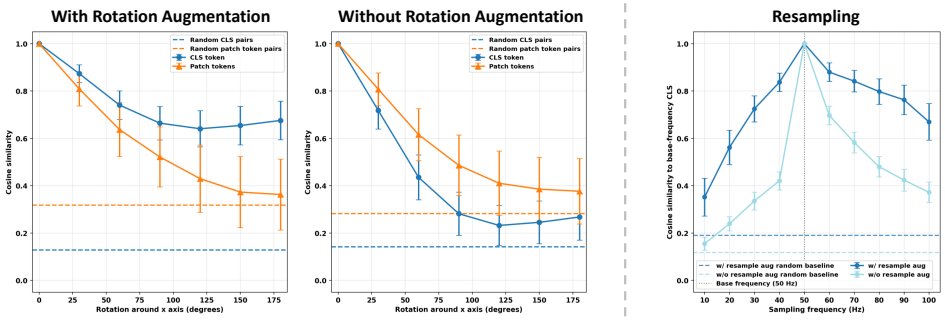


Figure 4: How CLS and patch tokens responds to perturbation. Our invariance-motivated view transformations induced in the CLS tokens the desired robustness to rotation and resampling. We further demonstrate in the rotation case that CLS robustness does not expand to patch tokens, which remain sensitive to rotation. Results aggregated over 700 examples from 7 HAR datasets.

In the model trained with rotation augmentation, the CLS similarity decreased only marginally when the input is rotated. The model trained without rotation sees a much more drastic decrease in similarity. In contrast, the patch token similarity trend is almost identical, decreasing to near-random level when we rotate 180°, indicating that orientation is retained in the dense features. In addition, we also demonstrate how the *resampling* transformation improve robustness to varying sampling rate. Again, data augmentation induced the desired invariance in global feature representation.

## 5 Conclusion

We introduced IMUDINO, a foundation model for inertial sensing. We adapted DINO self-distillation to inertial SSL through a set of global semantics-preserving view transformations, as well as a minimalist transformer architecture. We trained IMUDINO on a relatively modest corpus of 11600 hours of wearable inertial data covering various scenarios and sensor setups. Extensive experiments on HAR, health, motion capture and robot odometry demonstrated that this simple approach is remarkably effective, and produces generalizable global and dense features. To the best of our knowledge, it is the first inertial FM to demonstrate competitive performance on a broad range of global and dense prediction tasks, including OOD ones like robot odometry. IMUDINO demonstrated that it is possible to learn broadly generalizable motion representation from inertial data, and that inertial FMs do not need to narrowly focus on one sub area (e.g., HAR). We hope IMUDINO can empower inertial sensing applications and kickstart research into general-purpose inertial FMs.

### AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, some part of the Appendix has been polished by AI, which is then checked by the authors. We did not write or polish any part of the main text using generative AI. We have not used generative AI tools for designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, interpreting results, or implementing methods. Other tasks requiring disclosure are not applicable to this work. These include generating synthetic datasets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, and supporting qualitative and thematic data analysis. We have reviewed all AI-assisted work and take full responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We intend to make this work fully reproducible. The details needed to train IMUDINO are described in Sec. 3 and Appendices A and B. Furthermore, we include detailed description of all experiment protocols in Appendices D, I and J. The experiments in Appendices E and F also come with detailed descriptions.

## REFERENCES

Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C. Ranasinghe. Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(1), March 2021. doi: 10.1145/3448083. URL https://doi.org/10.1145/3448083.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, Jorge Luis Reyes-Ortiz, et al. A public domain dataset for human activity recognition using smartphones. In *Esann*, volume 3, pp. 3–4, 2013.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. URL https://arxiv.org/abs/2510.15821.

Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M. Hausdorff, Nir Giladi, and Gerhard Troster. Wearable assistant for parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine*, 14(2):436–446, 2010. doi: 10.1109/TITB.2009.2036165.

Billur Barshan and Murat Cihan Yüksek. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *The Computer Journal*, 57(11): 1649–1667, 2014.

Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. Wear: An outdoor sports dataset for wearable and egocentric activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–21, 2024.

Bingyi Cao, Koert Chen, Kevis-Kokitsi Maninis, Kaifeng Chen, Arjun Karpur, Ye Xia, Sahil Dua, Tanmaya Dabral, Guangxing Han, Bohyung Han, Joshua Ainslie, Alex Bewley, Mithun Jacob, René Wagner, Washington Ramos, Krzysztof Choromanski, Mojtaba Seyedhosseini, Howard Zhou, and Andre Araujo. Tipsv2: Advancing vision-language pretraining with enhanced patch-text alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29325–29335, June 2026.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Shing Chan, Yuan Hang, Catherine Tong, Aidan Acquah, Abram Schonfeldt, Jonathan Gershuny, and Aiden Doherty. Capture-24: A large dataset of wrist-worn activity tracker data collected in the wild for human activity recognition. *Scientific Data*, 11(1):1135, 2024.

Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. The opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters*, 34(15):2033–2042, 2013.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Julia F Christensen, Andrés Fernández, Rebecca A Smith, Georgios Michalareas, Sina HN Yazdi, Fahima Farahi, Eva-Madeleine Schmidt, Nasimeh Bahmanian, and Gemma Roig. Emokine: A software package and computational framework for scaling up the creation of highly controlled emotional full-body movement datasets. *Behavior Research Methods*, 56(7):7498–7542, 2024.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Jack H. Geissinger and Alan T. Asbeck. Motion inference using sparse inertial sensors, self-supervised learning, and a new dataset of unscripted human motion. *Sensors*, 20(21), 2020. ISSN 1424-8220. doi: 10.3390/s20216330. URL https://www.mdpi.com/1424-8220/20/21/6330.

Hristijan Gjoreski, Mathias Ciliberto, Francisco Javier Ordoñez Morales, Daniel Roggen, Sami Mekki, and Stefan Valentin. A versatile annotated dataset for multimodal locomotion analytics with mobile devices. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, SenSys '17, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450354592. doi: 10.1145/3131672.3136976. URL https://doi.org/10.1145/3131672.3136976.

Mattia Guidolin, Emanuele Menegatti, and Monica Reggiani. Unipd-bpe: Synchronized rgb-d and inertial data for multimodal body pose estimation and tracking. *Data*, 7(6), 2022. ISSN 2306-5729. doi: 10.3390/data7060079. URL https://www.mdpi.com/2306-5729/7/6/79.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, June 2022.

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 Trans. Graph.*, 37(6), December 2018. ISSN 0730-0301. doi: 10.1145/3272127.3275108. URL https://doi.org/10.1145/3272127.3275108.

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*, SA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394703. doi: 10.1145/3550469.3555428. URL https://doi.org/10.1145/3550469.3555428.

Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg. Anyloc: Towards universal visual place recognition. *IEEE Robotics and Automation Letters*, 9(2):1286–1293, 2023.

Mohammad Malekzadeh, Richard G. Clegg, Andrea Cavallaro, and Hamed Haddadi. Protecting sensory data against sensitive inferences. In *Proceedings of the 1st Workshop on Privacy by Design in Distributed Systems*, W-P2DS'18, pp. 2:1–2:6, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5654-1. doi: 10.1145/3195258.3195260. URL http://doi.acm.org/10.1145/3195258.3195260.

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

Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10), 2017. ISSN 2076-3417. doi: 10.3390/app7101101. URL https://www.mdpi.com/2076-3417/7/10/1101.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

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

Stylianos Paraschiakos, Ricardo Cachucho, Matthijs Moed, Diana van Heemst, Simon Mooijaart, Eline P Slagboom, Arno Knobbe, and Marian Beekman. Activity recognition using wearable sensors for tracking the elderly: S. paraschiakos et al. *User Modeling and User-Adapted Interaction*, 30(3):567–605, 2020.

Veronika Potter, Hoan Tran, Daniel Mobley, Suzanne M. Bertisch, Dinesh John, and Stephen Intille. The physical activity assessment using wearable sensors (paaws) dataset: Labeled laboratory and free-living accelerometer data. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 9(4), December 2025. doi: 10.1145/3770639. URL https://doi.org/10.1145/3770639.

Minghui Qiu, Cekai Weng, Mingming Fan, and Kaishun Wu. Towards customizable foundation models for human activity recognition with wearable devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 9(3), September 2025. doi: 10.1145/3749479. URL https://doi.org/10.1145/3749479.

Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Multi-task self-supervised learning for human activity detection. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(2), June 2019. doi: 10.1145/3328932. URL https://doi.org/10.1145/3328932.

Yoli Shavit and Itzik Klein. Boosting inertial-based human activity recognition with transformers. *IEEE Access*, 9:53540–53547, 2021. doi: 10.1109/ACCESS.2021.3070646.

Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul J. M. Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014. ISSN 1424-8220. doi: 10.3390/s140610146. URL https://www.mdpi.com/1424-8220/14/6/10146.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana,

Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. URL https://arxiv.org/abs/2508.10104.

Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pp. 127–140, 2015.

Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. Exploring contrastive learning in human activity recognition for healthcare. *arXiv preprint arXiv:2011.11542*, 2020.

Jamie Tolan, Hung-I Yang, Benjamin Nosarzewski, Guillaume Couairon, Huy V. Vo, John Brandt, Justine Spore, Sayantan Majumdar, Daniel Haziza, Janaki Vamaraju, Theo Moutakanni, Piotr Bojanowski, Tracy Johns, Brian White, Tobias Tiecke, and Camille Couprie. Very high resolution canopy height maps from rgb imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar. *Remote Sensing of Environment*, 300:113888, 2024. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2023.113888. URL https://www.sciencedirect.com/science/article/pii/S003442572300439X.

Eugene Vorontsov, Alican Bozkurt, Adam Casson, George Shaikovski, Michal Zelechowski, Kristen Severson, Eric Zimmermann, James Hall, Neil Tenenholtz, Nicolo Fusi, et al. A foundation model for clinical-grade computational pathology and rare cancers detection. *Nature medicine*, 30(10): 2924–2935, 2024.

Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, SenSys '21, pp. 220–233, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450390972. doi: 10.1145/3485730.3485937. URL https://doi.org/10.1145/3485730.3485937.

Maxwell Xu, Alexander Moreno, Hui Wei, Benjamin M Marlin, and James Rehg. Rebar: Retrieval-based reconstruction for time-series contrastive learning. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15440–15464, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/439bf902de1807088d8b731ca20b0777-Paper-Conference.pdf.

Maxwell Xu, Jaya Narain, Gregory Darnell, Haraldur Hallgrimsson, Hyewon Jeong, Darren Forde, Richard Fineman, Karthik Raghuram, James Rehg, and Shirley Ren. Relcon: Relative contrastive learning for a motion foundation model for wearable data. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 53188–53206, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/83eb339ed42297658fa24b5cec939285-Paper-Conference.pdf.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10371–10381, June 2024.

Zhoutong Ye, Yanwen Huang, Chun Yu, Yuntao Wang, Yuqi Luo, and Yuanchun Shi. Activityseeker: Towards collaborative personalized human activity discovery and recognition on smartphones. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, CHI '26, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400722783. doi: 10.1145/3772318.3791014. URL https://doi.org/10.1145/3772318.3791014.

Xinyu Yi, Yuxiao Zhou, and Feng Xu. Transpose: real-time 3d human translation and pose estimation with six inertial sensors. *ACM Trans. Graph.*, 40(4), July 2021. ISSN 0730-0301. doi: 10.1145/3450626.3459786. URL https://doi.org/10.1145/3450626.3459786.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13157–13168, 2022. doi: 10.1109/CVPR52688.2022.01282.

Hang Yuan, Shing Chan, Andrew P Creagh, Catherine Tong, Aidan Acquah, David A Clifton, and Aiden Doherty. Self-supervised learning for human activity recognition using 700,000 person-days of wearable data. *NPJ digital medicine*, 7(1):91, 2024.

Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. If-convtransformer: A framework for human activity recognition using imu fusion and convtransformer. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(2), July 2022. doi: 10.1145/3534584. URL https://doi.org/10.1145/3534584.

Ye Zhang, Longguang Wang, Qing Gao, Chaocan Xiang, Mohammed Bennamoun, and Yulan Guo. Triple spectral fusion for sensor-based human activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(9):10856–10873, 2026. doi: 10.1109/TPAMI.2026.3690949.

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 (CVPR)*, pp. –, 2024a.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1889–1899. IEEE, 2024b.

Shibo Zhao, Sifan Zhou, Raphael Blanchard, Yuheng Qiu, Wenshan Wang, and Sebastian Scherer. Tartan imu: A light foundation model for inertial positioning in robotics. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22520–22529, 2025. doi: 10.1109/CVPR52734.2025.02097.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.

Qin Zou, Yanling Wang, Qian Wang, Yi Zhao, and Qingquan Li. Deep learning-based gait recognition using smartphones in the wild. *IEEE Transactions on Information Forensics and Security*, 15:3197–3212, 2020. doi: 10.1109/TIFS.2020.2985628.

# A Implementation Details

## A.1 Motion Transformations

We report how each motion transformation is implemented below.

**Cropping.** We sample the global crop size from $U(0.5, 0.8)$ and the local crop size from $U(0.05, 0.4)$. We then randomly choose the starting point of each crop.

**Scaling.** Scaling is randomly applied to the views with a probability of $p = 0.5$. Both the speed factor and magnitude factor are sampled from $U(0.8, 1.2)$. Speed up / down is implemented with linear interpolation, while the magnitude factor is simply multiplied to the sensor readings. Empirically, mild scaling like this rarely change the global semantics of motion significantly.

**Rotation.** Rotation is randomly applied to the views with a probability of $p = 0.5$. We sample a random rotation matrix from $SO(3)$. This covers the full $-180^\circ$ to $180^\circ$ range on all three axes.

**Resampling.** Resampling is randomly applied to the views with a probability of $p = 0.5$. The new frequency (in Hz) is sampled from $U(30, 200)$. We use linear interpolation to implement resampling.

**Channel masking.** Channel masking is applied to each channel with a probability of $p = 0.2$. The masked channel is replaced with zeros.

### A.2 Models and Training Recipe

**Models.** Detailed model hyperparameters are in Table 5. IMUFormer-85M is identical to ViT-B except for the number of attention heads, which we increased from 12 to 24.

| | IMUFormer-2.7M | IMUFormer-6M | IMUFormer-25M | IMUFormer-85M |
|---|---|---|---|---|
| Patch size | 16 | 16 | 16 | 16 |
| Token dim | 192 | 256 | 512 | 768 |
| Num heads | 6 | 8 | 16 | 24 |
| Num layers | 6 | 8 | 8 | 12 |
| Register Tokens | 4 | 4 | 4 | 4 |
| MLP type | GELU MLP | GELU MLP | GELU MLP | GELU MLP |
| MLP ratio | 4 | 4 | 4 | 4 |
| Drop path | 0.0 | 0.0 | 0.1 | 0.3 |

Table 5: The 4 IMUFormer models used in this paper.

**Training Recipe.** We train the models on 6 Nvidia H100 GPUs using PyTorch DDP and BF16 AMP. We use a batch size of 1536 samples, leading to 3072 global views and 9216 local views per batch. To improve training efficiency, we limit global views to 1024 frames and local views to 512 frames. Frames exceeding this cutoff, generated when the *resample* transformation upsamples a crop, are discarded. We train the models for 60 epochs using the AdamW optimizer, with a 1000-step linear warmup, a post-warmup LR of 5e-4, a cosine annealing LR schedule and a weight decay of 5e-2. Similar to DINO, we warmup the teacher's softmax temperature linearly from 0.04 to 0.07 in the first 1000 steps, and gradually increase its EMA momentum from 0.996 to 1.0 following a cosine schedule spanning the entire training run. To improve training stability, we clip the gradient norm to 3.0. As in DINOv2, we compute $\mathcal{L}_{\mathrm{KoLeo}}$ separately for each GPU to maximize efficiency. Training takes 11 hours for IMUFormer-85M and 6 hours for IMUFormer-6M.

## B Pre-training Dataset Processing

In this section we present details on how we filtered the pre-training dataset. We generally follow this four step process: file-level quality control, unit normalization, cropping into 1024-frame samples, and sample-level quality control.

**Capture24.** We removed recordings where the sampling rate deviate from the normal 100 Hz as quality control. We further filtered out segments with null or invalid annotation. The remaining data has the unit of acceleration normalized to $\mathrm{m/s^2}$ (default unit is $g$).

**SHL.** We did not filter SHL data, as all SHL data has consistent sampling rate and no NaN values. We simply combine the train and validation sets from the SHL 2026 Challenge.

**PAAWS.** First, we removed all data where the "Sensor orientation issue(s)" field is not empty or "None" (352 files). We also removed data where the sampling rate deviates significantly from the expected 100 Hz (3 files). For the remaining files, we removed all NaN readings, and normalized the unit of acceleration from $g$ to $\mathrm{m/s^2}$ and that of angular velocity from $\mathrm{deg/s}$ to $\mathrm{rad/s}$. Finally, we filtered samples where with inconsistent activity / posture labels, as well as static samples with linear acceleration below 0.30 $\mathrm{m/s^2}$ **and** angular velocity below 0.08 $\mathrm{rad/s}$.

## C Visualization of IMUDINO Features

We present a t-SNE visualization of IMUDINO features in Fig. 5. IMUDINO features can distinguish different activities, as well as different people performing the same activity.

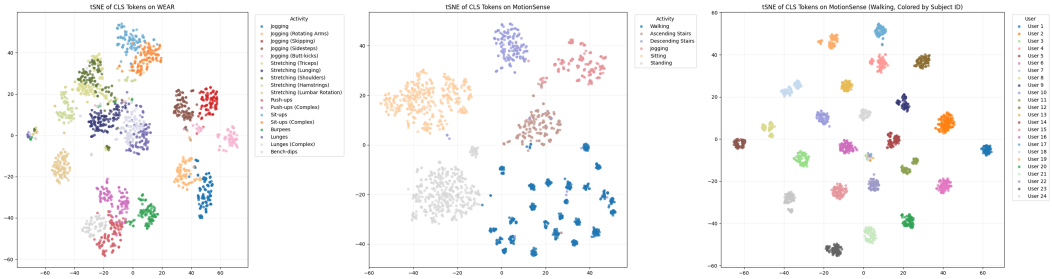


Figure 5: IMUDINO feature visualization. Samples are colored by label. Activities from WEAR (left) and MotionSense (middle) form distinct clusters. Furthermore, for the *walking* class in MotionSense (right), each user forms a distinct cluster, indicating that IMUDINO is capable of capturing the differences in gait between individuals. Further results on gait recognition are in Appendix F.

# D HAR Experiment Setup

## D.1 Datasets

Seven datasets are used in our HAR experiment: MotionSense (Malekzadeh et al., 2018), HAPT (Reyes-Ortiz et al., 2016), HHAR (Stisen et al., 2015), DSADS (Barshan & Yüksek, 2014), OPPORTUNITY (Chavarriaga et al., 2013), Shoaib et al. (2014), and WEAR (Bock et al., 2024). We provide a detailed description of the number of subjects, activity classes, and sensor setups of the datasets, as well as the protocols used in our experiments.

**MotionSense** is a smartphone IMU dataset covering 24 subjects and 6 activities (ascending stairs, descending stairs, walking, jogging, sitting, and standing). The smartphone was placed in the subjects' front pockets, and the experiment was done in a controlled setting. The sampling rate is 50 Hz.

**HAPT** is an extended version of the classic UCI-HAR (Anguita et al., 2013) smartphone IMU dataset covering 30 subjects and 12 activities. These include 6 normal activities (ascending stairs, descending stairs, walking, sitting, standing, and lying down), and 6 types of transitions (stand-to-sit, sit-to-stand, sit-to-lie, lie-to-sit, stand-to-lie, and lie-to-stand). The smartphone was waist-mounted and had a sampling rate of 50 Hz.

**HHAR** is a smartphone dataset covering 9 subjects and 6 activities (ascending stairs, descending stairs, walking, cycling, sitting, and standing). Each subject carried 8 smartphones (4 different models) in a tight pouch secured around their waist. The sampling rates ranged from 50 Hz to 200 Hz, depending on the phone model.

**DSADS** is a wearable IMU dataset covering 8 subjects and 19 activities (walking in a parking lot, 4 km/h walking on flat treadmill, 4 km/h walking on inclined treadmill, 8 km/h running on treadmill, ascending stairs, descending stairs, exercising on a stepper, exercising on a cross trainers, cycling on an exercise bike in horizontal position, cycling on an exercise bike in vertical position, rowing, jumping, playing basketball, sitting, standing, lying on back, lying on right side, standing still in an elevator, and moving around in an elevator). 5 IMUs were placed on each subject (torso, right arm, left arm, right leg, and left leg). The IMUs had a sampling rate of 25 Hz.

**OPPORTUNITY** is a wearable IMU dataset covering 4 subjects in a sensor rich environment. Following Zhang et al. (2026), we use the most challenging 18-class setting (null class, open door 1, open door 2, close door 1, close door 2, open fridge, close fridge, open dishwasher, close dishwasher, open drawer 1, open drawer 2, open drawer 3, close drawer 1, close drawer 2, close drawer 3, clean table, drink from cup, and toggle switch). We use the seven 6-axis IMUs (back, right upper arm, left upper arm, right lower arm, left lower arm, left shoe, right shoe) as the input. The data was collected at 30 Hz.

**WEAR** is a wearable IMU dataset covering 22 subjects and 18 workout activities, including 5 types of jogging, 5 types of stretching, and 8 types of strength training. It also included a null class. The data is collected from four 50 Hz accelerometers (one on each ankle and one on each wrist).

**Shoaib et al. (2014)**'s HAR dataset covers 10 subjects and 7 activities ((ascending stairs, descending stairs, walking, jogging, cycling, sitting, and standing). The data is collected from 5 smartphones (right jeans pocket, left jeans pocket, belt clipper, right upper arm, and right wrist) at 50 Hz.

Combined, these datasets cover a huge variety of sensor locations, including on the limbs and torso, as well as in pockets and pouches. IMUDINO's SOTA performance across all seven datasets indicates strong generalization across sensor locations.

### D.2 PROTOCOL

We train a 2-layer MLP probe on top of frozen IMUDINO features (or frozen features from the FM baselines). The MLP has the following structure: Dropout(0.2) $\rightarrow$ Linear($D \times S \rightarrow 512$) $\rightarrow$ GELU $\rightarrow$ Dropout(0.5) $\rightarrow$ Linear($512 \rightarrow K$), where $D$ is the feature dimension of the backbone, $S$ is the number of sensors (features are concatenated along the *channel* dimension) and $K$ the number of classes. We use a mild data augmentation scheme: random scaling ($p = 0.5$) identical to IMUDINO pretraining settings, random rotation ($p = 0.5$) along the three axes sampled from $U(-30°, 30°)$, and random jitter ($p = 0.5$, Gaussian noise from $\mathcal{N}(0, 0.05)$). We use the AdamW optimizer with a weight decay of 0.1 and a cosine LR schedule. Detailed data processing protocols, and the LR for each dataset, are in Table 6. All experiments follow the leave-one-subject-out (LOSO) cross validation protocol. The sampling rate, window size, window overlap, and LOSO protocol we used are identical to those of TSF (Zhang et al., 2026), making the HAR results directly comparable except for WEAR, an accelerometer-only dataset lacking the gyroscope channels required by TSF. For WEAR, we follow the 2-second window with 1-second overlap setup in the original dataset paper (Bock et al., 2024).

| **Datasets** | MotionSense | HAPT | HHAR | DSADS | WEAR | OPPORTUNITY | Shoaib et al. (2014) |
|---|---|---|---|---|---|---|---|
| Sampling Rate (Hz) | 50 | 50 | 100* | 25 | 50 | 30 | 50 |
| Window Size (s) | 2.56 | 2.56 | 2.00 | 5.00 | 2.00 | 1.60 | 0.96 |
| Overlap (s) | 1.28 | 1.28 | 1.00 | 0.00 | 1.00 | 0.80 | 0.48 |
| LR | 1e-3 | 1e-3 | 1e-3 | 1e-4 | 3e-4 | 3e-4 | 3e-4 |
| Epochs | 100 | 100 | 100 | 100 | 50 | 50 | 50 |

Table 6: Data processing protocol, LR, and the number of training epochs for each HAR dataset. We use the final epoch checkpoint for test set evaluation. *Phones with different sampling rate are resampled to 100 Hz.

## E ONLINE OPEN-SET HAR

In addition to conventional HAR, we consider the challenge of *personalization*. Online personalized HAR aims to solve the problem by learning user-specific patterns through actual use and interactions. One example is ActivitySeeker (Ye et al., 2026), a recent personalized HAR system that (1) extracts features from unlabeled IMU data, (2) discovers motion patterns through clustering, (3) asks the user for a label when the current activity matches a cluster, (4) labels the entire cluster with the user-provided information, and (5) trains lightweight classifiers using the personalized dataset. The performance of ActivitySeeker's *collaborative labeling* strategy is highly dependent on the feature extraction model. Moreover, the ActivitySeeker dataset covers diverse sensor locations and activities. These factors make ActivitySeeker a suitable test bench for IMUDINO.

In this experiment, we reproduce the simulated online learning experiment of Ye et al. (2026) and replace the original ResNet feature extraction model with IMUDINO. Note that IMUDINO-6M is around the same size as the ResNet in the original paper. We report the metrics used by Ye et al. (2026) in Table 7. All metrics are averaged over 13 users. The results demonstrate that motion patterns are revealed much more efficiently when IMUDINO features are used, leading to higher data utilization and better HAR performance.

| Backbone / Condition | Data Util. | Collab. Labeling Acc. | Activities Discovered | HAR mF1 |
|---|---|---|---|---|
| W/o Personalization | - | - | - | 62.6 |
| Supervised ResNet | 67.4 | 98.8 | 8.3 / 8.7 | 89.5 |
| IMUDINO-6M | 76.5 | 98.7 | **8.4 / 8.7** | 95.3 |
| IMUDINO-85M | **77.3** | **99.3** | **8.4 / 8.7** | **96.8** |

Table 7: Results on ActivitySeeker. Using IMUDINO off-the-shelf improves all metrics, demonstrating its potential as a general-purpose inertial feature extraction model.

# F Gait Analysis and Health

## F.1 Task Overview and Results

We evaluate IMUDINO on a variety of gait analysis and health tasks. These include activity intensity estimation on GOTOV (Paraschiakos et al., 2020), fall detection on UniMiB-SHAR (Micucci et al., 2017), freezing of gait (FOG) detection in Parkinson's patients on Daphnet FOG (Bachlin et al., 2010), and gait recognition on whuGAIT (Zou et al., 2020). We follow the experiment settings described in the original dataset papers and report the results in Table 8. Overall, IMUDINO delivers competitive performance on all 4 datasets, demonstrating its potential in diverse health and gait analysis applications.

| Method | GOTOV | | UniMiB-SHAR | | Daphnet FOG | | whuGAIT | |
|---|---|---|---|---|---|---|---|---|
| | RMSE↓ | $R^2$ | HAR Acc | Fall Acc | Spec. | Sens. | Acc | EER↓ |
| Official Baseline | 1.35 | 0.45 | 56.58 | 97.05 | 0.869 | **0.781** | 93.75 | - |
| IMUDINO-6M+MLP | 0.96 | 0.58 | 83.96 | 99.59 | 0.908 | 0.671 | 96.55 | 3.24 |
| IMUDINO-85M+MLP | **0.94** | **0.59** | **84.78** | **99.81** | **0.936** | 0.551 | **97.41** | **2.92** |

Table 8: Results on four health and gait analysis datasets. Official baseline results are from the original dataset paper, whose experiment protocols are followed in IMUDINO experiments.

## F.2 Common setup

**Input convention.** Each sensor stream is a 6-channel window: accelerometer $xyz$ in m/s$^2$ followed by gyroscope $xyz$ in rad/s. For acceleration-only datasets, the gyroscope channels are zero-filled, matching the acceleration-only streams seen in pretraining. The per-sample sampling rate is passed to the frequency-aware positional encoding, so each dataset keeps its native sampling rate unless stated otherwise.

**Probe and optimization.** Unless stated otherwise in the next section, the probe is a 2-layer MLP over the frozen class token(s) of the last layer: Dropout(0.2) $\rightarrow$ Linear($D \times S \rightarrow 512$) $\rightarrow$ GELU $\rightarrow$ Dropout(0.5) $\rightarrow$ Linear($512 \rightarrow K$), where $D$ is the feature dimension of the backbone, $S$ is the number of sensor streams (features are concatenated across streams) and $K$ the number of classes (1 for regression). A fresh probe is trained per fold with AdamW (weight decay 0.1), cosine learning-rate decay, bf16 autocast, and cross-entropy with label smoothing 0.2 (classification) or MSE on fold-standardized targets (regression). Training windows receive the same light augmentation as our HAR evaluation (random scaling, $\pm 30^\circ$ rotation, speed and jitter perturbations) unless stated otherwise. Subject splits follow each dataset's canonical protocol, defaulting to leave-one-subject-out cross validation (LOSO CV).

## F.3 Datasets and per-task settings

We provide a summary of the health and gait analysis task settings in Table 9.

**Energy expenditure (GOTOV (Paraschiakos et al., 2020)).** GOTOV provides synchronized accelerometry (three GENEActiv units at ankle, wrist and chest, resampled from 85.71 Hz to 50 Hz) and breath-by-breath EE from a COSMED K4b2 (kcal/min) for 31 older adults. We follow the protocol of Paraschiakos et al. (2020): EE labels are aggregated into 10 s bins, and each bin is predicted

| Dataset | Task | Protocol |
|---|---|---|
| whuGAIT | gait authentication | fixed split (20 unseen subjects) |
| UniMiB-SHAR | 17-way HAR / fall detection | LOSO |
| Daphnet | freeze of gait (FOG) detection | LOSO, 4 s/0.5 s windows |
| GOTOV | energy expenditure (EE) regression | LOSO, 120 s→10 s bins |

Table 9: Summary of the gait analysis and health evaluation. All experiments use frozen IMUDINO backbones with lightweight MLP probes.

from the preceding 120 s acceleration window (10,875 windows, labels 0.70–14.18 kcal/min), evaluated with LOSO CV. The probe is a $2304 \to 512 \to 1$ MLP (three concatenated sensor features) trained with MSE on fold-standardized targets, lr $10^{-3}$, 150 epochs, batch 256.

**Activity and fall recognition (UniMiB-SHAR (Micucci et al., 2017)).** UniMiB-SHAR contains 30 subjects performing 9 activities of daily living (ADL) and 8 fall types, recorded by a 50 Hz smartphone accelerometer and segmented into peak-centered 3 s windows (151 samples; truncated to 144 to align with the patch size). We run 17-way classification with LOSO CV (the dataset's own protocol), lr $10^{-3}$, 100 epochs, batch 1024.

**Freezing-of-gait detection (Daphnet FOG (Bachlin et al., 2010)).** Daphnet records 10 Parkinson's patients with three 64 Hz accelerometers (ankle, thigh, trunk; values converted from mg to m/s$^2$). We follow the canonical protocol of Bachlin et al. (2010): 4 s windows with 0.5 s step, a window is labeled *freeze* if at least 50% of its samples are annotated as freeze, and segments outside the protocol are discarded, giving 35,405 windows (9.6% freeze) evaluated with LOSO CV. We use balanced oversampling to mitigate the data imbalance.

**Gait authentication (whuGAIT (Zou et al., 2020), Dataset #5).** whuGAIT Dataset #5 contains 74,142 authentication pairs from 118 subjects (66,542 training pairs from 98 subjects; 7,600 test pairs from 20 disjoint subjects), each pair consisting of two horizontally aligned 128-sample two-step segments of 50 Hz smartphone accelerometer+gyroscope data. The task is binary same/different-subject verification on the fixed split. The input here are the concatenated interaction features $[a, b, |a - b|, a \cdot b]$ over per-segment features (layer-11 class token concatenated with mean-pooled patch tokens). We also use swap augmentation $(a, b) \to (b, a)$, an lr of $2.5 \times 10^{-4}$ with 60 epochs and early stopping on a 10% validation split carved from the *training* pairs only.

## G HAR WITH VARIABLE INPUT LENGTH AND SAMPLING RATE

In this section, we report MLP-probe results on MotionSense under different input length and sampling rate conditions sampled from the following grid: $\{1.28\text{s}, 2.56\text{s}, 5.12\text{s}, 10.24\text{s}\} \times \{25\text{Hz}, 50\text{Hz}, 100\text{Hz}, 200\text{Hz}\}$, in Table 10. The IMUDINO-85M variant is used. The results show that changing the sampling rate does not impact HAR performance by much, especially for larger windows (5.12s and 10.24s). Likewise, IMUDINO handles all 4 window sizes well, with larger windows bringing slight improvements as expected. In summary, this experiment demonstrates that IMUDINO does indeed support variable input length and sampling rate.

| Window | 25 Hz | 50 Hz | 100 Hz | 200 Hz |
|---|---|---|---|---|
| 1.28 s | 97.40 | 97.56 | 97.60 | 97.86 |
| 2.56 s | 98.26 | 98.54 | 98.67 | 98.72 |
| 5.12 s | 98.89 | 98.91 | 98.92 | 98.81 |
| 10.24 s | 99.16 | 99.34 | 99.33 | 99.24 |

Table 10: MotionSense mF1 of IMUDINO-85M with the 2-layer MLP probe for different input window lengths and sampling rates (resampled from the native 50 Hz). Supervised SOTA: 97.34 (2.56 s, 50 Hz, Zhang et al. (2026)).

# H SCALING IMUDINO

We explore how scaling affects IMUDINO performance and report the results in Fig. 6. For HAR and robot odometry, larger models consistently perform better than smaller ones. MoCap, however, seems to saturate at 25M parameters. This type of behavior is not uncommon in the DINO paradigm: DINOv2 ViT-L outperformed its larger ViT-g counterpart on instance-level recognition (Oquab et al., 2024), while DINOv3 ViT-L outperformed its larger ViT-H+ counterpart on segmentation, tracking, and keypoint matching (Siméoni et al., 2025). Overall, we believe that scaling up IMUDINO further is worth exploring as future work as there is still considerable potential in inertial SSL waiting to be unlocked.

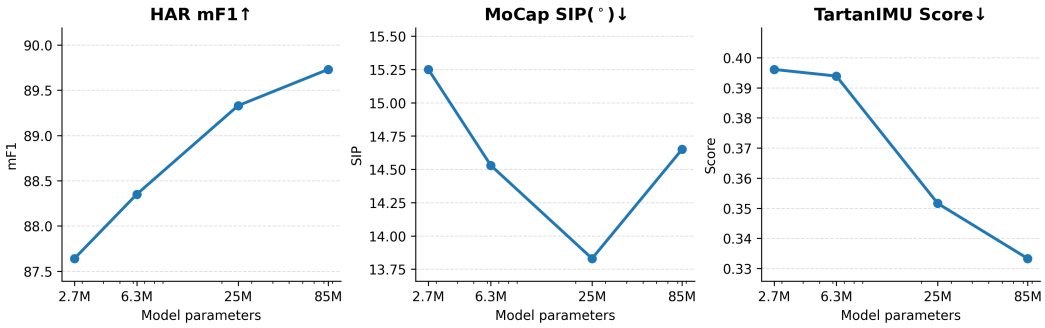


Figure 6: How IMUDINO models of different sizes perform in HAR, MoCap, and robot odometry.

# I MOCAP EXPERIMENT SETUP

## I.1 TASK AND DATASETS

Given six body-worn IMUs (pelvis, head, left/right lower arm, left/right lower leg; 60 Hz), the model regresses full-body articulation. Following DynaIP (Zhang et al., 2024a), we use the publicly available Xsens motion-capture datasets (AnDy (Maurice et al., 2019), CIP (Palermo et al., 2022), Emokine (Christensen et al., 2024), UNIPD (Guidolin et al., 2022), and Virginia Natural Motion (Geissinger & Asbeck, 2020); 1,106 training and 177 held-out test trials). Ground-truth articulation is the Xsens skeleton, mapped to the SMPL joint layout following DynaIP.

**Synthesizing Gyroscope Readings.** None of these datasets provide raw gyroscope readings. To match the six-axis input format of IMUDINO, we synthesize per-sensor angular velocity from the calibrated global-orientation sequence $R_t \in SO(3)$ via the relative rotation between consecutive frames: $\omega_t = \mathrm{Log}(R_t^\top R_{t+1})/\Delta t$, where $\Delta t = 1/60\,\mathrm{s}$ and $\mathrm{Log}(\cdot)$ is the $SO(3)$ logarithm map, yielding angular velocity in the sensor frame. The finite difference is computed on the full uncropped sequence so that frames near a later crop retain valid neighbors (the last frame repeats the previous value). Since DynaIP takes the $SO(3)$ sequence directly as input while IMUDINO uses the finite difference of the same sequence, the comparison is fair as neither model has access to information unavailable to the other.

## I.2 MODEL

The backbone is the pretrained IMUDINO-85M encoder, kept *frozen*. Per IMU we encode a 480-frame (8 s) window into 30 patch tokens (patch size 16). Tokens from the six IMUs are concatenated patch-wise ($6\times768$) and projected to 512 dimensions. The MLP probe applies a 2-layer MLP (dim 512), linearly interpolates the resulting sequence back to the input resolution (480 frames), and predicts per-frame 16-joint root-relative 6D rotations of the SMPL skeleton with a per-frame linear head (2.67M trainable parameters). The transformer probe inserts a 2-layer transformer decoder (dim 512, 8 heads, 7.44M trainable parameters) between the projection and the interpolation head: 30 learnable pose queries cross-attend to the projected patch tokens, then follow the same per-patch MLP, interpolation, and per-frame head. Predictions are assembled into 24-joint global rotations us-

ing the measured pelvis orientation for the root, measured calibrated orientations for IMU-equipped joints, and parent copying for IMU-less joints, following the assembly of DynaIP.

### I.3 Training

Decoders are trained with a 6D rotation MSE loss on the Xsens training split (additionally including DIP-IMU for the Xsens+DIP variants), using AdamW (lr 1e-3, weight decay 1e-4, cosine schedule with warmup), batch size 48, 30 epochs, windows of 480 frames with stride 240, and bf16 autocast. Main-text models use gravity-restored accelerometer inputs. For the finetuned variant († in Table 2), the backbone is unfrozen and trained jointly with the decoder at a lower learning rate (1e-4).

### I.4 Evaluation Protocol

The entire test sequence is predicted in one pass with sliding windows stitched at their centers. We use the official DynaIP evaluator and metrics: SIP error (upper-arm/upper-leg global rotation), global angular error (GAE, root and ignored joints replaced by ground truth), joint position error (JPE, cm, via SMPL/Xsens forward kinematics), and mesh vertex error (cm, SMPL only). Results are reported in Table 2 of the main text.

## J Robot Odometry Experiment Setup

### J.1 Task and Datasets

Given a 1 s window of 6-axis IMU data (accelerometer + gyroscope, 200 Hz, body frame, gravity retained), the model predicts the window-mean body-frame velocity. Training and validation use the TartanIMU challenge corpus (car/dog/drone/human; 395 training and 80 validation trajectories, 27 h). Labels are window means of the ground-truth body velocity.

### J.2 Model

The backbone is the pretrained IMUDINO encoder kept *frozen*. Each IMU stream is encoded into patch tokens, and the decoder operates on the final-layer features of the input window (12 patch tokens for the default 1 s input). We consider two decoders: a 2-layer transformer (dim 512, 8 heads, 6.96M parameters) that also consumes the `CLS` token and mean-pools over tokens, and a lighter MLP probe (3 layers, dim 512, 0.66M parameters) over the mean-pooled patch tokens. Both regress the 3D body velocity and receive no platform information. Training uses Huber loss ($\beta$=1), AdamW (lr $10^{-3}$, weight decay 0.05, warmup + cosine), 30 epochs, EMA (0.999), and platform-balanced sampling of training windows.

### J.3 Sliding Window Strategies

We consider two sliding window strategies: the sparse variant uses non-overlapping 1 s windows; the dense variant (main-text default) slides the training windows with a 0.2 s stride (5× windows). These settings only affect training, and the evaluation protocol (1-second windows, no overlap) is identical to the official TartanIMU Challenge evaluation script. Results are shown in Table 11.

| Window strategy | AVE↓ | | ATE20↓ | | Score↓ | |
|---|---|---|---|---|---|---|
| | MLP | Transformer | MLP | Transformer | MLP | Transformer |
| 1 s, non-overlapping | 0.283 | 0.262 | 1.103 | 0.992 | 0.373 | 0.341 |
| 1 s, 0.2 s stride (default) | 0.249 | 0.209 | 1.012 | 0.854 | 0.333 | 0.280 |

Table 11: Window-strategy ablation on the TartanIMU validation split (frozen IMUDINO-85M, macro-averaged over platforms). Denser training windows help both probes.

### J.4 Evaluation Protocol

Models are scored on the official validation split with the official TartanIMU metrics: per-platform AVE (m/s) and ATE20 (m), macro-averaged over the four platforms, and the official TartanIMU

Score ($0.6 \cdot \text{AVE}/0.7356 + 0.4 \cdot \text{ATE20}/3.116$; lower is better). IMUDINO is evaluated platform-agnostic (a single shared model for all platforms). Results are reported in Table 3 of the main text.

## K Additional MoCap Visualizations

We present addition visualized results from the motion capture task. IMUDINO dense features can support accurate inertial human pose estimation (HPE). With task-specific decoder architectures and training objective with physical and body priors, we expect IMUDINO to be a serious contender for SOTA inertial HPE.

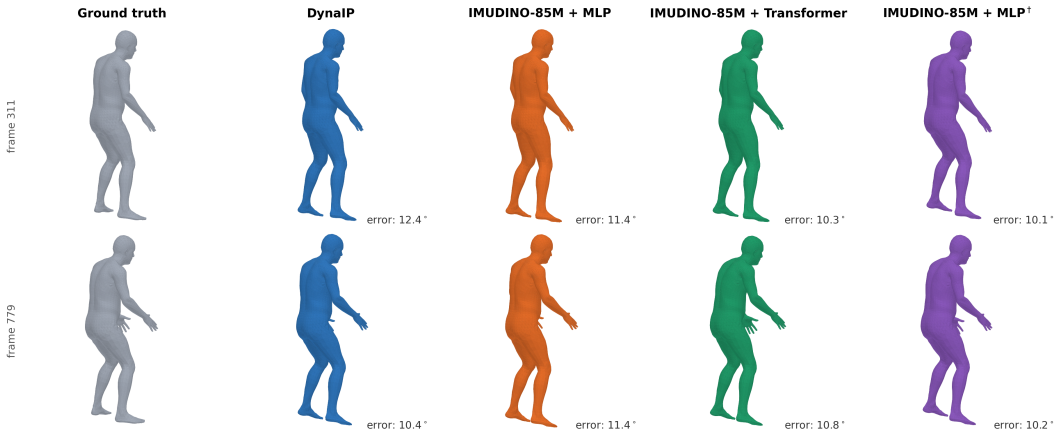


Figure 7: Results on a squatting sequence from the UNIPD test set.

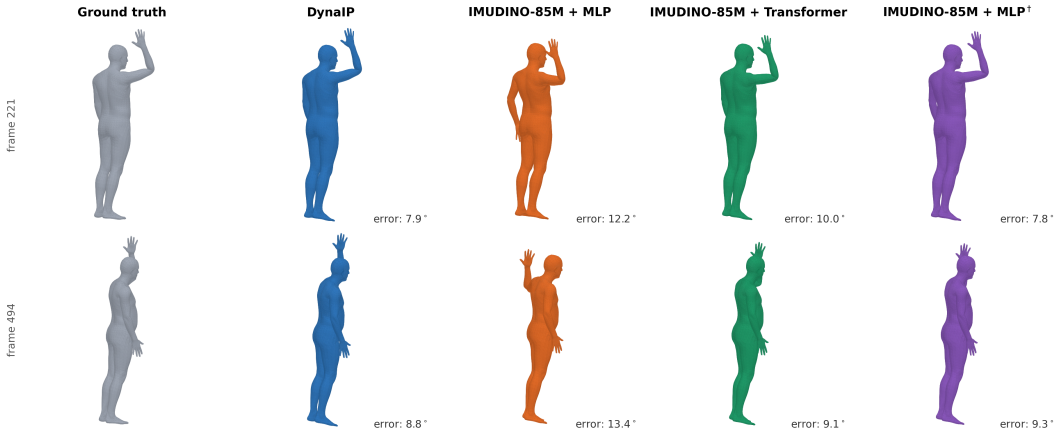


Figure 8: Results on a throwing sequence from the UNIPD test set.

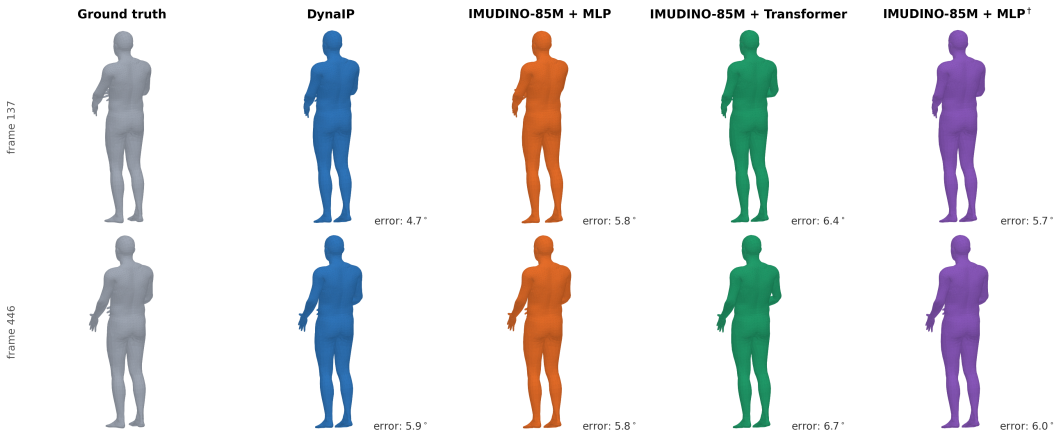


Figure 9: Results on a cross-arms sequence from the UNIPD test set.

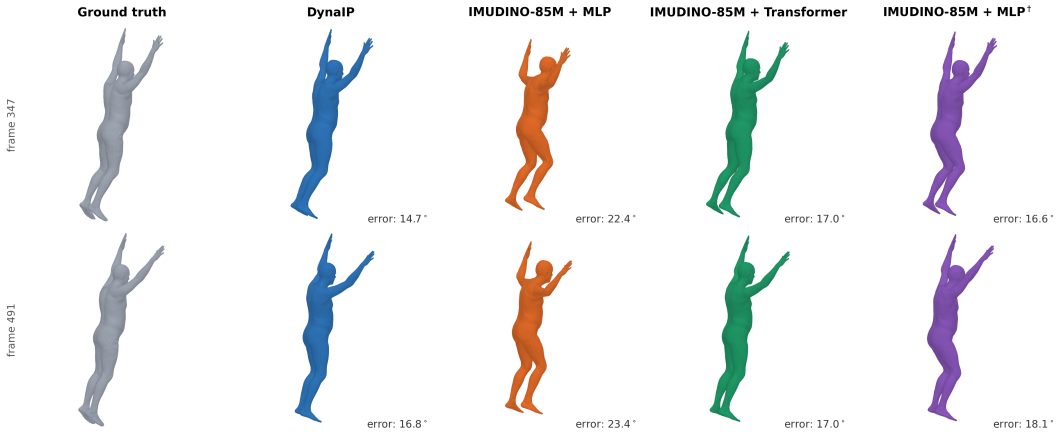


Figure 10: Results on a jumping sequence from the UNIPD test set.

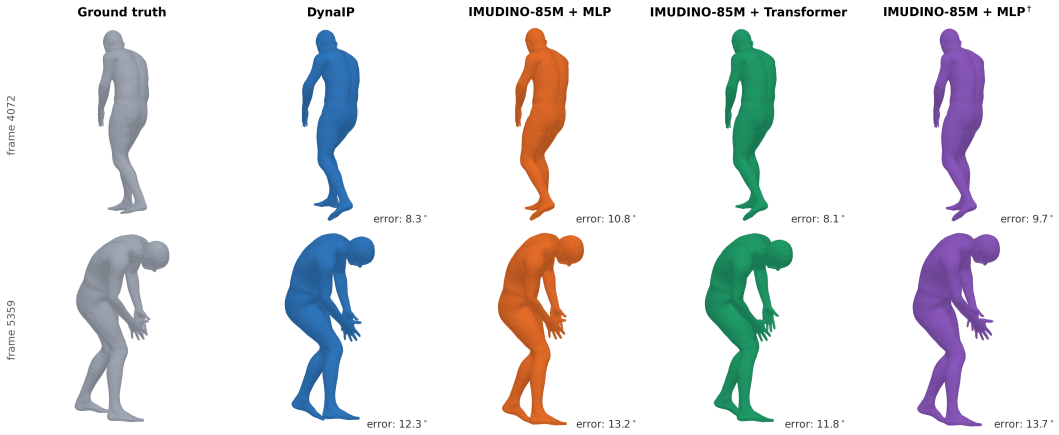


Figure 11: Results on a lifting sequence from the AnDy test set.

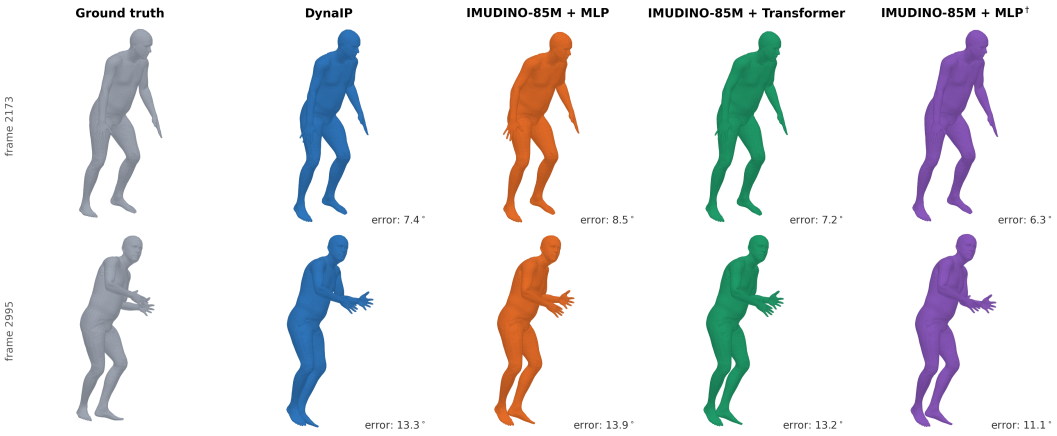


Figure 12: Results on a circuit sequence from the CIP test set.

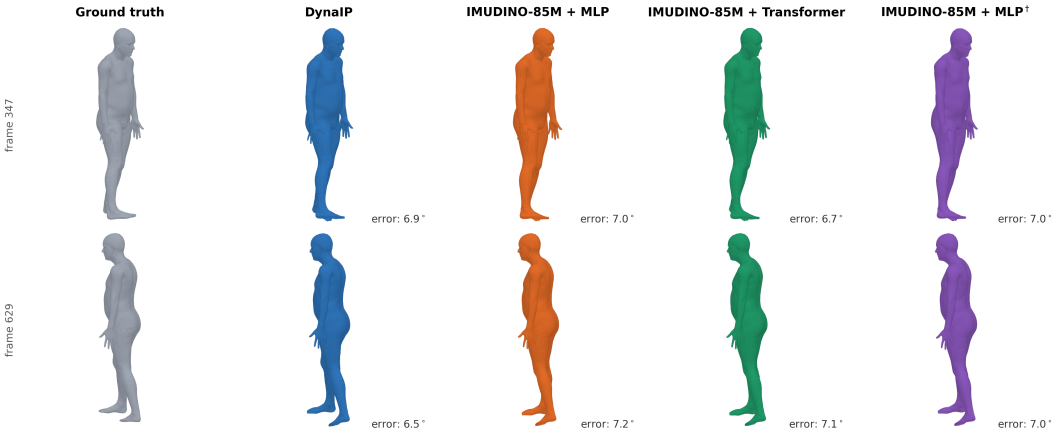


Figure 13: Results on a walking sequence from the UNIPD test set.