SetCon: Setup-Conditioned IMU-Language Learning from 3D Human Motion

Lennart Schiweck[1] Alaa Saleh[2] Cristóbal Curio[1]
[1]Cognitive Systems Group, Reutlingen University, Germany [2]Bosch Sensortec GmbH, Germany

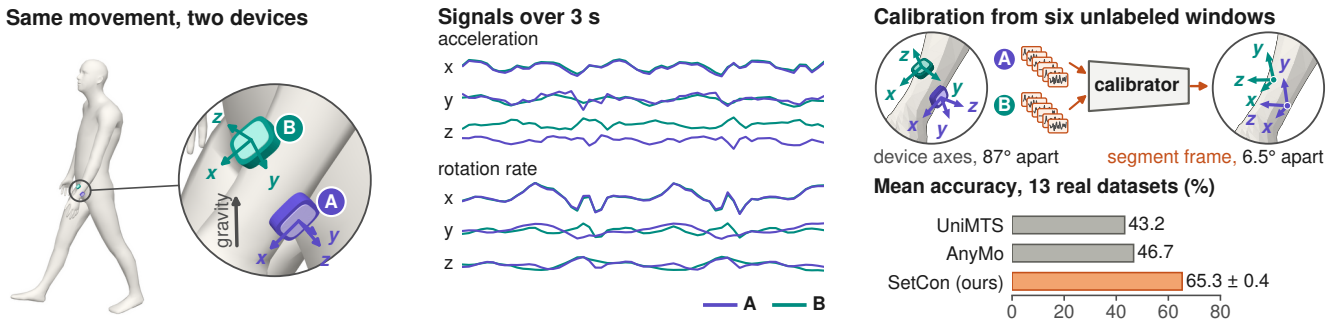


Figure 1. **Same movement, two devices.** Two virtual devices on one forearm (left), 4.6 cm apart and rotated by 87° against each other, record one Nymeria [21] walk differently. For this pair, chosen for illustration, the axes of the two devices are 6.5° apart in the forearm's segment frame (right) once SetCon's calibrator has estimated each device's setup reference from six unlabeled windows. The bars give the mean accuracy of UniMTS [50], AnyMo [3] and SetCon with anchors on 13 real datasets (Tab. 1, last row, ± sd over three seeds).

Abstract

Aligning inertial measurement unit (IMU) signals with text enables zero-shot activity recognition across wearable devices. A worn device's signal depends on its setup, the combination of body site and mounting. Leading IMU-language models make their encoders invariant to the mounting, which also discards the device's orientation relative to the body, such as the direction of gravity that separates sitting, standing and lying. We introduce SetCon, which replaces full invariance with selective invariance. It estimates each device's mounting on its known body site from a few unlabeled windows and conditions its encoder on this estimate. Because the mounting stays nearly fixed while the device is worn, the moving body serves as a calibration target. Both calibration and representation learning are supervised by 3D human motion, since one movement can be simulated under many known sensor setups. At test time, a class is represented by its name or by an anchor, which averages the text embeddings of the name and of activity descriptions and needs no labeled window. A language model built on the frozen encoder also answers how and in what order a movement is performed. Trained only on IMU simulated from the same Nymeria motion capture as AnyMo, SetCon reaches 56.6 % mean zero-shot accuracy on 13 real datasets with class names and 65.3 % with anchors, against 41.8 % and 46.7 % for AnyMo.

1. Introduction

An inertial measurement unit (IMU) in a smartwatch, a phone or any other wearable records how the body segment it sits on accelerates and rotates. If an encoder mapped 10-s windows of this signal into the embedding space of a text encoder [8, 27], as CLIP [32] does for images, each window could be classified zero-shot as the activity whose text embedding lies nearest, on any wearable and without labeled data. Because real IMU with activity labels is scarce, recent encoders learn from virtual IMU [15–17], simulated on 3D human body models driven by recorded motion [3, 50]. UniMTS [50] and AnyMo [3] align such virtual IMU with text and classify activities on real datasets from class-name prompts. We evaluate on the 13 real datasets of the UniMTS suite that record a gyroscope on every device, so that UniMTS, AnyMo and our model each receive their full input. On Mean-13, their mean accuracy, UniMTS and AnyMo reach 41.4 and 41.8 % with class names.

The obstacle is the sensor setup [12, 13, 49], which consists of the body site where a device is worn, such as a wrist or the hip, and its mounting, how it is attached and rotated there. Because the same movement produces different signals under different mountings (Fig. 1), UniMTS and AnyMo make their encoders invariant to the mounting. UniMTS uses random rotation augmentation and AnyMo learns motion tokens that do not depend on how a device is placed within its body segment. Invariance removes

the mounting but also the direction of gravity relative to the body. For a motionless wearer, this direction indicates whether trunk and thighs are upright or horizontal and thus separates sitting, standing and lying. A hip device worn upright on a standing person and the same device rotated by 90° on a person lying on the back record the same window, with gravity along the same device axis and zero angular velocity. Accordingly, on the 7 test datasets with at least 20 windows of each posture, UniMTS queried with anchors classifies on average 76 % of the sitting, standing and lying windows as standing, while AnyMo classifies 56 % of them as sitting.

Most deployments, however, provide the information that these models leave unused. The body site of a device is known and its mounting stays nearly fixed while the device is worn, so the windows it records during that time share one mounting. A few of these windows, unlabeled and drawn at random, suffice to estimate how the device is oriented on its segment, because the moving body serves as a calibration target [14, 54]. We therefore introduce SetCon (Fig. 2, top), which estimates the setup and provides it to its encoder instead of removing it. A calibrator estimates this orientation and its uncertainty from the body site and unlabeled windows of the device as the *setup reference* (Sec. 3.2), the rotation from device axes into the *segment frame*, axes fixed to the body segment (Fig. 1). The encoder receives every window rotated so that gravity points up [23, 25], which keeps the vertical axis explicit and with it the direction in which the wearer turns. It also receives each window's gravity in the segment frame, which distinguishes the two hip devices.

3D human motion makes this conditioning learnable. A device simulated on the 3D body model that defines the segment frame has a known mounting, which gives the calibrator its target and a measure of its own errors. One movement can be simulated under many setups, so that the encoder learns how it appears under each, whereas a real recording carries one mounting per device that datasets rarely record. Like prior work [3, 50], SetCon generates its training data from 3D motion, but also uses it to supervise the setup estimate.

The setup reference conditions the IMU side of the alignment. On the text side, the training captions and the class prompts determine how much of the separation in the embeddings carries over to zero-shot classification. SetCon trains on both the human narrations of the motion capture [21] and captions that describe kinematics, generated by a large language model (LLM) from the measured 3D motion of the same windows. At test, a class is represented by the text embedding of its name or by an anchor, a CLIP-style prompt ensemble that averages the text embeddings of the name, its nearest training narrations and LLM-written descriptions of the activity [24, 29], so no labeled window is needed. Because a class name does not say how or in what order a movement is performed, a small language model built on the frozen encoder [19] is trained to answer such questions.

Our contributions are as follows.

- We propose setup conditioning for IMU-language learning, which replaces full invariance with *selective invariance*. The sensor setup becomes an input, so that the encoder can discount how a device is mounted while it keeps the direction of gravity relative to the body and the direction in which the wearer turns. A calibrator estimates the setup from a few unlabeled windows of the device, which it selects by its tilt uncertainty (Secs. 3.2 and 3.5).
- We use existing 3D human motion to supervise the setup estimate of an IMU-language encoder, not only to generate training data. Real IMU datasets rarely record the mounting, whereas a simulated device has a known one. Simulation therefore provides the calibrator's target and lets the encoder learn how one movement appears under many setups (Secs. 3.1 to 3.3).
- We extend language access to the encoder beyond class names. For training, an LLM generates captions that describe kinematics from the measured 3D motion. At test, anchors need no labeled window and improve every model, most of all when the encoder is trained with these captions. A language model on the frozen encoder is more accurate than class-name prompts and answers how and in what order a movement is performed (Secs. 3.4 to 3.6 and 4.4).
- Trained only on simulated IMU, SetCon achieves state-of-the-art zero-shot accuracy on average over 13 real datasets, with class names and with anchors. Its embeddings also separate the activities best, measured without any text prompt (Sec. 4.2).

2. Related work

IMU-language pretraining. IMU2CLIP [27] and ImageBind [8] bind head-worn IMU to egocentric video and text. UniMTS [50] and AnyMo [3] align IMU simulated on body models with text from an LLM, while ZeroHAR [34], LanHAR [45], IMUZero [39] and HALO [5] add sensor or activity descriptions to training on labeled real IMU. ZeroHAR, HALO and AnyMo describe each sensor in text, whereas SetCon estimates from unlabeled windows how a test device is mounted on its known site.

Virtual IMU from body models. Inertial pose estimation, from DIP to MobilePoser [10, 26, 44, 47, 48], trains on IMU synthesized from AMASS [22] on SMPL bodies [20]. IMUTube [15] and IMUGPT [16, 17] derive virtual IMU from video and text-to-motion. WIMUSim [28], W2W [33] and IMUCoCo [53] model placement and device variation. SetCon simulates its devices on Momentum Human Rig (MHR) [6] fits of Nymeria [4, 21] and uses the body model

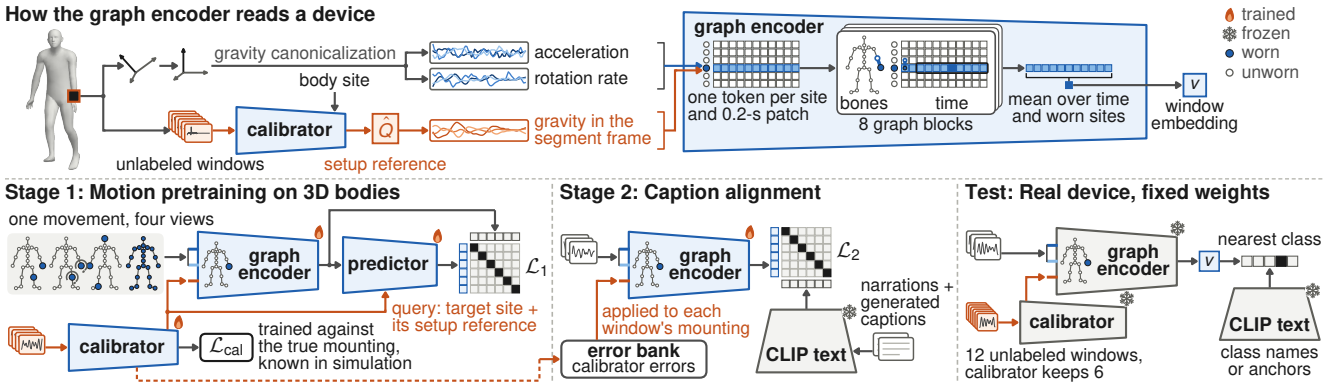


Figure 2. **Overview of SetCon.** The top row shows how the graph encoder, a spatio-temporal graph network over the body sites, reads a device. Every sample is rotated so that gravity points up and a calibrator estimates the setup reference $\hat{Q}$ (orange) from unlabeled windows and the body site, so that the encoder also receives gravity in the segment frame. In the bottom row, stage 1 learns how a movement appears under a target setup and stage 2 aligns the encoder with captions under calibrator errors from an error bank. The weights stay fixed at test.

not only to generate data and to vary the placement, as these works do, but also to supervise mounting estimation.

Placement, orientation and test-time adaptation. Displacement, placement and facing are long-standing problems in on-body recognition [12, 13], answered by invariance through orientation-removing transforms [49], rotation-invariant backbones [2] or augmentation [50], or by alignment with gravity through averaging [25] or attitude filters [23] and with the gait direction [14]. The Transformer IMU Calibrator [54], also trained with simulated offsets, estimates sensor-to-bone offsets from short windows for inertial motion capture and is closest to our calibrator. Test-time adaptation updates weights or statistics on unlabeled test data [40, 42, 52]. SetCon changes no weights at test and does not rotate the signal into the segment frame, whose forward axis is ambiguous under gait. An encoder pretrained to use the estimate receives it with the signal.

Descriptions and language models for IMU. Pratt *et al*. [29] and Menon and Vondrick [24] replace class-name prompts with LLM-written descriptions, while Ghosh [7] writes them by hand for sensor prototypes. HARGPT [11], LLaSA [37], SensorLLM [18], EgoLM [9], Ego4o [43] and Mojito [36] feed IMU or egocentric motion signals into LLMs. Zhang *et al*. [51] find that vision-language models classify images worse than the CLIP image encoder they are built on [32]. SetCon builds anchors from such descriptions for its frozen IMU embeddings and trains a language model on top of its encoder as in LLaVA [19].

3. Method

SetCon is trained on virtual IMU from 3D bodies and applied to real IMU with fixed weights (Fig. 2). For every device, virtual or real, a calibrator estimates the setup reference from its unlabeled windows and the encoder receives it with each window rotated so that gravity points up. Stage 1 pretrains the encoder on movements simulated under several views (sets of worn devices with their setups) and stage 2 aligns it with captions under calibrator errors.

3.1. Virtual devices on 3D bodies

A virtual device sits on a body site with a mounting, given by a skin point and a rotation M_d from joint to device axes. This mounting is known for every simulated device but rarely recorded for a real one. We train on Nymeria's motion capture [21], represented as MHR bodies [6] and cut into consecutive, non-overlapping windows of up to 10 s. Nymeria also provides wristband IMU and narrations. Stage 1 uses all 94,868 training windows and stage 2 the 71,125 of them that overlap a narration. As in UniMTS [50], signals are sampled at 20 Hz on a graph of 22 sites that take position and axes from MHR joints. Every site is a node of the encoder, worn or not. Each site has eight skin points on the MHR surface and each wrist has thirty on a ring around the distal forearm. In half of the windows the wrist device follows the hand and in the other half only the pronation of the forearm. In stage 1, M_d follows the skin surface as in AnyMo [3] and is drawn as $M_d = R_y(\epsilon_2)\, R_x(\epsilon_1)\, R_z(\psi)\, T_d^\top$, where the columns of T_d are two tangents and the outward normal of the skin at the device, R_x, R_y and R_z rotate about the coordinate axes, ψ is a uniform rotation about the normal and the tilts ϵ_1 and ϵ_2 are uniform within $\pm 10°$. Re-wearing a device draws a new skin point and a new M_d. The accelerometer reads the specific force f_d (the acceleration minus gravity in device axes) and the gyroscope reads the angular velocity ω_d, both computed by central differences over the sample interval Δt

as

$$f_d(t) = M_d R_d(t)^\top \big(\ddot{p}_d(t) - g\big),$$
$$\omega_d(t) = \frac{M_d R_d(t)^\top}{2\Delta t} \log\big(R_d(t{+}\Delta t)\, R_d(t{-}\Delta t)^\top\big)^\vee, \quad (1)$$

where R_d is the world orientation of the joint, $p_d = J_d + R_d\, r_d$ the device position given by the joint position J_d and the skin point r_d in joint axes, g gravity, log the matrix logarithm and $(\cdot)^\vee$ the vector of a skew-symmetric matrix. We then apply an 8-Hz low-pass filter and add white noise and a per-window constant bias. On 98 wrists of 49 held-out recordings, virtual magnitudes and those of Nymeria's real wristband correlate per window with a median of 0.78 for the specific force and 0.82 for the angular velocity, compared with at most 0.11 and 0.27 when a window is paired with another window, its time reverse or the other wrist.

3.2. Setup reference and gravity canonicalization

An accelerometer senses gravity, so the signal of every device contains its tilt. This tilt has two sources. The tilt of the body segment separates postures, whereas the way the device is mounted on it is arbitrary. Gravity therefore reaches the encoder by two routes (Fig. 2, top row). Gravity canonicalization removes the tilt from the motion signals. The setup reference returns the segment's part.

Gravity canonicalization. Every sample is rotated so that gravity points up, by the shortest rotation $R_\uparrow(t)$ that turns the estimated gravity direction $\hat{g}_d(t)$ in device axes onto the vertical axis. A complementary filter [23] with gain $1\,\mathrm{s}^{-1}$ estimates $\hat{g}_d$ from accelerometer and gyroscope. On accelerometer-only devices a zero-phase 0.3-Hz low-pass of the accelerometer replaces it. Because one rotation acts on both signals, the signed rotation rate about the vertical is preserved and distinguishes left from right turns. Canonicalization cannot, however, separate device tilt from segment tilt, so it removes both.

Setup reference. The reference describes how a device sits on its body segment (Fig. 1). On the 3D body model, each site has a segment frame W_d, measured once in joint axes on 60 simulated Nymeria recordings. We use the quarter of their 10-s windows with the most rotation, where the body is in motion. The up axis of W_d is the mean vertical direction in these windows and its forward axis the principal horizontal axis of their dynamic acceleration. The reference is the fixed rotation from device axes into W_d, which covers the mounting's rotation but not its skin point. For a simulated device it is known, $Q_d^* = W_d M_d^\top$. Applied to the gravity direction, it yields gravity in the segment frame, the tilt of the segment whatever the mounting.

Calibrator. A real device does not report its mounting. The calibrator C_ϕ therefore estimates the reference from k unlabeled windows $X_{d,1}, \ldots, X_{d,k}$ of one worn device, its *context*, raw in device axes with gravity included, together with a missing-gyroscope flag and the region and side ℓ_d of the site. Two cues make this possible. Because the up axis of W_d is the mean vertical in motion, gravity averaged over a few windows determines it in device axes. Gait fixes the forward axis, but only up to its sign, so Q_d^* and $R_y(\pi)Q_d^*$ are equally valid targets, where $R_y(\pi)$ is the rotation by π about the up axis of W_d, its y axis.

A 1-D convolutional network encodes each window, with embeddings of ℓ_d and the flag added, into a feature w_n. A multilayer perceptron (MLP) head on the order-invariant pooling $[\mathrm{mean}_n\, w_n; \max_n w_n]$ outputs a density $p_\phi(Q)$ over rotations, a mixture of two Matrix-Fisher components on $SO(3)$, so that the estimate carries its own uncertainty. The estimate $\hat{Q}_d = C_\phi\big(\{X_{d,n}\}_{n=1}^k, \ell_d\big)$ is the mode of the heavier component. Its tilt uncertainty $\sigma_\uparrow$ is the spread of that component's up axis. The calibration loss is the negative log-likelihood of the target, symmetrized over $R_y(\pi)$, plus an equivariance term $\mathcal{L}_{\mathrm{eq}}$,

$$\mathcal{L}_{\mathrm{cal}} = -\log \tfrac{1}{2}\big[p_\phi(Q_d^*) + p_\phi(R_y(\pi)\, Q_d^*)\big] + \lambda_{\mathrm{eq}}\, \mathcal{L}_{\mathrm{eq}}, \quad (2)$$

where $\lambda_{\mathrm{eq}} = 0.5$. The term $\mathcal{L}_{\mathrm{eq}}$ requires that rotating all raw vectors of the context by a uniformly random $S \in SO(3)$ turns the predicted density with them, from Q to $QS^\top$. The target Q_d^* requires the true mounting, which only a simulated device provides. The calibrator is therefore pretrained with this loss on simulated devices, refined in stage 1, then frozen.

Encoder input. The nine channels of the encoder input hold the canonicalized specific force and angular velocity (acceleration and rotation rate in Fig. 2) divided by fixed scales κ_f and κ_ω, followed by the reference channels $b_d = \hat{Q}_d \hat{g}_d$, gravity in the segment frame,

$$x_d(t) = \big[R_\uparrow f_d/\kappa_f;\; R_\uparrow\, \omega_d/\kappa_\omega;\; \hat{Q}_d\, \hat{g}_d\big](t), \quad (3)$$

where $R_\uparrow = R_\uparrow(t)$. The scales $\kappa_f = 6.20\,\mathrm{m/s^2}$ and $\kappa_\omega = 0.74\,\mathrm{rad/s}$ are the root-mean-square values per axis of canonicalized joint-center signals on 500 training clips. The reference channels thus carry the segment tilt, but nothing about segments without a device. Their horizontal part depends on the forward axis of W_d, which gait fixes only up to its sign, so this part is randomized in training (Sec. 3.3).

Walking frame. As a training-free alternative to the calibrator (Sec. 4.3), the Walking frame [14, 25] estimates the up axis and a sign-ambiguous forward axis in device axes, as for W_d, from the context windows that pass a gait test on the periodicity of their acceleration magnitude and leaves

the reference channels at zero if none does. Training simulates it from W_d with the measured spread of its up axis.

3.3. Encoder and motion pretraining

Encoder. The graph encoder E_θ (Fig. 2, top right) is a spatio-temporal graph network over the 22 sites with the bones as edges, as in ST-GCN [46] and AnyMo [3], but over tokens of 0.2-s patches rather than samples. A worn site holds one token per patch, a linear projection of its patch of x_d. An unworn site holds a learned mask token, which relays information between worn sites. Every token also receives learned embeddings of its site, time and gyroscope flag and a linear map of the 44 worn-site and gyroscope flags. Eight residual graph blocks each update every site from itself, its parent and its children with learned edge weights and then convolve along time over nine patches with stride one. The tokens of the worn sites, averaged over these sites and over all time steps that hold signal, projected and normalized, give the window embedding v. Tokens have width 256 and v has 512 dimensions.

Stage 1: Motion pretraining. The encoder learns to predict how a movement appears under a given setup, with the reference as the only cue to the mounting of the target device (Fig. 2, bottom left). Each batch simulates $N = 64$ movements under four views, namely a random set of 1 to 5 sites, the same set re-worn (Sec. 3.1), a second random set and all 22 sites. We adopt the predictor and contrastive loss of AnyMo [3], which predicts the pooled features of a second view, but let the predictor target one device and condition it on that device's setup reference. Each step also simulates 32 further movements on the same devices as calibrator context. The calibrator estimates one $\hat{Q}_d$ per device from a random subset of 6 to 32 of them. The reference of device d in view β is this estimate rotated about the up axis of W_d by an angle η, uniform except at the pelvis, legs and lower spine, where gait fixes the forward axis up to its sign and η is drawn from $\{0, \pi\}$. This keeps the encoder from relying on a forward direction the calibrator cannot estimate. The reference replaces $\hat{Q}_d$ in Eq. (3) and is zeroed for 20 % of the devices so that the encoder also works without one. For a source view α, a target view $\beta \neq \alpha$ and a target device d of β, the predictor P_ξ, a six-block Transformer over the 50 patches, takes the source features of movement i averaged over the worn sites $\mathcal{S}_\alpha$, with a query added at every patch that sums an embedding e_d of the site and an MLP of the first two columns of the rotated estimate,

$$\hat{z}_i = P_\xi\Big(\frac{1}{|\mathcal{S}_\alpha|}\sum_{s\in\mathcal{S}_\alpha} E_\theta(X_i^\alpha)_s,\; e_d + \mathrm{MLP}\big(R_y(\eta)\,\hat{Q}_d\big)\Big), \tag{4}$$

where $E_\theta(X)_s$ is the token sequence of site s for the input X. The predicted sequence $\hat{z}_i$ must find its movement among the N candidates $z_j = \mathrm{sg}\big[E_\theta(X_j^\beta)_d\big]$, the target features under a stop-gradient sg, by InfoNCE [41],

$$\mathcal{L}_1 = -\frac{1}{|\mathcal{G}|\,N}\sum_{(\alpha,\beta,d)\in\mathcal{G}}\sum_{i=1}^{N}\log\frac{\exp(\rho_{ii}/\tau)}{\sum_{j=1}^{N}\exp(\rho_{ij}/\tau)}, \tag{5}$$

where ρ_{ij} is the cosine between $\hat{z}_i$ and z_j averaged over the patches valid in both, $\mathcal{G}$ holds the triples (α, β, d) with up to 3 target sites per view pair and $\tau = 0.1$. Stage 1 minimizes $\mathcal{L}_1 + \mathcal{L}_{\mathrm{cal}}$ over encoder, predictor and calibrator.

3.4. Captions and caption alignment

Stage 2 aligns the encoder with text, using Nymeria's narrations (160,392 lines on the training windows) and captions generated from the measured motion of each of these windows. To generate captions, a rule-based program measures the MHR skeleton in a pelvis-centered frame and produces a time-stamped digest of states and changes, from which an LLM (Qwen3.6-27B [31]) writes four captions per window, instructed not to invent objects, scenery or movements. Because the digest carries the motion but not its purpose, the generated captions describe kinematics rather than activities. For one window the generated caption reads "a person bends forward deeply, shifts weight backward and forward, and rotates the torso", the narration "picking up another decorative item on the table". In every step, a window is paired with one of its narrations, if it has any, with probability 0.5 and with one of its generated captions otherwise. The narrations-only and generated-only models of Tab. 2 set this probability to one and zero.

Stage 2: Caption alignment. The encoder then learns to match captions (Fig. 2, bottom middle). It embeds window i as v_i, while the CLIP ViT-B/32 text encoder [32], as in UniMTS but frozen, embeds its caption as u_i. With a learned logit scale γ and a batch of B windows, the symmetric contrastive loss of CLIP reads

$$\begin{aligned}\mathcal{L}_2 = -\frac{1}{2B}\sum_{i=1}^{B}\Big[&\log\frac{e^{\gamma\, v_i^\top u_i}}{\sum_j e^{\gamma\, v_i^\top u_j}}\\ &+\log\frac{e^{\gamma\, u_i^\top v_i}}{\sum_j e^{\gamma\, u_i^\top v_j}}\Big],\end{aligned} \tag{6}$$

and stage 2 minimizes $\mathcal{L}_2 + \lambda_A\mathcal{L}_A$, where the auxiliary loss $\mathcal{L}_A$, weighted by $\lambda_A = 0.1$, keeps the motion of every worn joint recoverable from the encoder's tokens as the mean of $\big(h_A(z_{s,p}) - y_{s,p}\big)^2$ over the entries of worn sites s and valid patches p, with gyroscope entries only where a gyroscope is worn. Here h_A is a linear head on the token $z_{s,p}$. The target $y_{s,p}$ holds the noise-free specific force and angular velocity of the patch at the joint center, aligned with the true gravity direction and divided by κ_f and κ_ω. Because the calibrator's estimates are imperfect, stage 2 trains under its errors.

After stage 1, an error bank stores them once as the rotations $\Delta = M_b^\top \hat{Q}_b^\top W_b$ from the true mounting M_b of a simulated device b at the same site to the mounting $\hat{Q}_b^\top W_b$ that the frozen calibrator implies for it, for 2,000 devices per site and gyroscope condition, each from six held-out Nymeria windows. The median error with a gyroscope is 4.1° for the estimated up axis and 19° for the reference channels, whose horizontal part also depends on the forward sign that gait leaves open. Stage 2 draws M_d uniformly from $SO(3)$, covering every mounting. For every window and site, one error is drawn uniformly from the bank entries of that site and gyroscope condition and applied to its mounting, so that the calibrator's estimate in Eq. (3) is replaced by

$$\hat{Q}_d = W_d \, (M_d \Delta)^\top, \qquad b_d(t) = \hat{Q}_d \, \hat{g}_d(t), \tag{7}$$

which equals the estimate $\hat{Q}_b$ the calibrator made for device b whenever $M_d = M_b$. Half of the stage-2 windows are accelerometer-only at all sites.

3.5. Inference with class names and anchors

At test, all windows of one participant at one site are treated as one device, assumed to share one mounting (Fig. 2, bottom right). From twelve unlabeled windows $\mathcal{K}$ of the device, the frozen calibrator repeatedly drops the window whose removal leaves the smallest tilt uncertainty, $\mathcal{K} \leftarrow \mathcal{K} \setminus \{\arg\min_{n \in \mathcal{K}} \sigma_\uparrow(\mathcal{K} \setminus \{n\})\}$, until six remain and then estimates $\hat{Q}_d$ from them (Tab. 1, last row). Every scored window of the device is encoded with $b_d(t) = \hat{Q}_d \, \hat{g}_d(t)$ (Eq. (3)). Zero-shot classification assigns each window to the class whose text embedding has the highest cosine similarity to v, where the class text, embedded by each model's own frozen text encoder, is either the class name as given in the dataset or an anchor.

An anchor replaces the class-name embedding with an average of text embeddings, as in CLIP's prompt ensembling [32] with descriptions [24, 29]. It uses the class name of the target dataset but no labeled target window. Let n_c be the embedding of the class name, r_c the mean embedding of the ten training narrations nearest to the name in all-mpnet space [35, 38] and g_{cj} the mean embedding of ten descriptions of a person performing the activity, written by an LLM in sampling round $j \in \{1, 2, 3\}$. With U the unit normalization, the anchor of class c weighs the name and its nearest narrations equally against the LLM-written descriptions,

$$a_c = U\Big(U\big(n_c + U(r_c)\big) + U\Big(\tfrac{1}{3}\sum_j U(g_{cj})\Big)\Big), \tag{8}$$

and each window is assigned the class $\hat{c} = \arg\max_c v^\top a_c$.

3.6. Language model on the encoder

For questions a class name cannot answer (Sec. 1), we also train a language model (LM) on top of the encoder. An MLP projector maps the features of the frozen SetCon encoder, computed with the calibrator's estimate of the setup reference, to one token for the worn-site flags and one token per worn site and time patch, which precede a Qwen3.5-0.8B [30] prompt that lists the worn sites. The projector is first trained to caption virtual windows with the LM frozen, then jointly with the LM on an equal mix of captions and question-answer pairs that an LLM writes from the narrations and digest of each window. Both stages minimize the next-token loss of the answer, with contrastive and margin terms that tie the answer to the IMU tokens.

3.7. Implementation details

The encoder has 6.4 M parameters and the calibrator 0.48 M. Every stage uses AdamW. Stage 1 trains for 20 epochs at constant learning rates of $3 \cdot 10^{-4}$ for encoder and predictor and $3 \cdot 10^{-5}$ for the calibrator. Stage 2 trains the encoder with its new projection and heads for 100 epochs on batches of $B = 128$ windows at a learning rate of $1 \cdot 10^{-4}$ with weight decay 0.05 and a linear warm-up over the first 3 epochs. We evaluate the exponential moving average of its weights (decay 0.999). This schedule is the same for every model, variant and seed. Training and evaluating one model takes at most 9.2 GPU-hours on one NVIDIA H100.

4. Experiments

4.1. Experimental protocol

SetCon is trained only on virtual IMU from Nymeria [21] (Sec. 3.1), with every recording of twelve participants held out for validation. It is tested on the 13 real datasets of the UniMTS suite [50] that record a gyroscope on every device. Mean-13 (Sec. 1) averages the accuracy on the scored windows of each dataset, with macro-F1 averaged in the same way. The baselines are UniMTS [50] and AnyMo [3] with their released weights. UniMTS uses the accelerometer of one window, while AnyMo uses accelerometer and gyroscope with the body segment in its prompt. For its reference, SetCon also uses unlabeled context windows of the same device, never scored. In the last row of Tab. 1 the calibrator chooses six of twelve such windows (Sec. 3.5) and every other result uses six random ones unless stated otherwise, averaged over five draws.

4.2. Main results

With six of twelve unlabeled windows per device as context, SetCon reaches 56.6 % Mean-13 accuracy with class names and 65.3 % with anchors (Tab. 1), whereas AnyMo, the stronger baseline, reaches 41.8 % and 46.7 % on the same windows, a relative gain of 35 % and 40 %. The lead holds in macro-F1 and with random context windows. With six random windows, SetCon reaches 55.9 % accuracy and 50.2 % macro-F1 with class names, compared with 41.8 %

Table 1. **Zero-shot classification.** Mean-13 accuracy and macro-F1 (%) with class names and anchors, plus the Mean-13 accuracy of the class-mean readout (Sec. 4.2). The calibrator estimates the reference from unlabeled context windows of each device, never scored. In the last row it chooses six of twelve context windows. The first three rows share their scored windows. On the last row's scored windows, UniMTS reaches 41.4 and 43.2 and AnyMo 41.8 and 46.7 with names and anchors. SetCon rows give mean ± sd over three seeds.

			Class-mean	Mean-13 acc.		Mean-13 macro-F1	
Method	Gyro.	Context	readout	names	anchors	names	anchors
UniMTS (released)	no	none	80.0	41.6	43.0	35.3	35.0
AnyMo (released)	yes	none	83.5	41.8	46.8	34.8	40.2
SetCon (ours)	yes	6 random windows	91.5 ± 0.2	55.9 ± 0.4	64.7 ± 0.6	50.2 ± 0.6	60.5 ± 0.5
SetCon (ours)	yes	6 of 12 windows	**91.6** ± 0.2	**56.6** ± 0.6	**65.3** ± 0.4	**51.6** ± 0.7	**62.1** ± 0.4

Table 2. **Ablations.** Mean-13 accuracy (%) with six random context windows (Tab. 1, third row). Mean ± sd over three seeds. Free answers are judged against the class names (Sec. 4.4).

	Mean-13	
Variant	names	anchors
Reference at test		
Calibrator (SetCon)	55.9 ± 0.4	**64.7** ± 0.6
Walking frame	54.6 ± 0.2	62.7 ± 0.6
Zeroed	41.5 ± 0.7	48.0 ± 2.1
Reference in training		
None	52.6 ± 0.4	59.0 ± 1.1
Walking frame	54.4 ± 0.7	63.2 ± 0.5
Training captions		
Narrations + generated (SetCon)	55.9 ± 0.4	**64.7** ± 0.6
Narrations only	58.1 ± 0.9	63.7 ± 1.7
Generated only	38.5 ± 2.0	48.5 ± 0.9
Language model (free answers)	**59.2** ± 1.5	–

Table 3. **Accuracy per motion group.** Mean class accuracy (%) with class names over Mean-13 classes, grouped by name. SetCon uses six random context windows (three seeds, mean ± sd).

Group (no. of classes)	UniMTS	AnyMo	SetCon (ours)
Posture (34)	33.2	43.4	**53.3**±0.8
Stairs, elevator (19)	18.3	25.1	**53.2**±1.8
Left/right, circle (6)	2.1	7.6	**18.8**±3.8
Locomotion (27)	**76.1**	63.8	64.0±4.7
Hand and arm (35)	**18.0**	14.6	17.8±1.1
Other whole-body (21)	47.2	37.8	**56.2**±2.5

and 34.8 % for AnyMo.

SetCon's embeddings also separate the activities best. A class-mean readout tests this without class names, which takes the wording of the classes out of the comparison. It averages the embeddings of the labeled windows of each class and assigns every window to the class with the nearest mean. With class means computed from all scored test windows, it reaches 91.5 % Mean-13 accuracy for SetCon, compared with 83.5 % for AnyMo and 80.0 % for UniMTS. SetCon is therefore ahead of both baselines even when no class name is used.

4.3. Analysis of the setup reference

SetCon relies on its setup reference (Tab. 2). Zeroing it at test lowers Mean-13 accuracy from 55.9 % to 41.5 % with class names and from 64.7 % to 48.0 % with anchors. A model trained without a reference lies 3.3 and 5.6 points below SetCon. One context window already restores most of the drop (53.7 % with names). The training-free Walking frame of Sec. 3.2 comes close to the calibrator, with 54.6 % using the same weights and 54.4 % when the encoder is also trained with it, so the encoder relies on the setup reference itself rather than on one estimator of it. The learned calibrator matters where the context windows contain no gait and the Walking frame leaves the reference at zero. On these devices (15 % of the scored windows), it outperforms the zeroed reference by 9.6 points with names.

Built for selective invariance, SetCon leads where full invariance fails (Tab. 3). With names it reaches a mean class accuracy of 53.3 % on postures against 43.4 % for AnyMo and 53.2 % on stairs and elevators against 25.1 %. Both groups depend on the direction of gravity, which full invariance removes together with the mounting. Trained on the same data without a reference, the encoder reaches only 45.6 % on postures, close to AnyMo and below SetCon, while no other group of Tab. 3 changes by more than 4.0 points. On UCI-HAR [1], its embeddings mix lying with sitting as AnyMo's do, while SetCon's keep them apart (Fig. 3). With the reference zeroed, SetCon fails like the baselines in Sec. 1 and classifies, with anchors, on average 86 % of the sitting, standing and lying windows of the same datasets as lying.

4.4. Language access

Training captions must describe activities. Aligned with generated captions alone, which describe kinematics rather than activities, the encoder reaches 38.5 % Mean-13 accu-

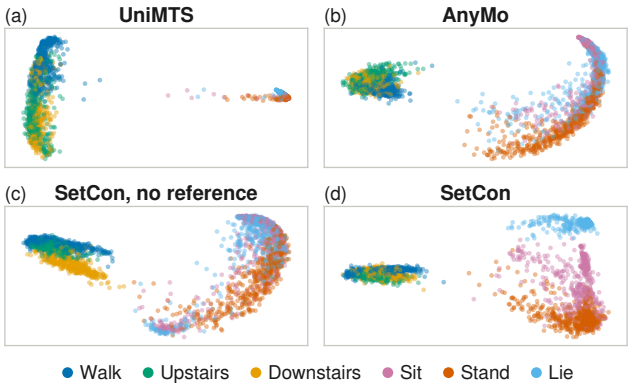


Figure 3. **Embeddings of UCI-HAR.** Each panel shows the first two principal components of one model's embeddings of the scored windows, with the first seed in (c) and (d). Lying mixes with sitting in (b) and (c) but forms its own cluster in (d).

racy with class names, 17.4 points below SetCon. Trained on narrations alone, it reaches 58.1 % with names, above SetCon's 55.9 %, but 63.7 % with anchors, below its 64.7 %. Generated captions trade a little class-name accuracy for a larger gain from anchors, which grows with their share from +5.6 points without them to +10.0 with them alone. The growth comes mainly from the descriptions half of the anchor in Eq. (8). The LLM-written descriptions, like the generated captions, say how the body moves, whereas the narrations in the other half describe activities, as class names do. Added to the class name, the descriptions bring +4.3, +6.3 and +8.4 points for the narrations-only model, SetCon and the generated-only model. Zero-shot classification thus works best with prompts that resemble the training text.

Class-name prompts recover only part of the separation in the embeddings. SetCon stays 36 points below the class-mean readout. Anchors, which use no labeled window, close part of the gap, adding 8.7 points for SetCon, compared with 4.9 for AnyMo and 1.4 for UniMTS (Tab. 1).

The language model of Sec. 3.6, trained on each of the three SetCon seeds, answers "What is the person doing?" in free text. An LLM judge [31] maps each answer to a class name of the dataset or to none. The language model reaches 59.2 % Mean-13 accuracy (Tab. 2), in every seed above the 55.9 % of the encoder itself with class-name prompts. Vision-language models, in contrast, classify images worse than their image encoder [51]. With anchors, however, the same encoder reaches 64.7 % and is more accurate in every seed. All three stay below the 91.5 % of the class-mean readout (Tab. 1). For a closed set of classes, cosine similarity to the anchors thus remains the most accurate and efficient zero-shot classifier, while the language model handles questions a class prompt cannot express. On held-out virtual Nymeria windows with candidate answers, it answers five-way questions on execution, order and body region correctly in 71, 64 and 46 % of cases (20 % chance) and selects the correct activity of four in 82 % (25 % chance). It also answers the five-way activity question written by the LLM correctly in 58 % of cases (20 % chance) and yes/no questions in 80 % (50 % chance). Its answers depend on the signal. On a subset of the windows of Tab. 2, they reach 5.6 % Mean-13 accuracy without the IMU tokens. Given the IMU tokens of another window, they give that window's class in 59 % of cases.

5. Conclusion

For IMU-language learning, the sensor setup is a source of information. An encoder invariant to how a device is worn loses the direction of gravity relative to the body, which separates postures. With selective invariance, SetCon retains it through the setup reference, which a calibrator estimates from unlabeled windows. Trained only on Nymeria, it outperforms UniMTS and AnyMo on average over 13 real datasets on which all three models receive their full input. Zero-shot accuracy also depends on the class prompt. Anchors bring +8.7 points for SetCon, compared with +5.6 for an encoder trained without generated captions. With anchors, the encoder is more accurate than a language model built on it (64.7 against 59.2 %). 3D human motion does more than provide training data. Real IMU datasets rarely record the mounting, so the calibrator's target comes from simulation. To learn the setup at scale, general-purpose IMU-language models therefore require 3D human motion.

Limitations. SetCon assumes that the body site of a device is known and that its mounting stays fixed while the device is worn. The 13 test sets are public benchmarks with known body sites. Unscripted daily life, in which devices slip and sites are unknown, remains to be tested.

Impact. With the best-separated embeddings in Tab. 1, SetCon can serve as a general-purpose IMU encoder for the phones, watches and glasses people already wear. Recognizing activities on any wearable without labels could lower the cost of health and rehabilitation monitoring. Beyond classification, its embeddings can support retrieval, captioning and question answering over long recordings. In egocentric multimodal models, it adds to vision and audio a signal that follows the body out of the camera's view. AI agents and robots could use SetCon to perceive what the people they cooperate with are doing. Estimating the setup may also help other body-worn sensors, such as cameras. SetCon learns from IMU simulated on 3D body motion, which motion capture and human mesh recovery from video both produce. Progress in 3D human motion therefore becomes training data for wearable sensing. Because SetCon reveals

what wearers do, deployment requires their informed consent.

Acknowledgments

This research was funded by the European Regional Development Fund (ERDF) within the project 'AIDA' (grant FEIH 2476184). L. Schiweck was supported by a scholarship from Bosch Sensortec GmbH. Views and opinions expressed are those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

The authors used writing and coding assistants (Claude by Anthropic, ChatGPT by OpenAI) to draft and edit text, captions, tables and figure scripts, to write and run experiment code and to search related work. The authors verified every number, claim and reference and take full responsibility for the content.

References

[1] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, Bruges, Belgium, 2013. 7

[2] Seungyeol Baek, Yoonbyung Chai, Yonghyeon Lee, Sungjoon Choi, and Sungho Suh. Rotation-invariant multi-IMU activity recognition under independent per-location orientation shifts. In *Proceedings of the 2026 ACM International Symposium on Wearable Computers (ISWC)*, pages 1–7, 2026. 3

[3] Baiyu Chen, Zechen Li, Wilson Wongso, Lihuan Li, Xiachong Lin, Hao Xue, Benjamin Tag, and Flora Salim. AnyMo: Geometry-aware setup-agnostic modeling of human motion in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026. arXiv:2605.22715. 1, 2, 3, 5, 6

[4] Daniel DeTone, Federica Bogo, Eric-Tuan Le, Duncan Frost, Julian Straub, Yawar Siddiqui, Yuting Ye, Jakob Engel, Richard Newcombe, and Lingni Ma. NymeriaPlus: Enriching Nymeria dataset with additional annotations and data. *arXiv preprint arXiv:2603.18496*, 2026. 2

[5] Zihan Ding, Liyu Zhang, and Xiaomin Ouyang. HALO: A heterogeneity-aware language-aligned IMU foundation model for open-set human activity recognition. *arXiv preprint arXiv:2608.27233*, 2026. 2

[6] Aaron Ferguson, Ahmed A. A. Osman, Berta Bescos, et al. MHR: Momentum Human Rig. *arXiv preprint arXiv:2511.15586*, 2025. 2, 3

[7] Anik Ghosh. Closing the modality gap in zero-shot HAR: Contrastive training and separability-optimized prototypes on IMU data. *arXiv preprint arXiv:2606.10789*, 2026. 3

[8] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023. 1, 2

[9] Fangzhou Hong, Vladimir Guzov, Hyo Jin Kim, Yuting Ye, Richard Newcombe, Ziwei Liu, and Lingni Ma. EgoLM: Multi-modal language model of egocentric motions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5344–5354, 2025. 3

[10] 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*, 37(6):1–15, 2018. 2

[11] Sijie Ji, Xinzhe Zheng, and Chenshu Wu. HARGPT: Are LLMs zero-shot human activity recognizers? In *IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things (FMSys)*, pages 38–43, 2024. 3

[12] Kai Kunze and Paul Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp)*, pages 20–29, 2008. 1, 3

[13] Kai Kunze and Paul Lukowicz. Sensor placement variations in wearable activity recognition. *IEEE Pervasive Computing*, 13(4):32–41, 2014. 1, 3

[14] Kai Kunze, Paul Lukowicz, Kurt Partridge, and Bo Begole. Which way am I facing: Inferring horizontal device orientation from an accelerometer signal. In *International Symposium on Wearable Computers (ISWC)*, pages 149–150, 2009. 2, 3, 4

[15] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. IMUTube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–29, 2020. 1, 2

[16] Zikang Leng, Hyeokhyen Kwon, and Thomas Plötz. Generating virtual on-body accelerometer data from virtual textual descriptions for human activity recognition. In *Proceedings of the 2023 ACM International Symposium on Wearable Computers (ISWC)*, pages 39–43, 2023. 2

[17] Zikang Leng, Amitrajit Bhattacharjee, Hrudhai Rajasekhar, Lizhe Zhang, Elizabeth Bruda, Hyeokhyen Kwon, and Thomas Plötz. IMUGPT 2.0: Language-based cross modality transfer for sensor-based human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(3):1–32, 2024. 1, 2

[18] Zechen Li, Shohreh Deldari, Linyao Chen, Hao Xue, and Flora D. Salim. SensorLLM: Aligning large language models with motion sensors for human activity recognition. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 354–379, 2025. 3

[19] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information*

Processing Systems (NeurIPS), pages 34892–34916, 2023. 2, 3

[20] 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): 1–16, 2015. 2

[21] Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David Soriano Fosas, C. Karen Liu, Ziwei Liu, Jakob Engel, Renzo De Nardi, and Richard Newcombe. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 445–465. Springer, 2024. 1, 2, 3, 6

[22] 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 (ICCV)*, pages 5441–5450, 2019. 2

[23] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5): 1203–1218, 2008. 2, 3, 4

[24] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *International Conference on Learning Representations (ICLR)*, 2023. 2, 3, 6

[25] D. Mizell. Using gravity to estimate accelerometer orientation. In *Seventh IEEE International Symposium on Wearable Computers (ISWC)*, pages 252–253, 2003. 2, 3, 4

[26] Vimal Mollyn, Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. IMUPoser: Full-body pose estimation using IMUs in phones, watches, and earbuds. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023. 2

[27] Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Aparajita Saraf, Amy Bearman, and Babak Damavandi. IMU2CLIP: Language-grounded motion sensor translation with multimodal contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13246–13253, 2023. 1, 2

[28] Nobuyuki Oishi, Phil Birch, Daniel Roggen, and Paula Lago. WIMUSim: Simulating realistic variabilities in wearable IMUs for human activity recognition. *Frontiers in Computer Science*, 7:1514933, 2025. 2

[29] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? Generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15691–15701, 2023. 2, 3, 6

[30] Qwen Team. Qwen3.5-0.8B. https://huggingface.co/Qwen/Qwen3.5-0.8B, 2026. Model card, accessed 22 October 2026. 6

[31] Qwen Team. Qwen3.6-27B. https://huggingface.co/Qwen/Qwen3.6-27B, 2026. Model card, accessed 22 October 2026. 5, 8

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 1, 3, 5, 6

[33] Lala Shakti Swarup Ray, Bo Zhou, and Paul Lukowicz. W2W: A simulated exploration of IMU placement across the human body for designing smarter wearable. In *Proceedings of the 2025 ACM International Symposium on Wearable Computers (ISWC)*, pages 170–176, 2025. 2

[34] Ranak Roy Chowdhury, Ritvik Kapila, Ameya Panse, Xiyuan Zhang, Diyan Teng, Rashmi Kulkarni, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. ZeroHAR: Sensor context augments zero-shot wearable action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16046–16054, 2025. 2

[35] Sentence-Transformers. all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2, 2021. Model card, accessed 22 October 2026. 6

[36] Ziwei Shan, Yaoyu He, Chengfeng Zhao, Jiashen Du, Jingyan Zhang, Qixuan Zhang, Jingyi Yu, and Lan Xu. Mojito: LLM-aided motion instructor with jitter-reduced inertial tokens. *arXiv preprint arXiv:2502.16175*, 2025. 3

[37] Sheikh Asif Imran Shouborno, Mohammad Nur Hossain Khan, Subrata Biswas, and Bashima Islam. LLaSA: A sensor-aware LLM for natural language reasoning of human activity from IMU data. In *Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion)*, pages 893–899, 2025. 3

[38] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 16857–16867, 2020. 6

[39] Jie Su, Fengtong Ge, Zhenyu Wen, Taotao Li, Yang Bai, Yejian Zhou, and Xiaoqin Zhang. IMUZero: Zero-shot human activity recognition by language-based cross modality fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(4):1–28, 2025. 2

[40] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. 3

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5

[42] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021. 3

[43] Jian Wang, Rishabh Dabral, Diogo Luvizon, Zhe Cao, Lingjie Liu, Thabo Beeler, and Christian Theobalt. Ego4o: Egocentric human motion capture and understanding from multi-modal input. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22668–22679, 2025. 3

[44] Vasco Xu, Chenfeng Gao, Henry Hoffmann, and Karan Ahuja. MobilePoser: Real-time full-body pose estimation and 3D human translation from IMUs in mobile consumer devices. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2024. 2

[45] Hua Yan, Heng Tan, Yi Ding, Pengfei Zhou, Vinod Namboodiri, and Yu Yang. Large language model-guided semantic alignment for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(4):1–25, 2025. 2

[46] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 5

[47] Xinyu Yi, Yuxiao Zhou, and Feng Xu. TransPose: Real-time 3D human translation and pose estimation with six inertial sensors. *ACM Transactions on Graphics*, 40(4):1–13, 2021. 2

[48] 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 (CVPR)*, pages 13167–13178, 2022. 2

[49] Aras Yurtman and Billur Barshan. Activity recognition invariant to sensor orientation with wearable motion sensors. *Sensors*, 17(8):1838, 2017. 1, 3

[50] Xiyuan Zhang, Diyan Teng, Ranak Roy Chowdhury, Shuheng Li, Dezhi Hong, Rajesh K. Gupta, and Jingbo Shang. UniMTS: Unified pre-training for motion time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 107469–107493, 2024. 1, 2, 3, 6

[51] Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 51727–51753, 2024. 3, 8

[52] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22520–22529, 2025. 3

[53] Haozhe Zhou, Riku Arakawa, Yuvraj Agarwal, and Mayank Goel. IMUCoCo: Enabling flexible on-body IMU placement for human pose estimation and activity recognition. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2025. 2

[54] Chengxu Zuo, Jiawei Huang, Xiao Jiang, Yuan Yao, Xiangren Shi, Rui Cao, Xinyu Yi, Feng Xu, Shihui Guo, and Yipeng Qin. Transformer IMU calibrator: Dynamic on-body IMU calibration for inertial motion capture. *ACM Transactions on Graphics*, 44(4):1–14, 2025. 2, 3