VARIABLE INPUT MASKED AUTOENCODERS FOR WRIST-WORN ACCELEROMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Large-scale repositories of wearable sensor data have enabled the development of foundation models for human activities and behavior. However, existing models are pretrained using fixed input durations and sampling rates, despite substantial heterogeneity in recording conditions, downstream tasks, and deployment constraints. We introduce `VarMAE`, a variable input masked autoencoder that learns a single representation across sampling rates and temporal contexts. During pretraining we construct two views of the same signal: a long duration high frequency view, and a short duration, lower frequency view, obtained via random cropping and downsampling. We perform masked reconstruction within each view and introduce a cross-resolution masked reconstruction objective that uses visible tokens from the long view to predict masked tokens from the short view. Cropping and downsampling impose complementary information bottlenecks, encouraging the model to learn representations that remain effective across input durations and sampling rate changes. We pretrain `VarMAE` on the large-scale NHANES dataset, producing a single pretrained encoder matching or outperforming controlled foundation model baselines on eight benchmark datasets. With only 0.7M parameters, `VarMAE` is parameter efficient and shows that a single foundation model can achieve broad transfer capabilities across sensing configurations.

1 INTRODUCTION

The popularity and widespread use of wearables has led to transformative progress across a wide gamut of health applications, including sleep analysis (Yuan et al., 2024b), monitoring Parkinson's disease (Del Din et al., 2016), fall detection Chatzaki et al. (2016), and predicting flu Merrill et al. (2023). Their ubiquitous nature allows biosignals to be collected in everyday settings, leading to vast amounts of sensor data (Schuetz et al., 2026). This in turn has motivated foundation model (FM) development, as unlabeled physiological data can be collected at scale whereas obtaining high quality annotations is both expensive and time consuming (Haresamudram et al., 2025).

Among wearable sensing modalities, accelerometry is widely used to capture human movements and underpins applications ranging from gesture recognition (Bonazzi et al., 2026) to analysis of daily activities and mobility (Yang & Hsu, 2010). However, accelerometer FMs face a fundamental challenge: *there is no single gold-standard input configuration in wearable sensing.* Downstream applications differ substantially in both the temporal context they require and the sampling rate at which the signal is sampled. For example, the OpenWatch gesture benchmark (Bonazzi et al., 2026) utilizes one second windows recorded at 100 Hz to recognize short gestures such as pinches and slides. In contrast, Yuan et al. (2024a) utilize ten second windows at 30 Hz to model daily living activities in naturalistic conditions. These differences stem from factors such as the timescale of the activity being observed, practical constraints such as the hardware platform, and other deployment requirements.

Existing accelerometer FMs largely sidestep this heterogeneity. As noted by Tarale et al. (2026), the window duration and sampling rate are typically fixed during pretraining and downstream evaluation occurs in similar input configurations. While there are adaptations like resampling or padding that can be applied, they undermine a key promise of foundation models: a *single pretrained model* that transfers broadly across many applications. It remains unclear how well representations learned under one input configuration transfer to others.

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.

Our approach, VarMAE, addresses this limitation through a variable input masked autoencoder (MAE) trained using two views of the same sensor data: *(i)* a long duration, high frequency view that captures the full context; and *(ii)* a short duration, lower frequency view which is randomly cropped from the long view, allowing the model to adapt to different frequencies. Both views are masked, and the masked regions in the short view are also masked in the long view to prevent information leakage about the masked target. In addition to within-view masked reconstruction, we introduce *cross-resolution masked reconstruction*, where the visible regions from the long view condition the prediction of the masked tokens of the short view.

Our key idea is to employ *cropping and downsampling as information bottlenecks.* On one hand, downsampling removes fine-grained temporal information while preserving the underlying signal; on the other, cropping removes the broader context surrounding the movements. Cross-resolution masked reconstruction bridges these bottlenecks by using information from the long context, high rate view to predict masked tokens at the low rate cropped view. This setup *exposes a single encoder to a range of sampling rates and window durations* during pre-training, enabling it to operate across a range of sensing configurations rather than specializing to a single one.

Architecturally, VarMAE retains the standard MAE's Transformer encoder-decoder, but extends it with rate flexibility in both input and output. We define patches over a fixed duration of time, causing the number of samples within each patch to vary based on the sampling rate. For example, a patch duration of 0.2s is 10 samples of sensor data at 50Hz but two samples at 10Hz. To accommodate this, a *variable-rate tokenizer* maps these rate dependent patches to a single fixed dimension token vector before the Transformer encoder, and correspondingly, a *variable-rate reconstruction head* reconstructs the decoder's outputs at the target sampling rate.

VarMAE is pretrained NHANES Centers for Disease Control and Prevention (CDC) (2010); Troiano (2008), which is a publicly available large-scale wrist accelerometer dataset comprising approx. one week of data from around $16k$ subjects. We evaluate our *single pretrained encoder* on eight HAR datasets covering a wide range of activities, subject counts, recording conditions, and input configurations. Despite using only 0.7M params, VarMAE outperforms general purpose time-series FMs that are 30-550x larger. Scaling our method to 3.82M params further improves performance, and the single encoder matches or outperforms FMs specialized to individual input configurations on all datasets. These results demonstrate that a single accelerometer FM can effectively transfer across substantially different downstream sensing configurations.

Our contributions are threefold:

(1) We introduce a variable input temporal masked autoencoder that enables a single Transformer encoder-decoder to operate across different sampling rates and window sizes, rather than specializing to a fixed sensing configuration.
(2) We propose *cross-resolution masked reconstruction*, to jointly learn across temporal context and resolution by using a longer, high rate view to reconstruct masked information in a shorter, lower-rate view. This improves representation quality, particularly at low sampling rates.
(3) Using a *single pretrained encoder*, we demonstrate broad and parameter-efficient transfer across across eight benchmarks spanning 1-10 second window sizes.

2 Related Work

In wearable computing, the development of FMs has received substantial interest in recent years, enabled by the availability of large-scale datasets such as UK Biobank (Doherty et al., 2017), NHANES (Troiano, 2008), and OpenMHC (Schuetz et al., 2026). The goal is to develop generic models that can support a wide range of downstream applications, e.g., gait analysis (Kavanagh & Menz, 2008; Tao et al., 2012), fall detection (Bagala et al., 2012), and activity recognition (Ordóñez & Roggen, 2016). Moreover, SSL has been shown to reduce reliance on annotated data (Haresamudram et al., 2022), and produce representations that are robust to class imbalance (Liu et al., 2021).

For accelerometry, the early SSL approach involved augmentation prediction (Saeed et al., 2019). Yuan et al. (2024a) later scaled this approach to 700k person days of data from UK Biobank, and the resulting representations were used for tasks like activity recognition (Yuan et al., 2024a), analysis of the association between sleep and mortality (Yuan et al., 2024b), and animal behavior recognition (Hoffman et al., 2024). More generally, two dominant paradigms emerged: reconstruction and

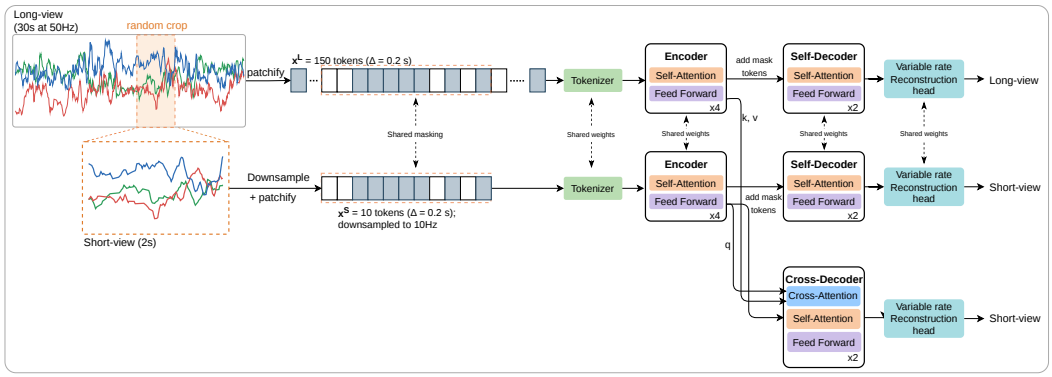


Figure 1: VarMAE is a variable input MAE trained using two views: first, a long duration, high frequency view, and second, lower frequency, short duration view. In addition to within-view masked reconstruction, we also perform cross-resolution masked reconstruction.

contrastive learning. Contrastive methods learn by training to reduce the distance between positive pairs while increasing between the negative pairs. Such pairs have been constructed using specialized augmentations (Chen et al., 2020; Tang et al., 2020), time-frequency consistency (Zhang et al., 2022), future timestep prediction (Haresamudram et al., 2021), and learned similarity functions (Xu et al., 2024; 2025a), among other strategies. Reconstruction-based methods recover either the entire input (Haresamudram et al., 2019) or intentionally masked portions of it, as in MAEs (He et al., 2022). This setup has recently shown great effectiveness when scaled to large datasets and models (Narayanswamy et al., 2025; Xu et al., 2025b; Tarale et al., 2026).

A limitation shared by most existing SSL approaches is that each pretrained model fixes the sensing configuration, i.e., the window size and sampling rate, throughout training and evaluation. For example, Yuan et al. (2024a) downsample UK-Biobank data to 30 Hz and pre-train/evaluate using either five- or ten-second windows. Similarly, Xu et al. (2025a) pre-train separate models for 2.56s and 2s windows in separate experiments. The input configs used by prior FMs are summarized in the Appendix (Table 4). Therefore, existing approaches largely address the differences in sensing configurations through configuration-specific pretraining. In contrast, we train a *single accelerometer FM* across sampling rates and window sizes. Our model operates directly on heterogeneous configurations, without requiring any resampling or setting specific pretraining.

3 Methodology

Our goal is to *pre-train a single encoder* that can reliably transfer across a variety of window durations and sampling rates. To achieve this, VarMAE has two key components: *(i)* a variable-rate masked autoencoder that processes accelerometer signals at different sampling rates and window durations; and *(ii)* a cross-resolution masked reconstruction objective where a short, downsampled view is reconstructed using context from a longer, higher frequency view of the same recording.

3.1 Variable-Rate Masked Autoencoder

Standard MAEs assume a fixed number of samples per patch, effectively tying the patch embedding to a particular sampling rate. Instead, we define patches *in physical time* (i.e., in *seconds*): each patch spans $\Delta = 0.2$s, and therefore contains $L_r = r\Delta$ samples, where r is the sampling rate. Consequently, a 0.2s patch contains 2 samples at 10 Hz and 10 samples at 50 Hz. Across sampling rates, fixing the patch duration in seconds enables consistency in the quantity of movement contained within a patch. Accelerometer data of window duration d is split into $N = d/\Delta$ non-overlapping patches, and each patch j covers $[j\Delta, (j+1)\Delta)$.

At a high-level, VarMAE follows the standard MAE pipeline, but makes it input flexible. A *variable-rate tokenizer* first maps the L_r samples in each visible patch to a single fixed size token vector. A *Transformer encoder* processes these tokens, and the temporal position is represented in physical

seconds rather than indices. The masked tokens are added back along with their positional embeddings, and a lightweight *decoder* reconstructs the masked tokens through a *variable-rate reconstructor head*, that produces L_r samples at the required rate r. In this way, the encoder-decoder remain identical across sampling rates as the input and output changes are handled by the tokenizer and reconstructor. Following Nie et al. (2022), we process *each accelerometer channel independently*.

1) Variable-Rate Tokenizer: As L_r varies with sampling rate, the conventional linear patch projection cannot be applied. Instead, we utilize a cross-attention tokenizer, similar to the Q-Former (Li et al., 2023). Each sample of accelerometer data is first projected to $d = 128$ and a single learned query vector then attends over L_r embedded samples within the patch $\mathbf{E} = (\mathbf{e}1, \ldots, \mathbf{e}L_r)$ using 4 attention heads, to produce a single patch token. The learned query is input as the query whereas the key-value are the linear projections of the sample embeddings in the cross-attention setup.

The query vector is conditioned on the number of samples L_r through an embedding of $\log L_r$ (a sinusoidal encoding followed by an MLP), thereby explicitly informing the tokenizer about the sampling rate. In lieu of fixed sinusoidal encodings based on sample indices, the position of each sample within the patch is described as an offset in seconds: the sample i sits at offset $\delta_i = (i-1)/r$ from the start of the patch. At 50 Hz, a 10-sample patch has offsets $[0, 0.02, 0.04, \ldots, 0.18]$ s; at 10 Hz, a 2-sample patch has offsets $[0, 0.1]$ s. Rotary Positional Embeddings (RoPE) are applied to the query and keys using these offset positions, and the learned query is positioned at the mean sample offset. No tokenizer parameters depend on L_r, and it can process patches at different rates. The tokenizer uses four attention heads, followed by a dense layer of size 256.

2) Encoder-Decoder: The encoder (f_{enc}) is a four layer pre-LayerNorm (LN) Transformer with dim=128, four attention heads, and feedforward dim=256, followed by a LayerNorm (LN). As in MAE (He et al., 2022), it only processes visible tokens. However, instead of indices, the tokens are positioned at the center t_j of its patch (0.2s), which is independent of the sampling rate.

We first project the encoder outputs from 128→64 dim, and inserts the masked tokens back at their original positions, to once again obtain N patches. Then, the self-decoder (g_{self}), a 2-layer pre-LN Transformer with dim=64, two attention heads, and feedforward dim=128 processes the N patches. The outputs at only the masked locations $\mathbf{h}_j$ are passed to the variable rate reconstruction head.

Position Encoding: All attention layers utilize rotary position embeddings (RoPE) (Su et al., 2024). The positions are specified in *seconds* rather than token indices. Hence, samples separated by one second have the same positional encoding regardless of the sampling rate (and as a result, the number of samples in between). For attention across patches, we use rotary periods spanning 0.04s to 120s, whereas within patch the maximum period is 1s. This is done following Scale-MAE (Reed et al., 2023) which similarly conditions position encoding relative to the physical scale of the input. In addition, we employ QK-norm throughout (Henry et al., 2020) as it has been shown to improve training stability (Dehghani et al., 2023). As RoPE depends solely on relative position, the encoder is invariant to a global shift in timestamps. This property enables us to separately encode the long and short views and subsequently align the embeddings for the cross resolution masked reconstruction.

3) Variable-Rate Reconstruction: As a consequence of the variable rate input, the number of samples to be reconstructed also varies with the rate r (0.2s at 10Hz is 2 samples but at 50Hz it is 10 samples). As a result, we replace the standard MAE's fixed reconstruction head with a small MLP conditioned on the patch's temporal offset (based on (Sitzmann et al., 2020)). For each masked patch, the transformer decoder produces a single representation $\mathbf{h}_j$, from which the MLP predicts the signal value for each temporal offset value in the patch. The predictions are conditioned on both the sampling rate and a sinusoidal encoding of the offset. Therefore, the MLP produces $r\Delta$ reconstructions at sampling rate r, allowing the same decoder to reconstruct signals at any sampling rate. The variable rate decoders are shared for the masked reconstruction of each view, and a separate decoder is instantiated for the cross-resolution masked reconstruction.

3.2 Cross-Resolution Masked Reconstruction

The variable-rate MAE allows the same architecture to process different sampling rates. We exploit this capability during pre-training by constructing two views of the same accelerometer recording: (i) a long duration, high rate view $\mathbf{x}^{\mathrm{L}}$, and a shorter, potentially downsampled view $\mathbf{x}^{\mathrm{S}}$. Both views

and encoded independently by the same encoder, with $\mathbf{x}^{\mathrm{L}}$ providing additional temporal context through the cross-resolution masked reconstruction objective.

1) Long and Short Views: $\mathbf{x}^{\mathrm{L}}$ has a fixed duration and sampling rate $(d_{\mathrm{L}}, r_{\mathrm{L}}) = (30\mathrm{s}, 50\mathrm{Hz})$, producing $N_{\mathrm{L}} = 150$ patches. For each mini-batch, we randomly sample a short view duration $d_{\mathrm{S}} \in \{1, 2, 5, 10\}\mathrm{s}$ and sampling rate $r_{\mathrm{S}} \in \{10, 20, 30, 40, 50\}\mathrm{Hz}$. The entire mini-batch uses the same $(d_{\mathrm{S}}, r_{\mathrm{S}})$. In $\mathbf{x}^{\mathrm{L}}$, we sample a start time s on the patch grid and extract the d_{S}-second crop, and then resample to r_{S} using polyphase resampling with an anti-aliasing filter to obtain $\mathbf{x}^{\mathrm{S}}$. Here, cropping and downsampling create complementary information bottlenecks. While cropping removes the surrounding temporal context, downsampling removes fine-grained temporal information within $\mathbf{x}^{\mathrm{S}}$.

2) Masking and Co-Masking: We first mask $\mathbf{x}^{\mathrm{S}}$ with a masking ratio of 0.75 using contiguous spans (Baevski et al., 2020) of 5 tokens. As $\mathbf{x}^{\mathrm{S}}$ is randomly cropped from $\mathbf{x}^{\mathrm{L}}$, the two views cannot be masked independently. A region masked in $\mathbf{x}^{\mathrm{S}}$, if visible in $\mathbf{x}^{\mathrm{L}}$ would allow the cross-decoder (g_{cross}) to directly observe the reconstruction target. Consequently, we perform masking on $\mathbf{x}^{\mathrm{S}}$ first, and match the masked locations in $\mathbf{x}^{\mathrm{L}}$ as well. Every patch masked in $\mathbf{x}^{\mathrm{S}}$ is also masked at the corresponding location in $\mathbf{x}^{\mathrm{L}}$, and the patches visible in $\mathbf{x}^{\mathrm{S}}$ remain visible in $\mathbf{x}^{\mathrm{L}}$. Additional patches in $\mathbf{x}^{\mathrm{L}}$ required to reach the 50% masking ratio are sampled uniformly from outside the short-view interval. This strategy of *co-masking* removes the direct reconstruction shortcut while retaining the information advantage of $\mathbf{x}^{\mathrm{L}}$ due to the surrounding context.

3) Cross-Resolution Decoder: The cross-decoder (g_{cross}) reconstructs masked patches of $\mathbf{x}^{\mathrm{S}}$. Following Gupta et al. (2023), the visible tokens from $\mathbf{x}^{\mathrm{L}}$ are used in the cross-attention setup as the key and value, whereas all tokens from $\mathbf{x}^{\mathrm{S}}$ (visible and masked) form the query. The cross-decoder comprises two Transformer layers, with cross attention using $\mathbf{x}^{\mathrm{L}}$ and self-attention over $\mathbf{x}^{\mathrm{S}}$, and a feedforward block. The reconstructed outputs at the masked locations of $\mathbf{x}^{\mathrm{S}}$ are converted back to raw signal values using a variable-rate reconstruction head. Due to the co-masking, the cross-decoder must use the visible tokens of $\mathbf{x}^{\mathrm{S}}$ and the additional context from $\mathbf{x}^{\mathrm{L}}$ to reconstruct the masked tokens.

4) Training Objective: We denote the visible tokens by $\mathcal{V}$, the masked patches by $\mathcal{M}$, and the encoder output by $\mathbf{Z} = f_\theta(\mathbf{x}\mathcal{V})$, where $\mathbf{x}$ is the input signal. Let $\mathbf{z}^S = f_{\mathrm{enc}}(\mathbf{x}^S_{\mathcal{V}^S})$ and $\mathbf{z}^L = f_{\mathrm{enc}}(\mathbf{x}^L_{\mathcal{V}^L})$ be the shared encoder's outputs on the visible patches of each view, with $\mathcal{M}^S \subseteq \mathcal{M}^L$ on the long view's time axis. P^S and P^L are the samples per patch in each view. The overall training loss is the sum of the three masked reconstruction losses, given below:

$$\mathcal{L}_{\mathrm{Short}} = \frac{1}{|\mathcal{M}^S|P^S} \sum_{t \in \mathcal{M}^S} \left\| g_{\mathrm{self}}(\mathbf{z}^S)_t - \mathbf{x}^S_t \right\|_2^2 \tag{1}$$

$$\mathcal{L}_{\mathrm{Long}} = \frac{1}{|\mathcal{M}^L|P^L} \sum_{t \in \mathcal{M}^L} \left\| g_{\mathrm{self}}(\mathbf{z}^L)_t - \mathbf{x}^L_t \right\|_2^2 \tag{2}$$

$$\mathcal{L}_{\mathrm{Cross}} = \frac{1}{|\mathcal{M}^S|P^S} \sum_{t \in \mathcal{M}^S} \left\| g_{\mathrm{cross}}(\mathbf{z}^S, \mathbf{z}^L)_t - \mathbf{x}^S_t \right\|_2^2 \tag{3}$$

$$\mathcal{L} = \mathcal{L}_{\mathrm{Cross}} + \mathcal{L}_{\mathrm{Short}} + \mathcal{L}_{\mathrm{Long}} \tag{4}$$

The within-view losses $\mathcal{L}_{\mathrm{Short}}$ and $\mathcal{L}_{\mathrm{Long}}$ reconstruct the masked samples of the short $\mathbf{x}^{\mathrm{S}}$ and long views $\mathbf{x}^{\mathrm{L}}$, respectively. $\mathcal{L}_{\mathrm{Short}}$ reconstructs the short view using only its own visible tokens, matching the downstream setting where no surrounding context is available. The cross-resolution loss $\mathcal{L}_{cross}$ also reconstructs the masked tokens of $\mathbf{x}^{\mathrm{S}}$, but conditions the predictions on the *visible tokens from the long view* $\mathbf{x}^{\mathrm{L}}$, through the cross-decoder.

4 Experimental Settings

Pre-training Data: We utilize data from the NHANES study Centers for Disease Control and Prevention (CDC) (2010) for pre-training, similar to Tarale et al. (2026). Specifically, we use all subjects from NHANES 2011-2012, NHANES 2013-2014, and NNYFS studies, leading to a total of $\sim$16k users, with approximately one week of data per user. The data were recorded at 80Hz and downsampled to 50Hz for pretraining.

Evaluation Data: We evaluate VarMAE on *eight HAR datasets*, spanning fine-grained gestures, locomotion-style activities, and activities of daily living. The datasets vary substantially in sensor location, number of participants, the number/types of classes, providing an effective testbed for evaluating transfer performance. They include OpenWatch (Bonazzi et al., 2026) and WatchHand (Kim et al., 2026), Mobiact (Chatzaki et al., 2016) and Motionsense (Malekzadeh et al., 2021), RealWorld (Sztyler & Stuckenschmidt, 2016) and HARTH (Logacjov et al., 2021), and Capture-24 (Chan et al., 2024) and PAAWS-SimFL (Potter et al., 2025). Full details in the Appendix (Section A).

4.1 Comparative Baselines

We consider two evaluation settings for the baselines: *(i)* controlled, where we pre-train each methods on NHANES using our setup, thereby providing a direct comparison; and *(ii)* public checkpoint, where we utilize publicly available pre-trained models to perform activity recognition on our downstream evaluation datasets. In this setting, all relevant data and method pre-processing remain identical to the original authors' implementations wherever applicable.

For the controlled evaluation, we consider three representative SSL methods, with detailed information in the Appendix (Section B.1):

(1) **Augmentation prediction** (AugPred, (Saeed et al., 2019; Yuan et al., 2024a), where pre-training is performed via multi-task prediction of whether augmentations have been applied or not.
(2) **SimCLR** (Chen et al., 2020; Tang et al., 2020), which involves contrastive learning between randomly augmented views of the same input window.
(3) **Vanilla MAE** (He et al., 2022), which uses the encoder-decoder is identical to VarMAE. We use the same patch duration (0.2s) at 50Hz, yielding a patch size of 10 samples. In contrast to VarMAE, it uses linear projections for the patch embedding and for reconstruction. This provides the most direct comparison to VarMAE.

For the public checkpoint evaluation, we use the public weights released by Yuan et al. (2024a), which were pre-trained on 700k person days of data from the UK Biobank. We evaluate on both the 5s and 10s variants. Further, we also compare against BioPM, which is a recently released MAE pretrained on NHANES using 10s windows. Finally, we compare against general-purpose time-series FMs, including Chronos (Ansari et al., 2024) and MOMENT (Goswami et al., 2024), to assess the benefit of specifically pretraining on accelerometer data. We embed each channel of accelerometer data separately and perform mean pooling for use in downstream activity recognition.

Evaluation Setup In line with prior work (Saeed et al., 2019; Haresamudram et al., 2022; Xu et al., 2025a) we utilize subject-disjoint train/val/test splits. For downstream datasets, we first randomly sample 20% of users to form the test set. Of the remaining users, 20% are once again sampled for validation and the rest is used for training. We repeat this process five times such that each user appears in the test set exactly once. For all results, we report the average macro-F1 score of the test set across folds. We provide all pretraining and evaluation parameters in the Appendix (Section B).

Linear evaluation: After pretraining, we freeze the encoder and train a single linear layer to predict activities. The encoder functions purely as a feature extractor, allowing us to evaluate the quality of learned representations without confounding performance with classifier's capacity. For all linear evaluations, we use Adam (Loshchilov & Hutter, 2017) with lr=1e-3 and no weight decay.

5 Results

5.1 Activity Recognition Performance

Public Checkpoint Evaluation: In Table 1, we compare VarMAE against publicly released checkpoints from Harnet-Oxford (Yuan et al., 2024a) and Bio-Inspired FM (Tarale et al., 2026). Both methods were pretrained at 30Hz, we therefore resample all evaluation datasets to 30Hz before feature extraction. Scores marked with † indicate settings where the downstream window duration differs from that used during pretraining. Harnet-Oxford provides two checkpoints pretrained using 5s and 10s windows. We use the checkpoint whose pre-training window most closely matches the evaluation window. For datasets with window sizes shorter than 5s, e.g., Mobiact, we perform circular padding to 5s before feature extraction to match the receptive field of the 5s Harnet-Oxford

Table 1: **Activity recognition performance:** we contrast VarMAE against end-to-end training, general purpose TSFMs, as well as accelerometer-based FMs. VarMAE matches or outperforms other FMs on six out of eight datasets. **Bold** is best and underline second best per dataset. † indicates settings where the window duration differs from that used during pretraining.

Approach	Enc #Params	1 s		2 s		5 s		10 s	
		Open Watch	Watch Hand	Motion Sense	Mobi Act	HARTH	Real World	PAAWS-SimFL	Capture-24
End-to-end training									
Conv encoder	80k	0.5785	0.6846	0.8850	0.8125	0.7679	0.7420	0.7817	0.4177
ResNet encoder	2.61M	0.6398	0.6286	0.8775	0.8389	0.7933	0.7046	**0.8364**	0.4516
TSFMs + linear eval									
Chronos-T5-small	21M	0.1182	0.4467	0.8303	0.5891	0.5876	0.6417	0.5894	0.4074
Chronos-2	120M	0.4466	0.6053	0.8163	0.7069	0.6530	0.7482	0.6782	0.4321
MOMENT	385M	0.3261	0.5572	0.7984	0.6071	0.6207	0.7305	0.6334	0.3893
Public Checkpoints + linear eval									
Harnet-Oxford	10M	0.2030†	0.4991†	0.7494†	0.6426†	0.6412	0.7269	0.7820	**0.5137**
Bio-Inspired FM	1.4M	0.5133†	**0.7369**†	0.8411†	0.7†	0.7384†	0.7526†	0.6334	0.4039
Controlled Specialist FMs + linear eval									
AugPred	4.23M	0.2666	0.4703	0.8704	0.7241	0.6732	0.7576	0.6465	0.4366
SimCLR	4.23M	**0.6498**	0.5473	0.8774	0.7633	0.6831	0.7574	0.6296	0.4344
Vanilla MAE	0.53M	0.5912	0.6513	0.8614	0.8313	0.8260	0.8068	0.7615	0.4519
Ours									
Ours (small)	0.70M	0.6258	0.6675	0.8873	0.8441	0.8241	0.8179	0.7624	0.4450
Ours (medium)	3.82M	**0.6491**	0.6795	**0.9108**	0.8643	**0.8278**	**0.8299**	0.7932	0.4617
Ours (large)	10.94M	0.6457	0.6781	0.9082	**0.8664**	0.8250	0.8278	0.8002	0.4545

model. We observe that this high quantity of padding (i.e., from 1s/2s to 5s) produces low performance. The preprocessing for BioPM matches the original paper and requires no padding.

At 5s, VarMAE-Small substantially outperforms Harnet-Oxford (>8%), despite using a much smaller encoder (0.7M vs 10M params). On PAAWS-SimFL, Harnet-Oxford shows an increase over VarMAE-Small (0.782 vs 0.762), while the Medium and Large models surpass it, achieving 0.793 and 0.80 respectively. All three variants of VarMAE outperform Bio-Inspired FM on SimFL and Capture-24. Interestingly, it obtains the highest performance for WatchHand (73.69%), despite substantial difference in window sizes. This demonstrates that VarMAE remains competitive against pretrained FMs with much higher capacity, while scaling further improves performance.

Controlled Evaluation: As the released checkpoints provide matched comparisons only at the selected window durations, we additionally perform a controlled evaluation by pretraining specialist models for each of 1s, 2s, 5s, and 10s windows. For AugPred and SimCLR, we use a shallower variant of the Resnet architecture from Yuan et al. (2024a), and adapt its receptive field to match the corresponding input duration (again, similar to Yuan et al. (2024a)). The Vanilla MAE uses the same encoder-decoder and patch duration as VarMAE, making it our most direct comparison. VarMAE-Medium matches or *outperforms all controlled baselines across datasets*. Vanilla MAE is the strongest controlled baseline, highlighting the strength of masked reconstruction as a SSL pretext task, while also demonstrating the additional benefit of our flexible training approach.

General-Purpose Time-Series FMs and End-to-End Training: All three variants of VarMAE outperform the TSFMs indicating the substantial advantage of pretraining specifically using accelerometer data. Despite freezing the encoder and only training a linear classifier, VarMAE is highly competitive with end-to-end training, and VarMAE-Small outperforms both end-to-end models on four datasets, with the larger variants increasing it to five datasets.

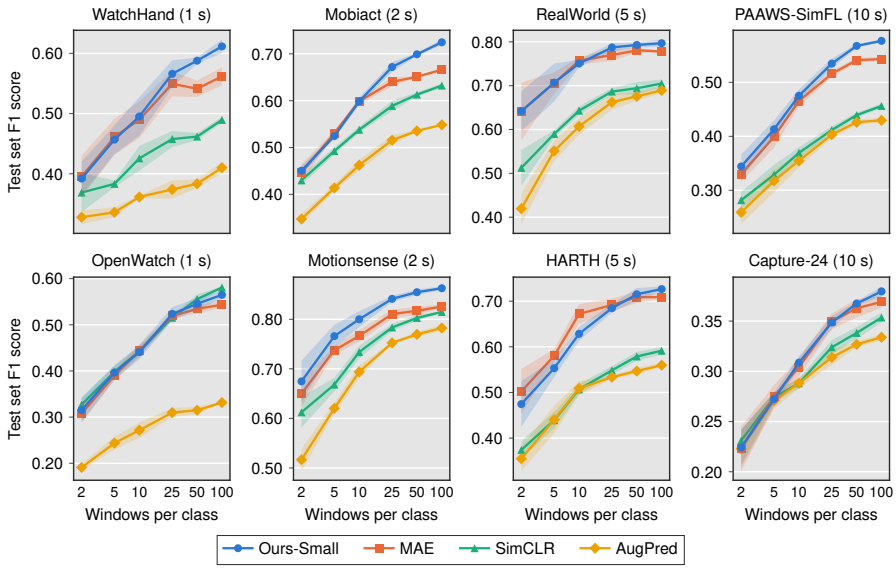


Figure 2: Few Shot Learning: During linear evaluation, we extract representations only for {2, 5, 10, 25, 50, 100} randomly sampled windows per activity, for five random seeds. Our approach matches or outperforms the specialized FM baselines.

5.2 Few-Shot Learning Evaluation

Annotating sensor data is a great challenge and is both costly and time consuming. Consequently, a key advantage of FMs is their ability to produce high performance when there is very limited access to labeled data, i.e., in the few-shot regime (Yuan et al., 2024a; Haresamudram et al., 2022). In Fig. 2 we contrast the few-shot learning performance of `VarMAE` against specialist FMs.

For each downstream dataset, we first randomly sample {2, 5, 10, 25, 50, 100} labeled windows per activity class from the train set, while validation and test sets remain untouched. We extract representations using the pre-trained FMs and train a linear classifier using only the labeled windows. As performance can vary substantially in the limited data regime, we repeat the sampling and linear evaluation with five random seeds and report the test set performance across seeds in Fig. 2.

Importantly, this setting is favorable to the specialist FMs as they are trained with the exact window duration and sampling rate required by the corresponding downstream datasets. In contrast, `VarMAE` uses *a single pre-trained model across datasets*. As expected, the performance improves rapidly with the availability of increasing amounts of annotated data. Overall, `VarMAE` is competitive with or outperforms all specialist FMs, despite using a single shared model. The vanilla MAE is the strongest specialist baseline and closest in performance in this few-shot regime. As more labeled data are available, `VarMAE` performs better, especially when the windows/activity >25. AugPred and SimCLR generally perform worse, often trailing by more than 10% F1-score.

5.3 Ablations

Contribution of the Losses: We ablate each component of the pretraining objective (Eq. (4)) and evaluate the impact on four HAR datasets. Beyond the native 50Hz, we resample the datasets to {5, 10, 25}Hz in Fig. 3 to assess the contribution of each component under changes in sampling rate. At 50Hz, removing any single loss has little effect on performance. The impact becomes substantially more pronounced as the sampling rate is progressively reduced.

The strongest effect is obtained with the removal of $\mathcal{L}_{\text{Short}}$. At 5Hz, it produces substantial degradation in performance across datasets, with performance drops exceeding 25% F1-score. This is consistent with $\mathcal{L}_{\text{short}}$ exposing the model to lower-rate signals during pretraining. Without this objective, representations learned using only the higher rate view transfer poorly to heavily downsampled inputs. The cross-resolution loss $\mathcal{L}_{\text{cross}}$ is also important, and its removal produces a consistent

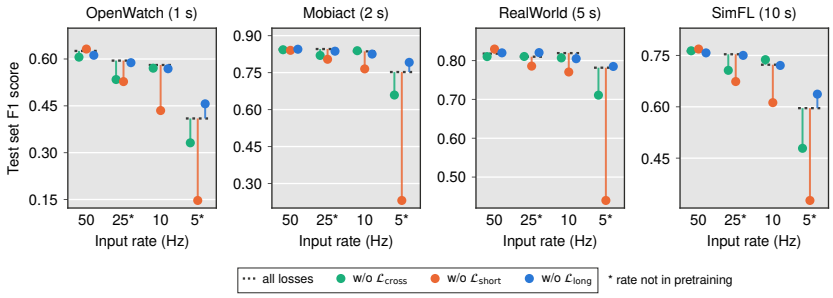


Figure 3: Impact of the pre-training losses: removing the short-view and cross-resolution losses has substantial impact at low frequencies.

Table 2: Varying the masking ratio during pre-training: we tabulate the mean test performance over four evaluation datasets, namely OpenWatch, Mobiact, RealWorld, and SimFL. The short view's masking ratio has substantial impact at low sampling rates. *Rate not used in pretraining.

Mask ratio						
Long view	Short view	50 Hz	25 Hz*	10 Hz	5 Hz*	Avg.
0.5	0.75	0.7730	0.7681	0.7564	0.6874	0.7462
0.25	0.75	0.7739 (+0.0009)	0.7707 (+0.0026)	0.7565 (+0.0001)	0.6709 (−0.0165)	0.7430 (−0.0032)
0.75	0.75	0.7703 (−0.0027)	0.7628 (−0.0053)	0.7503 (−0.0061)	**0.7166** (+0.0292)	0.7500 (+0.0038)
0.5	0.25	0.7668 (−0.0062)	0.7547 (−0.0134)	0.7466 (−0.0098)	0.5425 (−0.1449)	0.7027 (−0.0435)
0.5	0.5	**0.7751** (+0.0021)	**0.7718** (+0.0037)	**0.7616** (+0.0052)	0.4934 (−0.1940)	0.7005 (−0.0457)

reduction in performance. In contrast, $\mathcal{L}_{\text{Long}}$ only has a modest impact across sampling rates. As such, the utility of $\mathcal{L}_{\text{short}}$ and $\mathcal{L}_{\text{cross}}$ is primarily to retain performance as sampling rate decreases.

Masking ratios: The default configuration of `VarMAE` uses a 0.5 and 0.75 masking ratio for the long and short views, respectively. In Table 2, we vary the masking ratios and report the mean of five-fold test set performance across OpenWatch, Mobiact, RealWorld, and SimFL (full scores in the Appendix, Table 6). At 50, 25, and 10 Hz, the performance is largely insenstive to the masking ratio in both views. However, at the lowest rate (5 Hz), the masking ratio of the short-view has high impact. Reducing from the default 0.75 to 0.5/0.25 causes performance drop of 14.5% or higher. In comparison, changing the long view's masking ratio from the default 0.5 to 0.25 or 0.75 has a substantially smaller effect at 5Hz, yielding 0.6709 and 0.7166 respectively.

These results mirror the findings in Fig. 3, where the design choices associated with the short view have little effect at higher sampling rates but become more important upon downsampling. High masking ratios in the short-view encourage the models to retain their effectiveness even when less temporal resolution is available during evaluation.

6 Conclusion

We introduced `VarMAE`, a variable input MAE pretrained on large-scale accelerometry from ∼16k participants. `VarMAE` extends the standard MAE architecture with variable-rate tokenization and reconstruction, enabling the use of a shared Transformer encoder-decoder to operate across different sampling rates and window durations. During pretraining, we jointly optimized masked reconstruction within the long and short views, along with cross-resolution masked reconstruction, where the long-duration high frequency view provides context for reconstructing a shorter, downsampled view. We evaluated our approach on eight activity recognition datasets, covering a wide variety of activities, sensing locations, input window durations, and number of subjects. These results demonstrate that a single accelerometer representation can remain effective across substantially different sensing configurations, reducing the need for bespoke training at specific input configurations. Such flexibility is an important step towards the developement of more broadly usable FMs for wearable sensing.

AI USE STATEMENT

In this work, we used generative AI tools to help write code, and to polish grammar and fix typos in the paper. We did not use them to develop ideas, to design the method or experiments, or to interpret results. The authors reviewed all AI-assisted work and take full responsibility for the contents of the paper.

ETHICS STATEMENT

Our work aims to support personalized health applications through developing models on physiological signals. We recognize that the use of such data raises important ethical considerations, particularly regarding privacy and potential health disparities, as increasingly detailed physiological characterization may expose sensitive information or disproportionately benefit certain populations. The collection, storage, and use of physiological data should therefore be subject to strict governance to ensure appropriate data protection and preserve individuals' autonomy over their health information. In this study, all data are publicly available, and no personally identifiable information was used in our analyses. Overall, we believe that our work adheres to the ICLR Code of Ethics and contributes to more personalized health assessment, and, ultimately, improves the quality and safety of care.

REPRODUCIBILITY STATEMENT

To ensure complete reproducibility of our pre-training framework, model checkpoints and empirical findings, we will open-source all code, pretrained model weights, and downstream evaluation pipelines upon acceptance. Section 3 describes the model architecture and the training objective. Section 4 and the Appendix (Section B) document the pretraining and downstream datasets, preprocessing procedures, data splits, optimization and training hyperparameters, downstream evaluation protocol, and implementation details for all controlled baselines. We additionally summarize the downstream datasets in the Appendix A.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Fabio Bagala, Clemens Becker, Angelo Cappello, Lorenzo Chiari, Kamiar Aminian, Jeffrey M Hausdorff, Wiebren Zijlstra, and Jochen Klenk. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PloS one*, 7(5):e37062, 2012.

Pietro Bonazzi, Youssef Ahmed, Daniel Eckert, Andrea Ronco, Junjie Zeng, Dengxin Dai, and Michele Magno. Openwatch: A multimodal benchmark for hand gesture recognition on smartwatches. *arXiv preprint arXiv:2605.04791*, 2026.

Centers for Disease Control and Prevention (CDC). National health and nutrition examination survey data, 2010. URL `https://www.cdc.gov/nchs/nhanes.htm`.

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.

Charikleia Chatzaki, Matthew Pediaditis, George Vavoulas, and Manolis Tsiknakis. Human daily activity and fall recognition using a smartphone's acceleration sensor. In *International Conference on Information and Communication Technologies for Ageing Well and e-Health*, pp. 100–118. Springer, 2016.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PmLR, 2020.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International conference on machine learning*, pp. 7480–7512. PMLR, 2023.

Silvia Del Din, Alan Godfrey, Claudia Mazzà, Sue Lord, and Lynn Rochester. Free-living monitoring of parkinson's disease: Lessons from the field. *Movement disorders*, 31(9):1293–1313, 2016.

Aiden Doherty, Dan Jackson, Nils Hammerla, Thomas Plötz, Patrick Olivier, Malcolm H Granat, Tom White, Vincent T Van Hees, Michael I Trenell, Christoper G Owen, et al. Large scale population assessment of physical activity using wrist worn accelerometers: the uk biobank study. *PloS one*, 12(2):e0169649, 2017.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Agrim Gupta, Jiajun Wu, Jia Deng, and Fei-Fei Li. Siamese masked autoencoders. *Advances in Neural Information Processing Systems*, 36:40676–40693, 2023.

Harish Haresamudram, David V Anderson, and Thomas Plötz. On the role of features in human activity recognition. In *Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pp. 78–88, 2019.

Harish Haresamudram, Irfan Essa, and Thomas Plötz. Contrastive predictive coding for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(2):1–26, 2021.

Harish Haresamudram, Irfan Essa, and Thomas Plötz. Assessing the state of self-supervised human activity recognition using wearables. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–47, 2022.

Harish Haresamudram, Chi Ian Tang, Sungho Suh, Paul Lukowicz, and Thomas Ploetz. Past, present, and future of sensor-based human activity recognition using wearables: A surveying tutorial on a still challenging task. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(2):1–44, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 15979–15988. IEEE, 2022.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Benjamin Hoffman, Maddie Cusimano, Vittorio Baglione, Daniela Canestrari, Damien Chevallier, Dominic L DeSantis, Lorène Jeantet, Monique A Ladds, Takuya Maekawa, Vicente Mata-Silva, et al. A benchmark for computational analysis of animal behavior, using animal-borne tags. *Movement Ecology*, 12(1):78, 2024.

Justin J Kavanagh and Hylton B Menz. Accelerometry: a technique for quantifying movement patterns during walking. *Gait & posture*, 28(1):1–15, 2008.

Jiwan Kim, Chi-Jung Lee, Hohurn Jung, Tianhong Catherine Yu, Ruidong Zhang, Ian Oakley, and Cheng Zhang. Watchhand: Enabling continuous hand pose tracking on off-the-shelf smartwatches. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*, pp. 1–21, 2026.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Hong Liu, Jeff Z HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

Shengzhong Liu, Tomoyoshi Kimura, Dongxin Liu, Ruijie Wang, Jinyang Li, Suhas Diggavi, Mani Srivastava, and Tarek Abdelzaher. FOCAL: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Aleksej Logacjov, Kerstin Bach, Atle Kongsvold, Hilde Bremseth Bårdstu, and Paul Jarle Mork. Harth: a human activity recognition dataset for machine learning. *Sensors*, 21(23):7853, 2021.

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

Mohammad Malekzadeh, Richard Clegg, Andrea Cavallaro, and Hamed Haddadi. Dana: Dimension-adaptive neural architecture for multivariate sensor data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(3):1–27, 2021.

Mike A Merrill, Esteban Safranchik, Arinbjörn Kolbeinsson, Piyusha Gade, Ernesto Ramirez, Ludwig Schmidt, Luca Foschini, and Tim Althoff. Homekit2020: A benchmark for time series classification on a large mobile sensing dataset with laboratory tested ground truth of influenza infections. In *Conference on Health, Inference, and Learning*, pp. 207–228. PMLR, 2023.

Girish Narayanswamy, Xin Liu, Kumar Ayush, Yuzhe Yang, Xuhai Xu, Shun Liao, Jake Garrison, Shyam Tailor, Jacob Sunshine, Yun Liu, et al. Scaling wearable foundation models. In *International Conference on Learning Representations*, volume 2025, pp. 59382–59413, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

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

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. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 9(4):1–32, 2025.

Colorado J Reed, Ritwik Gupta, Shufan Li, Sarah Brockman, Christopher Funk, Brian Clipp, Kurt Keutzer, Salvatore Candido, Matt Uyttendaele, and Trevor Darrell. Scale-mae: A scale-aware masked autoencoder for multiscale geospatial representation learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4065–4076. IEEE, 2023.

Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Multi-task self-supervised learning for human activity detection. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–30, 2019.

Narayan Schuetz, Yuze Bai, Lianggang Pan, Edgar Eggert, Favour Nerrise, Juan Delgado-SanMartin, Max Rosenblattl, Milana Gurbanova, Mohammad Asadi, Anders Johnson, et al. Openmhc: Accelerating the science of wearable foundation models. *arXiv preprint arXiv:2607.16235*, 2026.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *2016 IEEE international conference on pervasive computing and communications (PerCom)*, pp. 1–9. IEEE, 2016.

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.

Weijun Tao, Tao Liu, Rencheng Zheng, and Hutian Feng. Gait analysis using wearable sensors. *Sensors*, 12(2):2255–2283, 2012.

Prithviraj Tarale, Kiet Chu, Abhishek Varghese, Kai-Chun Liu, Maxwell A Xu, Mohit Iyyer, and Sunghoon I Lee. Bio-inspired self-supervised learning for wrist-worn accelerometer data. *arXiv preprint arXiv:2603.10961*, 2026.

Richard Troiano. Physical activity in the united states measured by accelerometer. *Medicine & Science in Sports & Exercise*, 2008.

Astrid Ustad, Aleksej Logacjov, Stine Øverengen Trollebø, Pernille Thingstad, Beatrix Vereijken, Kerstin Bach, and Nina Skjæret Maroni. Validation of an activity type recognition model classifying daily physical behavior in older adults: The HAR70+ model. *Sensors*, 23(5):2368, 2023. doi: 10.3390/s23052368.

Maxwell Xu, Alexander Moreno, Hui Wei, Benjamin M Marlin, and James Rehg. Rebar: Retrieval-based reconstruction for time-series contrastive learning. In *International Conference on Learning Representations*, volume 2024, pp. 15440–15464, 2024.

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 *International Conference on Learning Representations*, volume 2025, pp. 53188–53206, 2025a.

Maxwell A Xu, Girish Narayanswamy, Kumar Ayush, Dimitris Spathis, Shun Liao, Shyam A Tailor, Ahmed Metwally, A Ali Heydari, Yuwei Zhang, Jake Garrison, et al. Lsm-2: Learning from incomplete wearable sensor data. *arXiv preprint arXiv:2506.05321*, 2025b.

Che-Chang Yang and Yeh-Liang Hsu. A review of accelerometry-based wearable motion detectors for physical activity monitoring. *Sensors*, 10(8):7772–7788, 2010.

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, 2024a.

Hang Yuan, Tatiana Plekhanova, Rosemary Walmsley, Amy C Reynolds, Kathleen J Maddison, Maja Bucan, Philip Gehrman, Alex Rowlands, David W Ray, Derrick Bennett, et al. Self-supervised learning of accelerometer data provides new insights for sleep and its association with mortality. *NPJ digital medicine*, 7(1):86, 2024b.

Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in neural information processing systems*, 35:3988–4003, 2022.

A DATASETS

Table 3: Overview of the evaluated datasets, grouped by window length. All signals are tri-axial accelerometry, downsampled to 50Hz (if their original frequencies are higher). The overlap is set to 50% for the 2s and 5s datasets, 20% for the 1s datasets, and 0 for Capture-24 and SimFL since they have a larger subject count.

Dataset	Win.	#Users	#Classes	Classes
OpenWatch	1 s	50	6	double clench, double pinch, pinch down, pinch up, slide, negative
WatchHand		23	3	wrist up, wrist down, wrist left
MobiAct	2 s	61	11	standing, walking, jogging, jumping, stairs up, stairs down, stand-to-sit, sitting, sit-to-stand, car step-in, car step-out
MotionSense		24	6	stairs down, stairs up, walking, jogging, standing, sitting
RealWorld	5 s	15	8	climbing down, climbing up, jumping, lying, running, sitting, standing, walking
HARTH		22	9	walking, running, shuffling, stairs up, stairs down, standing, sitting, lying, cycling
Capture-24	10 s	151	10	sleep, sitting, standing, walking, bicycling, vehicle, household chores, manual work, sports, mixed activity
PAAWS-SimFL		250	9	sitting, standing, lying, walking, stairs up, stairs down, gym exercises, biking, household chores

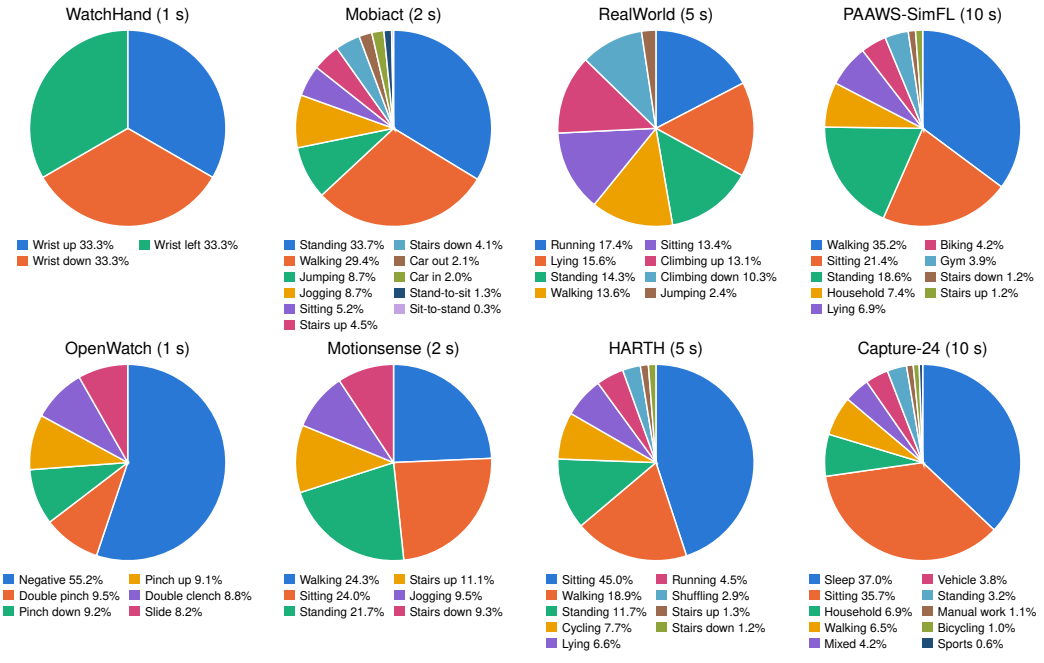


Figure 4: Visualizing the class proportions of the downstream datasets. Many datasets have substantial class imbalance.

We utilize the following window lengths for the datasets and provide a summary in Table 3: *(i)* OpenWatch (Bonazzi et al., 2026) and WatchHand (Kim et al., 2026) contain short wrist gestures such as pinches, clenches, and wrist rotations; following Bonazzi et al. (2026), we use a window

size of 1s. *(ii)* Mobiact (Chatzaki et al., 2016) and Motionsense (Malekzadeh et al., 2021) contain locomotion-style activities recorded from smartphones at the waist/in the pocket, including sitting, walking, and jogging; following Haresamudram et al. (2022), we use 2s windows. *(iii)* RealWorld (Sztyler & Stuckenschmidt, 2016) and HARTH (Logacjov et al., 2021) also comprise locomotion activities but have longer activity segments, motivating the use of longer, 5s windows, following (Liu et al., 2023; Ustad et al., 2023). Finally, *(iv)* Capture-24 (Chan et al., 2024) and PAAWS-SimFL (Potter et al., 2025) contain broader and more heterogenous activities of daily living, such as housework, cycling, and sleeping, evaluated using 10s windows, based on prior work (Yuan et al., 2024a; Potter et al., 2025). They are also the largest in terms of participant count. Overall, these datasets span both small- and large-cohort settings with temporal contexts ranging from brief gestures to sustained and heterogenous daily living activities. This enables us to systematically evaluate if a single pre-trained model can perform effectively across substantially different settings.

B Pre-training and Evaluation Settings

For `VarMAE` and all approaches in the controlled evaluation, we use the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 10^{-3} and no weight decay. MAE and our method utilize use $(\beta_1, \beta_2) = (0.9, 0.95)$, while SimCLR and AugPred use $(0.9, 0.999)$. We pre-train for a maximum of 250k steps. The learning rate is linearly warmed up over the first 20k steps and then decayed to zero using cosine-decay over the remaining steps. Training uses bf16 mixed precision and distributed data parallelism across 2 GPUs, with a total batch size of 1024 (512 / GPU).

For NHANES, we randomly split users into training and validation at a 80:20 ratio. We define an epoch as 10k training batches, followed by evaluation using 1k validation batches. We use early stopping with patience of five epochs, i.e., if the validation loss does not improve for five epochs, the training is terminated, and the checkpoint obtaining the lowest val loss is used for downstream evaluation.

Batch Construction To construct the batch on each GPU, we first uniformly sample 64 training users without replacement. Then, we sample one day per user and then draw weight 30s-windows from that day, yielding $64 \times 8 = 512$ windows per GPU. Windows are sampled with probability proportional to their standard deviation, so that stationary periods are sampled less often. This sampling process is similar to the one used by Yuan et al. (2024a).

For the specialist FMs, the input window duration is either $\{1, 2, 5, 10\}$ seconds, which are obtained after cropping from the 30s segments. The crop start time is also drawn with probability proportional to the standard deviation of the resulting crop. The sampling rate is set to 50Hz.

For `VarMAE`, each short-view has duration sampled randomly from $\{1, 2, 5, 10\}$ s and the input rate drawn uniformly from $\{10, 20, 30, 40, 50\}$ Hz. The resulting short-view is paired with its corresponding full 30s window sampled at 50 Hz.

Table 4: Input configurations for accelerometer FMs.

Foundation Model	Window (s)	Sampling rate (Hz)
Harnet-Oxford (Yuan et al., 2024a)	5/10	30
Bio-inspired FM (Tarale et al., 2026)	10	30
REBAR (Xu et al., 2024)	2.56	100
RelCon (Xu et al., 2025a)	2/2.56	100
Oxford-Sleep FM (Yuan et al., 2024b)	30	30

B.1 Baseline Implementation Details

The baselines in the controlled evaluation are trained within our experimental setup and therefore use identical data pre-processing and training settings (see details above).

1. AugPred: We follow the setup of Yuan et al. (2024a), and utilize three augmentations provided in their implementation (https://github.com/OxWearables/ssl-wearables). The augmentations include: random permutation, time flipping, and time warping.
2. SimCLR: During pretraining, we randomly sample a pair of transforms for each batch from the following set: adding gaussian noise, scaling, rotations, negation, time flipping, time warping, and channel shuffling. The temperature for the InfoNCE loss is set to 0.1. The transforms are taken from: https://github.com/iantangc/ContrastiveLearningHAR.
3. Vanilla MAE: this is the closest match to `VarMAE` and uses a masking ratio of 0.5. It differs from our approach as it only lacks the variable-rate tokenization and reconstruction heads.

In the Public Checkpoint evaluation, we use Oxford-Harnet Yuan et al. (2024a) and BioPM Tarale et al. (2026). Oxford-Harnet was pretrained at 30Hz using window sizes of 5 or 10s, while BioPM was pre-trained using 10s windows also at 30Hz. We use code from the original repositories (https://github.com/OxWearables/ssl-wearables) and (https://github.com/Prithvitarale/biopm), to pre-process the evaluation datasets and extract representations according the requirements of the public checkpoint. For BioPM we additionally follow the original filtering and linear movement-element extraction pipeline. The subsequent linear evaluation is performed within our setup, ensuring consistent evaluation protocol across methods.

C RESULTS

Table 5: Evaluating the impact of each loss component on performance. Bold: best within each dataset and rate. †Rate not used in pretraining.

	50 Hz	25 Hz†	10 Hz	5 Hz†
OpenWatch (1 s)				
All losses	0.626	**0.595**	**0.581**	0.410
w/o $\mathcal{L}_{\times}$	0.606 (−0.020)	0.534 (−0.061)	0.571 (−0.010)	0.332 (−0.078)
w/o $\mathcal{L}_{\text{crop}}$	**0.632** (+0.006)	0.528 (−0.067)	0.435 (−0.146)	0.147 (−0.263)
w/o $\mathcal{L}_{\text{tile}}$	0.612 (−0.014)	0.588 (−0.007)	0.569 (−0.012)	**0.456** (+0.046)
Mobiact (2 s)				
All losses	0.844	**0.846**	0.836	0.752
w/o $\mathcal{L}_{\times}$	0.843 (−0.001)	0.820 (−0.026)	**0.839** (+0.003)	0.659 (−0.093)
w/o $\mathcal{L}_{\text{crop}}$	0.840 (−0.004)	0.804 (−0.042)	0.765 (−0.071)	0.231 (−0.521)
w/o $\mathcal{L}_{\text{tile}}$	**0.845** (+0.001)	0.837 (−0.009)	0.825 (−0.011)	**0.792** (+0.040)
RealWorld (5 s)				
All losses	0.818	0.810	**0.819**	0.782
w/o $\mathcal{L}_{\times}$	0.810 (−0.008)	0.811 (+0.001)	0.807 (−0.012)	0.711 (−0.071)
w/o $\mathcal{L}_{\text{crop}}$	**0.830** (+0.012)	0.786 (−0.024)	0.771 (−0.048)	0.440 (−0.342)
w/o $\mathcal{L}_{\text{tile}}$	0.820 (+0.002)	**0.821** (+0.011)	0.805 (−0.014)	**0.785** (+0.003)
SimFL (10 s)				
All losses	0.762	**0.753**	0.723	0.596
w/o $\mathcal{L}_{\times}$	0.764 (+0.002)	0.707 (−0.046)	**0.738** (+0.015)	0.479 (−0.117)
w/o $\mathcal{L}_{\text{crop}}$	**0.769** (+0.007)	0.674 (−0.079)	0.612 (−0.111)	0.326 (−0.270)
w/o $\mathcal{L}_{\text{tile}}$	0.758 (−0.004)	0.750 (−0.003)	0.721 (−0.002)	**0.637** (+0.041)

Table 6: Evaluating the impact of the masking ratio of the long and short views. Bold: best within each dataset and rate. *Rate not used in pretraining.

	50 Hz	25 Hz*	10 Hz	5 Hz*
OpenWatch (1 s)				
main (tile 0.5, crop 0.75)	0.626	0.595	**0.581**	0.410
tile 0.25	0.611 (−0.015)	**0.609** (+0.014)	0.575 (−0.006)	0.384 (−0.026)
tile 0.75	**0.627** (+0.001)	0.597 (+0.002)	0.567 (−0.014)	**0.461** (+0.051)
crop 0.25	0.623 (−0.003)	0.583 (−0.012)	0.559 (−0.022)	0.222 (−0.188)
crop 0.5	0.610 (−0.016)	0.590 (−0.005)	0.568 (−0.013)	0.169 (−0.241)
Mobiact (2 s)				
main (tile 0.5, crop 0.75)	0.844	**0.846**	0.836	0.752
tile 0.25	0.843 (−0.001)	0.841 (−0.005)	0.839 (+0.003)	0.744 (−0.008)
tile 0.75	0.840 (−0.004)	0.835 (−0.011)	0.830 (−0.006)	**0.790** (+0.038)
crop 0.25	0.833 (−0.011)	0.829 (−0.017)	0.825 (−0.011)	0.583 (−0.169)
crop 0.5	**0.849** (+0.005)	0.843 (−0.003)	**0.842** (+0.006)	0.514 (−0.238)
RealWorld (5 s)				
main (tile 0.5, crop 0.75)	0.818	0.810	**0.819**	0.782
tile 0.25	0.819 (+0.001)	0.823 (+0.013)	0.817 (−0.002)	0.792 (+0.010)
tile 0.75	0.820 (+0.002)	0.822 (+0.012)	0.813 (−0.006)	**0.794** (+0.012)
crop 0.25	**0.831** (+0.013)	0.823 (+0.013)	0.813 (−0.006)	0.690 (−0.092)
crop 0.5	0.821 (+0.003)	**0.825** (+0.015)	**0.819** (0.000)	0.657 (−0.125)
SimFL (10 s)				
main (tile 0.5, crop 0.75)	0.762	0.753	0.723	0.596
tile 0.25	**0.766** (+0.004)	**0.758** (+0.005)	0.727 (+0.004)	0.538 (−0.058)
tile 0.75	0.754 (−0.008)	0.741 (−0.012)	0.716 (−0.007)	**0.649** (+0.053)
crop 0.25	0.752 (−0.010)	0.731 (−0.022)	0.723 (0.000)	0.447 (−0.149)
crop 0.5	0.765 (+0.003)	0.757 (+0.004)	**0.736** (+0.013)	0.422 (−0.174)