MoJEPA: Learning Transferable Motion Representations via Structured Dual-View Joint-Embedding Prediction

Anonymous authors
Paper under double-blind review

Abstract

A fundamental challenge in sensor-based Human Activity Recognition (HAR) is learning transferable motion representations from unlabeled inertial sensor signals for effective cross-domain adaptation. Existing self-supervised methods often rely on signal reconstruction or consistency across augmented views, which may capture low-level signal variations rather than the underlying physical structure of motion, limiting the transferability of the learned representations. To address this problem, Joint-Embedding Predictive Architectures (JEPAs) offer a framework for learning representations through latent-space prediction. Building on this framework, we propose MoJEPA, a dual-view joint-embedding predictive framework that learns transferable motion representations by explicitly modeling the underlying motion structure. MoJEPA introduces two structured masking views along the channel and temporal dimensions, guiding the model to predict masked latent representations from complementary motion cues. The channel view encourages the representation to capture cross-channel physical dependencies, while the temporal view captures temporal motion dynamics. Jointly optimizing these complementary objectives encourages representations that capture shared motion structure. Extensive experiments with pretraining on two large HAR datasets and evaluation across seven target datasets show that MoJEPA outperforms existing self-supervised baselines under frozen-encoder evaluation with few-shot and low-label adaptation.

1 Introduction

Wearable and mobile inertial sensors, such as accelerometers and gyroscopes, are widely used for human activity recognition (HAR). However, inertial signal patterns associated with the same activity can vary substantially across users, devices, and sensor placements. Such domain shifts make it difficult for HAR models trained in one domain to adapt to new sensing domains, particularly when only limited labeled data are available. A fundamental challenge is therefore to learn transferable motion representations from unlabeled inertial sensor signals that capture shared motion structure across domains.

Self-supervised learning (SSL) provides a effective way for addressing this challenge by learning from large-scale unlabeled data and adapting to new sensing domains with limited labeled data. Existing SSL-based approaches for HAR commonly rely on signal reconstruction or contrastive learning objectives (Logacjov & Bach, 2024; Hong et al., 2024). Signal reconstruction methods (Xu et al., 2021; Haresamudram et al., 2020; Cheng et al., 2024) learn representations by masking portions of inertial signals and reconstructing the corresponding signal values. However, inertial signals are not clean observations of motion and often contain high-frequency noise and stochastic fluctuations. Minimizing raw-signal reconstruction error may encourage the model to preserve fine-grained signal details, including noise and local variations, even when such details are irrelevant to the underlying motion structure. Contrastive learning methods typically learn representations by enforcing consistency across augmented views or aligning representations with semantic information. Augmentation-based methods (Khaertdinov et al., 2021; Eldele et al., 2023) apply transformations such as temporal shifting, scaling, jittering, and masking to generate different views of the same sample, encouraging their representations to remain close while distinguishing representations of

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.

different samples. This leads the model to learn representations that are invariant to these perturbations. However, the learned representations depend on whether the applied augmentations preserve the underlying motion structure (Calatrava-Nicolás et al., 2026). When transformations alter motion-relevant patterns or leave domain-specific signal variations unchanged, the model may learn invariances that do not reflect the underlying motion structure. Semantic-alignment methods (Zhang et al., 2024) provide a more explicit semantic learning signal by aligning motion representations with textual descriptions. However, obtaining such semantic supervision requires additional multimodal data or text generation pipelines, which introduces extra data and computational requirements.

These limitations suggest that the prediction target itself may be worth reconsidering. Joint-Embedding Predictive Architectures (JEPA) (Assran et al., 2023; 2025) provide a framework for self-supervised learning through latent-space prediction. Rather than reconstructing raw inertial signals and their variations (Cheng et al., 2026), JEPA predicts latent representations. This shifts the prediction target from individual signal values to contextualized representations in the latent space that reflect the underlying structure of motion. However, learning such representations from inertial signals is challenging because motion is jointly structured across the sensor-channel and temporal dimensions. Across sensor channels, measurements are physically related, with different axes and sensing modalities providing distinct observations of the same underlying movement. Along the temporal dimension, motion evolves over time, and its dynamics are reflected in the temporal patterns of sensor measurements. Many activities in HAR are sustained movements with recurring or periodic patterns rather than instantaneous events (Bulling et al., 2014). As a result, when only local portions of the signal are masked, nearby observations may already provide sufficient information to predict the masked representations. The model can therefore rely heavily on local temporal patterns without fully modeling the cross-channel physical dependencies and longer-range temporal dynamics of the underlying motion.

To explicitly model the physical structure and temporal dynamics of motion, we propose MoJEPA, a structured dual-view JEPA for learning transferable motion representations from inertial signals. MoJEPA introduces two complementary prediction views along the sensor-channel and temporal dimensions. In the channel view, a randomly selected subset of sensor channels is fully masked while the temporal context is preserved, and their latent representations are predicted from the remaining channels. This removes all temporal observations from the masked channels and requires the model to rely on cross-channel dependencies, where complementary measurements jointly characterize the underlying motion. In the temporal view, a continuous temporal block is masked across all channels, and its latent representations are predicted from the surrounding temporal context. By masking a contiguous block rather than isolated temporal patches, this view encourages the model to use temporal context beyond individual local observations and capture the continuity and dynamics of motion. Together, the two views impose complementary inductive biases on latent prediction, encouraging representations that encode cross-channel physical dependencies and temporal dynamics. Extensive experiments across seven target datasets, with pretraining on two large HAR datasets, show that MoJEPA outperforms existing self-supervised baselines under frozen-encoder evaluation with few-shot and low-label adaptation.

Our contributions are summarized as follows:

- We formulate self-supervised inertial HAR as latent motion representation prediction to learn transferable motion representations across domains.
- We propose MoJEPA, a dual-view JEPA that imposes complementary inductive biases to capture cross-channel physical dependencies and temporal dynamics.
- Experiments across seven cross-dataset HAR benchmarks show improvements over existing self-supervised baselines.

2 Related Work

2.1 Self-Supervised Learning for HAR

Self-supervised learning for HAR mainly explores signal reconstruction and contrastive learning. Reconstruction-based methods (Haresamudram et al., 2020; Qiu et al., 2025) recover masked sensor signals from visible context. LIMU-BERT (Xu et al., 2021) (Xu et al., 2021) and MaskCAE (Cheng

et al., 2024) apply masked reconstruction to inertial signals, while RCSMR (Logacjov & Bach, 2024) performs cross-sensor reconstruction, SWISS (Lim et al., 2024) and RobustHAR (Liu et al., 2025) investigate different masking strategies, and Bio-PM (Tarale et al., 2026) uses motion-aligned segments as reconstruction targets. Contrastive methods (Khaertdinov et al., 2021) learn representations by aligning positive views and separating negative instances. TS-TCC (Eldele et al., 2021) and CA-TCC (Eldele et al., 2023) exploit temporal and class-aware views, while ColloSSL (Jain et al., 2022) uses synchronized signals across devices. CrossHAR (Hong et al., 2024) combines signal reconstruction and contrastive learning for cross-dataset representation learning. UniMTS (Zhang et al., 2024) further introduces external text for semantic alignment. Existing methods mainly construct self-supervision through signal reconstruction or view-level alignment, whereas MoJEPA performs latent prediction directly from unlabeled inertial data and imposes complementary inductive biases along the sensor-channel and temporal dimensions.

2.2 Joint-Embedding Predictive Architecture

Joint-Embedding Predictive Architectures (JEPAs) have emerged as an alternative paradigm for self-supervised representation learning by predicting latent representations rather than reconstructing observations in the input space. I-JEPA (Assran et al., 2023) demonstrated this paradigm for images by predicting latent representations of masked image regions from visible context, while V-JEPA (Bardes et al., 2024) extended it to masked spatio-temporal regions in videos. A-JEPA (Fei et al., 2023) adapted latent prediction to audio by predicting masked time-frequency regions in spectrograms. These studies demonstrate the use of masked latent prediction for learning representations without reconstructing low-level input details. Recent work has extended JEPA to sequential and physiological signals. Time-series JEPA approaches (Ennadir et al., 2025; Girgis et al., 2026) explore latent prediction for temporal data, while ECG-JEPA (Weimann & Conrad, 2025; Kim, 2026) applies JEPA to physiological signals. For multichannel inertial signals, however, motion is jointly expressed through physically coupled sensor channels and evolves continuously over time. MoJEPA therefore constructs complementary latent prediction views along the sensor-channel and temporal dimensions, encouraging the learned representations to reflect both cross-channel physical dependencies and temporal motion dynamics.

3 Methodology

Unlabeled inertial recordings are multivariate time series collected from multiple sensor channels. We first resample each recording to a fixed sampling frequency and segment it into non-overlapping windows $\mathbf{W}_i$, where each window $\mathbf{W}_i \in \mathbb{R}^{C\times T}$ contains C sensor channels and T time steps. Our goal is to learn transferable motion representations from these unlabeled windows, so that activity recognition can be performed to support activity recognition in target environments with only a few labeled windows.

To achieve this goal, we propose MoJEPA, a joint-embedding predictive architecture trained by dual-view masked latent prediction. Figure 1 provides an overview of MoJEPA. The framework has two components: pretraining by dual-view masked latent prediction and a channel-pooled linear probe for downstream activity recognition. During pretraining, latent targets at masked positions are predicted from the unmasked tokens of the same window. For downstream recognition, the predictor is discarded and only a linear layer is trained on the frozen EMA encoder.

3.1 Structured Dual-view Masking

Tokenization. Each window $\mathbf{W}_i \in \mathbb{R}^{C\times T}$ is partitioned along the temporal dimension into $T_p = T/P$ non-overlapping patches of length P, while keeping the channels separate. This yields $C \times T_p$ channel–time patches, where patch (c, τ) is the length-P segment of channel c at temporal cell $\tau \in 1, \ldots, T_p$. Each patch is linearly projected to a D-dimensional token $\mathbf{z}_{c,\tau} \in \mathbb{R}^D$, where (c, τ) denotes its channel and temporal position (Dosovitskiy, 2020).

Channel view. To explicitly learn dependencies across sensor channels that jointly measure the same physical motion, we construct the channel view by masking entire channels. For each window, we randomly sample a subset $\mathcal{C} \subset \{1, \ldots, C\}$ and mask every token of those channels, where $|\mathcal{C}|$

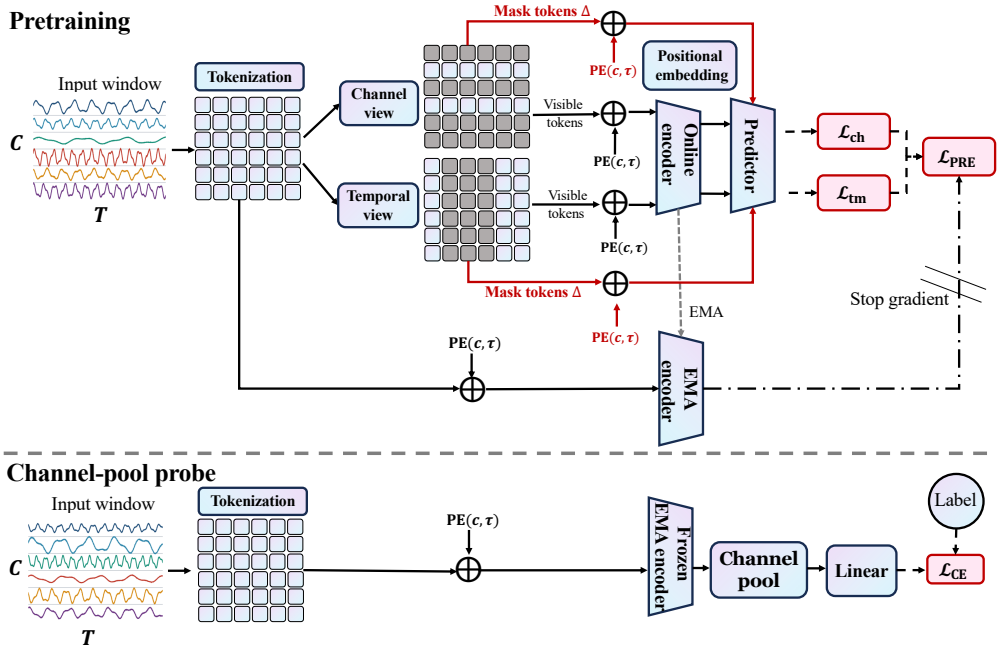


Figure 1: Overview of MoJEPA. **Pretraining** (top). We first pretrain the MoJEPA encoder on unlabeled inertial data using a dual-view masked latent prediction objective. An inertial window is patchified into channel–temporal tokens, and two masking views are formed from the same window. The channel view randomly masks a subset of entire channels, while the temporal view randomly masks a contiguous temporal block across all channels. For each view, the unmasked tokens are added to the corresponding $\mathrm{PE}(c, \tau)$. The shared encoder processes these tokens and outputs a latent representation for each. Next, learnable mask tokens are added to the corresponding $\mathrm{PE}(c, \tau)$ and concatenated with the encoder outputs. The shared predictor processes the concatenated sequence and outputs latent predictions at the masked positions. The target representations are computed by a momentum-updated EMA encoder, which processes every token of the unmasked window. The predictions are compared with these targets at the masked positions, giving $\mathcal{L}_{\mathrm{ch}}$ and $\mathcal{L}_{\mathrm{tm}}$ for the channel and temporal views. The pretraining objective $\mathcal{L}_{\mathrm{PRE}}$ is the average of the two losses. **Channel-pool probe** (bottom). After pretraining, the predictor is discarded and the EMA encoder is frozen. The full window is processed by the frozen EMA encoder. Representations are averaged over time within each channel, concatenated, and passed to a linear layer. Only this layer is trained with the activity labels, using $\mathcal{L}_{\mathrm{CE}}$.

denotes the number of masked channels. The masked token positions are defined as:

$$\mathcal{M}_{\mathrm{ch}} = \big\{(c, \tau) : c \in \mathcal{C},\ \tau \in \{1, \ldots, T_p\}\big\}, \tag{1}$$

where $\mathcal{M}$ch denotes the masked token positions and $\mathcal{V}$ch denotes the remaining visible token positions in the channel view. Masking an entire channel eliminates that channel's own temporal trace. The model must predict the latent representations at the masked positions from the visible channels. This prevents the model from relying solely on the temporal continuity of the masked channel and encourages it to use cross-channel dependencies.

Temporal view. To explicitly learn the temporal dynamics of the motion, we construct the temporal view by masking one contiguous temporal block across all channels. Specifically, we randomly sample a start index $\tau_0 \in \{1, \ldots, T_p - L + 1\}$ and mask the corresponding block of fixed length L across all C channels:

$$\mathcal{M}_{\mathrm{tm}} = \big\{(c, \tau) : c \in \{1, \ldots, C\},\ \tau_0 \le \tau < \tau_0 + L\big\}, \tag{2}$$

where $\mathcal{M}_{\mathrm{tm}}$ denotes the masked token positions and $\mathcal{V}_{\mathrm{tm}}$ denotes the visible token positions in the temporal view. Unlike scattered token masks, a contiguous masked block removes the local temporal context within the block. The visible tokens before and after the block provide temporal context,

from which the model predicts the latent representations of the masked tokens. This encourages the learned representation to capture longer-term temporal dependencies and the evolution of the underlying motion over time.

3.2 JOINT-EMBEDDING PREDICTION

Positional encoding. Sensor channels have distinct identities rather than an inherent order. For example, the x-, y-, and z-axis accelerometer channels are distinct measurements, not ordered positions. We therefore encode channel identity and temporal position separately. Each channel is treated as a distinct category and has a learned embedding $\mathbf{e}_c \in \mathbb{R}^D$, while each temporal position has a fixed sinusoidal code $\mathrm{sincos}(\tau)$ (Vaswani et al., 2017) that encodes its temporal order within the window. The same τ uses the same code across all channels. The two codes are added in the same D-dimensional space:

$$\mathrm{PE}(c, \tau) = \mathbf{e}_c + \mathrm{sincos}(\tau). \tag{3}$$

This positional encoding is added to each unmasked token and, at a masked position, to the mask token, so that both unmasked and masked tokens receive explicit channel and temporal positional information.

Online encoder. For each view, the visible tokens are fed into the shared Transformer encoder f_θ (Vaswani et al., 2017). Self-attention captures dependencies among the visible tokens within each view, and the encoder returns one contextualized latent at each visible position:

$$\mathbf{h}_{c,\tau} = \Big[f_\theta(\{\mathbf{z}_{c',\tau'} + \mathrm{PE}(c', \tau')\}_{(c',\tau') \in \mathcal{V}_{\mathrm{ch}}}) \Big]_{c,\tau}, \qquad (c, \tau) \in \mathcal{V}_{\mathrm{ch}}, \tag{4}$$

$$\mathbf{h}_{c,\tau} = \Big[f_\theta(\{\mathbf{z}_{c',\tau'} + \mathrm{PE}(c', \tau')\}_{(c',\tau') \in \mathcal{V}_{\mathrm{tm}}}) \Big]_{c,\tau}, \qquad (c, \tau) \in \mathcal{V}_{\mathrm{tm}}. \tag{5}$$

where $\mathbf{h}_{c,\tau} \in \mathbb{R}^D$. The two views are processed independently with shared weights. The encoder does not receive mask tokens. The resulting latents provide the visible context from which the masked tokens are predicted.

Predictor. One learnable mask token $\Delta \in \mathbb{R}^{D_p}$ is shared by the masked positions of a view and carries no input content. The encoder outputs are first projected from D to the predictor hidden width D_p. At that width, a positional code of the same form as Eq. 3, but defined in $\mathbb{R}^{D_p}$, is added to both the projected encoder outputs and Δ. For the channel view, the masked tokens $\Delta + \mathrm{PE}_p(c, \tau)$ at $\mathcal{M}_{\mathrm{ch}}$ are concatenated with the projected visible outputs along the token axis. The same operation is applied to the temporal view using $\mathcal{M}_{\mathrm{tm}}$. A shared Transformer predictor g_ϕ of width D_p processes the resulting sequence and produces D-dimensional latent predictions only at the masked positions.

EMA encoder. The prediction targets are produced by a momentum-updated EMA encoder $f_{\bar{\theta}}$ (Assran et al., 2023; 2025). Unlike the online encoder, it processes all tokens of the full window without masking:

$$\bar{\mathbf{h}}_{c,\tau} = \Big[f_{\bar{\theta}} \left(\{\mathbf{z}_{c',\tau'} + \mathrm{PE}(c', \tau')\}_{(c',\tau')} \right) \Big]_{c,\tau}, \qquad c \in \{1, \ldots, C\},\ \tau \in \{1, \ldots, T_p\}, \tag{6}$$

where $\bar{\mathbf{h}}_{c,\tau} \in \mathbb{R}^D$ denotes the target representation at position (c, τ). The parameters $\bar{\theta}$ are updated from the online encoder parameters θ using an exponential moving average:

$$\bar{\theta} \leftarrow \mu \bar{\theta} + (1 - \mu)\theta, \tag{7}$$

where $\mu \in [0, 1)$ is the momentum coefficient. The EMA encoder is not directly optimized by backpropagation, and its output targets are treated with stop-gradient when computing the prediction loss.

Latent prediction loss. Let $\hat{\mathbf{h}}_{c,\tau}$ denote the predictor output at a masked position. On the channel view it is trained to match the stop-gradient EMA latent from Eq. equation 6 at the same position:

$$\mathcal{L}_{\mathrm{ch}} = \frac{1}{|\mathcal{M}_{\mathrm{ch}}|} \sum_{(c,\tau) \in \mathcal{M}_{\mathrm{ch}}} \left\| \hat{\mathbf{h}}_{c,\tau} - \mathrm{sg}\big(\bar{\mathbf{h}}_{c,\tau}\big) \right\|_1. \tag{8}$$

$\mathcal{L}_{\mathrm{tm}}$ is defined analogously over $\mathcal{M}_{\mathrm{tm}}$:

$$\mathcal{L}_{\mathrm{tm}} = \frac{1}{|\mathcal{M}_{\mathrm{tm}}|} \sum_{(c,\tau)\in\mathcal{M}_{\mathrm{tm}}} \left\| \hat{\mathbf{h}}_{c,\tau} - \mathrm{sg}\big(\bar{\mathbf{h}}_{c,\tau}\big) \right\|_1 . \tag{9}$$

The pretraining objective averages the two views:

$$\mathcal{L}_{\mathrm{PRE}} = \frac{1}{2}\big(\mathcal{L}_{\mathrm{ch}} + \mathcal{L}_{\mathrm{tm}}\big). \tag{10}$$

We minimize $\mathcal{L}_{\mathrm{PRE}}$ with respect to θ and ϕ only. $\bar{\theta}$ receives no gradient from this loss. The two views share f_θ, g_ϕ, and $f_{\bar{\theta}}$, and neither is the target of the other.

3.3 Channel-pool linear probe

After pretraining, the predictor is discarded and the EMA encoder $f_{\bar{\theta}}$ is frozen (Figure 1, bottom). The full window is processed by $f_{\bar{\theta}}$, and the resulting representations are temporally averaged within each channel and then concatenated:

$$\mathbf{q} = \Big[\tfrac{1}{T_p}\sum_{\tau=1}^{T_p} \bar{\mathbf{h}}_{c,\tau}\Big]_{c=1}^{C} \in \mathbb{R}^{CD}. \tag{11}$$

$\mathbf{q}$ is fed to a linear layer that maps it to class logits. Only the linear layer is trained, using the activity labels and the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}$.

4 Results

4.1 Experimental Setup

Datasets. We consider eight common activities in real-world HAR applications: sitting, standing, lying, upstairs, downstairs, walking, running, and jumping. We use accelerometer and gyroscope signals as input. For self-supervised pretraining, we construct a combined unlabeled corpus from two publicly available HAR datasets, DSA (Barshan & Yüksek, 2014) and WISDM (Weiss, 2019), with their activity labels discarded. We evaluate transferability on seven unseen target datasets, Motion (Malekzadeh et al., 2019), HHAR, Shoaib (Shoaib et al., 2014), UCI (Reyes-Ortiz et al., 2016), PAMAP2 (Reiss & Stricker, 2012), USC-HAD (Zhang & Sawchuk, 2012), and mhealth (Banos et al., 2014), with each dataset evaluated independently. Detailed dataset statistics are provided in Appendix A.1.

Data processing and splits. We resample all inertial signals to 20 Hz and segment them into non-overlapping 6-second windows, yielding 120 time steps per window. Each channel is further divided into patches of length 6. For pretraining, all the data are unlabeled. For downstream evaluation, each dataset is split into 10% labeled training, 10% validation, and 80% test sets. The 5% setting uses half of the training split, while 5-shot randomly selects five samples per class. Validation and test sets are fixed across methods.

Model architecture. The main experiments use a Transformer encoder with 6 layers, hidden dimension 64, and 4 attention heads. The predictor has the same depth as the encoder, but uses a smaller hidden dimension of 32. The EMA target encoder shares the architecture of the online encoder and is updated by exponential moving average. In addition to this default configuration, we also evaluate smaller and larger model variants to study the effect of model capacity. Detailed architectural and training configurations are provided in Appendix A.2.

Baselines and variants. We compare MoJEPA with representative self-supervised learning methods for sensor-based HAR and time series, covering different self-supervised learning paradigms, including masked reconstruction, contrastive learning, and joint-embedding prediction. The baselines include LIMU-BERT (Xu et al., 2021), CrossHAR (Hong et al., 2024), TimeMAE (Cheng et al., 2026), PatchTST (Nie et al., 2022), HAR-FM (Qiu et al., 2025), TS-TCC (Eldele et al., 2023), and TS-JEPA (Ennadir et al., 2025). BioBankSSL (Yuan et al., 2024) is included as a large-scale

Table 1: Cross-dataset performance comparison using a frozen backbone and linear probe under limited labeled data. Results are averaged across seven datasets.

Method	Metric	5-shot	10-shot	5%	10%
LIMU-BERT	Acc.	0.306 ± 0.011	0.328 ± 0.010	0.364 ± 0.003	0.371 ± 0.005
	F1	0.271 ± 0.010	0.295 ± 0.013	0.238 ± 0.002	0.241 ± 0.008
CrossHAR	Acc.	0.489 ± 0.006	0.517 ± 0.023	0.445 ± 0.002	0.447 ± 0.005
	F1	0.454 ± 0.007	0.476 ± 0.023	0.300 ± 0.005	0.310 ± 0.014
PatchTST	Acc.	0.555 ± 0.015	0.625 ± 0.005	0.715 ± 0.005	0.755 ± 0.002
	F1	0.533 ± 0.012	0.606 ± 0.005	0.689 ± 0.008	0.733 ± 0.003
TimeMAE	Acc.	0.591 ± 0.018	0.624 ± 0.019	0.601 ± 0.002	0.608 ± 0.003
	F1	0.564 ± 0.019	0.595 ± 0.018	0.515 ± 0.004	0.520 ± 0.005
HAR-FM	Acc.	0.609 ± 0.003	0.689 ± 0.006	0.771 ± 0.005	0.790 ± 0.003
	F1	0.588 ± 0.003	0.667 ± 0.006	0.729 ± 0.009	0.749 ± 0.002
TS-JEPA	Acc.	0.614 ± 0.000	0.674 ± 0.014	0.767 ± 0.006	0.805 ± 0.005
	F1	0.591 ± 0.003	0.648 ± 0.015	0.724 ± 0.006	0.763 ± 0.007
BioBankSSL	Acc.	0.643 ± 0.005	0.660 ± 0.004	0.687 ± 0.005	0.725 ± 0.005
	F1	0.621 ± 0.010	0.626 ± 0.005	0.650 ± 0.005	0.698 ± 0.006
TS-TCC	Acc.	0.651 ± 0.015	0.699 ± 0.004	0.714 ± 0.006	0.724 ± 0.009
	F1	0.639 ± 0.019	0.683 ± 0.005	0.655 ± 0.009	0.661 ± 0.013
UniMTS	Acc.	0.740 ± 0.012	0.750 ± 0.012	0.738 ± 0.008	0.745 ± 0.006
	F1	0.733 ± 0.013	0.744 ± 0.011	0.722 ± 0.008	0.733 ± 0.002
MoJEPA	Acc.	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.929 \pm 0.002}$
	F1	$\mathbf{0.767 \pm 0.008}$	$\mathbf{0.827 \pm 0.006}$	$\mathbf{0.886 \pm 0.008}$	$\mathbf{0.920 \pm 0.005}$

pretraining baseline, using a model pretrained on large-scale wearable activity data. We also include UniMTS (Zhang et al., 2024), a multimodal method pretrained on paired inertial data and textual activity descriptions. In addition, we evaluate ablated variants of MoJEPA to examine the contributions of its dual-view design, complementary channel and temporal masking strategies, and latent-space prediction. Details of the baseline methods and their configurations are provided in Appendix A.3.

4.2 Transferability Performance

Table 1 shows cross-dataset results on seven unseen target datasets under limited-label settings. We freeze each pretrained backbone and train a linear probe using either 5 or 10 labeled samples per class, or 5% or 10% of the training data. Results are averaged across the seven datasets. MoJEPA achieves the highest Accuracy and macro-F1 in all four settings, with clear improvements over the other baselines. Per-dataset results are provided in Appendix B.1, and representation visualizations are provided in Appendix B.2.

4.3 Ablation Study

To verify whether the pretrained MoJEPA encoder learns transferable motion representations, Table 2 compares supervised training from scratch with pretrained MoJEPA. MoJEPA achieves higher performance across all label budgets, with larger gains in the few-shot settings. These results show that the pretrained encoder provides transferable representations that benefit downstream adaptation, particularly when only a small number of target labels are available.

To verify the contributions of the dual-view design and latent-space prediction, we evaluate random, single-view, and dual-view masking with both latent-space prediction and raw-signal reconstruction. All variants use the same frozen-encoder evaluation protocol and channel-pool linear probe. As shown in Table 3, for latent prediction, the dual-view model consistently achieves the highest performance across all label budgets. Removing either view causes a clear performance drop, while random masking also performs below the dual-view model. In contrast, the performance gap between random and dual-view masking is much smaller for raw-signal reconstruction. These results

Table 2: Supervised scratch training vs. pretrained MoJEPA.

Method	Metric	5-shot	10-shot	5%	10%
Scratch	Acc.	0.442 ± 0.011	0.481 ± 0.011	0.475 ± 0.007	0.492 ± 0.010
	F1	0.384 ± 0.007	0.399 ± 0.021	0.334 ± 0.018	0.358 ± 0.013
MoJEPA	Acc.	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.929 \pm 0.002}$
	F1	$\mathbf{0.767 \pm 0.008}$	$\mathbf{0.827 \pm 0.006}$	$\mathbf{0.886 \pm 0.008}$	$\mathbf{0.920 \pm 0.005}$

Table 3: Ablation study of the dual-view design and prediction targets.

Masking view	Prediction target	Metric	5-shot	10-shot	5%	10%
random masking	raw signal	Acc.	0.752 ± 0.005	0.824 ± 0.009	0.876 ± 0.004	0.904 ± 0.004
		F1	0.740 ± 0.006	0.814 ± 0.009	0.860 ± 0.005	0.897 ± 0.004
	latent	Acc.	0.714 ± 0.013	0.777 ± 0.007	0.876 ± 0.002	0.912 ± 0.003
		F1	0.698 ± 0.012	0.764 ± 0.005	0.856 ± 0.005	0.903 ± 0.004
w/o temporal view	raw signal	Acc.	0.718 ± 0.014	0.792 ± 0.006	0.870 ± 0.006	0.902 ± 0.006
		F1	0.702 ± 0.013	0.778 ± 0.008	0.853 ± 0.004	0.892 ± 0.007
	latent	Acc.	0.733 ± 0.013	0.800 ± 0.002	0.875 ± 0.001	0.905 ± 0.006
		F1	0.717 ± 0.013	0.788 ± 0.001	0.856 ± 0.004	0.893 ± 0.007
w/o channel view	raw signal	Acc.	0.625 ± 0.007	0.682 ± 0.013	0.757 ± 0.004	0.802 ± 0.006
		F1	0.605 ± 0.008	0.664 ± 0.010	0.721 ± 0.005	0.775 ± 0.008
	latent	Acc.	0.646 ± 0.010	0.697 ± 0.007	0.761 ± 0.006	0.806 ± 0.007
		F1	0.628 ± 0.010	0.684 ± 0.007	0.721 ± 0.009	0.780 ± 0.008
dual-view	raw signal	Acc.	0.709 ± 0.019	0.796 ± 0.007	0.870 ± 0.002	0.901 ± 0.003
		F1	0.698 ± 0.016	0.786 ± 0.005	0.851 ± 0.002	0.892 ± 0.003
	latent	Acc.	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.929 \pm 0.002}$
		F1	$\mathbf{0.767 \pm 0.008}$	$\mathbf{0.827 \pm 0.006}$	$\mathbf{0.886 \pm 0.008}$	$\mathbf{0.920 \pm 0.005}$

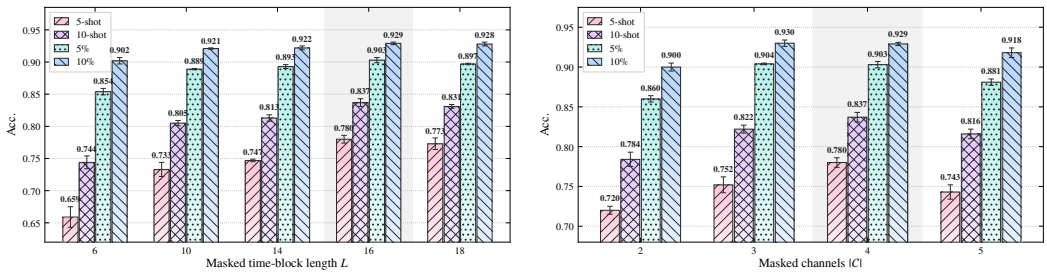
suggest that latent prediction benefits more from structured masking, as random masking may produce scattered visible and masked regions that provide less coherent context for learning high-level motion representations. The complementary temporal and cross-channel constraints introduced by the dual-view design therefore play a more important role in latent-space prediction than in signal-level reconstruction.

4.4 Masking and Patch Length Study

Figure 2 varies L and n independently to examine the effect of mask size. With smaller masks, such as L=6 or n=2, the prediction task becomes easier but downstream performance drops, especially in the few-shot setting. A larger channel mask (n=5) leaves less visible context and also hurts performance, although the drop is smaller. These results suggest that the amount of masking affects the quality of the learned motion representations. Masking too little makes the prediction task too easy, while masking too much removes useful context and makes prediction more difficult. A moderate masking level therefore provides a better balance for learning informative motion representations. Table 4 varies the patch length $P \in 4, 6, 8, 12$, with the temporal block scaled to keep the masked fraction comparable. P=6 gives the best 5-shot accuracy, while P=12 performs best in the 10-shot, 5%, and 10% settings. The differences are modest, and we keep P=6 as the default.

4.5 Encoder capacity study

Table 5 examines the effect of encoder width and depth on classification accuracy. For encoder width, we vary the width from 32 to 128, corresponding to 0.099M–1.512M total parameters in the encoder and predictor, with the predictor width fixed to half of the encoder width and all other settings unchanged. Increasing the width from 32 to 64 substantially improves few-shot performance,



(a) Varying L with n=4.

(b) Varying n with L=16.

Figure 2: Mask-parameter study.

Table 4: Effect of patch length (Acc.).

Patch length	5-shot	10-shot	5%	10%
4	0.772 ± 0.010	0.827 ± 0.002	0.894 ± 0.001	0.920 ± 0.005
6	$\mathbf{0.780 \pm 0.006}$	0.837 ± 0.006	0.903 ± 0.004	0.929 ± 0.002
8	0.759 ± 0.011	0.832 ± 0.004	0.907 ± 0.002	0.935 ± 0.002
12	0.780 ± 0.008	$\mathbf{0.842 \pm 0.003}$	$\mathbf{0.914 \pm 0.003}$	$\mathbf{0.938 \pm 0.005}$

Table 5: Effect of encoder width and depth (Acc.).

Setting	Params	5-shot	10-shot	5%	10%
			Encoder width		
32	0.099M	0.754 ± 0.012	0.794 ± 0.010	0.846 ± 0.002	0.870 ± 0.003
64	0.383M	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	0.903 ± 0.004	0.929 ± 0.002
96	0.855M	0.755 ± 0.006	0.829 ± 0.003	0.910 ± 0.005	0.938 ± 0.004
128	1.512M	0.765 ± 0.006	0.834 ± 0.005	$\mathbf{0.915 \pm 0.002}$	$\mathbf{0.939 \pm 0.004}$
			Encoder depth		
3	0.234M	0.767 ± 0.008	0.816 ± 0.004	0.889 ± 0.002	0.921 ± 0.002
6	0.383M	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	0.903 ± 0.004	0.929 ± 0.002
9	0.533M	0.764 ± 0.007	0.833 ± 0.008	$\mathbf{0.905 \pm 0.002}$	$\mathbf{0.934 \pm 0.006}$
12	0.683M	0.765 ± 0.010	0.831 ± 0.006	0.899 ± 0.002	0.929 ± 0.004

while further widening the encoder mainly benefits the 5% and 10% label settings. For encoder depth, we vary the number of layers in $3, 6, 9, 12$ while keeping the other settings fixed. Increasing the depth beyond 6 layers provides limited or inconsistent improvements across the four evaluation settings. Overall, an encoder width of 64 and depth of 6 provide a compact configuration with strong performance across different label regimes.

5 Conclusion

We presented MoJEPA, a dual-view joint-embedding predictive framework for learning transferable motion representations from unlabeled inertial signals. By performing prediction in the latent space, MoJEPA focuses self-supervised learning on motion structure beyond low-level signal details and variations. Its structured channel and temporal masking views impose complementary inductive biases, encouraging the representation to capture both cross-channel physical dependencies and temporal motion dynamics. Experiments across seven target datasets show that MoJEPA improves performance under few-shot and low-label adaptation. These results suggest that structured latent-space prediction with complementary inductive biases provides an effective way to learn motion representations that transfer across datasets.

AI USE STATEMENT

In this work, we used generative AI tools for **language polishing and improving the readability of the manuscript**. We also used generative AI tools to assist with the formal presentation, specifically for **checking notational consistency and formatting equations**. Generative AI tools were not used to generate research ideas, design the methodology, or produce experimental results. We have reviewed and verified all AI-assisted content and have checked the final manuscript, and we take full responsibility for the content of this work, including all text, claims, and artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

All datasets used in this work are publicly available. The source code is provided as supplementary material for review and will be publicly released upon acceptance. We provide details on the experimental settings in Appendix A and additional experiments in Appendix B.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *International workshop on ambient assisted living*, pp. 91–98. Springer, 2014.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv preprint arXiv:2404.08471*, 2024.

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

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):1–33, 2014.

Francisco M Calatrava-Nicolás, Todor Stoyanov, and Oscar Martinez Mozos. Siamese masked reconstruction with temporal rotation consistency for inertial-based human activity recognition. *Scientific Reports*, 16(1):25518, 2026.

Dongzhou Cheng, Lei Zhang, Lutong Qin, Shuoyuan Wang, Hao Wu, and Aiguo Song. Maskcae: Masked convolutional autoencoder via sensor data reconstruction for self-supervised human activity recognition. *IEEE Journal of Biomedical and Health Informatics*, 28(5):2687–2698, 2024.

Mingyue Cheng, Xiaoyu Tao, Zhiding Liu, Qi Liu, Hao Zhang, Rujiao Zhang, and Enhong Chen. Timemae: Self-supervised representations of time series with decoupled masked autoencoders. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 498–508, 2026.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv preprint arXiv:2106.14112*, 2021.

Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. Self-supervised contrastive representation learning for semi-supervised time-series classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12): 15604–15618, 2023.

Sofiane Ennadir, Siavash Golkar, and Leopoldo Sarra. Joint embeddings go temporal. *arXiv preprint arXiv:2509.25449*, 2025.

Zhengcong Fei, Mingyuan Fan, and Junshi Huang. A-jepa: Joint-embedding predictive architecture can listen. *arXiv preprint arXiv:2311.15830*, 2023.

Abanoub M Girgis, Alvaro Valcarce, and Mehdi Bennis. Time-series jepa for predictive remote control under capacity-limited networks. *IEEE Internet of Things Journal*, 2026.

Harish Haresamudram, Apoorva Beedu, Varun Agrawal, Patrick L Grady, Irfan Essa, Judy Hoffman, and Thomas Plötz. Masked reconstruction based self-supervision for human activity recognition. In *Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 45–49, 2020.

Zhiqing Hong, Zelong Li, Shuxin Zhong, Wenjun Lyu, Haotian Wang, Yi Ding, Tian He, and Desheng Zhang. Crosshar: Generalizing cross-dataset human activity recognition via hierarchical self-supervised pretraining. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(2):1–26, 2024.

Yash Jain, Chi Ian Tang, Chulhong Min, Fahim Kawsar, and Akhil Mathur. Collossl: Collaborative self-supervised learning for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(1):1–28, 2022.

Bulat Khaertdinov, Esam Ghaleb, and Stylianos Asteriadis. Contrastive self-supervised learning for sensor-based human activity recognition. In *2021 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–8. IEEE, 2021.

Sehun Kim. Learning general representation of 12-lead ecg with a joint-embedding predictive architecture. *Neural Networks*, pp. 109334, 2026.

Saerin Lim, Young Jae Lee, and Seoung Bum Kim. Swiss: Signal-wise self-supervised learning to learn a multi-signal interaction for human activity recognition. *Knowledge-Based Systems*, 287: 111464, 2024.

Xiao Liu, Guan Yuan, Yanmei Zhang, Shang Liu, and Qiuyan Yan. Robusthar: Multi-scale spatial-temporal masked self-supervised pre-training for robust human activity recognition. In *IJCAI*, pp. 8563–8571, 2025.

Aleksej Logacjov and Kerstin Bach. Self-supervised learning with randomized cross-sensor masked reconstruction for human activity recognition. *Engineering Applications of Artificial Intelligence*, 128:107478, 2024.

Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*, pp. 49–58, 2019.

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.

Minghui Qiu, Cekai Weng, Mingming Fan, and Kaishun Wu. Towards customizable foundation models for human activity recognition with wearable devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–29, 2025.

Attila Reiss and Didier Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*, pp. 108–109. IEEE, 2012.

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

Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, and Paul JM Havinga. Fusion of smartphone motion sensors for physical activity recognition. *Sensors*, 14(6):10146–10176, 2014.

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 imu signals. *arXiv e-prints*, pp. arXiv–2603, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Kuba Weimann and Tim OF Conrad. Self-supervised pre-training with joint-embedding predictive architecture boosts ecg classification performance. *Computers in Biology and Medicine*, 196: 110809, 2025.

Gary M Weiss. Wisdm smartphone and smartwatch activity and biometrics dataset. *UCI Machine Learning Repository: WISDM Smartphone and Smartwatch Activity and Biometrics Dataset Data Set*, 7(133190-133202):5, 2019.

Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. Limu-bert: Unleashing the potential of unlabeled data for imu sensing applications. In *Proceedings of the 19th ACM conference on embedded networked sensor systems*, pp. 220–233, 2021.

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

Mi Zhang and Alexander A Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pp. 1036–1043, 2012.

Xiyuan Zhang, Diyan Teng, Ranak R Chowdhury, Shuheng Li, Dezhi Hong, Rajesh K Gupta, and Jingbo Shang. Unimts: Unified pre-training for motion time series. *Advances in Neural Information Processing Systems*, 37:107469–107493, 2024.

A Additional Experimental Details

A.1 Datasets Overview

We use nine public HAR datasets in our experiments. DSA and WISDM provide the largest unlabeled pools under our unified preprocessing and are used exclusively for self-supervised pretraining, with activity labels discarded. The remaining seven datasets (PAMAP2, USC-HAD, HHAR, MHealth, Shoaib, UCI, and Motion) were held out for downstream evaluation and never used during pretraining. Table 6 summarizes the statistics of the nine datasets. Each dataset defines its own activity taxonomy, with the number of original classes ranging from 6 to 18 (Orig. in Table 6). To enable a shared evaluation protocol, we retain eight activities that recur across datasets: sitting, standing, lying, walking upstairs, walking downstairs, walking, running, and jumping. We map the labels of each dataset to this unified set (Sel. in Table 6). Datasets that do not contain a given class contribute neither windows nor false negatives for that class.

Table 6: Statistics of the HAR datasets used in the experiments.

Role	Dataset	Device	#Subj.	#Windows	Orig.	Sel.	Placement
Pretrain	DSA	IMU	8	16,000	18	8	chest, arms, legs
	WISDM	Phone; Watch	51	14,385	18	5	pocket, wrist
Eval	PAMAP2	IMU	9	5,796	18	8	wrist, chest, ankle
	USC-HAD	IMU	14	3,039	12	8	front-right hip
	HHAR	Phone; Watch	9	9,178	6	5	wrist, waist
	mhealth	IMU	10	1,452	12	7	chest, wrist, ankle
	Shoaib	Phone	10	8,850	7	6	wrist, pocket, waist, arm
	UCI	Phone	30	2,088	6	6	waist
	Motion	Phone	24	4,534	6	6	front pocket

A.2 Default configuration of MoJEPA

Table 7 summarizes the default MoJEPA configuration used throughout the paper. Unless otherwise stated, all experiments use the same pretraining and evaluation protocol, varying only the factor under study. All experiments are conducted on NVIDIA GeForce RTX 3090 GPUs using Python 3.10 and PyTorch 2.5.1.

A.3 Baselines and Evaluation Protocol

Each baseline follows its published backbone and pretext objective, with the authors' recommended hyperparameters used when available. For methods that support pretraining on our unlabeled data, pretraining is conducted on DSA and WISDM under the same data preprocessing and evaluation setting. Methods with publicly released pretrained checkpoints are evaluated using the corresponding official checkpoints as described above. All methods are evaluated with a frozen encoder followed by a linear probe. For downstream evaluation, we use channel-aware pooling when a method preserves an explicit channel or stream structure in its representation. Specifically, PatchTST, HAR-FM, and HAR-JEPA use the same channel-pool linear probe as MoJEPA. Methods that do not retain channel identity, including LIMU-BERT, TS-TCC, CrossHAR, BioBankSSL, TimeMAE, and TS-JEPA, use mean pooling over all output tokens before the linear probe. This protocol preserves explicit physical structure across sensor channels when available while avoiding additional architectural modifications to methods that do not maintain such structure.

LIMU-BERT (Xu et al., 2021). LIMU-BERT adapts BERT-style masked modeling to inertial sensor sequences by masking random spans of the multivariate series and reconstructing them in the signal domain with a Transformer encoder.

CrossHAR (Hong et al., 2024). CrossHAR is designed for cross-dataset HAR and uses a self-supervised Transformer with data augmentation and contrastive learning to learn transferable temporal representations across devices and subjects. We use its encoder with the original training objective and augmentation strategy.

Table 7: Default MoJEPA configuration and evaluation protocol.

Component	Item	Setting
Input	Channels	3-axis accelerometer, 3-axis gyroscope, C=6
	Windows	20 Hz, non-overlapping T=120 (6 s)
	Normalization	Per-channel z-score from the pretraining data
Architecture	Tokenization	Patch length P=6; T_p=20; 120 tokens per window
	Encoder	Transformer, 6 layers, D=64, 4 attention heads
	Predictor	Transformer, 6 layers, D=32, 4 attention heads
	EMA encoder	EMA momentum μ: $0.996 \rightarrow 1.0$
	Channel view	$\lvert\mathcal{C}\rvert$=4
	Temporal view	L=16
Pretraining	Source datasets	Unlabeled DSADS and WISDM (30,385 windows)
	Pretraining steps	200 epochs, batch size 128 ($\approx$ 47.4k steps)
	Optimizer	AdamW; peak learning rate 5×10^{-4} with linear warmup and cosine decay
	Weight decay	Cosine schedule from 0.04 to 0.4
Evaluation	Target datasets	PAMAP2, USC-HAD, HHAR, mhealth, Shoaib, UCI, MotionSense
	Protocol	Freeze the EMA encoder and train only a linear head on per-channel mean-pooled features.
	Linear probe optimizer	AdamW; learning rate 1×10^{-3}; weight decay 0.01; label smoothing 0.1
	Linear probe training	300 epochs; batch size 64; cosine annealing to 1×10^{-6}; select the checkpoint with the best validation accuracy
	Metrics	Accuracy and macro-F1, averaged across seven target datasets and three random seeds $\{42, 123, 456\}$

PatchTST (Nie et al., 2022). PatchTST treats each channel as an independent univariate series and applies a shared Transformer to temporal patches under masked patch reconstruction with RevIN. We follow its self-supervised setting with patch length 12, stride 12, d=128, 3 layers, and mask ratio 0.4. Because its tokens retain a channel$\times$time layout, channel-pool probes are applicable.

TimeMAE (Cheng et al., 2026). TimeMAE is a masked autoencoder for time series in which a Transformer encodes partially masked patches and a decoder reconstructs the masked signal, together with an alignment objective in the original formulation. We use the official configuration.

HAR-FM (Qiu et al., 2025). HAR-FM is a wearable foundation model with a CNN tokenizer and Transformer encoder, pretrained using an unlabeled masked-modeling stage on accelerometer and gyroscope streams. We use only the unlabeled pretraining stage and exclude the activity-text alignment component. Its representations retain the sensor-stream structure, allowing channel-pool evaluation.

TS-JEPA (Ennadir et al., 2025). TS-JEPA applies JEPA to multivariate time series by predicting EMA latents of randomly masked temporal patches with an L_1 loss. We pretrain the official encoder on DSADS and WISDM and evaluate it with a frozen mean-pool linear probe under the shared splits.

TS-TCC (Eldele et al., 2023). TS-TCC learns representations through temporal contrasting between two augmented views of each window, encouraging consistency across nearby timesteps while distinguishing negative samples. We use the published TS-TCC encoder with its default architecture and training objective.

BioBankSSL Yuan et al. (2024). BioBankSSL learns wearable representations through self-supervised training on large-scale activity data with masked and contrastive objectives. We include

it as a large-scale pretraining baseline because it is pretrained on substantially larger-scale activity data. We use the official OxWearables `harnet5` checkpoint pretrained on UK Biobank wrist accelerometer data. Since the backbone supports accelerometer data only, we retain the three accelerometer axes, resample the input from 20 to 30,Hz, and crop each window to 5,s to match the backbone input. The backbone is frozen, and a linear probe is trained on the target data.

UniMTS (Zhang et al., 2024). UniMTS aligns inertial representations with natural-language activity descriptions via CLIP-style contrastive learning. Its pretraining uses inertial sequences synthesized from motion-capture skeletons, paired with LLM-paraphrased activity text. Because these multimodal resources are unavailable in DSA and WISDM, we cannot retrain UniMTS under the same unlabeled inertial-only setting as the other baselines. We therefore use the official pretrained checkpoint without further pretraining on DSA and WISDM, freeze the ST-GCN encoder, and train a linear probe under our evaluation protocol.

MoJEPA Variants. For the ablation experiments, we evaluate MoJEPA variants under the same pretraining and evaluation protocol, changing only the factor under study:

- **Masking strategies:** *random masking*, *w/o temporal view*, and *w/o channel view*. Random masking replaces the structured channel masking and contiguous temporal-block masking with random token masking at matched ratios, while the latter two remove one of the two masking views.
- **Prediction targets**: compare *latent prediction* and *signal reconstruction* under each masking strategy, replacing the EMA-based latent prediction with raw signal reconstruction.

B Additional Experimental Results

B.1 Per-Dataset Performance

Table 8 reports the performance of all methods on each target dataset under the 5-shot, 10-shot, 5%, and 10% protocols. Under the 5% and 10% settings, MoJEPA achieves the highest performance on all seven target datasets. Under the 10-shot setting, MoJEPA leads on six datasets, with TS-TCC obtaining a slightly higher result on HHAR. The 5-shot results show greater variation across datasets: UniMTS achieves the highest performance on PAMAP2, MHealth, and Shoaib, while TS-TCC leads on HHAR. MoJEPA achieves the highest performance on the remaining three datasets, USC-HAD, UCI, and Motion, and maintains the highest unweighted average across all seven datasets. The unweighted average across the seven target datasets consistent with the results reported in Table 1.

B.2 Representation Visualization

Figure 3 visualizes the frozen channel-pool features on USC-HAD using t-SNE. From left to right, the five panels correspond to Scratch, random masking, w/o temporal view, w/o channel view, and MoJEPA with dual-view. Colors denote the eight unified activity classes. The dual-view MoJEPA representation shows more clearly separated activity clusters than the other variants, indicating that the complementary channel and temporal views improve the organization of the learned representation.

B.3 Predictor Capacity Study

Table 9 examines the effect of predictor width and depth on classification accuracy. For predictor width, a width of 32 achieves the best performance across all four label settings, while increasing the width to 64 consistently reduces performance. For predictor depth, increasing the depth from 2 to 4 improves performance across all settings, whereas further increasing it to 6 provides little additional benefit. The 4-layer predictor achieves the highest accuracy under the 5% and 10% label settings, while the 6-layer predictor performs slightly better in the few-shot settings. These results suggest that a moderate predictor capacity is sufficient for MoJEPA, with larger predictor width or depth providing limited benefits.

Table 8: Per-dataset accuracy under 5-shot, 10-shot, 5%, and 10% labeled data.

Method	PAMAP2	USC-HAD	HHAR	mhealth	Shoaib	UCI	Motion
				5-shot			
LIMU-BERT	0.194 ± 0.029	0.280 ± 0.042	0.428 ± 0.013	0.264 ± 0.021	0.329 ± 0.024	0.320 ± 0.005	0.323 ± 0.032
CrossHAR	0.255 ± 0.042	0.620 ± 0.027	0.613 ± 0.015	0.451 ± 0.032	0.435 ± 0.030	0.522 ± 0.032	0.524 ± 0.029
PatchTST	0.373 ± 0.039	0.552 ± 0.019	0.592 ± 0.011	0.584 ± 0.047	0.573 ± 0.043	0.562 ± 0.055	0.648 ± 0.008
TimeMAE	0.479 ± 0.063	0.619 ± 0.037	0.666 ± 0.018	0.517 ± 0.022	0.584 ± 0.057	0.563 ± 0.032	0.706 ± 0.034
HAR-FM	0.377 ± 0.026	0.660 ± 0.007	0.601 ± 0.056	0.674 ± 0.024	0.505 ± 0.016	0.742 ± 0.025	0.702 ± 0.020
TS-JEPA	0.455 ± 0.020	0.717 ± 0.044	0.603 ± 0.032	0.439 ± 0.059	0.629 ± 0.027	0.763 ± 0.046	0.690 ± 0.004
BioBankSSL	0.474 ± 0.045	0.649 ± 0.024	0.705 ± 0.004	0.688 ± 0.045	0.671 ± 0.022	0.579 ± 0.013	0.732 ± 0.029
TS-TCC	0.459 ± 0.019	0.747 ± 0.040	$\mathbf{0.774 \pm 0.041}$	0.568 ± 0.026	0.554 ± 0.037	0.748 ± 0.068	0.705 ± 0.018
UniMTS	$\mathbf{0.594 \pm 0.023}$	0.735 ± 0.055	0.710 ± 0.076	$\mathbf{0.814 \pm 0.029}$	$\mathbf{0.792 \pm 0.004}$	0.666 ± 0.043	0.871 ± 0.006
MoJEPA	0.586 ± 0.017	$\mathbf{0.823 \pm 0.049}$	0.720 ± 0.065	0.797 ± 0.038	0.775 ± 0.039	$\mathbf{0.871 \pm 0.029}$	$\mathbf{0.889 \pm 0.013}$
				10-shot			
LIMU-BERT	0.198 ± 0.027	0.314 ± 0.048	0.429 ± 0.019	0.342 ± 0.040	0.342 ± 0.012	0.328 ± 0.023	0.346 ± 0.032
CrossHAR	0.308 ± 0.036	0.648 ± 0.015	0.619 ± 0.034	0.486 ± 0.054	0.481 ± 0.014	0.553 ± 0.053	0.526 ± 0.043
PatchTST	0.469 ± 0.018	0.621 ± 0.028	0.630 ± 0.026	0.651 ± 0.018	0.654 ± 0.021	0.641 ± 0.033	0.711 ± 0.009
TimeMAE	0.529 ± 0.009	0.635 ± 0.042	0.667 ± 0.039	0.613 ± 0.028	0.623 ± 0.028	0.585 ± 0.026	0.719 ± 0.031
HAR-FM	0.452 ± 0.022	0.743 ± 0.006	0.690 ± 0.035	0.771 ± 0.017	0.616 ± 0.004	0.794 ± 0.033	0.754 ± 0.027
TS-JEPA	0.515 ± 0.049	0.782 ± 0.026	0.644 ± 0.062	0.460 ± 0.034	0.699 ± 0.015	0.849 ± 0.045	0.769 ± 0.045
BioBankSSL	0.548 ± 0.009	0.672 ± 0.012	0.710 ± 0.005	0.647 ± 0.033	0.710 ± 0.019	0.582 ± 0.007	0.749 ± 0.027
TS-TCC	0.556 ± 0.048	0.766 ± 0.032	$\mathbf{0.802 \pm 0.027}$	0.623 ± 0.012	0.621 ± 0.004	0.790 ± 0.035	0.734 ± 0.022
UniMTS	0.634 ± 0.010	0.779 ± 0.036	0.688 ± 0.039	0.823 ± 0.017	0.790 ± 0.024	0.664 ± 0.038	0.871 ± 0.007
MoJEPA	$\mathbf{0.668 \pm 0.006}$	$\mathbf{0.897 \pm 0.017}$	0.786 ± 0.021	$\mathbf{0.864 \pm 0.012}$	$\mathbf{0.829 \pm 0.012}$	$\mathbf{0.896 \pm 0.019}$	$\mathbf{0.919 \pm 0.031}$
				5%			
LIMU-BERT	0.253 ± 0.015	0.361 ± 0.003	0.455 ± 0.007	0.320 ± 0.007	0.376 ± 0.008	0.328 ± 0.017	0.452 ± 0.003
CrossHAR	0.258 ± 0.012	0.430 ± 0.010	0.483 ± 0.014	0.301 ± 0.011	0.496 ± 0.033	0.608 ± 0.047	0.543 ± 0.004
PatchTST	0.580 ± 0.007	0.698 ± 0.009	0.798 ± 0.012	0.637 ± 0.010	0.800 ± 0.002	0.678 ± 0.013	0.816 ± 0.010
TimeMAE	0.474 ± 0.035	0.596 ± 0.011	0.656 ± 0.007	0.585 ± 0.025	0.649 ± 0.018	0.594 ± 0.033	0.656 ± 0.007
HAR-FM	0.589 ± 0.006	0.778 ± 0.019	0.807 ± 0.003	0.811 ± 0.026	0.756 ± 0.012	0.832 ± 0.027	0.821 ± 0.005
TS-JEPA	0.629 ± 0.001	0.818 ± 0.013	0.792 ± 0.000	0.563 ± 0.064	0.838 ± 0.004	0.882 ± 0.037	0.850 ± 0.002
BioBankSSL	0.571 ± 0.003	0.668 ± 0.008	0.761 ± 0.012	0.672 ± 0.026	0.795 ± 0.001	0.574 ± 0.023	0.771 ± 0.016
TS-TCC	0.532 ± 0.003	0.780 ± 0.021	0.798 ± 0.023	0.625 ± 0.020	0.708 ± 0.017	0.814 ± 0.018	0.744 ± 0.004
UniMTS	0.646 ± 0.007	0.728 ± 0.012	0.727 ± 0.003	0.804 ± 0.023	0.787 ± 0.002	0.674 ± 0.053	0.802 ± 0.015
MoJEPA	$\mathbf{0.784 \pm 0.011}$	$\mathbf{0.932 \pm 0.016}$	$\mathbf{0.919 \pm 0.003}$	$\mathbf{0.876 \pm 0.014}$	$\mathbf{0.947 \pm 0.002}$	$\mathbf{0.917 \pm 0.011}$	$\mathbf{0.946 \pm 0.018}$
				10%			
LIMU-BERT	0.265 ± 0.013	0.359 ± 0.002	0.455 ± 0.005	0.331 ± 0.004	0.390 ± 0.008	0.344 ± 0.017	0.457 ± 0.005
CrossHAR	0.256 ± 0.011	0.431 ± 0.010	0.486 ± 0.014	0.282 ± 0.042	0.506 ± 0.026	0.623 ± 0.021	0.546 ± 0.006
PatchTST	0.610 ± 0.006	0.743 ± 0.005	0.824 ± 0.010	0.692 ± 0.012	0.822 ± 0.004	0.736 ± 0.013	0.859 ± 0.004
TimeMAE	0.473 ± 0.025	0.594 ± 0.012	0.658 ± 0.002	0.600 ± 0.023	0.648 ± 0.020	0.624 ± 0.047	0.659 ± 0.014
HAR-FM	0.608 ± 0.008	0.796 ± 0.017	0.815 ± 0.009	0.848 ± 0.008	0.765 ± 0.009	0.862 ± 0.019	0.839 ± 0.006
TS-JEPA	0.647 ± 0.004	0.842 ± 0.024	0.823 ± 0.005	0.674 ± 0.015	0.864 ± 0.005	0.908 ± 0.010	0.880 ± 0.009
BioBankSSL	0.598 ± 0.010	0.712 ± 0.012	0.787 ± 0.010	0.756 ± 0.003	0.819 ± 0.002	0.598 ± 0.021	0.809 ± 0.012
TS-TCC	0.547 ± 0.004	0.774 ± 0.022	0.798 ± 0.022	0.652 ± 0.027	0.711 ± 0.018	0.840 ± 0.011	0.747 ± 0.009
UniMTS	0.654 ± 0.006	0.743 ± 0.006	0.725 ± 0.003	0.810 ± 0.023	0.789 ± 0.002	0.686 ± 0.067	0.809 ± 0.005
MoJEPA	$\mathbf{0.816 \pm 0.010}$	$\mathbf{0.958 \pm 0.003}$	$\mathbf{0.949 \pm 0.002}$	$\mathbf{0.908 \pm 0.010}$	$\mathbf{0.963 \pm 0.001}$	$\mathbf{0.940 \pm 0.002}$	$\mathbf{0.966 \pm 0.006}$

B.4 Hyperparameter Sensitivity

Pretraining steps. Figure 4a shows the effect of pretraining steps. Few-shot accuracy is highest at 200 epochs (≈ 47.4k steps) and decreases slightly with longer training, while the 5% and 10% settings show a slight improvement. The performance differences across 100–400 epochs are small, so we use 200 epochs as the default setting.

Pretraining learning rate. Figure 4b shows the effect of the pretraining learning rate. Pretraining uses a peak learning rate of 5×10^{-4} with linear warmup for the first 10% of training steps, followed by cosine decay. Few-shot accuracy is highest at 5×10^{-4}, whereas the 5% and 10% settings

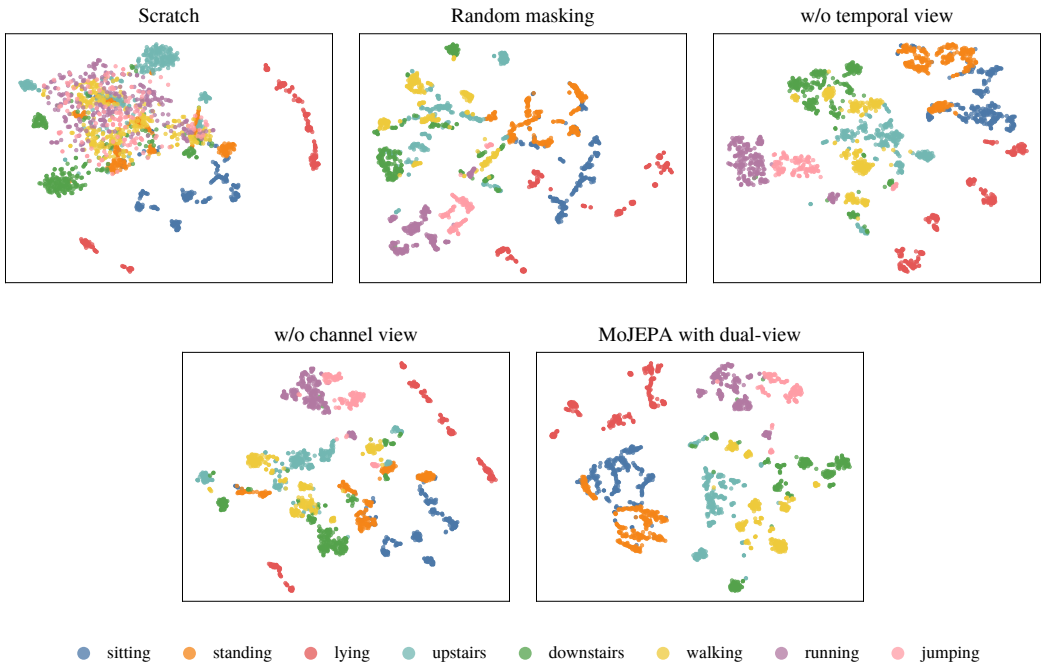
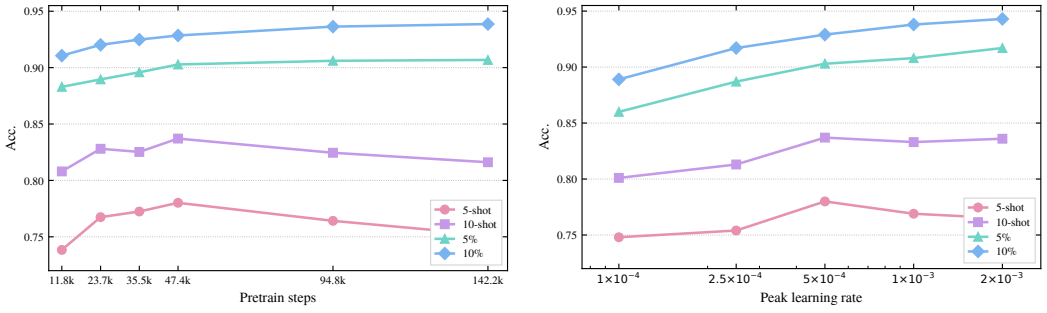


Figure 3: t-SNE visualization of frozen channel-pool features on USC-HAD.

Table 9: Effect of predictor width and depth (accuracy).

Setting	Params	5-shot	10-shot	5%	10%
		Predictor width			
16	0.324M	0.744 ± 0.011	0.808 ± 0.003	0.886 ± 0.003	0.919 ± 0.004
32	0.383M	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.929 \pm 0.002}$
64	0.612M	0.726 ± 0.009	0.797 ± 0.007	0.886 ± 0.004	0.917 ± 0.005
		Predictor depth			
2	0.333M	0.711 ± 0.010	0.771 ± 0.003	0.848 ± 0.003	0.878 ± 0.001
3	0.345M	0.745 ± 0.001	0.815 ± 0.012	0.894 ± 0.004	0.923 ± 0.004
4	0.358M	0.770 ± 0.002	0.823 ± 0.003	$\mathbf{0.905 \pm 0.004}$	$\mathbf{0.931 \pm 0.003}$
6	0.383M	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	0.903 ± 0.004	0.929 ± 0.002



(a) Varying pretraining steps.

(b) Varying peak learning rate.

Figure 4: Performance with different pretraining steps and peak learning rates.

show slightly higher accuracy at larger peak learning rates. Across the tested learning rates, the performance differences are small. We use 5×10^{-4} as the default setting.

Table 10: Performance with different pretraining batch size.

Batch size	Metric	5-shot	10-shot	5%	10%
128	Acc.	$\mathbf{0.780 \pm 0.006}$	$\mathbf{0.837 \pm 0.006}$	$\mathbf{0.903 \pm 0.004}$	$\mathbf{0.929 \pm 0.002}$
	F1	$\mathbf{0.767 \pm 0.008}$	$\mathbf{0.827 \pm 0.006}$	$\mathbf{0.886 \pm 0.008}$	$\mathbf{0.920 \pm 0.005}$
256	Acc.	0.754 ± 0.010	0.810 ± 0.007	0.894 ± 0.002	0.921 ± 0.003
	F1	0.739 ± 0.011	0.799 ± 0.010	0.876 ± 0.002	0.911 ± 0.003
512	Acc.	0.737 ± 0.006	0.805 ± 0.012	0.881 ± 0.003	0.918 ± 0.003
	F1	0.718 ± 0.006	0.792 ± 0.016	0.863 ± 0.004	0.909 ± 0.003

Batch size. Table 10 varies the pretraining batch size in $\{128, 256, 512\}$. Accuracy falls as the batch grows, with the largest gap in few-shot (≈ 4 points vs. 512). We keep batch size 128 as the default.