MORPHCL: MORPHOLOGICAL CONTRASTIVE LEARNING FOR INERTIAL-BASED HUMAN ACTIVITY RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the ubiquity of sensors in wearable and mobile devices and the abundance of human movement data they generate, translating unlabeled recordings into foundational motion models remains an open challenge. Self-supervised learning (SSL) has alleviated the need for costly annotations, yet existing approaches leave the global structure of large-scale motion data largely untapped, relying on randomly sampled batches and local comparisons that become particularly problematic for in-the-wild inertial data dominated by stationary, low-variance behaviors. Here we introduce *Morphological Contrastive Learning (MorphCL)*, a self-supervised pretraining framework that uses structure-aware grouping to inject explicit modeling of *global structure* into inertial-based SSL approaches. Building on two well-established pillars of motion analysis, the discovery of motion primitives, or *motifs*, and domain-specific feature descriptors, we show that MorphCL substantially improves linear probing and finetuning results of learned encoders by up to 15 percentage points in F1-score. In a comparison with existing foundation models, we demonstrate that MorphCL-pretrained encoders match or surpass them models in linear probing performance while trained on $4600\times$ less data. Qualitative analysis of the resulting embedding spaces further reveals morphologically meaningful cluster structure, with improved separation of kinematically similar activity classes. Code and model weights are publicly available via: `https://anonymous.4open.science/r/mcl-AC4B`.

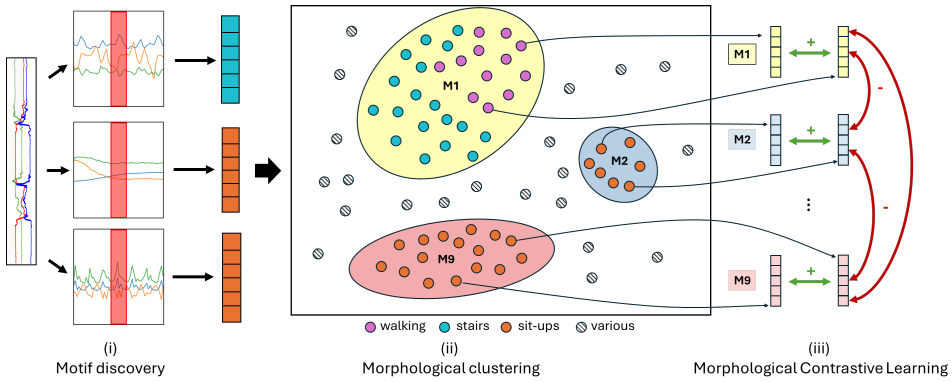

Figure 1: **Morphological Contrastive Learning (MorphCL)** explicitly models global motion structure by (i) discovering recurring motifs (red) from raw accelerometer data, (ii) clustering them into morphological groups across the pretraining corpus, (iii) defining a group-based contrastive objective that drives agreement beyond instance-level comparisons.

1 INTRODUCTION

Activities and movements performed during daily life encode rich information about an individual's behavioral patterns, physiological state, and habitual routines (Tryon, 2006). The automatic identi-

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.

fication of such behaviors, commonly referred to as Human Activity Recognition (HAR), has therefore attracted growing interest across a range of health-related research fields (Patterson et al., 2005; Ward et al., 2006). Inertial Measurement Units (IMUs), now ubiquitous in consumer wearables and mobile devices, have emerged as a particularly practical sensing modality for this purpose, yielding discriminative representations well-suited to activity recognition systems (Bulling et al., 2014). However, while other domains such as computer vision have seen the rise of large-scale foundation models, which can generalize easily to unseen tasks being trained on vast amounts of data, inertial-based HAR research remains largely confined to small-scale benchmark studies. This gap is driven in part by the difficulty of annotating raw IMU signals retrospectively without additional context, making the construction of labeled datasets (Reiss & Stricker, 2012; Roggen et al., 2010) costly and predominantly restricted to controlled laboratory environments. Although such datasets have enabled verification of methodological progress on relatively small neural networks (Ordóñez & Roggen, 2016), they lack the scale and diversity required to train modern architectures and fail to capture the variability of real-world human behavior.

The introduction of self-supervised learning (SSL) to inertial HAR has presented a paradigm shift. By no longer requiring annotations of human activities, it has unlocked the use of the large corpora of unlabeled data, generated by everyday wearables. While cleverly crafted pretext tasks such as contrasting differently augmented views of input signals have shown to perform well on various downstream tasks (Tang et al., 2021), such methods typically rely on instance-level discrimination, treating each segment as independent. However, this assumption is often violated in IMU data, where recurring motion primitives are shared across activities and subjects, making instance-level objectives poorly aligned with the underlying structure (Xu et al., 2024).

Recent work relaxes this constraint by defining positives beyond simple augmentations. Methods like REBAR and RelCon (Xu et al., 2024; 2025) retrieve similar segments across a timeseries and introduce ranking-based objectives. While improving pair selection, they do not use information about dataset composition and still rely on randomly sampled batches. This becomes particularly problematic in large-scale, in-the-wild datasets, which are highly imbalanced and dominated by repetitive, low-variance activities. For example, in the largest open dataset CAPTURE-24 (Chan et al., 2024), approximately 75% of the data corresponds to stationary behaviors, while more informative activities are rare. As a result, randomly sampled batches are likely to contain many near-identical segments, sustaining the false negative problem (Chuang et al., 2020) at the batch level even with well-chosen pairs. This can lead to weak training signals and limits contrastive objectives from learning discriminative representations (as shown in Section 5).

Based on this observation, we argue that a key missing component in current SSL approaches for HAR is the explicit modeling of *global structure* in motion data. Human movement exhibits recurring patterns, such as periodic walking cycles or characteristic transitions, that appear across time, activities, and individuals (Minnen et al., 2006). The automatic discovery of these *motifs* has long served as a fundamental concept in motion analysis, acting as atomic building blocks for recognizing complex activities (Minnen et al., 2006; Hamid et al., 2007). We therefore introduce *Morphological Contrastive Learning (MorphCL)*, a self-supervised pretraining framework that can serve as a plug-in solution to inject explicit modeling of *global structure* into existing SSL approaches (see Figure 1). By combining motif discovery with domain-specific feature descriptors, MorphCL constructs a morphological clustering of the pretraining corpus and uses the resulting grouping to define a group-based contrastive objective that encourages agreement between segments sharing similar motion patterns. Our contributions are three-fold:

- We propose a novel morphological clustering system for accelerometer data, which uses a combination of motif-discovery and domain-specific precomputed features to extract morphological structures within an unlabelled pretraining corpus.
- Using the extracted clusters, we devise Morphological Contrastive Learning (MorphCL), a group-based contrastive objective that optimizes agreement between windows with similar motion morphology, moving beyond instance-level comparisons.
- We demonstrate that including explicit modeling of morphological structure can improve downstream results if combined with existing SSL frameworks (up to 15% in F1-score) and yields representations competitive with a foundation model that was trained on $4600\times$ more data (Yuan et al., 2024), suggesting that domain-informed inductive bias can complement large scale training.

2 Related Work

Inertial-based Contrastive Learning. Contrastive learning has become a prominent SSL approach for addressing the limited availability of annotated data in Human Activity Recognition (Haresamudram et al., 2022; Tang et al., 2021). Methods such as SimCLR (Tang et al., 2021) follow an *instance-discrimination* setup, where positive pairs are created by augmenting the same temporal window (e.g., jittering, scaling, permutation) and all other samples in the batch are treated as negatives. While effective, this formulation assumes that augmentations sufficiently capture semantic similarity, which is often not the case for complex and noisy motion signals (Xu et al., 2024). To overcome the rigidity of augmentation-defined positives, recent work has explored more flexible similarity measures. REBAR (Xu et al., 2024) shifts away from purely temporal or augmentation-defined positives by using a learned reconstruction error as a proxy for semantic similarity, and RelCon (Xu et al., 2025) introduces a relative ranking loss to preserve hierarchical relationships between motions, avoiding the limitations of rigid positive-negative binaries. However, these approaches remain *instance-centric* and depend on *randomly sampled batches*, without explicitly modeling global structure in motion data. In contrast, our work introduces a structure-aware, group-based objective that captures recurring morphological patterns across large-scale, in-the-wild datasets.

Group-based Contrastive Clustering. Several studies have shown that incorporating group-level structure can improve contrastive learning. Supervised Contrastive Learning (SupCon) (Khosla et al., 2020) extends the contrastive framework by defining positives as all samples sharing the same label, leading to more stable and semantically meaningful representations. A parallel line of work jointly optimizes cluster assignments and representations (Asano et al., 2020; Van Gansbeke et al., 2020; Caron et al., 2020). These methods were developed primarily for vision, where grouping is typically inferred from raw inputs or learned embeddings. In sensor-based HAR, however, domain features are rich, physically grounded, and known to be discriminative across a wide range of activities (Van Der Donckt et al., 2023). Closest to our approach, PaPaGei (Pillai et al., 2025) introduced a domain-specific proxy (sVRI) to define groups of related samples without explicit labels in the context of Photoplethysmogram (PPG) signals. MorphCL extends this intuition to inertial data, where the higher variability and multi-axis dynamics prevent the use of a single scalar proxy, instead combining motif discovery with a broader domain feature set to construct morphological groups. MorphCL decouples clustering from representation learning entirely, providing a plug-in solution that is semantically grounded and complements rather than replaces existing instance-level objectives. To our knowledge, no prior SSL framework for inertial data has explicitly leveraged such structure-aware grouping to inject domain knowledge into the contrastive objective.

Foundation Models for Motion Data. Recent efforts have begun scaling SSL approaches to large wearable datasets to train foundation models for motion data. Narayanswamy et al. (2025) and Zhang et al. (2025) train wearable foundation models using minute-level handcrafted features instead of raw signals, which, however, makes them less applicable for temporally precise motion tasks such as HAR. Yuan et al. (2024) use 700k person-days of data from UK Biobank to train an inertial representation model. SlotFM (Park et al., 2025) introduces a slot-based attention mechanism to decompose motion signals into disentangled latent components. Among these models, only Yuan et al. (2024) open-source model weights so we include a comparison with the model.

3 Methodology

We propose Morphological Contrastive Learning (MorphCL), a self-supervised pretraining framework for inertial time series that injects dataset-level global structure into existing SSL approaches. Building on two well-established pillars of motion analysis, MorphCL constructs morphology-aware pseudo-labels from unlabeled motion data and uses these labels to define positives beyond standard augmentation pairs. As shown in Figure 1, MorphCL consists of three stages: (i) motif discovery, (ii) clustering of domain-specific features, and (iii) a group-based contrastive objective. We describe each component below.

3.1 Motif-based Pre-filtering

Let

$$\mathbf{T} = \{\mathbf{x}_t\}_{t=1}^{N}, \quad \mathbf{x}_t \in \mathbb{R}^D \tag{1}$$

denote a multivariate inertial time series of length N with D sensor axes. Prior work in activity recognition has long viewed complex activities as compositions of shorter repeated patterns, or *motifs* (Minnen et al., 2006; Hamid et al., 2007). We thus use motif discovery to identify prominent repeated patterns ("motion primitives") within $\mathbf{T}$ and extract their occurrences. Since activities are typically expressed through coordinated motion across axes (Yeh et al., 2017), we perform motif discovery using *multi-dimensional matrix profiling* as proposed by Yeh et al. (2017) rather than analyzing each channel independently. Since the full time series can be very large, we perform motif discovery independently on temporally contiguous chunks, identifying up to m distinct motif patterns per chunk and extracting r instances of each which do not exceed a pre-defined motif distance threshold σ, as measured by the matrix profiling. This way we identify subsequences that are simultaneously similar across a subset of sensor axes. Specifically, we require each motif to be supported by at least two of the three accelerometer axes.

Since different motifs may occur with highly varying frequencies, naïvely selecting r reoccurences of motifs can result in many near-duplicate patterns originating from a single, frequently occurring activity. To promote diversity among the discovered motifs, we adopt a greedy masking strategy, masking all reoccurrences of a motif that do not exceed σ within the current chunk before discovering the next most prominent motif. This prevents repeatedly selecting highly similar instances of the same pattern and encourages the discovery of motifs corresponding to a broader range of activities and motion primitives.

Let

$$\mathcal{I} = \{i_1, \ldots, i_M\} \subset \{1, \ldots, N\} \tag{2}$$

denote the center indices of the M motif occurrences collected across all chunks.

For each $i \in \mathcal{I}$, we extract a *motif window* $\mathbf{w}_i = \{x_{i-W/2}, x_{i+1-W/2} \ldots, x_{i+W/2}\}^T$ and a *context window* $\mathbf{c}_i = \{x_{i-C/2}, x_{i+1-C/2} \ldots, x_{i+C/2}\}^T$, given a predetermined window size W and context window size C ($C \geq W$). To avoid redundancy across motifs, we retain only motif windows that do not overlap more than 50% with any other motif window. The union of all retained motif windows ($\mathcal{W} = \{\mathbf{w}_i \mid i \in \mathcal{I}\}$), and context windows ($\mathcal{C} = \{\mathbf{c}_i \mid i \in \mathcal{I}\}$), forms the input to the subsequent morphological clustering stage.

3.2 Morphological Clustering

Having identified repeated patterns in $\mathbf{T}$, we next group these patterns according to their morphology. Unlike physiological signals such as Photoplethysmogram (PPG), which can often be summarized using a small number of domain-specific proxy measures (e.g., sVRI (Pillai et al., 2025)), accelerometer data exhibit highly variable, multi-axis dynamics that cannot be captured by a single descriptor (Van Der Donckt et al., 2023). We therefore represent each context window $\mathbf{c}_i \in \mathcal{C}$ as a feature vector $\mathbf{z}_i = \Phi(\mathbf{c}_i) \in \mathbb{R}^F$ where $\Phi(\cdot)$ concatenates statistical features commonly used in classical machine-learning-based activity recognition. Specifically, we reuse the feature set described by (Van Der Donckt et al., 2023) as it has shown to provide strong discriminative capabilities in describing numerous activities in inertial-based data reliably for downstream classification purposes (Van Der Donckt et al., 2023; 2024b;a).

Since for example flat-line windows may lead to undefined feature values, we further postprocess $\Phi(\mathbf{c}_i)$ by replacing undefined values with 0 or an appropriate upper/lower bound. The details are in the Appendix. Next, we cluster the set of extracted morphological feature vectors $\mathcal{Z} = \{\mathbf{z}_i \mid i \in \mathcal{I}\}$ using HDBSCAN (McInnes et al., 2017), a density-based clustering algorithm well suited for clusters of varying size and the presence of outliers. Formally, HDBSCAN induces a mapping

$$\kappa : \mathbb{R}^F \to \{1, \ldots, K\} \cup \{\varnothing\} \tag{3}$$

where K denotes the number of morphological clusters discovered in $\mathcal{Z}$, and $\varnothing$ indicates outliers rejected by the algorithm. Importantly, the number of clusters K is not fixed a priori but is determined automatically by HDBSCAN based on the density structure of the data and the chosen support parameter ρ. The resulting motif-filtered and morphologically labeled dataset (excluding outliers)

$$\mathcal{D} = \{(\mathbf{w}_i, y_i) \mid y_i \neq \varnothing, i \in \mathcal{I}\}, \tag{4}$$

with $y_i = \kappa(\phi(\mathbf{c}_i))$, thus represents a collection of motif windows, each assigned to one of the K morphological groups, defining which windows are treated as positives (same group) and negatives

(different groups) in our pretraining corpus. A sensitivity analysis on chosen clustering parameters can be found in the Appendix.

3.3 Morphological Contrastive Learning (MorphCL)

With morphological pseudo-labels at hand, we can now define a contrastive objective, Morphological Contrastive Loss (MorphCL), that exploits dataset-level group structure. Given a batch $\mathcal{B} \subset \mathcal{D}$ of motif windows, we encode each motif window in $\mathcal{B}$ using a feature encoder (f_θ) as well as a projection head (g_θ) resulting in latent vectors $\mathbf{u}_i \in \mathbb{R}^E$. Note that, as in Khosla et al. (2020), we extend $\mathcal{B}$ by extending it with an augmented version of each window $\mathbf{w}_i$, randomly drawn from a set of augmentations (noise, scaling, rotation, and negation). These augmentations are a subset of those proposed by Tang et al. (2021), selected because they preserve our definition of morphology.

For a given window $\mathbf{w}_i$ and its corresponding morphological label y_i, we define the set of windows considered to be positive and negative instances of $\mathbf{w}_i$ as

$$\mathcal{P}(i) = \{p \in \mathcal{B} \setminus \{i\} \mid y_p = y_i\}, \qquad \mathcal{N}(i) = \{p \in \mathcal{B} \setminus \{i\} \mid y_p \neq y_i\}. \tag{5}$$

That is all windows in $\mathcal{B}$ that share (or do not share) the same morphological label as $\mathbf{w}_i$.

Based on our definition of $\mathcal{P}$ and $\mathcal{N}$, we define our MorphCL loss as a group-based variant of the Supervised Contrastive Loss (SCL) (Khosla et al., 2020)

$$\mathcal{L}_{\text{morph}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} -\log \left(\frac{\sum_{p \in \mathcal{P}(i)} \exp\left(S_{ip}\right)}{\sum_{j \in \mathcal{B} \setminus \{i\}} \exp\left(S_{ij}\right)} \right) \tag{6}$$

where $\exp$ is the exponential and $S_{ij} = \frac{\text{sim}(\mathbf{u}_i, \mathbf{u}_j)}{\tau}$ is the pairwise similarity matrix of embeddings $\mathbf{u}_i$ and $\mathbf{u}_j$ using the cosine similarity function $\text{sim}(\cdot)$ and a temperature value τ. Unlike SCL, which averages log-probabilities computed for each positive pair individually, our formulation thus adopts a group-based perspective. Specifically, SCL treats each positive pair (i, p) independently by normalizing over all other samples in the batch, resulting in a sum over positive pairs outside the logarithm, while MorphCL encourages agreement at the level of morphological groups rather than isolated pairs. An ablation study, comparing MorphCL with SCL as well as a group-based formulation of the NTXentLoss as employed in Pillai et al. (2025) can be found in the Appendix.

4 Experiments

4.1 Pretraining

Model. We assess a wide-range of SSL pretraining pretext tasks (Tang et al., 2021; Saeed et al., 2019; Haresamudram et al., 2020; 2019; Xu et al., 2024; 2025) both in combination with and without our proposed MorphCL framework. As encoder networks we use the DeepConvLSTM (Ordóñez & Roggen, 2016) as well as a repurposed ViT transformer (Kolesnikov et al., 2021) as recently proposed in Hoddes et al. (2025). Specifications of both architectures can be found in the Appendix.

Datasets. As pretraining corpora we use the WISDM (Kwapisz et al., 2011) and CAPTURE-24 (Chan et al., 2024) dataset. The WISDM dataset provides accelerometer and gyroscope data collected by a smartphone and smartwatch with a sampling rate of 20Hz of 51 participants doing various activities within an hour-long recording protocol. The CAPTURE-24 dataset provides 24-hour long, in-the-wild accelerometer recordings of a smartwatch sampling at 100Hz of around 150 participants. To ensure comparability among the two corpora, we only use the smartwatch data of the WISDM dataset and resample both datasets to a common sampling rate of 50Hz. We preprocess both the WISDM and CAPTURE-24 dataset by shifting a sliding window of size 128 samples (2.56s) over each participant's data stream without any overlap among the windows. Each pretraining corpus is normalized using the global mean and standard deviation across all windows of the train dataset. Further details on hyperparameters and optimization schemes of each pretraining framework are provided in the Appendix.

Training. We apply MorphCL on a subset of the pretraining corpus, $\mathcal{D}$. The relative size of the motif- and morphological subsets of the WISDM and CAPTURE-24 dataset created by applying

Table 1: **Downstream datasets.** #Sbjs: participants; #Cls: activities (+1 indicates dataset provides a NULL-class). Recording duration is averaged across participants. Activities are categorized into periodic (recurring, periodic movements), non-periodic (sporadic, often short movements/gestures) and complex activities (sequential combinations of periodic and non-periodic movements).

Dataset	Scenario	Activity Types	#Sbjs	#Cls	Avg. Recording (min)
Bock et al. (2024)	outdoor sports workouts	periodic, complex	22	18(+1)	52.52 ($\pm$ 17.89)
Scholl et al. (2015)	wetlab DNA extraction	(non-)periodic	22	8(+1)	47.93 ($\pm$7.48)
Sztyler & Stuckenschmidt (2016)	locomotion	periodic	13	8	75.84 ($\pm$4.55)
Hoelzemann et al. (2023)	basketball practice and game	(non-)periodic	24	5(+1)	60.59 ($\pm$10.84)
Reiss & Stricker (2012)	locomotion and ADL	periodic, complex	9	18(+1)	50.46 ($\pm$17.79)

the MorphCL framework are summarized in the Appendix. Unlike other frameworks which draw windows randomly from the entire pretraining corpus or available data of a participant, our framework creates batches by evenly sampling from the K morphological clusters. As cluster sizes differ, we allow windows of smaller clusters to be sampled multiple times during an epoch such that each window within the largest morphological cluster can be sampled exactly once. Similarly, we allow batches during MorphCL training to be sampled multiple times, such that the tested SSL frameworks, which use the full training data, can sample each window exactly once during each epoch. Ablation studies on these design choices can be found in the Appendix.

4.2 Downstream Tasks

We chose a total of five popular HAR benchmark datasets for our downstream evaluation (see Table 1). To make sensor information transferable from our pretraining corpora, we chose only the (right) wrist sensor provided in each of the datasets. We further adjust the sampling rate of each dataset to be 50Hz by resampling accordingly. Unlike existing downstream evaluations (Yuan et al., 2024), we include more complex activity recognition tasks such as sports (Bock et al., 2024; Hoelzemann et al., 2023) and process-related tasks (Scholl et al., 2015), and do not artificially remove background activity classes (NULL-class) of the downstream datasets, as the accurate differentiation between this class and relevant activities is a necessary trait of activity recognition systems for real-world deployment (Bulling et al., 2014).

We evaluate both linear probing as well as finetuning capabilities of our pretrained backbones. Details on the chosen hyperparameters and optimization parameters can be found in the Appendix. All results mentioned in this paper are reported as fold-averaged Leave-One-Subject-Out (LOSO) cross-validation results, where each participant in the downstream dataset becomes the test set exactly once, and seed-averaged across three runs.

5 Results

5.1 Complementarity to Existing SSL Methods

To investigate whether MorphCL can serve as a plug-in solution to inject explicit modeling of global structure in motion data, we evaluate models for both linear probing and finetuning combining loss terms of existing frameworks with our proposed MorphCL loss term as an unweighted sum. Ablation experiments using different weighting schemes can be found in the Appendix. Results in Table 2 and 3 show that by making SSL training aware of the morphological similarity through the additional MorphCL loss term, results consistently improve across nearly all methods with linear probing improvements ranging up to as much as 15% in F1-score.

Notably, REBAR and RelCon, two recent contrastive learning methods that explicitly address positive pair selection in inertial HAR, benefit most substantially from the addition of MorphCL, suggesting that while improved pair retrieval is valuable, it does not fully compensate for the lack of global structural awareness in the embedding space. Similarly, MTL, used to train foundation models for HAR as introduced by Yuan et al. (2024), benefits from the structure-aware contrastive loss by as much as 5% when pretraining on CAPTURE-24. SimCLR benefits from MorphCL on the evaluated subsets (see Section 5.2), while no improvement is observed under full-dataset pretraining. We hypothesize that this partly reflects conflicting pair assignments caused by SimCLR's negative pair

Table 2: Linear probing macro F1 (%), averaged over the five downstream datasets. Standard deviations across seeded runs appear below the means. **Bold**: higher mean per method and encoder/pretraining setting. Dataset-level results are provided in Appendix B.

	DeepConvLSTM WISDM			ViT-S WISDM			ViT-S CAPTURE-24		
Method	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ
MTL	38.31 ±0.06	**48.59** ±0.25	+10.28	47.41 ±0.22	**50.53** ±0.45	+3.13	47.29 ±0.12	**52.12** ±0.33	+4.83
SimCLR	46.27 ±0.09	**47.85** ±0.26	+1.58	48.78 ±0.57	**50.82** ±0.50	+2.04	**51.36** ±0.20	50.97 ±0.23	−0.39
Masked AE	–	–	–	44.28 ±0.87	**50.57** ±0.50	+6.29	47.56 ±0.38	**50.05** ±0.37	+2.49
Denoising AE	–	–	–	47.73 ±0.49	**49.32** ±0.33	+1.59	48.27 ±0.23	**51.13** ±0.33	+2.86
REBAR	37.81 ±0.19	**46.43** ±0.17	+8.62	41.48 ±0.68	**49.81** ±0.39	+8.32	37.09 ±0.13	**48.27** ±0.49	+11.18
RelCon	35.19 ±0.36	**45.88** ±0.20	+10.69	34.61 ±0.33	**49.96** ±0.40	+15.36	40.52 ±0.98	**49.93** ±0.30	+9.41

Table 3: Finetuning macro F1 (%), averaged over the five downstream datasets (mean ± standard deviation across seeded runs). **Bold**: higher mean per method and encoder/pretraining setting. Dataset-level results are provided in Appendix B.

	DeepConvLSTM WISDM			ViT-S WISDM			ViT-S CAPTURE-24		
Method	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ
MTL	54.18 ±0.44	**58.07** ±0.22	+3.89	57.51 ±0.27	**58.23** ±0.16	+0.71	60.26 ±0.15	**60.68** ±0.11	+0.42
SimCLR	57.32 ±0.31	**57.57** ±0.28	+0.24	55.28 ±0.25	**56.25** ±0.19	+0.97	58.22 ±0.39	**58.72** ±0.19	+0.51
Masked AE	–	–	–	54.36 ±0.32	**56.18** ±0.16	+1.81	55.61 ±0.32	**57.98** ±0.13	+2.37
Denoising AE	–	–	–	53.74 ±0.24	**55.27** ±0.13	+1.53	55.31 ±0.29	**58.35** ±0.25	+3.04
REBAR	53.23 ±0.29	**57.01** ±0.04	+3.78	52.95 ±0.21	**54.83** ±0.24	+1.88	54.05 ±0.13	**58.12** ±0.16	+4.07
RelCon	52.82 ±0.04	**56.18** ±0.40	+3.35	53.43 ±0.30	**53.95** ±0.15	+0.52	53.50 ±0.40	**56.66** ±0.42	+3.16

definition, whereby all in-batch instances are treated as negatives. Detailed results of all experiments are provided in the Appendix, including using only MorphCL as loss.

5.2 Ablation Experiments

Subset evaluation. To assess the contributions of data filtering and auxiliary MorphCL training, we train the base SSL objectives on motif-filtered and morphological subsets with and without MorphCL. The motif-filtered subset consists of all motif windows that were identified to contain a motif as per the motif-discovery. Morphological subset consists of all windows that are part of a morphological cluster and use the same balanced sampler as MorphCL. Table 4 reports results for MTL, SimCLR and Masked AE pretrained on WISDM. Adding MorphCL increases results compared to the based objective. Complete results are provided in Appendix Tables 14 and 15.

Hyperparameter sensitivity. We independently vary the motif-discovery parameters (σ, r) and clustering parameters (ρ, support) on WISDM and test linear probing results on our five chosen downstream tasks (see Figure 2; see Appendix B for full results and additional analysis). While our chosen default configurations show highest results, our implementation remains stable across various setups. At $\rho = 400$, all samples are classified as noise, preventing downstream evaluation.

Table 4: Subset pretraining on WISDM (linear probing macro F1 (%), averaged over five downstream datasets; mean ± standard deviation across seeded runs). + MorphCL adds the auxiliary MorphCL loss computed on the Morph subset. **Bold**: higher mean per SSL-subset pair.

	MTL			SimCLR			Masked AE		
Subset	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ	Base	+ MorphCL	Δ
Full	47.41 ±0.22	**50.53** ±0.45	+3.13	48.78 ±0.57	**50.82** ±0.50	+2.04	44.28 ±0.87	**50.57** ±0.50	+6.29
Motif	46.13 ±0.36	**49.70** ±0.79	+3.57	47.75 ±0.62	**50.53** ±0.69	+2.78	43.56 ±0.66	**50.06** ±0.47	+6.50
Morph	46.86 ±0.36	**49.42** ±0.49	+2.56	47.88 ±0.62	**49.80** ±0.50	+1.92	42.42 ±0.87	**50.12** ±0.88	+7.70

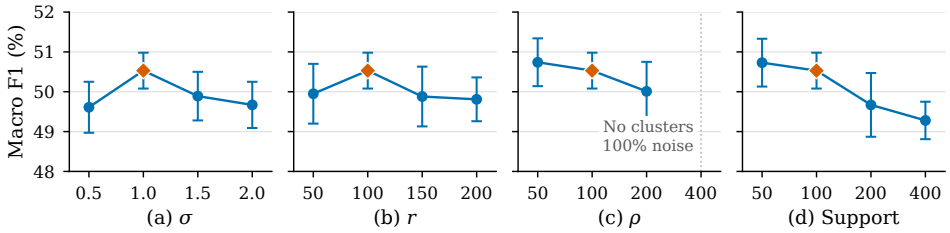

Figure 2: Sensitivity of MTL + MorphCL on WISDM. Mean linear-probing macro F1 (%) over five downstream datasets. Diamonds mark defaults: $\sigma = 1$, $r = \rho = \text{support} = 100$.

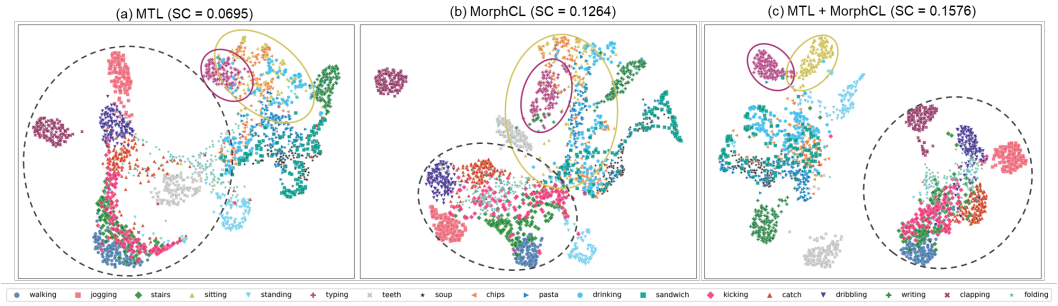

Figure 3: UMAP (McInnes et al., 2020) embeddings of a WISDM participant from ViT-S models pretrained on CAPTURE-24 using (i) MTL, (ii) MorphCL and (iii) MTL + MorphCL. SC: silhouette coefficient ($[-1, 1]$; higher is better). On unseen data, MTL yields entangled clusters, while MorphCL captures global structure. Their combination improves cluster compactness and separation (higher SC), distinguishing sitting from typing (yellow/purple circles) while preserving coherent manifolds among vigorous activities (dashed circle).

5.3 Qualitative Results

To better illustrate what MorphCL learns, Figure 3 shows how representations pretrained on CAPTURE-24 transfer to WISDM for a sample participant, visualized against activity annotations unseen during training. In (a) MTL, some coarse grouping exists, but the embedding space remains largely entangled: kinematically similar activities, e.g., typing vs. sitting (yellow and purple circles), overlap substantially. In (b) MorphCL, the global structure becomes more coherent, with samples organized into smoother manifolds that reflect morphological consistency, e.g. vigorous activities (dashed circle) separate more clearly from seated activities, yet class boundaries remain diffuse, and activities sharing similar motion patterns occupy contiguous, poorly separated regions. The combined objective ((c) MorphCL + MTL) yields both structure and separability: clusters are compact and well-delineated, with clearer boundaries even between kinematically similar actions.

This qualitative trend is further supported quantitatively by silhouette coefficients (Rousseeuw, 1987) provided with the plot and computed across all downstream datasets (full results in appendix), which consistently increase with MorphCL. Linear probing results confirm this further (see Appendix B.7 for confusion matrices) showing that residual misclassifications concentrate among semantically related activities rather than being scattered indiscriminately.

5.4 Comparison to Existing Accelerometer Foundation Models

While several efforts have been made in other communities advancing towards foundational models which can be used to extract discriminative features, open-sourced inertial-based foundation models remain sparse. Most existing approaches are closed-source due to data privacy constraints (Xu et al., 2025; Park et al., 2025; Narayanswamy et al., 2025; Zhang et al., 2025) and/or are applied using window sizes of several minutes (Logacjov et al., 2024), making them not suited for accelerometer-based in-situ activity recognition. To our knowledge, the only publicly available model for this

Table 5: Comparison with pretrained models from Yuan et al. (2024). Macro F1 (%; mean ± standard deviation across seeded runs) is reported for linear probing and finetuning.

	Hours of Data	Model Size	WEAR		Wetlab		PAMAP2		Hangtime		RWHAR		*Avg.*	
			LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine
Yuan et al. (2024) - MTL														
ResNet18	3.7K	10.4M	33.77 ±0.24	77.68 ±0.56	18.53 ±0.35	41.38 ±0.37	54.15 ±0.68	75.01 ±2.76	24.54 ±0.32	29.75 ±0.14	50.22 ±0.61	79.21 ±0.46	36.24 ±0.18	60.61 ±0.60
ResNet18	16.8M	10.4M	71.45 ±0.16	83.86 ±0.41	35.00 ±0.49	48.13 ±0.42	77.06 ±0.62	84.00 ±1.38	29.80 ±0.22	29.68 ±0.07	76.70 ±0.60	86.30 ±0.63	58.00 ±0.05	66.39 ±0.39
Ours - MTL + MorphCL														
ResNet18	3.7K	10.4M	72.40 ±0.21	80.61 ±0.14	39.98 ±0.61	47.02 ±0.54	75.15 ±1.07	75.80 ±1.71	29.64 ±0.07	28.73 ±0.09	76.03 ±1.05	80.74 ±1.59	58.64 ±0.31	62.58 ±0.27
ViT-S	3.7K	1M	65.82 ±0.48	75.85 ±0.69	31.68 ±0.19	44.92 ±0.23	75.28 ±0.48	79.88 ±1.40	29.57 ±0.35	29.90 ±0.20	75.46 ±0.28	77.10 ±0.47	55.56 ±0.16	61.53 ±0.31
ViT-M	3.7K	7.9M	63.94 ±0.26	79.61 ±0.26	30.66 ±0.85	47.47 ±0.82	72.03 ±0.55	81.71 ±1.19	30.20 ±0.15	29.89 ±0.36	77.80 ±0.21	82.42 ±0.85	54.92 ±0.18	64.22 ±0.52
ViT-L	3.7K	63.7M	63.42 ±0.46	79.35 ±0.38	26.25 ±0.76	46.37 ±0.59	73.17 ±0.32	79.40 ±2.76	29.65 ±0.36	17.96 ±2.09	71.89 ±0.75	78.40 ±0.51	52.87 ±0.07	60.30 ±0.15

setting is provided by Yuan et al. (2024), who release ResNet-18 weights pretrained on the UK Biobank dataset under controlled access (100k participants, one-week wrist accelerometer recordings) and public CAPTURE-24 dataset using the MTL framework.

Table 5 compares model weights provided by Yuan et al. (2024) with a selection of different sized ViT (as used by Hoddes et al. (2025)) and a ResNet-18 backbone pretrained using MTL combined with MorphCL on CAPTURE-24. Because Yuan et al. (2024) use larger input windows (10s vs. our 2.56s), we average embeddings over four consecutive windows (10.24s) for comparability. Despite using a smaller model (ViT-S), we achieve comparable probing performance to the Biobank model and surpass it when using the same ResNet-18 backbone, despite training on a dataset $4600\times$ smaller. This result indicates that incorporating global structural information can partially compensate for reduced pretraining scale by using the available data more efficiently.

5.5 Limitations

To evaluate our proposed MorphCL framework, we use CAPTURE-24, the largest publicly available dataset for this setting, but its scale remains modest relative to private pretraining corpora. This limitation is most apparent for larger ViT variants: while finetuning improves for ViT-M, linear probing does not. While our motif discovery and clustering hyperparameters transfer across CAPTURE-24 and WISDM, we have not studied scalability to substantially larger datasets. We leave this to future work, as we expect MorphCL to support more stable scaling on larger in-the-wild corpora. Regardless of the size of pretraining data, WISDM results indicate that already a few instances of activity executions per person can be sufficient for learning underlying morphological structures. Further, while we only test our MorphCL framework on accelerometer data, matrix profiling frameworks such as Yeh et al. (2017) can be analogously applied to magnetometer and gyroscope data.

6 Conclusion

This work introduced MorphCL, a group-based contrastive objective that uses two well-established pillars of motion analysis to inject domain knowledge into self-supervised pretraining for inertial data. Compared to existing SSL frameworks our approach grounds its notion of similarity in domain-informed feature descriptors and moves beyond instance-level and random comparisons, leveraging latent morphological structures present in inertial pretraining datasets. Concretely, we combine motif discovery, a technique with a long history in HAR for identifying the building blocks of complex motion, with domain-specific feature clustering to partition two large-scale pretraining corpora, WISDM and CAPTURE-24, into morphological groups, and use these groups to define an auxiliary contrastive loss that complements rather than replaces existing SSL frameworks. Downstream experiments demonstrate that incorporating MorphCL as an additional loss head improves representation quality of existing SSL frameworks, with gains of up to 15% in F1-score. A comparison to the accelerometer foundation model trained on the Biobank dataset (Yuan et al., 2024) shows that our method is capable of beating probing results while being trained on $4600\times$ less data.

AI use statement

In this work, we used generative AI tools for editing (e.g., grammar, spelling, word choice) and basic code assistance. We have not used generative AI tools for research ideation, developing of conceptual frameworks or execution and evaluation of experiments. All datasets used in this submission are

already published benchmark datasets and non-synthetic. We have reviewed all AI-assisted work by proofreading suggested grammatical changes. LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

With this paper proposing a new self-supervised learning loss for wearable activity recognition that is benchmarked solely on existing, publicly available benchmark datasets, we do not see ethical concerns in our work that would need to be addressed.

Reproducibility statement

Code to reproduce experiments mentioned in the main paper and supplementary material is available via an anonymized repo for reviewing (`https://anonymous.4open.science/r/mcl-AC4B`). Data downloads of all public datasets are mentioned in the README file of the repository.

Acknowledgments

Anonymized for review

References

Yuki Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyx-jyBFPr.

Marius Bock, Hilde Kuehne, Kristof Van Laerhoven, and Michael Moeller. WEAR: An Outdoor Sports Dataset for Wearable and Egocentric Activity Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 8(4):1–21, 2024. ISSN 2474-9567. doi: 10.1145/3699776. URL https://dl.acm.org/doi/10.1145/3699776.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *ACM Computing Surveys*, 46(3), 2014. doi: 10.1145/2499621. URL https://doi.org/10.1145/2499621.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Vancouver, BC, Canada, 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6.

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), 2024. doi: 10.1038/s41597-024-03960-3. URL https://doi.org/10.1038/s41597-024-03960-3.

Ching-Yao Chuang, Joshua Robinson, Lin Yen-Chen, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, Nips '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 978-1-7138-2954-6.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231, Portland, Oregon, 1996. AAAI Press.

Raffay Hamid, Siddhartha Maddi, Aaron Bobick, and Irfan Essa. Structure from Statistics - Unsupervised Activity Analysis using Suffix Trees. In *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, Rio de Janeiro, Brazil, 2007. IEEE. ISBN 978-1-4244-1630-1. doi: 10.1109/ICCV.2007.4408894. URL http://ieeexplore.ieee.org/document/4408894/.

Harish Haresamudram, David V. Anderson, and Thomas Plötz. On the role of features in human activity recognition. In *Proceedings of the 23rd International Symposium on Wearable Computers*, pp. 78–88, London United Kingdom, 2019. ACM. ISBN 978-1-4503-6870-4. doi: 10.1145/3341163.3347727. URL https://dl.acm.org/doi/10.1145/3341163.3347727.

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 *International Symposium on Wearable Computers*. ACM, 2020. doi: 10.1145/3410531.3414306. URL https://doi.org/10.1145/3410531.3414306.

Harish Haresamudram, Irfan Essa, and Thomas Plötz. Assessing the State of Self-Supervised Human Activity Recognition Using Wearables. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–47, 2022. URL https://doi.org/10.1145/3550299.

Tom Hoddes, Alex Bijamov, Saket Joshi, Daniel Roggen, Ali Etemad, Robert Harle, and David Racz. Scaling laws in wearable human activity recognition. *CoRR*, abs/2502.03364, 2025. doi: 10.48550/arXiv.2502.03364. URL https://doi.org/10.48550/arXiv.2502.03364.

Alexander Hoelzemann, Julia Lee Romero, Marius Bock, Kristof Van Laerhoven, and Qin Lv. Hang-Time HAR: A Benchmark Dataset for Basketball Activity Recognition Using Wrist-Worn Inertial Sensors. *Sensors*, 23(13), 2023. ISSN 1424-8220. doi: 10.3390/s23135879. URL https://www.mdpi.com/1424-8220/23/13/5879.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021. doi: 10.48550/arXiv.2010.11929. URL https://arxiv.org/abs/2010.11929.

Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *ACM SIGKDD Explorations Newsletter*, 12(2):74–82, 2011. ISSN 1931-0145, 1931-0153. doi: 10.1145/1964897.1964918. URL https://dl.acm.org/doi/10.1145/1964897.1964918.

Sean M. Law. STUMPY: A powerful and scalable python library for time series data mining. *The Journal of Open Source Software*, 4(39):1504, 2019. doi: 10.21105/joss.01504.

Aleksej Logacjov, Sverre Herland, Astrid Ustad, and Kerstin Bach. SelfPAB: Large-scale pretraining on accelerometer data for human activity recognition. *Applied Intelligence*, 54(6):4545–4563, 2024. ISSN 0924-669X, 1573-7497. doi: 10.1007/s10489-024-05322-3. URL https://link.springer.com/10.1007/s10489-024-05322-3.

Leland McInnes, John Healy, and Steve Astels. Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. ISSN 2475-9066. doi: 10.21105/joss.00205. URL http://joss.theoj.org/papers/10.21105/joss.00205.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, 2020. URL http://arxiv.org/abs/1802.03426.

David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. Discovering Characteristic Actions From On-Body Sensor Data. In *10th IEEE International Symposium on Wearable Computers*, 2006. URL https://doi.org/10.1109/iswc.2006.286337.

Davoud Moulavi, Pablo A. Jaskowiak, Ricardo J. G. B. Campello, Arthur Zimek, and Jörg Sander. Density-Based Clustering Validation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pp. 839–847. Society for Industrial and Applied Mathematics, April 2014. ISBN 978-1-61197-344-0. doi: 10.1137/1.9781611973440.96. URL https://epubs.siam.org/doi/10.1137/1.9781611973440.96.

Girish Narayanswamy, Xin Liu, Kumar Ayush, Yuzhe Yang, Xuhai Xu, shun liao, Jake Garrison, Shyam A. Tailor, Jacob Sunshine, Yun Liu, Tim Althoff, Shrikanth Narayanan, Pushmeet Kohli, Jiening Zhan, Mark Malhotra, Shwetak Patel, Samy Abdel-Ghaffar, and Daniel McDuff. Scaling wearable foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yb4QE6b22f.

Francisco Javier Ordóñez and Daniel Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *MDPI Sensors*, 16(1), 2016. doi: 10.3390/s16010115. URL https://doi.org/10.3390/s16010115.

Junyong Park, Oron Levy, Rebecca Adaimi, Asaf Liberman, Gierad Laput, and Abdelkareem Bedri. SlotFM: A Motion Foundation Model with Slot Attention for Diverse Downstream Tasks, 2025. URL http://arxiv.org/abs/2509.21673.

Donald J. Patterson, Dieter Fox, Henry Kautz, and Matthai Philipose. Fine-Grained Activity Recognition by Aggregating Abstract Object Usage. In *International Symposium on Wearable Computers*. IEEE, 2005. doi: 10.1109/ISWC.2005.22. URL https://doi.org/10.1109/ISWC.2005.22.

Arvind Pillai, Dimitris Spathis, Fahim Kawsar, and Mohammad Malekzadeh. PaPaGei: Open foundation models for optical physiological signals. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kYwTmlq6Vn.

Attila Reiss and Didier Stricker. Introducing a New Benchmarked Dataset for Activity Monitoring. In *IEEE 16th International Symposium on Wearable Computers*, 2012. URL https://doi.org/10.1109/ISWC.2012.13.

Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millàn. Collecting Complex Activity Datasets in Highly Rich Networked Sensor Environments. In *International Conference on Networked Sensing Systems*. IEEE, 2010. doi: 10.1109/INSS.2010.5573462. URL https://doi.org/10.1109/INSS.2010.5573462.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 03770427. doi: 10.1016/0377-0427(87)90125-7. URL https://linkinghub.elsevier.com/retrieve/pii/0377042787901257.

Aaqib Saeed, Tanir Ozcelebi, and Johan Lukkien. Multi-task Self-Supervised Learning for Human Activity Detection. *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2), 2019. doi: 10.1145/3328932. URL https://doi.org/10.1145/3328932.

Philipp M. Scholl, Matthias Wille, and Kristof Van Laerhoven. Wearables in the Wet Lab: A Laboratory System for Capturing and Guiding Experiments. In *International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2015. doi: 10.1145/2750858.2807547. URL https://doi.org/10.1145/2750858.2807547.

Timo Sztyler and Heiner Stuckenschmidt. On-Body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *International Conference on Pervasive Computing and Communications*. IEEE, 2016. doi: 10.1109/PERCOM.2016.7456521. URL https://doi.org/10.1109/PERCOM.2016.7456521.

Chi Ian Tang, Ignacio Perez-Pozuelo, Dimitris Spathis, and Cecilia Mascolo. Exploring Contrastive Learning in Human Activity Recognition for Healthcare. *CoRR*, abs/2011.11542, 2021. doi: 10.48550/arXiv.2011.11542. URL https://doi.org/10.48550/arXiv.2011.11542.

Warren W. Tryon. Activity Measurement. In *Clinician's Handbook of Adult Behavioral Assessment*, Practical Resources for the Mental Health Professional, pp. 85–120. Academic Press, 2006. ISBN 18730450. URL https://doi.org/10.1016/B978-012343013-7/50006-9.

Jeroen Van Der Donckt, Jonas Van Der Donckt, Emiel Deprost, Nicolas Vandenbussche, Michael Rademaker, Gilles Vandewiele, and Sofie Van Hoecke. Do not sleep on traditional machine learning. *Biomedical Signal Processing and Control*, 81:104429, 2023. ISSN 17468094. doi: 10.1016/j.bspc.2022.104429. URL https://linkinghub.elsevier.com/retrieve/pii/S1746809422008837.

Jeroen Van Der Donckt, Jonas Van Der Donckt, and Sofie Van Hoecke. Magnitude and Rotation Invariant Detection of Transportation Modes with Missing Data Modalities. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 597–602, Melbourne VIC Australia, 2024a. ACM. ISBN 979-8-4007-1058-2. doi: 10.1145/3675094.3678463. URL https://dl.acm.org/doi/10.1145/3675094.3678463.

Jonas Van Der Donckt, Jeroen Van Der Donckt, and Sofie Van Hoecke. Left-Right Swapping and Upper-Lower Limb Pairing for Robust Multi-Wearable Workout Activity Detection. In *Companion of the 2024 on ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 545–550, Melbourne VIC Australia, 2024b. ACM. ISBN 979-8-4007-1058-2. doi: 10.1145/3675094.3678453. URL https://dl.acm.org/doi/10.1145/3675094.3678453.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to Classify Images Without Labels. In *Computer Vision – ECCV 2020*, volume 12355, pp. 268–285. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58606-5 978-3-030-58607-2. doi: 10.1007/978-3-030-58607-2_16. URL https://link.springer.com/10.1007/978-3-030-58607-2_16.

Jamie A. Ward, Paul Lukowicz, Gerhard Tröster, and Thad E. Starner. Activity Recognition of Assembly Tasks Using Body-Worn Microphones and Accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1553–1567, 2006. doi: 10.1109/TPAMI.2006.197. URL https://doi.org/10.1109/TPAMI.2006.197.

Maxwell Xu, Alexander Moreno, Hui Wei, Benjamin Marlin, and James Matthew Rehg. REBAR: Retrieval-based reconstruction for time-series contrastive learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3zQo5oUvia.

Maxwell A. Xu, Jaya Narain, Gregory Darnell, Haraldur T. Hallgrimsson, Hyewon Jeong, Darren Forde, Richard Andres Fineman, Karthik Jayaraman Raghuram, James Matthew Rehg, and Shirley You Ren. RelCon: Relative Contrastive Learning for a Motion Foundation Model for Wearable Data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=k2uUeLCrQq.

Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn Keogh. Matrix Profile VI: Meaningful Multidimensional Motif Discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 565–574, New Orleans, LA, 2017. IEEE. ISBN 978-1-5386-3835-4. doi: 10.1109/ICDM.2017.66. URL http://ieeexplore.ieee.org/document/8215529/.

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. URL https://doi.org/10.1038/s41746-024-01062-3.

Yuwei Zhang, Kumar Ayush, Siyuan Qiao, A. Ali Heydari, Girish Narayanswamy, Maxwell A Xu, Ahmed Metwally, Jinhua Xu, Jake Garrison, Xuhai Xu, Tim Althoff, Yun Liu, Pushmeet Kohli, Jiening Zhan, Mark Malhotra, Shwetak Patel, Cecilia Mascolo, Xin Liu, Daniel McDuff, and Yuzhe Yang. SensorLM: Learning the language of wearable sensors. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TrHeq0yFhv.

A Technical Appendices and Supplementary Material

A.1 Motif-discovery & Morphological Clustering

In order to discover repeated patterns within the WISDM and CAPTURE-24 dataset we make use of the motif-discovery tool STUMPY (Law, 2019) which implements *multi-dimensional matrix profiling* as suggested by Yeh et al. (2017). STUMPY efficiently computes the matrix profile, which stores for any subsequence its distance to the most similar non-trivial subsequence, based on pairwise, z-score normalized Euclidean distances. This allows determining the r closest reoccurrences of a motif m_i which do not exceed a pre-defined motif distance threshold σ. As datasets such as the CAPTURE-24 dataset are highly unbalanced in the occurrences of their repeated patters, with e.g. flat line motifs dominating large stretches of the recorded timeseries, we mask out any additional subsequence whose matrix profile distance also lies within σ from the selected motif by setting their matrix profile values to infinity before extracting the next most prominent motif in the input timeseries. In order to scale our motif discovery to large inertial datasets, we use STUMPY's Dask-backed implementation, which enables the matrix profile to be computed in a chunked and parallelized fashion across multiple CPU cores or nodes. For both the WISDM and CAPTURE-24 dataset we extract $r = 100$ occurrences of the top $m = 100$ most prominent motifs per chunk, with σ being set to 1.0. We set the repeated subsequence of each motif to be exactly 0.2 seconds in length extracting a motif window of $W = 2.56$ seconds and context window $C = 7.68$ seconds around it. Note while sequentially extracting windows based on the extracted motif centers we ignore windows which overlap more than 50% with already extracted windows. In case of the WISDM dataset we set the chunks to be full-participant recording timelines (as this roughly equates to one hour per participant), while for CAPTURE-24 we set it to roughly one hour of data.

Once all extracted motif windows of each participant and data chunk are collected, we extract features that describe the morphology of each window. We make use of the features as reported by Van Der Donckt et al. (2023) which have shown to reapply these field-tested features to various application scenarios from sleep scoring (Van Der Donckt et al., 2023), mode of transport recognition (Van Der Donckt et al., 2024a), as well as sports activity recognition (Van Der Donckt et al., 2024b). Features are grouped into three complementary categories:

1. *Distribution features* capture the overall amplitude characteristics and statistical shape of the signal within a window. These include *minimum*, *maximum*, *peak-to-peak* amplitude, *interquartile range*, *standard deviation*, *skewness*, and *kurtosis*.

2. *Temporal and dynamical features* characterize the short-term evolution and structural complexity. *Hjorth mobility* and *complexity* quantify signal smoothness and oscillatory behavior, while *mean crossing rate* and *Petrosian fractal dimension* capture temporal variability and irregularity in the signal trajectory.

3. *Spectral and complexity features* summarize the distribution and organization of signal energy. *Spectral entropy*, computed from the Fourier spectrum, and *differential entropy* quantify signal uncertainty and spread, while *Katz fractal dimension* provides a complementary geometric measure of complexity.

The resulting feature vectors describing each window are basis for the morphological clustering. Undefined or non-finite feature values arising from degenerate windows (e.g., flat-line segments) are handled per feature as follows: *skewness* is set to 0, *kurtosis* to 3 (the Gaussian reference value), *Hjorth mobility* and *complexity* to 0, and *differential entropy* to a lower bound defined as the minimum value of said feature among all computed values minus $\mu = 0.001$. As reported in the main paper we use HDBSCAN as our clustering algorithm of choice. Being an extension of DBSCAN (Ester et al., 1996), HDBSCAN identifies clusters from the density structure of the feature vectors and treats unassigned windows as noise. We use Euclidean distance and leaf cluster selection, with $\rho = 100$ (min_cluster_count) and support$= 100$ (min_samples) in the default configuration. Sensitivity to these parameters is examined in Appendix Tables 12 and 13. With computed domain-specific features possessing widely varying natural scales, we apply z-score normalization on the computed feature vectors. Table 6 reports sizes of the created pretraining corpora.

Table 6: Size comparison of the different pretraining corpora. We compare the unfiltered version of the WISDM and CAPTURE-24 dataset, created by evenly splitting the dataset in windows of $2.56s$ without overlap, with our proposed motif-filtered variant as described in Chapter 3.1 and the morphological-clustered variant as described in Chapter 3.2. Values are provided as hours of data.

Pretraining Corpus	Unfiltered		Motif-filtered		Morph-filtered	
	Train	Test	Train	Test	Train	Test
WISDM	35.5	9.7	15.2	4.4	3.2	0.8
CAPTURE-24	3004.6	787.4	783.5	197.7	47.3	39.0

A.2 PRETRAINING

During pretraining DeepConvLSTM and ViT backbone architectures were trained using an Adam optimizer with the AMSGrad variant enabled and the initial learning rate set to $5e^{-4}$ for 100 epochs. Details of each backbone architecture can be found in their original publications (Ordóñez & Roggen, 2016; Hoddes et al., 2025) or within our code repository. To stabilize training a linear warm-up schedule is applied for the first 10 epochs followed by a learning rate scheduler that monitors the validation loss and reduces the learning rate by a factor of 0.5 if it did not improve for 5 consecutive epochs. The minimum learning rate is clipped at $1e^{-6}$ and a relative improvement threshold of $1e^{-4}$ is used. During all pretraining experiments we employ a batch size of 512 windows. Note that the REBAR (Xu et al., 2024) and RelCon (Xu et al., 2025) SSL framework follow a different batch definition, as their losses are calculated based on one randomly drawn anchor window as well as 20 candidate windows of each participant during an epoch. To make batch sizes used during REBAR and RelCon thus comparable to other frameworks mentioned in this paper, we draw from 24 participants to create a REBAR batch, with each participant appearing exactly once per epoch. As this makes epochs restricted in size by the number of participants in the pretrain dataset, we increase the total number of epochs to 1000 for both REBAR and RelCon, making the total number of windows drawn per pretraining experiment comparable to that of other pretraining framework such as MTL. In case of our MorphCL framework, to make each batch-wise gradient update an assessment of how well our feature encoder is capable of mapping to a latent space that respects morphological structures of our pretrain corpus, we design batches used during training of the MorphCL head in a way that each morphological group label y_i is represented in a balanced fashion. As groups, however, vary in size, we allow windows to occur multiple times across batches allowing for each window of each morphological group to appear at least once during training. Note that our proposed MorphCL framework uses the same model head structure as SimCLR (Tang et al., 2021), differing only in the set of possible augmentations (as mentioned in the main paper) and is thus comparable in complexity to said method. During all experiments we set the temperature value τ of our MorphCL head to 0.1.

In total we investigated six SSL frameworks:

Multi-task learning (MTL) (Saeed et al., 2019; Yuan et al., 2024): binary prediction of whether a set of pre-defined augmentations have been applied to a accelerometer signal or not. As proposed by the original authors the transformations are: flipping the data in the time-dimension, scaling values by a random factor, random permutation of sensor axes and warping the input data. Same as the original publication, we make each augmentation prediction head a 2-layered MLP with a hidden dimension of 128 and ReLU activation.

SimCLR (Tang et al., 2021): Similar to image data, the SimCLR framework for accelerometer signals is applied by creating positive pairs as two augmented views of an input signal and negative pairs as all other instances within a batch. As suggested by Tang et al. (2021) we chose a total of eight sensor-specific augmentations as our set of possible augmentations. Same as Tang et al. (2021) we use a 2-layered MLP with a 512 hidden dimension, ReLU activation, temperature $\tau = 0.1$ and 128 output dimension.

Denoising Autoencoder (Haresamudram et al., 2019): consists of using an encoder-decoder structure to recover an original input signal. We additionally randomly augment input signals, choosing from the same set of augmentations used in SimCLR, with the model needing to recover the unaugmented view. We reimplement the decoder as detailed in Hoddes et al. (2025), using 2 decoding layers, 4 attention heads, 0.1 dropout, GELU activation and setting the decoder dimension to half the input dimension and feedforward dimension to twice the decoder dimension.

Masked Autoencoder (Haresamudram et al., 2020; Hoddes et al., 2025): aims to learn useful latent features by having models reconstruct masked parts of an input signal through patterns extracted from surrounding unmasked parts of the input signal. Same as the denoising decoder, we reimplement the decoder as detailed in Hoddes et al. (2025), applying also a token-based masking ratio of 0.7.

REBAR (Xu et al., 2024): Extending the idea of SimCLR (Xu et al., 2024), introduced REBAR, which provides a learned metric to determine positive pairs within a batch. Similar to our approach, their work is based on the assumption that windows which are capable of reconstructing masked versions of each other well share inherent patterns. That way, the framework provides a way to define positive pairs across signals as well as participants. Note that negative pairs are defined the same way as in SimCLR. We reuse the implementation of REBAR from the authors repository, setting the embedding dimension to 256 and double receptive field parameter to 1. To calculate the REBAR loss we use $\tau = 0.1$ and $\alpha = 0.0$ as done in the original publication.

RelCon (Xu et al., 2025): In addition to small improvements to the training of the similarity metric and loss calculation, RelCon softens the negative pair definition of REBAR by replacing it with a function that dynamically adjusts the negative pairs based on the current positive pairing. This allows RelCon to dynamically learn a ranking among signals in a batch rather than relying on hard positive and negative pair decisions. We reuse the implementation of RelCon from the authors repository. To train the involved REBAR metric we reuse settings from the REBAR repository, i.e. setting the embedding dimension to 256 and double receptive field parameter to 1. To calculate the RelCon loss we use $\tau = 0.1$ as done in the original publication.

A.3 Downstream Experiments

During all downstream experiments, pretraining heads, such as our proposed MorphCL head, were removed from the backbone and replaced by a single linear classification layer, which maps the output dimension of the last backbone layer directly to the number of classes present in the downstream dataset. In case of linear probing experiments, we freeze the weights of the backbone allowing only the linear classification layer to be trained, while for finetuning experiments, we additionally allow all layers of the backbone to be adjusted. We use an Adam optimizer with the initial learning rate being $1e^{-1}$ for the linear classification head and $1e^{-4}$ for finetuning the pretrained backbone networks. In line with best practices we employ a Leave-One-Subject-Out cross-validation, where each participant in the dataset becomes the test set exactly once while remaining participants are used for training. Each dataset is segmented using a sliding window of of length 2.56 seconds with an overlap of 50%. Batches consist of 128 randomly-shuffled windows and are z-score normalized per-axis based on the training data. Each split is trained for 30 epochs with learning rates being decreased by a factor of 0.5 if the validation loss, computed on a excluded participant in the train dataset, does not improve over 5 epochs. Loss calculation is based on a class-weighted cross-entropy loss to accommodate the unbalancedness of the tested downstream datasets. For comparison we also provide fully-supervised training results on each dataset and backbone using a unified initial learning rate for backbone and classifier ($1e^{-3}$).

A.4 Licensing & Runtime Environment

All datasets were downloaded from their respective code repositories and websites as of submitting this paper. A complete list of download links can be found in our code repository. Processed versions (motif-filtered and morph-clustered) of the pretraining datasets are provided as supplementary download. All datasets are hosted under either the CC-BY 4.0 (CAPTURE-24, Hang-Time, PAMAP2, RWHAR, Wetlab, WISDM) or the CC BY-NC-SA 4.0 license (WEAR). All experiments were performed on a single-GPU (NVIDIA Tesla V100) with 2 AMD EPYC 7452 CPUs and 256 GB RAM. Larger ViT variants were trained on larger single GPU (NVIDIA A6000) with two AMD EPYC 7413 and 1008 GB RAM. Note that the footprint and computational complexity of each model is dependent on the chosen SSL framework. Motif discovery was performed on one hour chunks of data at a time and took around 170 seconds per chunk performed on a single, standard CPU (Intel Core Ultra 9 285K).

B SUPPLEMENTARY EXPERIMENTS

B.1 DETAILED MAIN RESULTS

Table 7 reports detailed results of the linear probing and finetuning experiments across the five downstream datasets comparing as a standalone framework MorphCL with the tested instance-based learning schemes. Table 8 and Table 9 provide detailed results of the relative linear probing and finetuning improvements achieved when combining MorphCL with the instance-based learning schemes. Note that results are provided as macro F1-scores averaged across the individual LOSO splits. Each experiment was repeated three times using a set of three different random seeds. The last column of each table further provides the average across all downstream datasets. To characterize variability across the three random seeds, we report standard deviations of all experiments.

Table 7: Downstream result comparison of the proposed morphological learning with instance-based self-supervised learning methods. Results are reported for five downstream benchmark datasets. Pretraining was performed on both the WISDM and CAPTURE-24 dataset using different encoder backbones. Fully-supervised learning refers to training the backbone without prior pretraining directly on the downstream dataset. Macro F1 is reported in % as mean ± standard deviation across seeded runs. Shaded rows denote standalone MorphCL.

	WEAR		Wetlab		PAMAP2		Hangtime		RWHAR		*Avg.*	
	LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine	LProbe	Fine
Encoder: DeepConvLSTM												
Fully-supervised	*61.42* ±0.63		*29.34* ±0.38		*65.46* ±0.93		*52.04* ±0.31		*63.89* ±0.70		*54.43* ±0.17	
Pretraining: WISDM												
MTL	37.72 ±0.34	57.86 ±0.53	17.63 ±0.07	27.99 ±0.19	54.84 ±0.18	69.69 ±0.16	27.03 ±0.56	48.59 ±0.23	54.31 ±0.44	66.77 ±1.46	38.31 ±0.06	54.18 ±0.44
SimCLR	46.74 ±0.11	60.66 ±0.25	21.21 ±0.41	32.58 ±0.25	61.47 ±0.22	73.64 ±0.80	38.64 ±0.36	52.40 ±0.12	63.29 ±0.45	67.35 ±0.89	46.27 ±0.09	57.32 ±0.31
REBAR	33.64 ±0.26	56.64 ±0.20	18.16 ±0.05	27.76 ±0.08	53.86 ±0.28	67.06 ±0.64	24.25 ±0.22	50.89 ±0.39	59.15 ±0.77	63.81 ±0.75	37.81 ±0.19	53.23 ±0.29
RelCon	30.80 ±0.11	56.43 ±0.15	16.19 ±0.11	27.22 ±0.54	48.90 ±0.63	65.24 ±1.01	25.39 ±0.24	50.87 ±0.23	54.66 ±1.23	64.37 ±1.24	35.19 ±0.36	52.82 ±0.04
MorphCL	43.99 ±0.42	60.00 ±0.37	24.24 ±0.11	31.62 ±0.41	60.85 ±0.23	70.27 ±1.37	32.84 ±0.52	52.16 ±0.50	65.20 ±1.10	69.47 ±1.46	45.42 ±0.13	56.71 ±0.17
Encoder: ViT-S												
Fully-supervised	*63.76* ±0.15		*25.20* ±0.51		*66.44* ±0.46		*34.10* ±1.80		*63.90* ±0.47		*50.68* ±0.25	
Pretraining: WISDM												
MTL	50.05 ±0.59	66.02 ±0.33	22.39 ±0.24	33.25 ±0.30	63.84 ±0.42	70.05 ±0.70	33.71 ±0.98	49.67 ±0.15	67.03 ±0.80	68.58 ±0.45	47.41 ±0.22	57.51 ±0.27
SimCLR	53.08 ±0.62	63.12 ±0.41	25.94 ±0.41	32.71 ±0.24	64.00 ±1.25	67.35 ±0.87	35.73 ±0.44	47.96 ±0.30	65.14 ±1.44	65.25 ±0.85	48.78 ±0.57	55.28 ±0.25
Masked AE	46.58 ±0.80	60.57 ±0.07	17.43 ±1.04	31.09 ±0.08	57.86 ±0.65	66.59 ±1.82	37.19 ±1.60	47.65 ±0.14	62.32 ±1.03	65.92 ±0.16	44.28 ±0.87	54.36 ±0.32
Denoising AE	49.70 ±0.59	60.51 ±0.08	20.92 ±0.93	31.25 ±0.16	62.53 ±0.25	65.63 ±1.36	38.35 ±0.38	47.21 ±0.24	67.15 ±1.08	64.11 ±0.64	47.73 ±0.49	53.74 ±0.24
REBAR	42.59 ±0.68	58.84 ±0.06	16.80 ±0.40	30.13 ±0.30	59.01 ±0.38	64.95 ±1.14	29.12 ±0.38	46.70 ±0.33	59.89 ±2.57	64.11 ±0.68	41.48 ±0.68	52.95 ±0.21
RelCon	37.53 ±0.63	59.42 ±0.16	13.70 ±0.61	30.20 ±0.11	43.71 ±1.19	64.54 ±0.95	25.88 ±0.26	47.30 ±0.01	52.21 ±1.18	65.69 ±0.87	34.61 ±0.33	53.43 ±0.30
MorphCL	52.38 ±0.09	61.67 ±0.24	26.45 ±0.52	32.31 ±0.15	66.21 ±0.85	67.52 ±1.06	34.86 ±0.81	47.55 ±0.25	67.92 ±0.35	64.19 ±0.70	49.56 ±0.42	54.65 ±0.30
Pretraining: CAPTURE-24												
MTL	49.34 ±0.74	69.07 ±0.12	22.44 ±0.20	35.79 ±0.04	66.56 ±0.81	70.79 ±0.15	34.86 ±0.34	51.02 ±0.14	63.27 ±0.57	74.64 ±0.60	47.29 ±0.12	60.26 ±0.15
SimCLR	55.96 ±0.10	66.87 ±0.16	27.43 ±0.24	35.10 ±0.18	68.85 ±0.77	70.73 ±1.41	39.04 ±0.68	49.12 ±0.33	65.53 ±0.56	69.26 ±0.72	51.36 ±0.20	58.22 ±0.39
Masked AE	48.30 ±0.20	61.94 ±0.25	20.32 ±1.19	32.19 ±0.14	63.04 ±0.72	69.15 ±0.82	39.49 ±0.08	48.25 ±0.09	66.65 ±0.94	66.51 ±0.94	47.56 ±0.38	55.61 ±0.32
Denoising AE	50.17 ±0.29	63.67 ±0.25	23.53 ±0.47	32.44 ±0.08	62.68 ±0.54	65.87 ±0.75	39.60 ±0.12	48.23 ±0.28	65.38 ±0.24	66.33 ±0.82	48.27 ±0.23	55.31 ±0.29
REBAR	36.76 ±0.52	59.77 ±0.12	16.80 ±0.69	30.89 ±0.11	53.47 ±1.35	65.18 ±0.54	24.50 ±0.82	46.08 ±0.15	53.91 ±0.98	68.36 ±0.04	37.09 ±0.13	54.05 ±0.13
RelCon	41.62 ±0.10	59.06 ±0.33	14.87 ±1.34	30.41 ±0.14	52.97 ±1.03	65.23 ±0.33	33.38 ±0.47	47.33 ±0.03	59.77 ±2.83	65.48 ±1.57	40.52 ±0.98	53.50 ±0.40
MorphCL	51.68 ±0.06	65.34 ±0.23	25.88 ±0.96	35.16 ±0.20	67.41 ±0.29	70.02 ±0.69	36.52 ±1.52	49.55 ±0.24	63.03 ±0.62	69.40 ±0.47	48.90 ±0.66	57.90 ±0.23

B.2 SENSITIVITY ANALYSIS ON MOTIF DISCOVERY PARAMETERS

We investigate the sensitivity of our motif discovery on the WISDM dataset to hyperparameters along two dimensions. First, Table 10 reports the discovered motifs for different motif distance thresholds σ. Second, in Table 11 we ablate different numbers of minimum occurrences r of a motif to be considered a motif. Note that we do not specifically ablate on motif sizes larger than 0.2s as we argue that (1) larger motifs can be considered a subset of these and (2) smaller motifs are too small given our fixed sampling rate of 50 Hz (less than 10 samples) to contain any meaningful motion. One can see that motif discovery remains relatively stable across differently chosen σ and r values. For smaller σ, we see less motifs and windows extracted, as fewer motif candidates are considered close enough to each other. For larger distances, number of extracted windows slightly decreases, which we attribute to our applied masking strategy of additional motif candidates ($> k$) as detailed in the paper. As we apply motif discovery in one hour chunks to both CAPTURE-24 and WISDM, we expect results below computed on WISDM to be representative for CAPTURE-24 as well. Downstream results (MTL + MorphCL) show our MorphCL framework being stable across different motif discovery hyperparameters. Note that we apply our initial configuration for clustering ($\rho = 100$ and support $= 100$) during these experiments. Our chosen configuration of $\sigma = 1.0$ and $r = 100$ returns most stable results for motifs of length 0.2 seconds.

Table 8: Absolute percentage point improvements on downstream linear probing results of instance-based self-supervised learning methods if combined with an additional morphological loss term. Results are reported for five downstream benchmark datasets. Pretraining was performed on both the WISDM and CAPTURE-24 dataset using different encoder backbones. Macro F1 is reported in % as mean ± standard deviation across seeded runs.

	WEAR	Wetlab	PAMAP2	Hangtime	RWHAR	*Avg.*
Encoder: DeepConvLSTM; **Pretraining:** WISDM						
MTL	37.72 ±0.34	17.63 ±0.07	54.84 ±0.18	27.03 ±0.56	54.31 ±0.44	38.31 ±0.06
MTL + MorphCL	50.58 ±0.34 (+12.86)	23.72 ±0.33 (+6.09)	66.99 ±0.54 (+12.16)	36.03 ±0.59 (+9.00)	65.61 ±1.12 (+11.30)	48.59 ±0.25 (+10.28)
SimCLR	46.74 ±0.11	21.21 ±0.41	61.47 ±0.22	38.64 ±0.36	63.29 ±0.45	46.27 ±0.09
SimCLR + MorphCL	48.41 ±0.49 (+1.67)	23.43 ±0.42 (+2.22)	63.57 ±0.61 (+2.10)	39.52 ±0.43 (+0.88)	64.33 ±0.31 (+1.04)	47.85 ±0.26 (+1.58)
REBAR	33.64 ±0.26	18.16 ±0.05	53.86 ±0.28	24.25 ±0.22	59.15 ±0.77	37.81 ±0.19
REBAR + MorphCL	45.01 ±0.25 (+11.37)	25.15 ±0.15 (+6.99)	65.29 ±0.14 (+11.43)	32.62 ±0.31 (+8.37)	64.08 ±0.71 (+4.93)	46.43 ±0.17 (+8.62)
RelCon	30.80 ±0.11	16.19 ±0.11	48.90 ±0.63	25.39 ±0.24	54.66 ±1.23	35.19 ±0.36
RelCon + MorphCL	44.50 ±0.22 (+13.70)	23.16 ±0.34 (+6.96)	65.25 ±0.61 (+16.35)	34.27 ±0.26 (+8.88)	62.23 ±0.34 (+7.57)	45.88 ±0.20 (+10.69)
Encoder: ViT-S; **Pretraining:** WISDM						
MTL	50.05 ±0.59	22.39 ±0.24	63.84 ±0.42	33.71 ±0.98	67.03 ±0.80	47.41 ±0.22
MTL + MorphCL	53.55 ±0.23 (+3.50)	26.53 ±0.30 (+4.14)	67.17 ±1.20 (+3.33)	35.82 ±0.51 (+2.11)	69.60 ±1.20 (+2.57)	50.53 ±0.45 (+3.13)
SimCLR	53.08 ±0.62	25.94 ±0.41	64.00 ±1.25	35.73 ±0.44	65.14 ±1.44	48.78 ±0.57
SimCLR + MorphCL	56.30 ±0.21 (+3.22)	25.98 ±0.54 (+0.04)	66.00 ±0.20 (+2.00)	37.36 ±1.08 (+1.63)	68.46 ±2.54 (+3.31)	50.82 ±0.50 (+2.04)
Masked AE	46.58 ±0.80	17.43 ±1.04	57.86 ±0.65	37.19 ±1.60	62.32 ±1.03	44.28 ±0.87
Masked AE + MorphCL	53.20 ±0.48 (+6.62)	27.47 ±1.01 (+10.04)	66.88 ±0.83 (+9.02)	36.91 ±0.33 (−0.28)	68.37 ±2.13 (+6.05)	50.57 ±0.50 (+6.29)
Denoising AE	49.70 ±0.59	20.92 ±0.93	62.53 ±0.25	38.35 ±0.38	67.15 ±1.08	47.73 ±0.49
Denoising AE + MorphCL	53.88 ±0.11 (+4.18)	25.38 ±0.27 (+4.45)	63.80 ±0.31 (+1.27)	37.23 ±0.45 (−1.12)	66.30 ±1.09 (−0.85)	49.32 ±0.33 (+1.59)
REBAR	42.59 ±0.68	16.80 ±0.40	59.01 ±0.38	29.12 ±0.38	59.89 ±2.57	41.48 ±0.68
REBAR + MorphCL	52.77 ±0.47 (+10.18)	26.20 ±1.25 (+9.40)	65.13 ±0.45 (+6.11)	36.45 ±1.07 (+7.33)	68.48 ±0.88 (+8.59)	49.81 ±0.39 (+8.32)
RelCon	37.53 ±0.63	13.70 ±0.61	43.71 ±1.19	25.88 ±0.26	52.21 ±1.18	34.61 ±0.33
RelCon + MorphCL	50.91 ±0.50 (+13.37)	24.93 ±0.46 (+11.23)	66.04 ±0.74 (+22.34)	39.30 ±0.21 (+13.42)	68.62 ±2.13 (+16.41)	49.96 ±0.40 (+15.36)
Encoder: ViT-S; **Pretraining:** CAPTURE-24						
MTL	49.34 ±0.74	22.44 ±0.20	66.56 ±0.81	34.86 ±0.34	63.27 ±0.57	47.29 ±0.12
MTL + MorphCL	56.53 ±0.29 (+7.19)	26.65 ±0.74 (+4.22)	69.70 ±0.65 (+3.14)	38.90 ±1.01 (+4.03)	68.83 ±0.86 (+5.56)	52.12 ±0.33 (+4.83)
SimCLR	55.96 ±0.10	27.43 ±0.24	68.85 ±0.77	39.04 ±0.68	65.53 ±0.56	51.36 ±0.20
SimCLR + MorphCL	56.67 ±0.24 (+0.71)	26.93 ±0.24 (−0.50)	68.29 ±0.29 (−0.56)	39.02 ±1.53 (−0.02)	63.94 ±1.07 (−1.59)	50.97 ±0.23 (−0.39)
Masked AE	48.30 ±0.20	20.32 ±1.19	63.04 ±0.72	39.49 ±0.08	66.65 ±0.94	47.56 ±0.38
Masked AE + MorphCL	54.21 ±0.03 (+5.91)	27.08 ±1.07 (+6.76)	65.55 ±0.74 (+2.52)	37.13 ±0.95 (−2.36)	66.26 ±0.27 (−0.40)	50.05 ±0.37 (+2.49)
Denoising AE	50.17 ±0.29	23.53 ±0.47	62.68 ±0.54	39.60 ±0.12	65.38 ±0.24	48.27 ±0.23
Denoising AE + MorphCL	56.41 ±0.59 (+6.24)	25.96 ±0.64 (+2.43)	67.23 ±0.57 (+4.55)	38.72 ±1.33 (−0.87)	67.32 ±1.25 (+1.94)	51.13 ±0.33 (+2.86)
REBAR	36.76 ±0.52	16.80 ±0.69	53.47 ±1.35	24.50 ±0.82	53.91 ±0.98	37.09 ±0.13
REBAR + MorphCL	50.83 ±0.48 (+14.07)	25.39 ±1.40 (+8.59)	66.97 ±0.25 (+13.51)	38.50 ±0.56 (+14.00)	59.67 ±1.74 (+5.75)	48.27 ±0.49 (+11.18)
RelCon	41.62 ±0.10	14.87 ±1.34	52.97 ±1.03	33.38 ±0.47	59.77 ±2.83	40.52 ±0.98
RelCon + MorphCL	54.19 ±0.72 (+12.58)	27.29 ±0.43 (+12.42)	64.35 ±0.45 (+11.38)	37.13 ±0.86 (+3.74)	66.69 ±0.99 (+6.92)	49.93 ±0.30 (+9.41)

B.3 Sensitivity Analysis on Clustering Parameters

We ablate ρ and cluster support on the WISDM dataset. To investigate the stability across changes in hyperparameters we report DBCV (Density-Based Clustering Validation), a cluster-validity measure as suggested by Moulavi et al. (2014) that does not require ground-truth labels ($[-1, 1]$, 1 being well-separated clusters) and AMI (Adjusted Mutual Information) against activity labels ($[0, 1]$ 1 being perfect agreement between clusters and activity labels) alongside cluster count and noise rate. Both DBCV and AMI remain stable across $\rho \in [50, 200]$ and support $\in [50, 200]$, indicating the method is not narrowly tuned to a single operating point. At tested extremes (400), quality predictably degrades as excessively high ρ exceeds the WISDM dataset's density scale, while excessively high cluster support forces merging of distinct sub-clusters into a few coarse groups. Downstream results (MTL + MorphCL) show that our MorphCL framework is stable across different clustering hyperparameters. Our chosen configuration of $\rho = 100$ and support$= 100$ returns the most stable results.

B.4 Subset Evaluation

To separate the contribution of the MorphCL loss from that of the data filtering, we train the base objectives (MTL, SimCLR, Denoising AE, Masked AE, REBAR and RelCon) directly on the motif-filtered subset ("Motif") and the morphological-clustered subset using the same balanced sampler as MorphCL ("Morph"), with and without the MorphCL loss. We also test random subsets matched in size and repeated samples, to isolate the effect of size alone. Numbers are averaged over the five downstream tasks across three seeds.

On WISDM (see Table 14), filtering alone results in comparable results to using the full dataset, with few exceptions slightly improving or degrading in performance. Adding MorphCL improves results across all subsets, demonstrating that the gain is attributable to the objective, not the subset.

Table 9: Absolute finetuning percentage point improvement of existing SSL methods if combined with MorphCL. Results are seed-averaged across 3 runs. All frameworks benefit from being combined with MorphCL with the MTL framework being the best performing method. Macro F1 is reported in % as mean ± standard deviation across seeded runs.

	WEAR	Wetlab	PAMAP2	Hangtime	RWHAR	*Avg.*
Encoder: DeepConvLSTM; **Pretraining:** WISDM						
MTL	57.86 ±0.53	27.99 ±0.19	69.69 ±0.16	48.59 ±0.23	66.77 ±1.46	54.18 ±0.44
MTL + MorphCL	62.59 ±0.46 (+4.74)	33.36 ±0.17 (+5.37)	72.37 ±1.04 (+2.67)	52.99 ±0.54 (+4.40)	69.06 ±0.92 (+2.29)	58.07 ±0.22 (+3.89)
SimCLR	60.66 ±0.25	32.58 ±0.25	73.64 ±0.80	52.40 ±0.12	67.35 ±0.89	57.32 ±0.31
SimCLR + MorphCL	62.33 ±0.28 (+1.67)	33.46 ±0.30 (+0.88)	71.68 ±0.97 (−1.96)	53.15 ±0.22 (+0.75)	67.21 ±1.41 (−0.14)	57.57 ±0.28 (+0.24)
REBAR	56.64 ±0.20	27.76 ±0.08	67.06 ±0.64	50.89 ±0.39	63.81 ±0.75	53.23 ±0.29
REBAR + MorphCL	59.52 ±0.62 (+2.88)	31.88 ±0.34 (+4.12)	69.79 ±0.07 (+2.73)	52.62 ±0.20 (+1.73)	71.23 ±0.37 (+7.42)	57.01 ±0.04 (+3.78)
RelCon	56.43 ±0.15	27.22 ±0.54	65.24 ±1.01	50.87 ±0.23	64.37 ±1.24	52.82 ±0.04
RelCon + MorphCL	60.21 ±0.11 (+3.78)	31.98 ±0.29 (+4.76)	69.50 ±2.62 (+4.26)	52.40 ±0.21 (+1.53)	66.79 ±0.62 (+2.42)	56.18 ±0.40 (+3.35)
Encoder: ViT-S; **Pretraining:** WISDM						
MTL	66.02 ±0.33	33.25 ±0.30	70.05 ±0.70	49.67 ±0.15	68.58 ±0.45	57.51 ±0.27
MTL + MorphCL	66.62 ±0.24 (+0.60)	34.46 ±0.24 (+1.21)	70.82 ±0.16 (+0.77)	49.30 ±0.09 (−0.38)	69.93 ±0.22 (+1.35)	58.23 ±0.16 (+0.71)
SimCLR	63.12 ±0.41	32.71 ±0.24	67.35 ±0.87	47.96 ±0.30	65.25 ±0.85	55.28 ±0.25
SimCLR + MorphCL	64.23 ±0.05 (+1.11)	33.69 ±0.03 (+0.98)	68.62 ±0.34 (+1.28)	48.78 ±0.47 (+0.82)	65.93 ±1.15 (+0.68)	56.25 ±0.19 (+0.97)
Masked AE	60.57 ±0.07	31.09 ±0.08	66.59 ±1.82	47.65 ±0.14	65.92 ±0.16	54.36 ±0.32
Masked AE + MorphCL	62.98 ±0.12 (+2.42)	32.98 ±0.28 (+1.89)	70.20 ±0.29 (+3.61)	48.17 ±0.24 (+0.52)	66.55 ±0.87 (+0.63)	56.18 ±0.16 (+1.81)
Denoising AE	60.51 ±0.08	31.25 ±0.16	65.63 ±1.36	47.21 ±0.24	64.11 ±0.64	53.74 ±0.24
Denoising AE + MorphCL	62.88 ±0.26 (+2.37)	32.37 ±0.16 (+1.12)	68.72 ±0.89 (+3.08)	48.11 ±0.23 (+0.90)	64.28 ±0.99 (+0.18)	55.27 ±0.13 (+1.53)
REBAR	58.84 ±0.06	30.13 ±0.30	64.95 ±1.14	46.70 ±0.33	64.11 ±0.68	52.95 ±0.21
REBAR + MorphCL	61.95 ±0.07 (+3.11)	32.37 ±0.12 (+2.24)	67.01 ±0.62 (+2.06)	48.46 ±0.16 (+1.76)	64.33 ±0.78 (+0.22)	54.83 ±0.24 (+1.88)
RelCon	59.42 ±0.16	30.20 ±0.11	64.54 ±0.95	47.30 ±0.01	65.69 ±0.87	53.43 ±0.30
RelCon + MorphCL	60.78 ±0.05 (+1.37)	31.50 ±0.25 (+1.31)	65.43 ±0.55 (+0.89)	47.42 ±0.12 (+0.11)	64.64 ±0.54 (−1.05)	53.95 ±0.15 (+0.52)
Encoder: ViT-S; **Pretraining:** CAPTURE-24						
MTL	69.07 ±0.12	35.79 ±0.04	70.79 ±0.15	51.02 ±0.14	74.64 ±0.60	60.26 ±0.15
MTL + MorphCL	70.47 ±0.30 (+1.39)	36.26 ±0.04 (+0.47)	72.04 ±0.25 (+1.25)	51.58 ±0.25 (+0.56)	73.07 ±0.49 (−1.57)	60.68 ±0.11 (+0.42)
SimCLR	66.87 ±0.16	35.10 ±0.18	70.73 ±1.41	49.12 ±0.33	69.26 ±0.72	58.22 ±0.39
SimCLR + MorphCL	67.32 ±0.27 (+0.45)	35.76 ±0.26 (+0.67)	72.07 ±0.20 (+1.34)	50.26 ±0.39 (+1.14)	68.20 ±0.55 (−1.06)	58.72 ±0.19 (+0.51)
Masked AE	61.94 ±0.25	32.19 ±0.14	69.15 ±0.82	48.25 ±0.09	66.51 ±0.94	55.61 ±0.32
Masked AE + MorphCL	66.55 ±0.25 (+4.60)	35.06 ±0.21 (+2.87)	69.78 ±0.72 (+0.62)	49.66 ±0.27 (+1.41)	68.87 ±0.33 (+2.36)	57.98 ±0.13 (+2.37)
Denoising AE	63.67 ±0.25	32.44 ±0.08	65.87 ±0.75	48.23 ±0.28	66.33 ±0.82	55.31 ±0.29
Denoising AE + MorphCL	68.00 ±0.33 (+4.33)	35.11 ±0.16 (+2.67)	69.66 ±0.64 (+3.79)	50.67 ±0.22 (+2.44)	68.30 ±0.61 (+1.97)	58.35 ±0.25 (+3.04)
REBAR	59.77 ±0.12	30.89 ±0.11	65.18 ±0.54	46.08 ±0.15	68.36 ±0.04	54.05 ±0.13
REBAR + MorphCL	66.13 ±0.41 (+6.35)	34.93 ±0.36 (+4.04)	70.79 ±0.34 (+5.62)	49.99 ±0.45 (+3.91)	68.76 ±0.27 (+0.40)	58.12 ±0.16 (+4.07)
RelCon	59.06 ±0.33	30.41 ±0.14	65.23 ±0.33	47.33 ±0.03	65.48 ±1.57	53.50 ±0.40
RelCon + MorphCL	64.43 ±0.36 (+5.37)	33.82 ±0.21 (+3.42)	69.42 ±1.80 (+4.19)	48.57 ±0.26 (+1.25)	67.05 ±0.16 (+1.57)	56.66 ±0.42 (+3.16)

Table 10: Hyperparameter sweep for σ with fixed $r = 100$ for a motif length of 0.2s.

σ	Discovered motifs	Extracted windows	Relative to full dataset	Downstream F1
0.5	636	15,816	31.69%	49.61 ±0.64
1.0	758	21,318	42.71%	50.53 ±0.45
1.5	788	23,202	46.49%	49.89 ±0.61
2.0	690	20,793	41.66%	49.67 ±0.58

Table 11: Hyperparameter sweep for r with fixed $\sigma = 1.0$ for a motif length of 0.2s.

r	Discovered motifs	Extracted windows	Relative to full dataset	Downstream F1
50	1,145	19,943	40.00%	49.95 ±0.75
100	758	21,318	42.71%	50.53 ±0.45
150	656	22,899	45.88%	49.88 ±0.75
200	544	22,956	45.99%	49.81 ±0.55

Table 12: Hyperparameter sweep for ρ with fixed support $= 100$.

ρ	n_{clusters}	Noise	DBCV	AMI (vs. activity)	Downstream F1
50	19	74.70%	0.3220	0.6954	50.74 ±0.60
100	8	78.83%	0.2434	0.7122	50.53 ±0.45
200	6	82.38%	0.3287	0.6595	50.01 ±0.74
400	–	100%	–	–	–

On CAPTURE-24 (see Table 15), motif-based filtering has a positive effect on SimCLR, MTL and REBAR and a negative effect on RelCon. Nevertheless, MorphCL adds a comparable further gain to all frameworks. For example, for MTL, motif filtering contributes a gain of +2.44 and adding MorphCL on the same subset achieves +2.89 more, for a total improvement of +5.33 over the

Table 13: Hyperparameter sweep support with fixed $\rho = 100$.

Support	n_{clusters}	Noise	DBCV	AMI (vs. activity)	Downstream F1
50	8	78.83%	0.2434	0.7122	50.73 ±0.60
100	8	78.83%	0.2434	0.7122	50.53 ±0.45
200	6	80.16%	0.2245	0.6674	49.67 ±0.80
400	2	14.64%	0.0622	0.0357	49.28 ±0.47

full-corpus baseline. Filtering and the MorphCL loss each contribute roughly half the total gain, confirming that gains reported in the main paper are not merely on the data selection.

Table 14: Downstream F1 scores (%; mean ± standard deviation) for different subset pretraining on the WISDM dataset.

Method	Full	Motif	Morph	Random (N=Motif)	Random (N=Morph)
SimCLR	48.78 ±0.57	47.75 ±0.62	47.88 ±0.62	46.69 ±0.56	46.29 ±0.43
SimCLR + MorphCL	**50.82** ±0.50	**50.53** ±0.69	**49.80** ±0.50	**49.51** ±0.36	**49.16** ±0.58
MTL	47.41 ±0.22	46.13 ±0.36	46.86 ±0.36	45.80 ±0.42	45.96 ±0.41
MTL + MorphCL	**50.53** ±0.45	**49.70** ±0.79	**49.42** ±0.49	**49.46** ±0.59	**48.50** ±0.41
Denoising AE	47.73 ±0.49	47.11 ±0.71	48.68 ±0.48	46.64 ±0.66	45.94 ±0.64
Denoising AE + MorphCL	**49.32** ±0.33	**49.75** ±0.59	**50.25** ±0.74	**49.47** ±0.86	**49.68** ±0.71
Masked AE	44.28 ±0.87	43.56 ±0.66	42.42 ±0.87	43.54 ±0.64	42.76 ±0.53
Masked AE + MorphCL	**50.57** ±0.50	**50.06** ±0.47	**50.12** ±0.88	**50.07** ±0.65	**50.08** ±0.64
REBAR	41.48 ±0.68	43.06 ±0.90	41.63 ±0.41	39.85 ±0.98	42.17 ±0.79
REBAR + MorphCL	**49.81** ±0.39	**50.17** ±1.01	**50.22** ±0.54	**50.22** ±0.54	**50.22** ±0.54
RelCon	34.61 ±0.33	34.41 ±0.71	36.40 ±1.18	36.72 ±0.83	32.20 ±1.19
RelCon + MorphCL	**49.96** ±0.40	**48.83** ±0.56	**48.83** ±0.56	**48.83** ±0.56	**48.83** ±0.56

Table 15: Downstream F1 scores (%; mean ± standard deviation) for different subset pretraining on the CAPTURE-24 dataset.

Method	Full	Motif	Morph	Random (N=Motif)	Random (N=Morph)
SimCLR	**51.36** ±0.20	52.05 ±0.47	45.93 ±0.44	50.14 ±0.47	46.51 ±0.46
SimCLR + MorphCL	50.97 ±0.23	**52.16** ±0.48	**48.12** ±0.80	**51.34** ±0.45	**49.26** ±0.43
MTL	47.29 ±0.12	49.73 ±0.85	49.48 ±0.59	45.08 ±0.47	46.61 ±0.53
MTL + MorphCL	**52.12** ±0.33	**52.62** ±0.52	**51.41** ±0.46	**50.92** ±0.41	**50.61** ±0.60
Denoising AE	48.27 ±0.23	46.65 ±0.42	46.21 ±0.58	46.62 ±0.57	44.77 ±0.68
Denoising AE + MorphCL	**51.13** ±0.33	**50.26** ±0.61	**49.11** ±0.51	**49.90** ±0.67	**49.57** ±0.41
Masked AE	47.56 ±0.38	47.39 ±0.65	46.46 ±0.49	46.20 ±0.68	45.60 ±0.77
Masked AE + MorphCL	**50.05** ±0.37	**49.28** ±0.42	**48.83** ±0.65	**49.57** ±0.38	**49.22** ±0.41
REBAR	37.09 ±0.13	42.63 ±0.62	42.53 ±0.87	36.39 ±0.67	32.21 ±0.61
REBAR + MorphCL	**48.27** ±0.49	**48.86** ±0.63	**49.22** ±0.41	**48.03** ±0.43	**48.00** ±0.46
RelCon	40.52 ±0.98	26.38 ±1.33	27.25 ±0.92	41.95 ±0.41	40.30 ±0.88
RelCon + MorphCL	**49.93** ±0.30	**50.44** ±0.75	**50.33** ±0.45	**50.30** ±0.65	**50.33** ±0.45

B.5 MorphCL Sampling Method

We also investigate the effects of the balancing and oversampling of MorphCL batches. We remove the balancing sampler components from MorphCL training on WISDM: (i) undersampling instead of oversampling for balance, and (ii) no balancing at all (uniform batches). Numbers are averages over the five downstream tasks, averaged across three seeds. Results pretraining on the WISDM dataset (see Table 16) suggest stability of our method even when not using oversampling and balanced batches. Same as on WISDM, results on the CAPTURE-24 dataset show robustness of our method to different sampling techniques (see Table 17). One can see that SimCLR slightly benefits from unbalanced MorphCL batches, which we hypothesize is due to a lower likelihood that MorphCL loss gradients collide with negative-pair gradients in the SimCLR loss.

B.6 Loss Term Weighting

To investigate the robustness of our results presented in the main paper to different scaled MorphCL loss terms, we report sample linear probing results using three scaling factors ($\lambda \in [5, 10, 20]$) using a ViT-S backbone in combination with MTL (see Table 18). In general, observing tracked gradient magnitudes of each individual loss, the MorphCL loss is on average around 5 to 10 order of

Table 16: Downstream F1 scores for different balancing strategies for the MorphCL loss pretraining on the WISDM dataset.

Method	Balanced + Oversampling	Balanced + Undersampling	Unbalanced
SimCLR + MorphCL	50.82 ±0.50	50.67 ±0.28	50.57 ±0.60
MTL + MorphCL	50.53 ±0.45	50.42 ±0.51	50.28 ±0.48

Table 17: Downstream F1 scores for different balancing strategies for the MorphCL loss pretraining on the CAPTURE-24 dataset.

Method	Balanced + Oversampling	Balanced + Undersampling	Unbalanced
SimCLR + MorphCL	50.97 ±0.23	51.59 ±0.65	52.08 ±0.57
MTL + MorphCL	52.12 ±0.33	52.31 ±0.57	52.24 ±0.60

magnitude smaller than the SSL objectives. Results in Table 18 show that unweighted summation ($\lambda = 1$) of MorphCL with MTL shows no significant difference compared to weighted sums ($\lambda > 1$), while inclusion of the loss term ($\lambda > 0$) shows a clear improvement. This shows the robustness of our results reported in the main paper to the scaling factor of our proposed MorphCL loss.

Table 18: Ablation on different loss scaling factors for MorphCL loss term on WISDM dataset.

$\lambda = 0$	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 20$
47.41 ±0.22	50.53 ±0.45	49.96 ±0.99	49.33 ±0.66	48.98 ±0.84

B.7 Extended Qualitative Analysis

To underline qualitative results we provide additional plots extracted from linear probing experiments using a ViT-S pretrained on CAPTURE-24 using either the (i) MTL, (ii) MorphCL or (iii) MTL + MorphCL as loss function. We provide extracted embeddings UMAP features from sample participants of the Hang-Time (see Figure 6) and RWHAR dataset (see Figure 4) along with confusion matrices (see Figure 5) that underline the improved recognition performances and decreased confusion among even difficult classes. Table 19 further provides all computed silhouette coefficients (SC) computed on the extracted embeddings across all participants of a dataset per pretrained model.

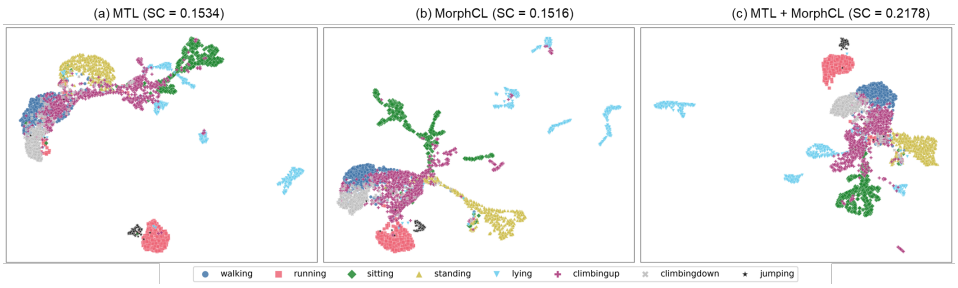

Figure 4: UMAP features and Silhouette coefficients (SC) extracted from a ViT-S pretrained on CAPTURE-24 applied to the RWHAR dataset using (i) MTL, (ii) MorphCL and (iii) MTL + MorphCL as pretext task. One can see a clear improvement when combining MorphCL with MTL with even kinematically close classes such as walking stairs ("climbingup"/"climbingdown") and walking are better separated.

B.8 SupCon vs. Group-based averaging

As discussed in the introduction and related work of this paper, both Khosla et al. (2020) and Pillai et al. (2025) rely on computing losses based on an inherent grouping among the samples during their SSL approaches. With Khosla et al. (2020) relying on actual class labels to define the grouping, Pillai

Table 19: Silhouette coefficients (SC ∈ [-1, 1], higher is better) based on activity classes computed on embeddings extracted from pretrained models using each tested SSL framwork (with and without MorphCL). We report scores based on all participants in each downstream dataset using cosine similarity and excluding background classes if present. Adding MorphCL increases the mean silhouette coefficient across datasets for each evaluated configuration.

Framework	WEAR	Wetlab	PAMAP2	Hangtime	RWHAR	WISDM	*Avg*
Encoder: DeepConvLSTM							
Pretraining: WISDM							
MTL	-0.1776	-0.2189	0.0533	-0.0471	-0.0639	-0.2035	-0.1096
MTL + MorphCL	-0.2019	-0.2025	0.1088	-0.0493	-0.0872	-0.1682	-0.1001
SimCLR	-0.1990	-0.2002	0.0449	-0.0742	-0.0928	-0.1814	-0.1171
SimCLR + MorphCL	-0.1906	-0.1964	0.0741	-0.0780	-0.1096	-0.1961	-0.1161
REBAR	-0.2003	-0.1809	0.0713	-0.0204	-0.1492	-0.1862	-0.1109
REBAR + MorphCL	-0.1681	-0.1947	0.0698	-0.0810	-0.1039	-0.1723	-0.1084
RelCon	-0.1803	-0.2131	0.0383	-0.0598	-0.1068	-0.1926	-0.1190
RelCon + MorphCL	-0.2039	-0.2199	0.0648	-0.0561	-0.0983	-0.1697	-0.1139
Encoder: ViT-S							
Pretraining: WISDM							
MTL	-0.1826	-0.1274	0.1650	-0.1269	-0.0887	-0.1309	-0.0819
MTL + MorphCL	-0.1155	-0.0845	0.2326	-0.0444	-0.0163	-0.0912	-0.0199
SimCLR	-0.0270	-0.0906	0.1833	-0.0534	-0.0468	-0.1614	-0.0326
SimCLR + MorphCL	-0.0135	-0.0966	0.2321	0.0110	0.0337	-0.0983	0.0114
DenoisingAE	-0.1140	-0.1854	0.0271	0.0766	-0.0658	-0.1648	-0.0710
DenoisingAE + MorphCL	-0.0022	-0.1440	0.0991	0.0437	-0.0098	-0.1360	-0.0249
MaskedAE	-0.1728	-0.2079	0.0627	0.0208	-0.1098	-0.1949	-0.1003
MaskedAE + MorphCL	-0.0236	-0.0661	0.2075	-0.0071	0.0444	-0.0535	0.0169
REBAR	-0.1541	-0.1383	0.0460	-0.2109	-0.0296	-0.1546	-0.1069
REBAR + MorphCL	-0.0429	-0.0846	0.2100	-0.0525	0.0208	-0.0682	-0.0029
RelCon	-0.2373	-0.2158	0.0838	-0.0550	-0.0971	-0.1789	-0.1167
RelCon + MorphCL	-0.0739	-0.1378	0.1177	0.0323	-0.0612	-0.1009	-0.0373
Pretraining: CAPTURE-24							
MTL	-0.0484	-0.2233	0.1206	0.0053	-0.0972	-0.1813	-0.0707
MTL + MorphCL	-0.0195	-0.0564	0.3024	-0.0436	0.0160	-0.0213	0.0296
SimCLR	-0.0265	-0.0611	0.3186	-0.1853	-0.0170	-0.0252	0.0006
SimCLR + MorphCL	-0.0171	-0.0498	0.2893	-0.1576	-0.0102	-0.0140	0.0068
DenoisingAE	-0.0856	-0.1759	0.1162	-0.0177	-0.1154	-0.1202	-0.0664
DenoisingAE + MorphCL	-0.0170	-0.0684	0.3211	-0.0333	0.0484	-0.0368	0.0357
MaskedAE	-0.1598	-0.2133	0.0468	0.0638	-0.1263	-0.2060	-0.0991
MaskedAE + MorphCL	0.0027	-0.0582	0.2968	-0.1234	0.0317	-0.0399	0.0183
REBAR	-0.3077	-0.2113	0.0675	-0.2416	-0.1708	-0.3194	-0.1972
REBAR + MorphCL	-0.1188	-0.0972	0.2125	-0.1885	-0.0864	-0.1385	-0.0695
RelCon	-0.2234	-0.1757	0.0012	0.0457	-0.0166	-0.2586	-0.1045
RelCon + MorphCL	-0.0199	-0.1248	0.2279	-0.1514	-0.0528	-0.0831	-0.0340

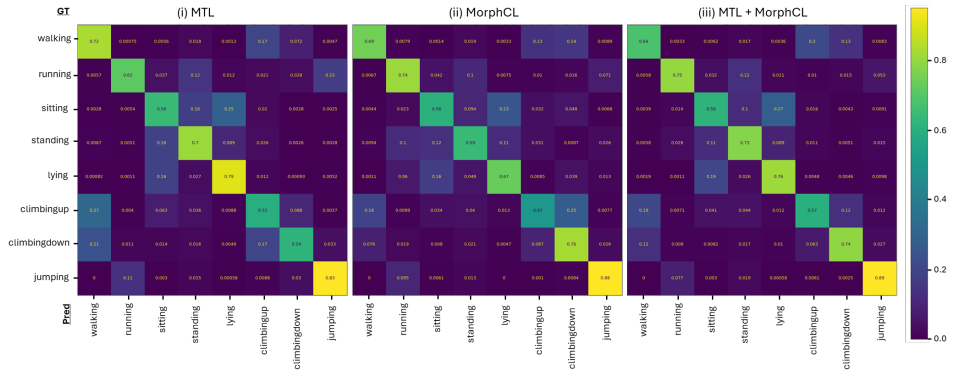

Figure 5: Linear probing results of a ViT-S pretrained on CAPTURE-24 applied to the RWHAR dataset using (i) MTL, (ii) MorphCL and (iii) MTL + MorphCL as pretext task. When combining MorphCL with MTL overall confusion across both kinematically close (e.g., ”jumping” and ”running”) is decreased.

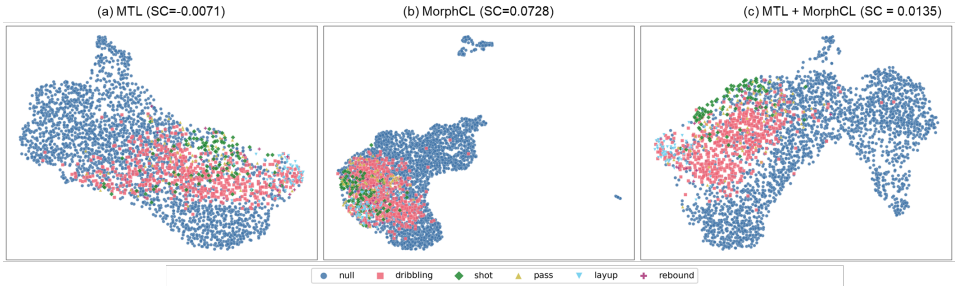

Figure 6: UMAP features and Silhouette coefficients (SC) extracted from a ViT-S pretrained on CAPTURE-24 applied to the Hang-Time dataset using (i) MTL, (ii) MorphCL and (iii) MTL + MorphCL as pretext task. Even with the presence of a dominant background class ("null"), combining MorphCL with MTL still lead to overall better separation of all classes.

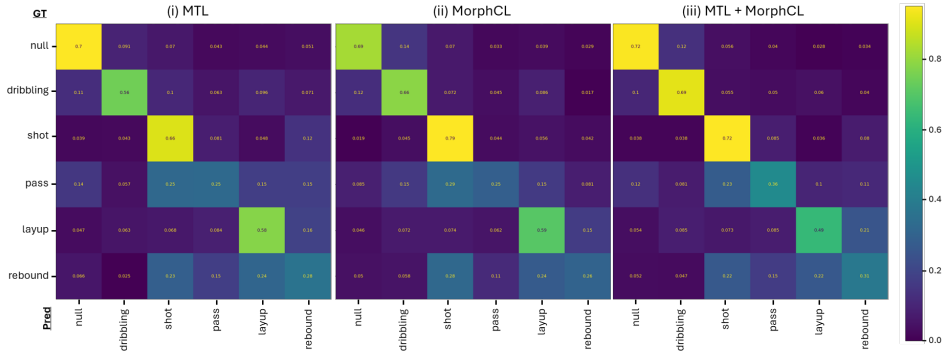

Figure 7: Linear probing results of a ViT-S pretrained on CAPTURE-24 applied to the Hang-Time dataset using (i) MTL, (ii) MorphCL and (iii) MTL + MorphCL as pretext task. One can see a clear improvement when combining MorphCL with MTL with even sporadic classes like passes being recognizable.

et al. (2025) use a pre-computed measure (sVRI) and group together records based on a pre-defined binning of said measure. With their methods inherently differing from our MorphCL framework they further apply different loss calculations based on the grouping. Khosla et al. (2020) uses an instance-based formulation in which the loss is computed independently for each positive pair. For a given anchor, the denominator aggregates similarity scores over all other samples in the batch regardless of label, while the numerator isolates the similarity to a single positive instance. The resulting per-pair log-probabilities are then averaged first over all positives of each anchor, and subsequently over all valid anchors in the batch. Pillai et al. (2025) adopt a structurally similar per-pair formulation, computing the loss as the difference between the raw similarity of each positive pair and the log-sum-exp of similarities to all non-self samples, averaged uniformly across all positive pairs in the batch rather than first averaging within each anchor. In both cases the loss treats each positive pair independently, normalizing against the full batch. Table 20 compares results of switching our proposed loss computation with the losses as used by Khosla et al. (2020) and Pillai et al. (2025) within our framework. One can see that our group-based loss formulation (with few exceptions) results in the on average highest F1-scores across the different downstream experiments. Note that results were computed same as the main results of this paper and are also seed-averaged across three runs.

Table 20: Ablation study on using losses as reported by Khosla et al. (2020) and Pillai et al. (2025). Results are averaged across the MorphCL framework itself as well as its combination with the different SSL frameworks reported in the main paper. Best results are in **bold**.

	WEAR	Wetlab	PAMAP2	Hangtime	RWHAR
Encoder: DeepConvLSTM; **Pretraining:** WISDM					
Khosla et al. (2020)	44.98 ±0.34	23.89 ±0.27	63.40 ±0.43	34.80 ±0.42	64.68 ±0.71
Pillai et al. (2025)	44.94 ±0.30	23.96 ±0.23	63.34 ±0.44	34.59 ±0.41	64.25 ±0.48
Ours	46.50 ±0.30	23.94 ±0.24	64.39 ±0.50	35.06 ±0.40	64.29 ±0.55
Encoder: ViT-S; **Pretraining:** WISDM					
Khosla et al. (2020)	53.20 ±0.52	25.75 ±0.53	66.77 ±0.89	35.71 ±0.86	68.18 ±0.87
Pillai et al. (2025)	53.05 ±0.39	26.06 ±0.62	66.68 ±1.09	35.52 ±0.75	67.64 ±1.13
Ours	53.28 ±0.30	26.13 ±0.62	65.89 ±0.65	36.85 ±0.64	68.25 ±1.47
Encoder: ViT-S; **Pretraining:** CAPTURE-24					
Khosla et al. (2020)	53.46 ±0.42	25.75 ±0.63	66.88 ±0.54	37.24 ±0.81	66.58 ±0.58
Pillai et al. (2025)	53.44 ±0.40	25.46 ±0.77	66.56 ±0.86	37.70 ±1.05	66.90 ±0.90
Ours	54.36 ±0.34	26.45 ±0.78	67.07 ±0.46	37.99 ±1.11	65.10 ±0.97