Open-World Continuous Human Activity Recognition with Novel Class Discovery

Jacky Hao Jiang
jackyjiang@utexas.edu
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas, USA

Edison Thomaz
ethomaz@utexas.edu
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas, USA

Abstract

Human activity recognition (HAR) is typically developed under a closed-set assumption that all activity classes are known a priori. This is unrealistic for long-term wearable sensing, where users adopt new routines, perform unseen activities, and exhibit behavioral variation across individuals, devices, and environments. While open-set methods can detect unfamiliar samples and continual-learning methods can incorporate new classes, both generally assume that novel samples have already been identified and grouped into meaningful categories. We instead formulate HAR as an open-world lifelong learning problem that unifies known-activity recognition, unknown detection, novel-activity discovery, and continual adaptation in a single streaming framework. We propose DOW (Decoupled Open-World), which separates recognition from discovery: a classification embedding supports prototype-based recognition and novelty scoring, while a discovery embedding clusters samples rejected as unknown into candidate activity classes that are validated and incrementally integrated through replay, distillation, and prototype updates. Across the WISDM, PAMAP2, and DSADS datasets and under within-user, cross-user, and pooled-user streaming settings, DOW-exclude achieves the strongest average post-promotion Macro-F1, while full DOW remains competitive with—and often surpasses—oracle incremental-learning baselines despite solving the harder problem of autonomous activity discovery. These results identify autonomous discovery as the primary bottleneck toward practical open-world HAR.

CCS Concepts

• **Human-centered computing** → **Ubiquitous computing**; • **Computing methodologies** → *Cluster analysis.*

Keywords

human activity recognition, open-world learning, continual learning, novel class discovery, open-set recognition, wearable sensing, sensor-based activity recognition



This work is licensed under a Creative Commons Attribution 4.0 International License.
HumanSys '26, Austin, TX, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2959-1/2026/10
https://doi.org/10.1145/3842436.3843804

1 Introduction

Human activity recognition (HAR) underpins wearable and mobile sensing applications such as health monitoring, fitness tracking, and context-aware assistants. Most HAR models, however, are trained under a closed-set assumption—all classes appearing at deployment are known during training—which is hard to satisfy in long-term use, where systems meet new routines, unseen activities, and distribution shifts across users, days, devices, and environments.

Existing paradigms address only parts of this problem. Open-set recognition can reject samples that do not match the currently known classes, where rejection means routing a low-confidence window to an unknown outcome rather than assigning it to one of the active known classes [4, 5, 14], but rejection alone does not explain what an unknown represents or how to learn from it. Continual learning can incorporate new classes over time, but typically assumes new-class samples are already identified, grouped, or labeled before adaptation [6, 8, 10, 13]. This removes the critical upstream challenge in wearable HAR: deciding which incoming samples are unknown, which unknowns belong together, and when a recurring unknown behavior should become a new activity class.

We therefore study HAR as an open-world lifelong learning problem that couples four usually-separate challenges in a single streaming pipeline: known-class recognition, unknown detection, novel-activity discovery, and continual integration. The model is trained only on an initial set of base classes $\mathcal{Y}_0$ and then receives a task-free stream of unlabeled windows. At any time t, the active known label set $\mathcal{Y}_t$ contains the base classes and any novel classes already promoted by the system. A stream window is therefore known if its activity belongs to $\mathcal{Y}_t$; otherwise, it is currently unknown to the model, even though it may later become known after discovery and promotion.

Our DOW framework rests on one key design: decoupling recognition from discovery. Because embeddings optimized for known-class classification need not preserve the cluster

structure of unseen activities, DOW uses class prototypes—representative embeddings of activity classes—for recognition and novelty scoring, and a separate discovery embedding for grouping buffered unknowns. The main contributions of this work are:

- We propose DOW, a decoupled prototype-based framework with separate classification and discovery embeddings: the former supports recognition and novelty scoring, the latter preserves cluster-friendly structure for grouping rejected samples.
- We introduce a dual-embedding overcomplete discovery and integration pipeline that clusters buffered rejected samples by first over-segmenting discovery embeddings with local K-means centers, then merging centers with highly overlapping soft assignments to form flexible candidate activity clusters for promotion and continual integration.
- We evaluate DOW on three public HAR benchmarks (WISDM, PAMAP2, DSADS) under streaming protocols that separate oracle incremental learning from autonomous open-world adaptation.

2 Related Work

Continual and online HAR adapts recognition models to nonstationary sensor streams while preserving performance on previously learned activities, increasingly under task-free streaming settings where old and new activities arrive without explicit task boundaries [15, 16]. Existing class-incremental methods, including rehearsal-, distillation-, and prototype-based approaches [1, 6, 8, 10, 13], provide strong continual-learning baselines but assume that novel-class samples are already identified, grouped, or labeled. In contrast, our setting requires the system to autonomously discover recurring novel activities before learning them.

Open-set recognition addresses samples outside the currently known class set, which in our framework evolves as newly discovered activities are promoted. Prior work has characterized open-space risk [14], demonstrated the limitations of softmax confidence for rejecting windows that cannot be confidently assigned to any active known class [4], and framed recognition under incomplete class knowledge as a fundamental challenge [5]. However, rejection alone is insufficient for lifelong HAR, where unknown samples may correspond to recurring activities that should become new classes. Novel class discovery groups unlabeled samples into coherent categories [9], but existing methods are largely offline or batch-based.

Prototype-based methods are particularly attractive for open-world HAR because prototypes provide compact, extensible class memories: new activities can be incorporated by estimating prototypes from promoted samples, while distance to existing prototypes naturally supports novelty detection. Yet embeddings optimized for classification do not necessarily preserve structure among unseen activities. We therefore decouple recognition and discovery, using a classification space for prediction, novelty scoring, and prototype maintenance, and a separate discovery space for clustering buffered unknown samples into candidate novel activities.

3 Method

Figure 1 summarizes the pipeline. The model is initialized on base classes $\mathcal{Y}_0$; during deployment, it maintains an active known label set $\mathcal{Y}_t$ that expands over time. Each incoming window is either recognized against prototypes of the current active classes or routed to UNK when it cannot be confidently matched. The discovery module periodically clusters these rejected samples, validates stable candidate groups, promotes recurring novel activities into $\mathcal{Y}_t$, and integrates them through continual finetuning.

3.1 Representation Learning

Each input sample is a fixed-length sensor window constructed from the sensing modality provided by each dataset. The encoder operates on whatever channels each benchmark exposes, normalized using statistics from the initial training split only.

The encoder first maps each window x_i into a classification embedding $z_{\text{cls}}^{(i)}$. This embedding is used for prototype-based classification, novelty scoring, and prototype refresh. For samples that are rejected as unknown, a projection head maps the classification embedding into a normalized discovery embedding:

$$z_{\text{disc}}^{(i)} = \frac{g_\phi\left(z_{\text{cls}}^{(i)}\right)}{\left\| g_\phi\left(z_{\text{cls}}^{(i)}\right) \right\|_2},$$

where $g_\phi(\cdot)$ is an MLP projection head. Thus, the discovery embedding is not computed directly from the raw sensor signal; it is a normalized projection of the classification embedding. During streaming inference, z_{disc} is only required for samples that are routed into the unknown buffer.

Base training follows three stages. First, the encoder is warmed up using supervised classification on base classes. Second, class prototypes are initialized from training embeddings in the classification space. Third, the model is trained with a prototype-based classification objective, combined with discovery-space regularization:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}.$$

Here, $\mathcal{L}_{\text{CE}}$ is the prototype classification loss in the classification space, $\mathcal{L}_{\text{sup}}$ encourages class-consistent structure in the

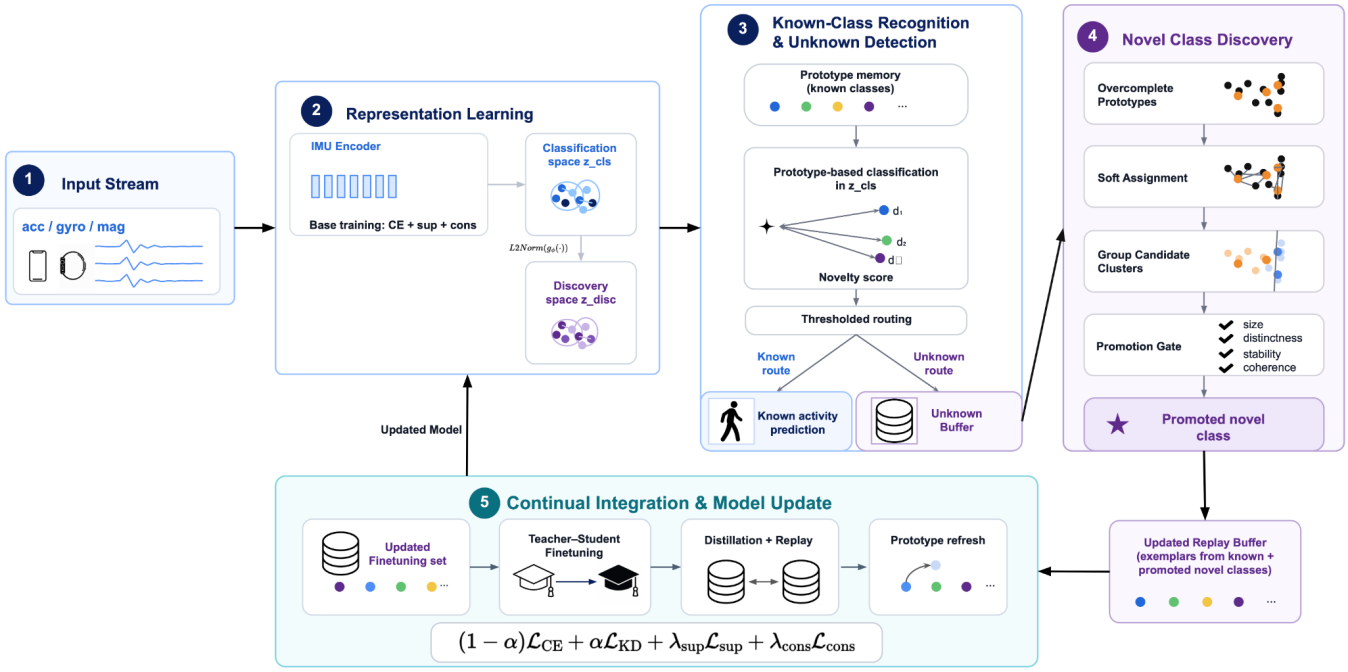


Figure 1: Overview of the proposed open-world lifelong HAR framework. A streaming sensor window from the input stream is first encoded into classification and discovery embeddings through representation learning (1, 2). Prototype-based known-class recognition then assigns high-confidence samples to known activities, while low-confidence samples are rejected into an unknown buffer for later processing (3). Buffered unknown samples are clustered in the discovery space, validated by a promotion gate, and promoted as novel activity classes (4). Promoted classes are finally incorporated into the model through continual integration with replay, distillation, and prototype refresh (5).

discovery space, and $\mathcal{L}_{\text{cons}}$ stabilizes discovery embeddings under small input perturbations.

3.2 Known-Class Recognition and Unknown Routing

The model maintains a multi-prototype memory for the active label set $\mathcal{Y}_t$. For each class c, we collect the classification embeddings of available training or replay samples from that class and run K-means in the classification space to obtain a prototype set:

$$\mathcal{P}_c = \{p_{c,1}, \ldots, p_{c,M_c}\}.$$

Using multiple prototypes allows the memory to represent intra-class variation, which is common in sensor-based HAR because the same activity may be performed at different speeds, by different users, or under different sensor placements.

For an incoming window x_i, the encoder first produces a classification embedding $z_{\text{cls}}^{(i)}$. The distance from the sample to class c is defined as the distance to the nearest prototype of that class:

$$d_{ic} = \min_k \left\| z_{\text{cls}}^{(i)} - p_{c,k} \right\|^2,$$

and the predicted known class $\hat{y}_i$ is the one that minimizes d_{ic}. The same nearest-prototype distance is used for unknown detection. If the minimum distance to the active prototype memory is below a threshold γ, the sample is treated as consistent with a known activity and assigned to $\hat{y}_i$. Otherwise, the sample is rejected as unknown:

$$r_i = \begin{cases} \hat{y}_i, & \min_{c \in \mathcal{Y}_t} d_{ic} \le \gamma, \\ \text{UNK}, & \min_{c \in \mathcal{Y}_t} d_{ic} > \gamma. \end{cases}$$

The threshold γ is selected using validation data from the base classes to balance known-class retention and unknown sensitivity.

Rejected samples are stored in the unknown buffer together with their raw window, classification embedding, and normalized discovery embedding. The discovery embedding is obtained by passing the classification embedding through the projection head defined in Section 3.1. The raw window is used for later continual finetuning, the classification embedding is used to initialize prototypes for promoted classes and

compare candidate clusters against known classes, and the discovery embedding is used to cluster buffered unknown samples.

3.3 Novel Class Discovery and Integration

Unknown rejection alone is insufficient for open-world HAR because rejected samples may correspond to recurring activities rather than isolated anomalies. The discovery module therefore transforms recurring patterns among rejected samples into new activity classes. Discovery operates periodically on the unknown buffer. Discovery embeddings are over-segmented using K-means to generate local centers, and rejected samples are softly assigned using cosine similarity. Centers with highly overlapping assignments are merged to form candidate clusters. Candidate clusters are promoted when they satisfy recurrence, distinctiveness from existing classes, temporal stability, and internal coherence criteria. Promoted clusters are assigned new labels and integrated into the active label set. New-class prototypes are initialized by clustering the promoted samples in the classification space and adding the resulting prototypes to the multi-prototype memory.The model is then finetuned on promoted samples $\mathcal{D}$new and replay data $\mathcal{D}$replay using replay and knowledge distillation:

$$\mathcal{L}_{\mathrm{CL}} = (1-\alpha)\mathcal{L}_{\mathrm{CE}} + \alpha\mathcal{L}_{\mathrm{KD}} + \lambda_{\mathrm{sup}}\mathcal{L}_{\mathrm{sup}} + \lambda_{\mathrm{cons}}\mathcal{L}_{\mathrm{cons}}.$$

Here, $\mathcal{L}_{\mathrm{KD}}$ preserves prior knowledge, while the discovery-space losses maintain cluster structure after adaptation. Following finetuning, prototypes and replay memory are updated, and streaming inference resumes with the expanded label set.

4 Experimental Setup

We use a two-stage protocol: pre-deployment training, then task-free streaming. Classes are split into *base* and *novel*; novel classes are excluded from *all* pre-deployment steps, training, validation, prototype computation, novelty threshold calibration, and held-out base-class construction. For the main comparison against prior methods, we use a 70%/30% base/novel split by class count. As an openness analysis, we additionally sweep the number of base classes from 1 to $C-1$. Base-class windows from training-eligible users are split approximately 50%/10%/40% into initial-training, validation, and held-out-base sets; the validation split is used for model selection, threshold calibration, and later known-validation retention checks, but is never used for streaming adaptation or replay, and novel-class windows are never seen during this stage.

We evaluate three user-level settings. *Within-user (WU)* restricts the pool to a single user and measures personalized open-world adaptation. *Cross-user (XU)* trains on source users and evaluates on held-out users—no held-out-user data is used for training, validation, normalization, prototypes, or calibration—measuring generalization to unseen users. *Pooled-user (PU)* still initializes on source-user base data but streams a mixture of source- and held-out-user windows (both base and novel), reflecting a heterogeneous multi-user deployment.

For streaming, base-class and novel-class windows are shuffled by default so that initially trained activities and held-out activities interleave rather than arrive as clean task blocks, $\mathcal{S}_{\mathrm{stream}} = \mathcal{S}_{\mathrm{base}} \cup \mathcal{S}_{\mathrm{novel}}$.

4.1 Baselines and Comparison Protocol

Datasets and preprocessing. We evaluate DOW on three public sensor-based HAR benchmarks: WISDM [7], PAMAP2 [11, 12], and DSADS [2, 3]. WISDM contains phone accelerometer recordings from 36 users over 6 activities; we use the raw three-axis acceleration signal sampled at 20 Hz. PAMAP2 contains multi-sensor recordings from hand, chest, and ankle IMUs over 12 activities from 9 users; we use 27 inertial channels, including accelerometer, gyroscope, and magnetometer signals sampled at 100 Hz. DSADS contains body-worn inertial measurements from 8 users over 19 daily and sports activities, recorded at 25 Hz. For all datasets, we segment streams into 3 s windows with 50% overlap: 60-sample windows for WISDM, 300-sample windows for PAMAP2, and 75-sample windows for DSADS. Normalization statistics are computed only from base-class training windows and are then applied unchanged to validation, replay, and streaming windows.

Dataset-specific user splits follow the WU, XU, and PU protocols defined above. For WISDM, XU and PU use 28 source users and 8 held-out users. For PAMAP2, WU and XU use 8 users while PU uses all 9 users. For DSADS, all settings use the 8 available users.

To isolate the effect of decoupling recognition from discovery, we also include *DOW-no-dual*, an ablation of DOW that removes the separate discovery embedding. DOW-no-dual uses the classification embedding for both prototype-based recognition and clustering of rejected samples, while keeping the same unknown routing, overcomplete clustering procedure, promotion criteria, replay, distillation, and prototype refresh. This ablation is evaluated under the same autonomous open-world protocol as full DOW.

Within each setting, all methods share preprocessing, base /novel and user splits, initial-training and validation data, normalization, stream order, and the starting base-class checkpoint. The shared recognizer is a prototype-based model whose IMU branch stacks three 1D-convolutional layers, a two-layer bidirectional GRU, and attention pooling, producing a 256-dimensional embedding classified by nearest-prototype distance. Unless a method requires otherwise,

all use batch size 32, Adam at 10^{-3}, and 50 initial-training epochs.

4.2 Evaluation Metrics

We report streaming Macro-F1, which weights classes equally and is therefore sensitive to rare activities. Samples are evaluated relative to the active label set when they arrive: base-class and post-promotion novel samples must be classified as their ground-truth activity, while pre-promotion novel samples are correct only when identified as unknown.

We report three evaluation views. **Oracle incremental-learning (IL) baselines** use ground-truth novel-class groupings before adaptation and are evaluated only on samples whose classes are already active. **DOW-exclude** uses the same post-introduction recognition-only view, but class introduction is performed autonomously by DOW, so incorrect or mistimed promotion can still hurt performance. **Full DOW** evaluates all stream samples and therefore captures known-class recognition, unknown detection, novel-class discovery, and post-promotion recognition.

To assess catastrophic forgetting, we additionally track *known-validation retention*. At the beginning of each experiment, the initial base-class data is split into training, validation, and stream portions. Macro-F1 on the fixed validation set containing only the original base classes, evaluated after initial training, after each promotion update, and after the stream ends. Drops in this score indicate forgetting of the original known classes.

5 Results and Conclusion

Table 1 summarizes streaming Macro-F1. DOW-exclude achieves the highest average post-promotion performance across all evaluation settings, demonstrating that once novel activities are discovered and promoted, the framework integrates them effectively with existing classes. Full DOW performs lower, as expected, because it is evaluated on the entire open-world stream, including the more difficult pre-promotion period. The gap between DOW and DOW-exclude indicates that autonomous discovery remains the primary challenge.

Comparing DOW with DOW-no-dual highlights the benefit of decoupling recognition and discovery. While both perform similarly within-user, DOW-no-dual degrades more noticeably under cross-user and pooled-user evaluation, suggesting that embeddings optimized for classification do not reliably preserve the structure needed for novel activity discovery under distribution shift.

Figure 2 shows that performance generally declines as openness increases. Nevertheless, DOW and DOW post-discovery remain among the strongest methods across most base/novel partitions, particularly on DSADS and PAMAP2.

Table 1: Streaming Macro-F1 across DSADS, PAMAP2, and WISDM. Avg. is the mean over the three datasets within each setting; best per column in bold. Proposed DOW variants are bolded in the Method column.

Method	DSADS	PAMAP2	WISDM	Avg.
Within-user				
DOW	0.8510	0.6976	0.6029	0.7171
DOW-exclude	**0.8546**	**0.7636**	**0.7107**	**0.7763**
DOW-no-dual	0.8419	0.6539	0.6280	0.7079
LwF [8]	0.7413	0.5677	0.4621	0.5904
Replay [13]	0.8336	0.6516	0.5354	0.6735
iCaRL [10]	0.7916	0.5122	0.4876	0.5971
ILOS [6]	0.8344	0.6240	0.5191	0.6592
LAPNet-HAR [1]	0.3869	0.3192	0.2571	0.3211
Cross-user				
DOW	0.6148	0.5180	0.4410	0.5246
DOW-exclude	**0.6320**	**0.6317**	**0.5855**	**0.6164**
DOW-no-dual	0.5194	0.3946	0.3459	0.4200
LwF [8]	0.6066	0.4287	0.4030	0.4794
Replay [13]	0.5969	0.5072	0.4330	0.5124
iCaRL [10]	0.6157	0.4447	0.3717	0.4774
ILOS [6]	0.6083	0.4657	0.3905	0.4882
LAPNet-HAR [1]	0.5037	0.3930	0.3450	0.4139
Pooled-user				
DOW	0.7561	0.5978	0.4197	0.5756
DOW-exclude	**0.7619**	**0.6460**	0.4468	**0.6092**
DOW-no-dual	0.7256	0.5545	0.3607	0.4715
LwF [8]	0.5786	0.4410	0.2648	0.4281
Replay [13]	0.6777	0.5334	**0.4881**	0.5664
iCaRL [10]	0.5764	0.4169	0.3893	0.4609
ILOS [6]	0.6393	0.4853	0.4380	0.5209
LAPNet-HAR [1]	0.5077	0.3950	0.3322	0.4116

Figure 3 further shows that DOW exhibits the least forgetting on the fixed known-validation set, indicating that replay, distillation, and prototype refresh effectively preserve previously learned classes while incorporating new ones.

Several limitations remain. Discovery quality depends on unknown detection, clustering, and promotion thresholds, and our evaluation uses benchmark-derived streams rather than longitudinal deployments. Future work should explore adaptive calibration, temporal modeling, lightweight user feedback, and real-world wearable studies. Overall, DOW advances HAR beyond fixed activity vocabularies by unifying open-set recognition, autonomous discovery, and continual learning within a single lifelong framework.

References

[1] Rebecca Adaimi and Edison Thomaz. 2022. Lifelong Adaptive Machine Learning for Sensor-Based Human Activity Recognition Using Prototypical Networks. *Sensors* 22, 18 (2022), 6881. doi:10.3390/s22186881

[2] Billur Barshan and Kerem Altun. 2013. Daily and Sports Activities. UCI Machine Learning Repository. doi:10.24432/C5C59F Dataset.

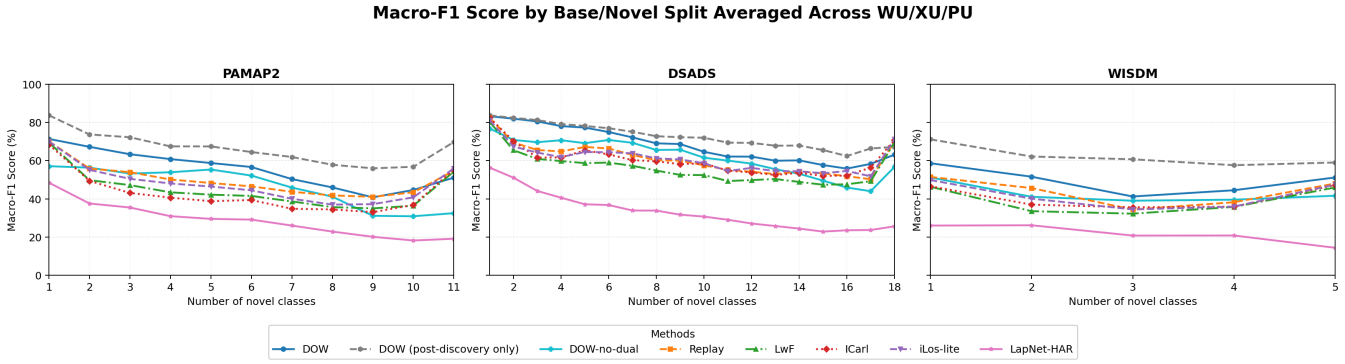


Figure 2: Macro-F1 under different base/novel class partitions. DOW denotes the full open-world streaming pipeline, while DOW post-discovery only corresponds to the DOW-exclude evaluation view.

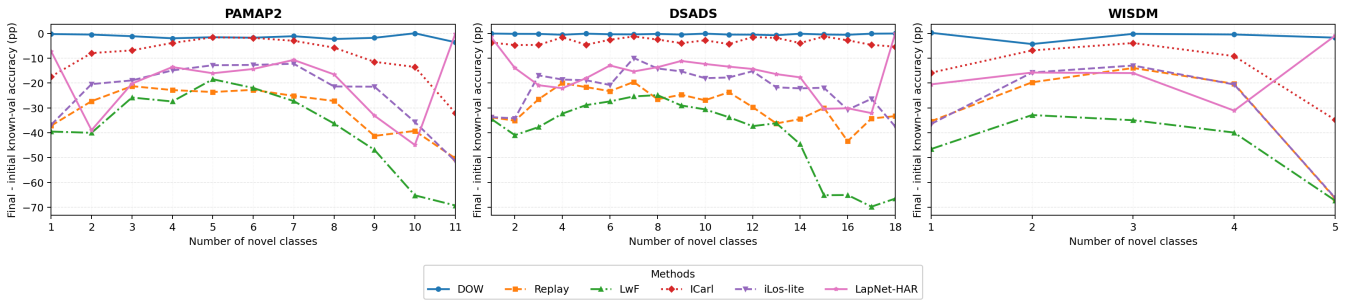


Figure 3: Catastrophic forgetting measured as final minus initial known-validation accuracy, averaged across WU, XU, and PU. Values closer to zero indicate stronger retention of the original known classes, while more negative values indicate greater forgetting.

[3] Billur Barshan and Murat Cihan Y"uksek. 2014. Recognizing Daily and Sports Activities in Two Open Source Machine Learning Environments Using Body-Worn Sensor Units. *Comput. J.* 57, 11 (2014), 1649–1667. doi:10.1093/comjnl/bxt075

[4] Abhijit Bendale and Terrance E. Boult. 2016. Towards Open Set Deep Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. arXiv:1511.06233

[5] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. 2020. A Survey on Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 10 (2020), 3614–3632.

[6] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. 2020. Incremental Learning in Online Scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13926–13935.

[7] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity Recognition Using Cell Phone Accelerometers. *ACM SIGKDD Explorations Newsletter* 12, 2 (2011), 74–82. doi:10.1145/1964897.1964918

[8] Zhizhong Li and Derek Hoiem. 2018. Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2018), 2935–2947. doi:10.1109/TPAMI.2017.2773081

[9] Jiaming Liu, Yangqiming Wang, Tongze Zhang, Yulu Fan, Qinli Yang, and Junming Shao. 2023. Open-world Semi-supervised Novel Class Discovery. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*.

[10] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. 2017. iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2001–2010.

[11] Attila Reiss. 2012. PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. doi:10.24432/C5NW2H Dataset.

[12] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *Proceedings of the 16th IEEE International Symposium on Wearable Computers*. 108–109. doi:10.1109/ISWC.2012.13

[13] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*, Vol. 32. 348–358.

[14] Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. 2013. Toward Open Set Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 7 (2013), 1757–1772. doi:10.1109/TPAMI.2012.256

[15] Martin Schiemer, Lei Fang, Simon Dobson, and Juan Ye. 2023. Online Continual Learning for Human Activity Recognition. *Pervasive and Mobile Computing* 93 (2023), 101817. doi:10.1016/j.pmcj.2023.101817

[16] Yao Zhang, Souza Leite Clayton, and Yu Xiao. 2025. Efficient Online Continual Learning in Sensor-Based Human Activity Recognition. arXiv:2511.05566 [cs.CV]