Article

Lifelong Adaptive Machine Learning for Sensor-Based Human Activity Recognition Using Prototypical Networks

Rebecca Adaimi * and Edison Thomaz

Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712, USA
* Correspondence: rebecca.adaimi@utexas.edu

Abstract: Continual learning (CL), also known as lifelong learning, is an emerging research topic that has been attracting increasing interest in the field of machine learning. With human activity recognition (HAR) playing a key role in enabling numerous real-world applications, an essential step towards the long-term deployment of such systems is to extend the activity model to dynamically adapt to changes in people's everyday behavior. Current research in CL applied to the HAR domain is still under-explored with researchers exploring existing methods developed for computer vision in HAR. Moreover, analysis has so far focused on task-incremental or class-incremental learning paradigms where task boundaries are known. This impedes the applicability of such methods for real-world systems. To push this field forward, we build on recent advances in the area of continual learning and design a lifelong adaptive learning framework using Prototypical Networks, *LAPNet-HAR*, that processes sensor-based data streams in a task-free data-incremental fashion and mitigates catastrophic forgetting using experience replay and continual prototype adaptation. Online learning is further facilitated using contrastive loss to enforce inter-class separation. *LAPNet-HAR* is evaluated on five publicly available activity datasets in terms of its ability to acquire new information while preserving previous knowledge. Our extensive empirical results demonstrate the effectiveness of *LAPNet-HAR* in task-free CL and uncover useful insights for future challenges.

Keywords: continual learning; lifelong learning; prototypical networks; catastrophic forgetting; intransigence; task-free; incremental learning; online learning; human activity recognition

1. Introduction

Advances in mobile devices, sensors and computational methods over the last decade have brought Human Activity Recognition (HAR) to the forefront. Recently, HAR has begun to play a central role in enabling numerous emerging real-world applications ranging from mobile health [1,2] and disorder diagnosis [3] to personal assistance and smart environments [4,5]. For instance, wearable sensors have been featured in automated dietary monitoring to track food intake gestures [6] and in health assessment applications to track sedentary activities of post-surgical patients [7]. However, despite the progress the field has experienced, HAR is hampered by important learning shortcomings. HAR models today are typically built and evaluated with data offline under the assumption that the environment, people and their corresponding behaviors and activities remain constant over time. While this has been essential in establishing a methodological foundation for HAR and steady progress is made in the field, state-of-the-art approaches remain unable to close the gap between validation in controlled conditions and the realities and challenges of the real world. Life is dynamic, forcing people to change careers, move to a new city and develop new hobbies. Starting a family and adopting a pet introduces new habits and routines, and unexpected health issues might pose mobility and cognitive challenges that cause significant changes in how one performs everyday activities. A smart kitchen assistant should be able to recognize cooking activities even when individuals move to a new home. Similarly, a health monitoring system should be able to accurately

track older adults' behaviors and assess their ability to live independently even after the onset of impairments that cause them to move more slowly. Consider a smartphone application designed to monitor and provide just-in-time interventions to individuals suffering from post-traumatic stress disorder (PTSD). Ideally, the application should be adaptive and responsive to patients' behaviors, health states and medication changes over time. The COVID-19 outbreak provides another example of the need for human-centered systems that adapt to changing conditions. In a matter of months, the pandemic deeply altered human behaviors and routines around the world; millions of people changed jobs and social patterns, found new ways to exercise, learned new hobbies and much more.

A key challenge in building adaptive HAR systems is that trained models must incorporate new knowledge while avoiding forgetting previously learned knowledge. Neural networks tend to exhibit loss of previously acquired knowledge when incorporating new information, a phenomenon known as *catastrophic forgetting (CF)*. Finding this balance between neural stability in remembering old classes and plasticity in learning new classes is the main hurdle for continual learning, which is a key ingredient for a long-term sustainable HAR system. To address this challenge, we leverage *Continual Learning (CL)*, a learning paradigm that incrementally extending a neural network's acquired knowledge, whether learning new input domains or new classes and tasks. Continual learning, which has also been referred to as lifelong learning [8] or incremental learning [9], has been attracting increasing interest in the field of machine learning, with researchers developing different methods for mitigating catastrophic forgetting when incrementally learning new tasks over time [10,11]. These approaches have achieved promising results in computer vision [12,13] and robotics [14].

While CL has shown much promise in the context of HAR [15,16], emphasis has been given to CL approaches that are either task- or class-incremental. Task-incremental setting refers to data arriving sequentially in batches with one batch corresponding to one task, which is usually comprised of two or more classes. In other words, the model observes all data belonging to one task and is optimized for it before moving to another task. These assumptions confine the scope and applicability of a continual learning system since, in a real-world setting, it not only encounters new classes but also revisits old classes in a non-uniform dynamic way. Moreover, new activity classes are often introduced with a limited number of data samples, and the system needs to incrementally learn these new activities from a few samples in order to recognize them again at later times. With that in mind, we advocate for a framework that processes data in an *online task-free incremental* fashion with the ability to incorporate new classes with limited data. Leveraging incremental prototypical networks [17], we utilize metric-based learning to optimize class prototype representations in an online fashion. Prototype networks are compelling because they support new class representations with limited data as well as online prediction using nearest mean classification. Moreover, it provides a more general learning setting that eliminates the notion of tasks.

In this paper, we propose a Lifelong Adaptive machine learning framework for HAR using Prototypical Networks, *LAPNet-HAR*. We demonstrate that the framework (1) processes data streams in a task-free data-incremental online fashion, (2) incorporates a prototype memory scheme that evolves continually with the data stream enabling both online learning and prediction paradigm, (3) mitigates CF by employing experience replay using replay memory, and (4) prevents prototype obsolescence with replay-based prototype adaptation. We perform an extensive analysis of our framework across multiple datasets measuring not only performance but also trade-offs between learning new knowledge versus retaining previously acquired information. The contributions of this paper can be summarized as follows:

- A continual learning model for HAR using Prototypical Networks, *LAPNet-HAR*, that incrementally evolves with streaming data in a task-free data-incremental paradigm and mitigates catastrophic forgetting using experience replay with continual prototype evolution.

- An extensive evaluation of *LAPNet-HAR* on five commonly used HAR datasets. The evaluation has demonstrated the effectiveness of *LAPNet-HAR* in continual activity recognition as well as in handling the catastrophic forgetting and model intransigence trade-off across all datasets. This further uncovered findings in dealing with various challenges of task-free continual learning in HAR.

2. Related Work

Wearable devices have become ubiquitous in people's daily lives with the rising need for continuous human monitoring. This has led to an increasing demand for developing advanced learning algorithms to tackle practical challenges and introduce new state-of-the-art wearable capabilities in various fields. A fitting example is the COVID-19 outbreak that pushed researchers to develop new wearable devices and monitoring systems [18]. Deep learning techniques have also gained increased attention in clinical medicine in a wide range of applications including screening and triage, diagnosis, and treatment recommendation [19,20]. While all this work has helped advance several areas related to human monitoring, it has also highlighted the issue of model generalizability when faced with changing human behavior and environment. Thus, recent work have shifted focus towards developing systems and models that continually learn from new data and people. For example, Amrollahi et al. proposed a continual learning algorithm that can incrementally learn from new patient data for sepsis prediction [21]. Continual learning has also been studied in other fields, such as navigation [22], text-to-speech synthesis [23] and prosthetic control [24], to just name a few. In the field of navigation for instance, a robot needs to adapt when facing different navigation environments, a problem known as lifelong navigation [22]. In the speech domain, continual learning enables extending a multi-speaker text-to-speech system with new speakers without forgetting previous speakers, thus improving speaker diversity [23].

While wearables, deep learning and continual learning are important concepts used in various domains, in this work, we focus on the problem of continual learning especially for human activity recognition. As such, we further discuss, in Sections 2.1 and 2.2, previous work that relate to our *LAPNet-HAR* framework, specifically work in incremental/continual learning for sensor-based human activity recognition as well as metric-based learning using prototypical networks.

2.1. Continual Learning in HAR

While most HAR-related prior work have mainly focused on offline supervised training, several methods with incremental learning abilities have been developed in the past years [25,26]. This incremental learning paradigm has not only been used to improve activity recognition performance [27], but also to personalize HAR models across users by adapting to distribution shifts [26–28]. For example, Learn++ was first introduced by Poliker et al. [9] and used in [29] to improve activity recognition systems when dealing with differences across individuals. Learn++ is an ensemble-based incremental learning algorithm that applies data weighting based on classification performance. There is also an increasing amount of work devoted to discovering and recognizing new activities in HAR. One popular approach is the use of clustering and one-class classifiers wherein a new cluster and classifier is added for every activity [30,31], enabling incremental learning. Gjoreski et al. [30] applied an agglomerative clustering technique to perform real-time clustering of streaming data. Using temporal assumptions on human activities, short outliers were filtered out allowing a more accurate discovery of meaningful clusters. Wu et al. [31] further extended this method with a self-supervised learning of activities labeled through one-shot interaction which optimizes the accuracy and user burden trade-off.

However, simply creating a cluster or one-class classifier for every new activity while keeping data representation fixed can lead to the overlapping of clusters, which often would require re-training the model. Thus, another main component towards continual learning is evolving the HAR models with new data and activities. Fang et al. [32] employed a von

Mises–Fisher-based hierarchical mixture model to model every type or pattern of activity as a component of the mixture model. Cheng et al. [33] presented a zero-shot learner to adapt to new activities with limited training data. In this setup, semantic relationship between high-level activities and low-level sensor attributes are encoded using a knowledge-driven model; domain experts and developers must manually add new attributes and update the activity–attribute matrix for every new activity.

All methods discussed thus far require a certain amount of re-engineering effort and therefore do not enable the automatic evolution of an activity model. For an end-to-end continual HAR system, a form of feedback loop is required where a new activity is discovered and fed into a network for evolution in an automatic and continual fashion while mitigating catastrophic forgetting of previously learned activities. Jha et al. [15,16] explored whether continual learning techniques previously used in computer vision, and which possess these attributes, can be applied to HAR. They performed a comprehensive empirical analysis of several regularization-based and rehearsal-based approaches on various HAR datasets showing the challenges faced when moving towards sensor-based time series data. Moving towards dynamic architectures for activity models, Ye et al. [34] proposed EMILEA, a technique to evolve an activity model over time by integrating dynamic network expansion Net2Net [35] for enhancing the model's capacity with increasing number of activities and Gradient Episodic Memory [36] for mitigating the effect of catastrophic forgetting. They further proposed another continual activity recognition system, HAR-GAN, that leverages Generative Adversarial Networks (GAN) to generate sensor data on previous classes without the need to store historical data [37].

It is clear that the field is still under-explored and researchers studying continual learning in HAR have barely scratched the surface. To date, proposed methods, such as EMILEA [34] and HAR-GAN [37], have relied on dynamically expanding network architectures with increasing number of classes. While shown effective, this leads to increasing number of model parameters with the growing set of activities, which can be limiting for a deployable resource-constraint solution. In contrast with the typical softmax classifier, *LAPNet-HAR* has a prediction head with fixed capacity, by applying representation learning of class prototypes for nearest-mean classification. This inherently supports a large number of classes without changing the network architecture. Moreover, while task-incremental learning has dominated continual learning HAR research, it is not a good match for real-world situations in which old activities are revisited and new activities are introduced in a non-uniform way and with limited training data. Thus, HAR systems with a (1) continual, (2) online, and (3) task-free learning paradigm offer increased real-world practicality and applicability. A step forward towards such a system was presented in iCARL [12] for class-incremental learning. While it is also prototype-based, it requires task boundaries for knowledge distillation. Furthermore, iCARL applies a computationally expensive herding technique, which requires recalculation of the feature means on each change of the memory size or network parameters. *LAPNet-HAR*, on the other hand, applies continual prototype adaptation which enables online prediction and also a contrastive representation loss that improves inter-class separation with the addition of new classes.

2.2. Metric-Based Learning Using Prototypical Networks

Metric-based approaches aim to learn a set of embedding functions and a metric space that measures the similarity between support and target samples which in turn is used for classification. Researchers have proposed several ways to learn this metric space. For instance, Koch et al. [38] proposed a deep convolutional siamese network to compute a similarity score between the image pairs. The learned distance was then used to solve the one-shot learning problem via a K-nearest neighbors classification. Vinyals et al. [39] presented an end-to-end trainable matching network that applies k-nearest neighbors using the cosine distance on the learned embedding features. Snell et al. [40] extended the matching network and proposed the prototypical network that uses the Euclidean distance to learn prototype representations of each class. Breaking away from traditional few-shot

learning, Ren et al. [41] extended the prototypical networks to semi-supervised few-shot learning. Incremental few-shot learning introduces the ability to jointly predict based on a set of previously learned classes as well as additional novel classes [42,43]. Moving towards online contextualized few-shot learning, Ren et al. [17] proposed using a contextual prototypical memory for encoding contextual information and storing previously learned classes. De Lange et al. [44] extended prototypical networks to an online data incremental learning setting with prototypes continually evolving with non-i.i.d data streams.

Leveraging prototypical networks from online continual few-shot learning [17,44], our work aims to build and analyze such a system for lifelong adaptive learning in HAR. Utilizing prototype memory with prototype evolution, our *LAPNet-HAR* framework continually learns class representations and optimizes them using online averaging to account for distribution shift. Moreover, prototype adaptation is applied to accommodate changes in the latent space when new classes are introduced. This, in turn, prevents prototypes from becoming outdated (prototype obsolescence) which leads to incorrect class representation in the embedding space. To the best of our knowledge, we are also the first to analyze a continual learning system for HAR in a completely task-free continual setting. Table 1 compares our proposed approach with past HAR-related CL work based on various properties of a continual learning paradigm.

Table 1. Comparison of past CL paradigms applied to the HAR domain vs. our approach.

Method	Continual Learning	Online Evaluation	Experience Replay	Task-Free Learning
CL-HAR-1 [15]	✓	✗	✗	✗
CL-HAR-2 [16]	✓	✗	✓	✗
EMILEA [34]	✓	✗	✓	✗
HAR-GAN [37]	✓	✗	✓	✗
LAPNet-HAR (ours)	✓	✓	✓	✓

3. LAPNet-HAR Framework

In this work, we introduce *LAPNet-HAR* that enables continual adaptive learning for sensor-based data. We define continual learning in activity recognition as incrementally learning new patterns belonging to both old and novel activities in a sequential manner starting from a base model trained to classify a set of base classes. Figure 1 depicts an overview of *LAPNet-HAR*. To reiterate, continual learning is enabled using (i) a prototypical network with prototype memory for incremental learning, (ii) experience replay of a small number of old classes' data stored in a replay buffer to prevent catastrophic forgetting, (iii) continual replay-based prototype adaptation to prevent prototype obsolescence, and (iv) a contrastive loss [45] that pushes samples from different classes apart to improve inter-class separation.

The workflow starts with an initial model f_{θ_0} and a set of base data $D_0 = (\{(x_i, y_i)\}_{i=1}^N, C_{base})$ where, x_i is a raw data frame with y_i its corresponding label and C_{base} is a set of base activities. During the continual learning process, a batch of training data $D_t = (X_t, Y_t)$ is presented at every time step t. The goal for *LAPNet-HAR* is to continually learn and adapt a time-series sensor-based model f_{θ_t} with new incoming data while mitigating catastrophic forgetting. A secondary but important goal is learn with as few examples as possible. Algorithm 1 presents the overall *LAPNet-HAR* framework. In the following, we give the algorithmic details of the key components of *LAPNet-HAR*.
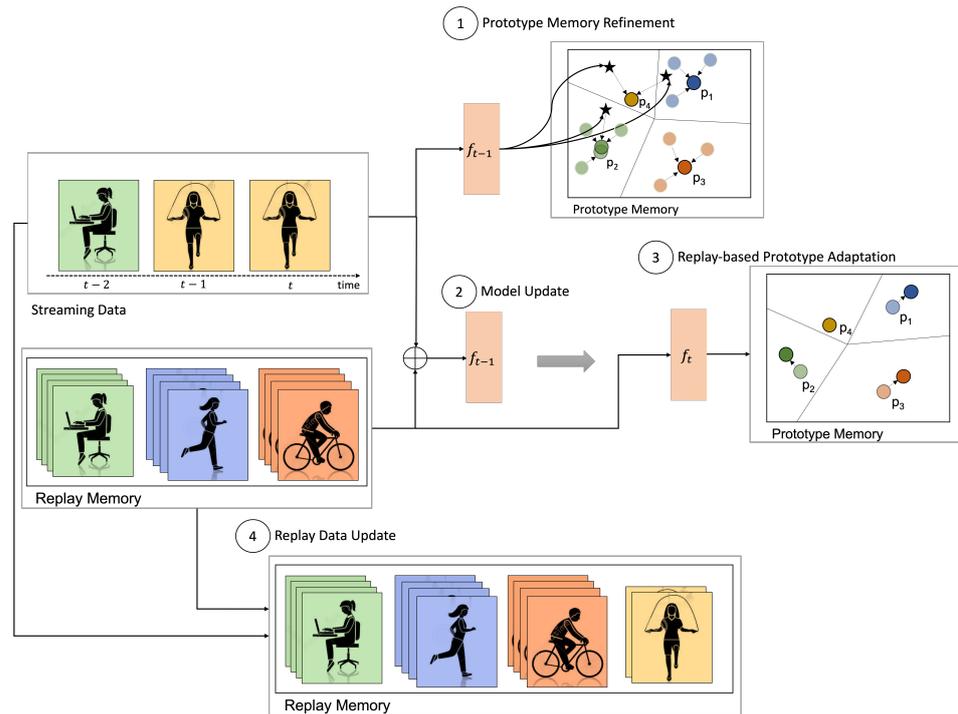
Figure 1. *LAPNet-HAR* Overview. Given a pretrained model at time $t-1$ with a prototype memory containing prototypes of old classes as well as a replay buffer with a subset of old data samples, the new batch of streaming data is first used to update the prototype memory by using online averaging on old classes and creating prototypes of new classes (step 1). Then, with the updated prototypes, the data is combined with the replay data to update the model parameters (step 2). To prevent prototype obsolescence and adapt to the new latent space, the prototypes are adapted using the new embeddings of the replay data (step 3). Lastly, the replay memory is updated by random sampling from the current batch and extended to include new classes (step 4).

Algorithm 1: LAPNet-HAR

Input: Initial network parameters θ_0, hyperparameter α, replay buffer size/class b
Data: Training data $(X, Y) = D$
$(X_0, Y_0) = D_0$ `// get base data for pretraining`
C_{base} = set of base classes in D_0
$M_p = \{\}; M_r = \{\}$ `// initialize empty prototype memory and replay`
 `buffer`

```
/* offline pretraining process                                          */
```
Update θ_0 with (X_0, Y_0)
Store prototypes p_k for $k \in C_{base}$ in M_p using D_0
Sample data from D_0 and store in M_r for replay

while *continually learning* **do**
    $(X_t, Y_t) = D_t$
    $M_p \leftarrow \text{UpdatePrototypeMemory}(D_t, M_p, \theta_{t-1})$ `// Figure 2a &`
     `Equation (3)`
    $Q \leftarrow D_t \cup M_r; (X_q, Y_q) = Q$ `// form combined query set`
    Incur loss $\mathcal{L}(f_{\theta_{t-1}}(X_q), Y_q)$ `// Equation (5)`
    Update model θ_t with (X_q, Y_q)
    $M_p \leftarrow \text{PrototypeAdaptation}(M_r, M_p, \theta_t, \alpha)$ `// Equation (6)`
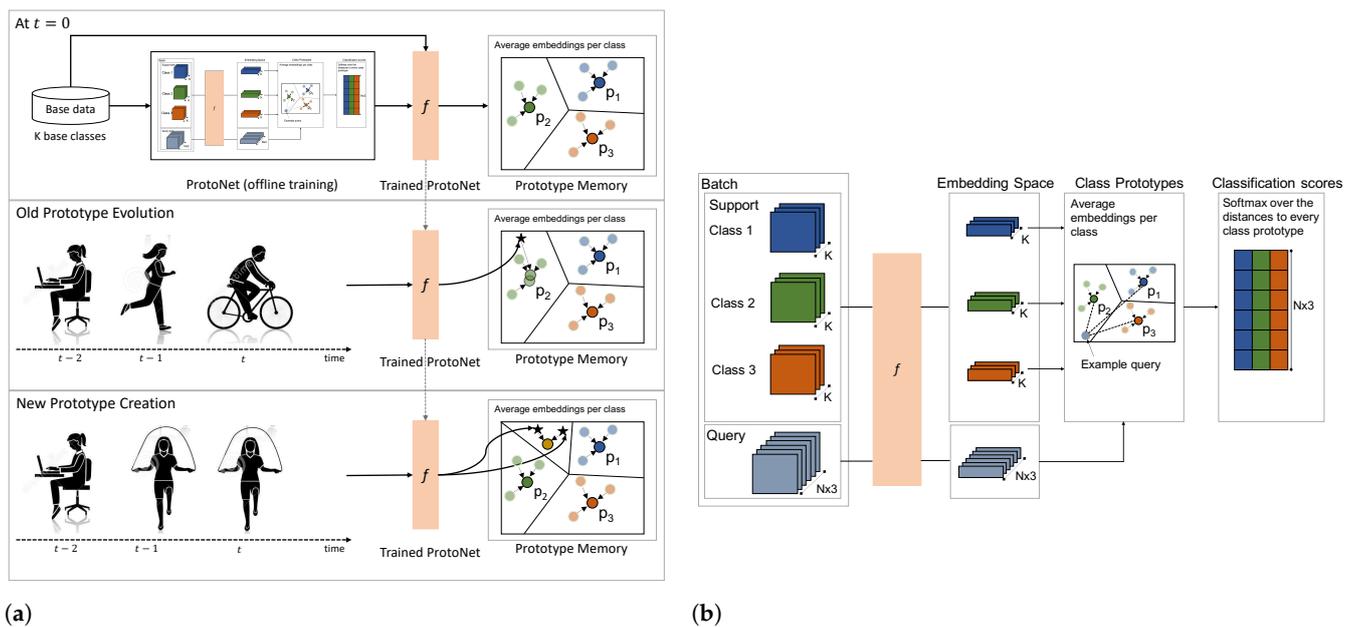 $M_r \leftarrow \text{UpdateReplayBuffer}(Q, M_r, \theta_t, b)$
end

(**a**)   (**b**)

Figure 2. (**a**) Incremental prototype memory refinement: old prototype evolution using online averaging and new prototype creation when encountering a new activity. (**b**) Offline Pretraining of Prototypical Network using standard few-shot learning process.

3.1. Prototypical Network with Prototype Memory

First, an evident requirement of our continual learning approach is the ability to keep track of classes learned over time without having to expand the network architecture. To that end, in order to form a knowledge base memory that keeps track of previously learned classes as well as enabled learning from limited number of samples, we leveraged Prototypical Networks [40] adapted for incremental continual learning (Figure 2). This type of network learns a metric space prototype representation for each class. It is adapted to an online setting by storing previously learned class prototypes in a prototype memory. The prototype memory serves as a knowledge memory of previously seen classes that are each represented by a prototype vector corresponding to the average representation of all support samples of the corresponding class.

3.1.1. Offline Pretraining

The first step of our framework is to create a base model able to classify a set of base classes. More specifically, in our continual learning scenario, we assume at time $t = 0$ that a subset of classes C_{base} and data samples D_0 are available to train a base network f_{θ_0} following the traditional few-shot learning process (Figure 2b).

To reiterate, a prototypical network constructs an embedding space wherein samples cluster around a prototype representation for each class. Training is done episodically wherein, in each episode, a set of classes is provided with a few labeled samples per class. This forms the *support* set S used to form the prototype $p[k]$ of each class k which is defined as the average of all samples belonging to that class:

$$p[k] = \frac{1}{|S_k|} \sum_{x \in S_k} f_{\theta_0}(x) \qquad (1)$$

Given the prototypes of each class, another unlabeled *query* set is used for evaluation. The prototypical network produces a distribution over the classes in memory for the current query sample $f_{\theta_0}(x)$ based on a softmax over the distances to the class prototypes in the

embedding space (Equation (2)). The training is then performed by minimizing the negative log-likelihood probability of assigning the correct class.

$$\hat{y}_k = softmax(-\left\| f_{\theta_0}(x) - p[k] \right\|^2) \tag{2}$$

It is important to note that in offline pretraining at every training episode, a support set and query set are randomly sampled every time from the current batch. While this has inspired several meta-learning algorithms, this setup does not allow classes learned in one episode to be carried forward to the next, which is critical for continual incremental learning. Thus, we leverage an incremental learning paradigm where classes learned are stored in a *Prototype Memory* (M_p) [17]. As such, after training our base model f_{θ_0} on D_0, we form a base prototype memory M_p that holds base class prototypes computed using all the data samples of each class in D_0. More specifically, all data samples in D_0 belonging to a class k are passed through the pretrained model f_{θ_0}, and their average is computed to form the corresponding class prototype that is stored in M_p. This is depicted in the top part of Figure 2a.

3.1.2. Incremental Prototype Memory Refinement

At this point, we have a base model f_{θ_0} trained to classify the base classes C_{base} for which a corresponding prototype is stored in M_p. Now, while the prototype memory allows keeping track of learned classes, when switching to processing data streams in an online fashion, another obvious requirement is to keep the prototype memory up-to-date by progressively refining it with new incoming data. At every time step t, we assume we already have stored a few prototypes corresponding to previously encountered classes in the memory represented by $p_t[k]$ where k denotes the corresponding class. With class prototypes estimating the average representation of a class across all support samples, they can be incrementally updated at every time step using online averaging. This can be considered similar to incremental real-time clustering. Given input features $f_{\theta_t}(X)$ for input batch X at time t, the prototype is updated using the following:

$$p_t[k] = \frac{c_{t-1}[k]}{c_t[k]} p_{t-1}[k] + \frac{f_{\theta_t}(X) \mathbb{1}[Y = k]}{c_t[k]} \tag{3}$$

where $c_t[k] = c_{t-1}[k] + |f_{\theta_t}(X) \mathbb{1}[Y = k]|$ is a count scalar indicating the number of samples added up to time t. This enables classes learned to be carried forward incrementally to future time steps. For novel classes, their prototypes are created by averaging their corresponding embedded samples in the current batch and storing them in the prototype memory. This is further illustrated in Figure 2a.

3.2. Continual Prototypical Network with Replay Memory

While prototype memory refinement allows to incrementally update the prototypes with new data and new classes, the feature embedding space initially learned by the base model was trained only on the base classes. Thus, when new classes are presented, we observe an incompatible feature space between the new and base classes. This is illustrated in Figure 3a where the base classes are well-separated whereas the feature spatial distribution of the new class is wider and overlaps with the base classes. This poses a challenge in the classification of both new and base classes. It is evident that the feature extractor needs to learn a new joint embedding space that clusters both old and new classes. Updating the model with streaming data using simple online finetuning narrows down the feature spatial distribution of new classes but loses the discriminative clustering of base classes (Figure 3b).
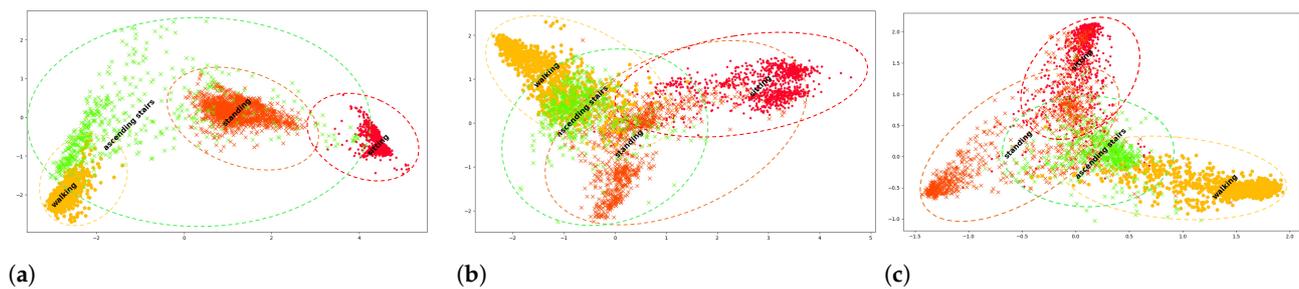
(**a**)　　　　　　　　　　　　(**b**)　　　　　　　　　　　　(**c**)

Figure 3. Feature embeddings of base classes—walking (yellow), sitting (red), standing (orange)—from PAMAP2 dataset learned during offline pretraining and new unseen class—ascending stairs (green). (**a**) After offline pretraining on base classes only: we observe how the feature spatial distribution of new classes is incompatible with the base classes that are well-distinguished and separated. (**b**) After online finetuning of base and new classes: we observe overlap in the clusters belong to base classes compared to offline pre-training but a narrowing in the feature spatial distribution of the new class. (**c**) After continual learning of base and new classes using *LAPNet-HAR*: compared to online finetuning, we observe a more compact feature distribution when applying contrastive loss. We still observe some overlap in the class clusters especially for classes that share similar characteristics (such as sitting and standing). We observe some improvement in the new class cluster being separated from the base clusters.

In order to solve the above challenges and force the model to learn a joint embedding space for old and new classes in a continual learning process, we proposed to employ a replay memory M_r that holds a subset of previously encountered data samples per class. In our implementation, we applied a random sampling approach of previously observed data, wherein a fixed small subset of samples per class is stored in memory. This replay memory, in turn, is used in the three remaining main components of our *LAPNet-HAR* framework: experience replay, continual replay-based prototype adaptation, and learning using a contrastive loss. The interplay between these components together was shown to improve the model's ability to learn new classes while also mitigating catastrophic forgetting of old classes. In our framework, catastrophic forgetting is incurred at the network parameter level and the prototype representation level. *Experience Replay* addresses forgetting at the network level by retaining observed data distribution, while *replay-based prototype adaptation* addresses forgetting at the stored representation level preventing prototypes from becoming obsolete. In Figure 3c, we observe some improvement in the clustering of base and new classes when applying *LAPNet-HAR*. It is important to keep in mind that the clustering performance differs for different classes depending on how distinct they are from each other. In Figure 3c, the classes *sitting* and *standing* can share similar characteristics that makes it more challenging to distinguish them apart. In what follows, we present and discuss the utility of each component of *LAPNet-HAR*.

3.2.1. Experience Replay

Experience replay presents a well-known method for addressing catastrophic forgetting by mixing exemplar data from previous classes with current data. Re-training the network with previous data helps constrain the network update from becoming biased towards new classes. It also enables the model to better discriminate between both old and new classes. However, this method requires extra memory space for storing exemplar samples per class which can be challenging for on-the-edge applications with limited resources. In our framework, we demonstrate that we do not need to store many exemplar samples in memory to achieve the desired solution. The replay memory M_r is designed to store an equal number of samples per class up to a fixed maximum size (six samples/class). Moreover, we employ a random sampling approach for selecting the exemplar samples, as it has been shown to be enough for selecting replay samples [16], and thus avoid computationally expensive herding techniques. At the end of every time step, the replay memory is updated by randomly sampling exemplar samples per class from the current batch and

the current data stored in M_r. It is also extended when new classes are observed in the current batch.

3.2.2. Contrastive Loss

As mentioned in Section 3.1.1, the prototypical network f_θ is optimized to learn a feature embedding space that clusters data samples closer to their corresponding class prototypes. This is done by minimizing the negative log-likelihood. While this enables the network to learn an embedding space in which several points cluster around a single prototype representation for each class, during the continual learning process, the data distribution shifts and the addition of new classes incurs drastic changes in the feature space, as was observed in Figure 3b. Therefore, one question that remains, *can we improve the embedding representations learned to adapt to the changing feature space during continual learning?*

Contrastive loss has been shown effective in learning semantic embeddings by leveraging pairwise relations of data samples [45,46]. It has been shown to be more effective than cross entropy loss in terms of its robustness to noisy labels and enhancement of class margins. By leveraging additional semantic similarities between data samples, the model can learn more discriminative embedding representations for each class. This also helps in faster online adaptation when encountering new class samples. Moreover, contrastive loss have been shown to overcome learning capabilities of traditional cross-entropy based classification especially when dealing with small number of training samples [38]. This is particularly useful in our continual learning setting where only a small batch of data is introduced at every time step. There are several formulations of contrastive loss based on metric distance learning or triplets [46,47] that have been used to learn powerful representations. From our observations in Figure 3b, our focus was to push dissimilar samples belonging to different classes (old and new classes) apart while also ensuring similar samples are pulled closer together. Towards that end, we employ a contrastive loss (Equation (4)) [45] that forces the network to narrow down the feature spatial distribution of classes while enlarging the distances among the different classes to be greater than a margin value $m > 0$. It is important to note that, while there have been several recent contrastive loss formulations for different purposes [48,49], we observed superior performance using the margin-based contrastive loss. Combined with experience replay, the contrastive loss was observed to further enforce inter-class variance between old and new classes. Enforcing inter-class variance reduces interference of new class prototypes on old classes, but at the same time, helps in faster learning of new classes.

Thus, given a current batch of data $B = D_t \cup M_r$ that includes new samples mixed with replayed samples, similar and different samples are grouped into positive and negative pairs using the ground truth labels. Thus, for a pair of samples (x_i, y_i) and (x_j, y_j),

$$\mathcal{L}_{contrastive} = \begin{cases} \left\| f_\theta(x_i) - f_\theta(x_j) \right\|^2 & if \ y_i = y_j \\ \max(0, m - \left\| f_\theta(x_i) - f_\theta(x_j) \right\|^2) & if \ y_i \neq y_j \end{cases} \tag{4}$$

Finally, the network parameters are updated by minimizing a loss function (Equation (5)) comprised of a softmax-based cross entropy loss encourages samples to be closer to their corresponding prototypes, and a contrastive loss that forces similar samples to be closer to each other (intra-class variance) while dissimilar samples to be at a distance greater than some margin m (inter-class variance).

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{contrastive} \tag{5}$$

3.2.3. Continual Replay-Based Prototype Adaptation

Lastly, with the embedding network evolving with new classes, it is important to prevent the class prototype representations already stored in memory from becoming obsolete, especially for classes not observed in the current batch. This is caused by the changes in

data distribution as the model is continually updated which can lead to catastrophic forgetting of old classes. Thus, after the model update and leveraging the replay memory M_r, we incorporate a replay-based approach to provide additional information about the current updated state of the embedding space and improving the class prototype representations. More specifically, we adapt a weighted update of previous $p_t[k]$ prototypes with replayed exemplars of classes stored in M_r which enables a stabilized prototype optimization. Given (X_r, Y_r) from the replay buffer M_r and network parameters f_{θ_t} at time t, the prototype $p_t[k]$ for class k is updated using the following:

$$p_t[k] = \alpha p_t[k] + (1 - \alpha) \frac{f_{\theta_t}(X_r) \mathbb{1}[Y_r = k]}{|f_{\theta_t}(X_r) \mathbb{1}[Y_r = k]|} \tag{6}$$

where α is a refresh ratio that acts as a learning rate that controls the amount of update applied to the prototypes to incorporate the new latent space while preserving information learned from the previous latent space. Equation (6) can be seen as equivalent to Equation (3) with $\alpha = \frac{c_{t-1}[k]}{c_{t-1}[k] + |h_t \mathbb{1}_{y_t = k}|}$. The difference between both equations is that $\alpha \in [0, 1]$ is a fixed hyperparameter in Equation (6). This helps in stabilizing the prototype trajectory in the ever-evolving latent space. In Section 6.1.2, we discuss and empirically analyze the effect of α on the continual learning process.

4. Experiments

The main objective the following experiments is to asses the effectiveness of our *LAPNet-HAR* framework in balancing the model's plasticity and stability with incremental streaming data of new as well as old classes. In this section, we detail the datasets used, the network architecture of the feature extractor, and the evaluation process. Source code for *LAPNet-HAR* and all the experiments is made available to the community at https://github.com/Human-Signals-Lab/LAPNet-HAR.git (accessed on 1 September 2022).

4.1. Datasets

To present a comprehensive analysis of our proposed framework, we assess its performance on five widely used, publicly available HAR datasets. Table 2 summarizes the main characteristics of each dataset.

Table 2. Summary of Selected Datasets.

Datasets	Activity Type	# of Sensor Channels	# of Classes	Balanced
Opportunity [50]	Daily Gestures	113	17	✗
PAMAP2 [51]	Physical Activities	52	12	✗
DSADS [52]	Daily & Sports Activities	45	19	✓
Skoda [53]	Car Maintenance Gestures	30	10	✗
HAPT [54]	Daily Activities & Postural Transitions	6	12	✗

4.1.1. Opportunity Dataset

The Opportunity dataset [50] is a public state-of-the-art HAR dataset. It comprises the readings of motion sensors recorded while 4 subjects executed typical daily activities. From the Opportunity challenge, we replicated the preprocessing steps and train/test split for recognizing 17 different gestures using 113 sensor channels. We applied a sliding window of 800 ms with 400 ms step size.

4.1.2. Physical Activity Monitoring Dataset (PAMAP2)

The PAMAP2 Physical Activity Monitoring dataset [51] contains 18 different physical activities captured using 3 inertial measurement units and a heart rate monitor from 9 subjects. Our implementation modeled 12 activity classes as defined in [51]. We split the dataset into training and testing by setting the data from participants 5 and 6 as the

test data while remaining was used for training. We applied a sliding window of 1 s with 500 ms step size.

4.1.3. Daily and Sports Activities Dataset (DSADS)

The DSADS dataset [52] comprises motion sensor data of 19 daily and sports activities performed by 8 subjects. Each activity was captured for 5 min at 25 Hz sampling frequency. The 5-min signals for each activity were divided into 5-s segments. The tri-axial acceleration, gyroscope, and magnetometer data was captured using 5 MTx orientation trackers placed on the torso, right arm, left arm, right leg, and left leg. The dataset was split into training and testing with the data from subject 7 and 8 used as test data.

4.1.4. Skoda Dataset

The Skoda dataset [53] contains data for 10 quality checks performed by assembly-line workers in a car maintenance scenario. The data was recorded using 10 body-worn accelerometers at 98 Hz sampling rate. The training set consists of 90% of data from each class, while the remaining 10% formed the test data. A 1 s sliding window is used to segment the data with 50% overlap.

4.1.5. Human Activities and Postural Transitions Dataset (HAPT)

The HAPT dataset [54] contains 12 daily activities collected from 30 subjects with a smartphone on their waist. The activities include basic static and dynamic activities such as standing, sitting, walking, etc. as well as postural transitions between static postures such as stand-to-sit, sit-to-stand, etc. Data captured 3-axial linear acceleration and 3-axial angular velocity at a constant rate of 50 Hz. A 2.56-s sliding window with 50% overlap was used to segment the data. The dataset was split into training and testing with the data from participants 29 and 30 used as test data.

4.2. Task-Free Data Incremental Protocol

Data incremental learning is a general paradigm that, unlike task-incremental learning, makes no assumption on the data stream. Thus, it provides a more practical and naturalistic scenario for continual learning systems since, at every time step, non-i.i.d data samples are encountered that may contain activity classes already observed before as well as completely new classes. Thus, in addition to addressing the catastrophic forgetting problem, we also consider concept drift [55] for previously learned classes due to the fact that data distribution in real life application dynamically change over time.

The goal is to continue to learn from new data streams starting from a base pretrained model trained on a set of base classes. We propose the following protocol: for a multi-class HAR dataset, classes are randomly split into a set of base classes and new classes. For each base class, the training data is further split into base training data used for offline pre-training and new training data used for streaming. After every time step, each new batch of data is used only once for training. Moreover, the updated model is evaluated on (1) test data containing the base classes only and (2) test data containing new classes already seen so far. In addition to evaluating the model's performance on new classes and base classes separately, we also measure the overall test accuracy on all classes seen so far (base and new combined).

In our experiments, we simulate a data stream where classes are presented randomly in a batch at every time step. We observed that simply applying random sampling over all classes sometimes resulted in a batch containing all classes and 1 sample per class, especially given the small batch size (e.g., 20 samples). Thus, in order to ensure a more practical streaming behavior, we restricted a batch to include at most 5 classes from remaining old and new classes combined.

4.3. Architecture and Training Configuration

While the network f_θ can be any type of architecture, in our experiments we apply the same architecture across all datasets in order to maintain a fair comparison. Deep-ConvLSTM [56], which has been widely used for HAR datasets, uses a combination of convolutional and recurrent layers. Raw sensor signals are processed by four 1D convolutional layers, of 64 filters with kernel size 1×5, which extract feature maps from the wearable data. Then, the feature maps are passed to two LSTM layers with 128 hidden units for sequential processing. The original DeepConvLSTM network includes a softmax classification layer at the end. However, in our prototypical network, the softmax layer is removed as the model is trained to learn a metric space in which classification can be performed by computing distances to the prototype representations of each class.

While maintaining a fair comparison premise across datasets by using the same network architecture, we empirically optimize hyperparameter configurations for each dataset to achieve the highest performance. Table 3 lists the optimized hyperparameters used for both pre-training and continual learning processes in our *LAPNet-HAR* framework. We discuss the effect of tuning some hyperparameters on the learning process in Section 6.

Table 3. Hyperparameter Configuration.

		Opportunity	PAMAP2	DSADS	Skoda	HAPT
Offline Pretraining	Batch size	200	200	200	200	200
	# Base classes	5	5	5	5	5
	% Training data/class	50%	50%	50%	50%	50%
	Epochs	100	100	100	100	100
	Learning rate	0.001	0.001	0.001	0.001	0.001
Continual learning	Batch size	20	20	20	20	20
	Replay buffer size/class	6	6	6	6	6
	Contrastive margin (m)	3	2	1	1	1
	Refresh ratio (α)	1.0	0.2	0.8	0.5	0.5
	Learning rate	0.001	0.001	0.001	0.001	0.001

4.4. Evaluation Metrics

In order to evaluate *LAPNet-HAR*'s ability to continually learn and adapt to new classes while retaining knowledge about previous classes, we compute several measures on the held-out test data.

4.4.1. Performance Measures

Using the held-out test data as explained in Section 4.2, we compute three performance measures, more specifically using average F1-score (Equation (7)): (1) *Base* performance which measures the model's ability to recognize the base classes C_{base} it was pretrained on, (2) *New* performance which measures the model's ability to recognize the new classes seen so far, and (3) *Overall* performance which measures the overall ability of the model to recognize both base and new classes seen so far. The base performance will indicate the model's stability while the new performance describes the plasticity of the model.

$$F1\text{-}score = \frac{2}{C} \sum_{i=1}^{C} \frac{prec_i \times recall_i}{prec_i + recall_i} \tag{7}$$

4.4.2. Forgetting Measure

Forgetting of a given task is defined as the difference between the maximum knowledge gained by the model about the task throughout the past learning process and the current knowledge the model has [57]. This metric helps provide a quantitative measure to better understand the model's stability and ability to retain knowledge about previous tasks.

While this measure was defined for a task-incremental learning scenario, we leverage the same formulation but adapt a modified forgetting measure that can be suitable for our data-incremental learning protocol. More specifically, in our learning setup, we do not deal

with incrementally adding tasks but rather incrementally adding data that can include new classes or new observations of old classes. Thus, after the model is trained incrementally till time step t, the forgetting score f_t^j is computed as the difference between the maximum performance gained for class j throughout the learning process up to $t - 1$ and the current performance for class j at time t. Adapting the normalized version introduced by [16], the forgetting measure is defined as:

$$f_t^j = 1 - \frac{a_{t,j}}{\max\limits_{l \in 1,\dots,t-1} a_{l,j}} \tag{8}$$

where $a_{t,j}$ is the performance measure for class j at current time step t. We compute the forgetting measure for each of the base classes the model was pretrained on. Then, we take the average forgetting measure across all base classes,

$$F_t = \frac{1}{|C_{base}|} \sum_{j \in C_{base}} f_t^j \tag{9}$$

A high forgetting measure indicates severe catastrophic forgetting of base classes.

4.4.3. Intransigence Measure

While the forgetting measure is crucial for understanding the catastrophic forgetting effect observed in continual learning systems, another key aspect that also needs to be analyzed is the model's ability to learn new classes. This indicates the plasticity of the model and typically interplays with the model's forgetting of old classes. Thus, analyzing both measures together provides a better understanding of our framework's effectiveness in continual learning. Again, we leverage the *intransigence* measure defined by Chaudhry et al. [57] that is defined as the inability of a model to learn new tasks.

Similar to our forgetting measure, we modify the intransigence measure to fit out learning setup by considering time steps instead of tasks. Therefore, we compare the performance measure of the model ($a_{t,j}$) at time t on new class j to a reference model's performance (a_j^*) on the same class. We define the reference model to be the prototypical network trained offline on all the training data that contains all classes. The reference model is basically the standard classification model that serves as the upper bound of the continual learning performance. Therefore, at time t, the intransigence measure for class j is,

$$I_t^j = a_j^* - a_{t,j} \tag{10}$$

Similar to Equation (9), we take the mean across the intransigence measures of all new classes excluding base classes. A model with lower intransigence score indicates better ability to learn new classes.

4.5. Baselines

We compare our framework against two baselines: (1) offline supervised, which implements a standard offline training of our prototypical network on all training datasets, and (2) online finetuning , which applies a simple update of the model with cross entropy loss using only the data as it is streamed without experience replay, prototype adaptation, and contrastive loss. Online finetuning is essentially the prototypical network with prototype memory, where the prototypes are updated online using online averaging as explained in Section 3.1.2. Establishing these two conditions as baselines allows us to assess how our method performs in relation to these two established measures. Thus, the offline supervised baseline serves as an upper bound while online finetuning baseline serves as a lower bound.

5. Results

In this section, we present the experimental results on the five HAR datasets. We perform ablation experiments on *experience replay*, *replay-based prototype adaptation*, and *contrastive loss*. The goal of these components in our *LAPNet-HAR* framework is to enable the model to continually learn new classes while mitigating catastrophic forgetting and adapt the feature embedding space to distribution shifts that occur during streaming. Thus, we evaluate the contribution of each component to the overall performance. It is important to note that the boost in the continual learning process is not explained by any single component of *LAPNet-HAR*, but by their composition.

5.1. Performance Analysis

Table 4 reports the mean and standard deviation of the *Base*, *New*, and *Overall* F1-score for each dataset over five runs. From the experimental results, we observe that the combination of *experience replay*, *replay-based prototype adaptation*, and *contrastive loss* significantly improves the recognition performance of new as well as base classes across all datasets. This in turn leads to higher *Overall* performance across all datasets when applying *LAPNet-HAR*.

As lower bound, the online finetuning baseline suffers the most catastrophic forgetting with *Base* performance dropping by 45% on average across all datasets compared to the offline upper bound *Base* performance. We observe that incorporating replay-based prototype adaptation (*LAPNet-HAR‡*) significantly reduced catastrophic forgetting of base classes compared to online finetuning specifically for PAMAP2, DSADS, and Skoda, while performance on new classes was comparable or less than the lower baseline. This demonstrates the benefit of updating and stabilizing the prototype representation progression as the embedding network evolves at every time step. However, the model's ability to learn new classes is still lacking. This can be due to the randomness in the class and data sequence at every time step which results in the scarcity of data samples when a new class is introduced. More specifically, the model does not converge on a new class or task before moving to the next one. While this reflects a more naturalistic learning scenario, this hinders the model's ability to properly separate the classes seen. This is in stark contrast to task-incremental learning, where all data samples of a new class or task is observed before moving forward. Employing experience replay to further mitigate catastrophic forgetting (*LAPNet-HAR†*) resulted in further increase in *Base* performance for all datasets, improving by a range of 14% to 23% compared to *LAPNet-HAR‡*. On the other hand, *New* performance slightly improved for HAPT, Skoda, and DSADS, while remained comparable for Opportunity and PAMAP2. In terms of the *Overall* performance, while *LAPNet-HAR†* significantly improved compared to online finetuning for all datasets, it was still low relative to the offline upper bound, reaching a minimum gap of 23%. This indicates the clustering ability of the model is weakened during continual learning of new and old classes. Thus, as mentioned previously, a contrastive loss is employed to improve inter-class separation during learning which is evident in the significant improvement in the *Base*, *New*, and *Overall* performances of *LAPNet-HAR* across all datasets. Skoda achieved an *Overall* performance of 74.78%, only a 6% difference compared to the offline performance. For Opportunity and HAPT, on the other hand, we observe lower performance indicating a more challenging continual learning process. This can be attributed to the highly imbalanced nature of the datasets, the larger number of activity classes, as well as the larger similarity and overlap in the classes. For example, the gestures activities (*OpenDoor1*/*OpenDoor2* or *CloseDoor1*/*CloseDoor2*) in Opportunity and the postural transitions (*sit to stand*/*stand to sit*) in HAPT can lead to overlapping cluster representations due to similarities in the activities.

5.2. Interplay of Forgetting and Intransigence

To further inspect the catastrophic forgetting and plasticity of our framework, we study the interplay of forgetting and intransigence in Figure 4. To reiterate, there is an inherent trade-off between a model's ability to retain information about old classes and

its ability to learn new classes without interfering with previous knowledge. Intuitively, if a model is heavily pushed to preserve past knowledge, it will forget less but have high intransigence. If, in contrast, preserving past information is weak, while intransigence will be low, the model will suffer from catastrophic forgetting. An ideal continual learning model would have low forgetting and intransigence measures, thus efficiently leveraging the model's learning capacity.

Table 4. Summary Results. ‡: without experience replay and contrastive loss, †: without contrastive loss.

(a) Base Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
Offline	80.56 ± 5.84	75.62 ± 4.53	62.68 ± 2.23	66.85 ± 10.86	76.46 ± 14.13
LAPNet-HAR ‡	46.94 ± 19.64	23.02 ± 17.24	14.21 ± 9.20	34.64 ± 22.21	42.48 ± 14.59
LAPNet-HAR †	63.29 ± 12.86	42.92 ± 19.78	31.07 ± 7.98	49.57 ± 14.93	65.05 ± 13.90
LAPNet-HAR	**69.66 ± 9.11**	**48.73 ± 10.87**	**35.47 ± 11.46**	**51.84 ± 22.27**	**74.28 ± 4.36**
Online finetuning	34.04 ± 30.80	20.02 ± 15.16	19.73 ± 13.01	26.43 ± 21.79	35.19 ± 12.78
(b) New Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
Offline	83.33 ± 2.09	67.69 ± 3.46	55.92 ± 3.40	64.70 ± 3.67	89.04 ± 10.69
LAPNet-HAR ‡	54.17 ± 6.07	14.68 ± 10.13	25.33 ± 4.35	35.27 ± 14.72	56.09 ± 26.34
LAPNet-HAR †	64.50 ± 13.54	20.57 ± 4.53	23.77 ± 4.62	36.24 ± 9.28	60.04 ± 26.72
LAPNet-HAR	**71.70 ± 6.61**	**28.28 ± 6.53**	**32.94 ± 4.66**	**45.63 ± 12.22**	**82.14 ± 15.46**
Online finetuning	51.92 ± 27.57	19.61 ± 10.99	23.82 ± 2.47	31.26 ± 16.97	55.31 ± 27.17
(c) Overall Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
Offline	78.33 ± 1.90	64.57 ± 3.09	52.88 ± 4.29	61.61 ± 4.90	80.34 ± 7.16
LAPNet-HAR ‡	45.85 ± 7.56	14.86 ± 10.97	18.81 ± 2.05	31.21 ± 7.08	42.56 ± 15.79
LAPNet-HAR †	58.76 ± 6.26	26.00 ± 8.52	21.79 ± 4.74	36.45 ± 6.70	57.33 ± 16.66
LAPNet-HAR	**65.92 ± 5.16**	**32.17 ± 5.33**	**28.41 ± 3.64**	**43.57 ± 10.26**	**74.78 ± 8.98**
Online finetuning	38.84 ± 23.83	16.62 ± 11.13	19.67 ± 4.28	26.62 ± 9.10	37.46 ± 14.69

We observe in Figure 4 that, across all datasets, *LAPNet-HAR* exhibits the best trade-off between forgetting and intransigence with PAMAP2 and Skoda losing around 18–19% of their best performance across all classes throughout the continual learning process. Focusing on the forgetting measure, the Opportunity dataset exhibits a more challenging learning process with forgetting measure reaching 51% when applying *LAPNet-HAR* which still outperforms other methods and the reference baseline. As expected, online finetuning exhibits the highest forgetting reaching ~60% for PAMAP2, Skoda, and DSADS and exceeding 70% for the Opportunity and HAPT datasets. While maintaining low forgetting, *LAPNet-HAR* is also observed to have the lowest intransigence across all datasets indicating that the model is also capable of continually learning new classes. It is evident that forgetting and intransigence of a model can be affected by dataset characteristics, as was also observed in the performance measures, especially with how old and new classes interfere with each other.
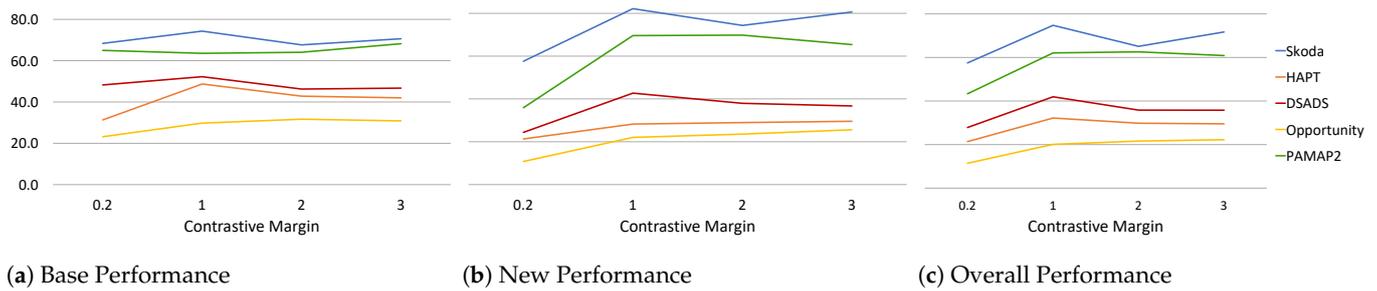
Figure 4. Interplay between forgetting and intransigence. The methods framed in red denote the method with the best trade-off between forgetting and intransigence for each dataset. ‡: without experience replay and contrastive loss, †: without contrastive loss.

6. Discussion

6.1. Parameter Configurations

In this section, we analyze the effect of tuning the (1) contrastive margin and (2) the refresh ratio α on the performance measures and the continual learning process.

6.1.1. Effect of Contrastive Margin

As explained in Section 3.2.2, the contrastive loss forces samples that belong to different classes to be farther apart by some predefined margin m enforcing inter-class separation. Keeping all other parameters fixed (Table 3) and varying the margin $m \in \{0.2, 1, 2, 3\}$, we analyze its effect on the performance measures across all datasets (Figure 5). The range was defined empirically based on observations of the intra-class euclidean distance of embeddings. We observe that the margin has a larger effect on the *New* performance with a significant increase when moving from margin $= 0.2$ to 1 across all datasets. For Skoda, DSADS, and PAMAP2, the *New* performance increases by at least 18% and at most 34% compared to low margin $m = 0.2$. The *Base* performance for all datasets, except for HAPT and PAMAP2, did not have a significant change ($\sim$5%) with the increase in margin. This can be due to the fact that the model is already pre-trained to recognize and separate the base classes, and thus during the continual learning process more emphasis is placed on learning to separate the new classes. Since a small number of random samples are streamed (20 samples), fast adaptation to new classes using only cross entropy is limited and challenging. Thus, incorporating a contrastive loss forces the model to learn an embedding space that separates different classes even with limited samples. Overall, a contrastive margin $m = 1$ resulted in a boost in *Overall* performance with a slight increase when further increasing the margin for some datasets.

(**a**) Base Performance (**b**) New Performance (**c**) Overall Performance

Figure 5. Effect of contrastive margin on (**a**) base, (**b**) new, and (**c**) overall performance across all datasets.

6.1.2. Effect of Refresh Ratio

The refresh ratio, employed in our continual replay-based prototype adaptation, allows a stabilizing update of the prototypes to incorporate the current new latent space while also preserving information from the previous latent space. Analyzing Equation (6), a large $\alpha > 0.5$ gives more emphasis to the previous embedding space, while $\alpha < 0.5$ pushes the prototype embeddings to the new latent space. We analyze its effect on the performance measures in Figure 6 while keeping all other parameters fixed.

(**a**) Base Performance (**b**) New Performance (**c**) Overall Performance
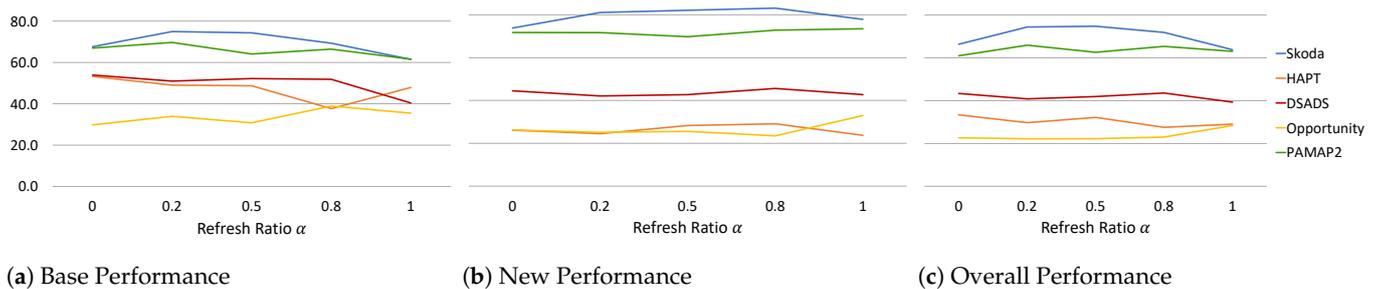
Figure 6. Effect of refresh ratio α on (**a**) base, (**b**) new, and (**c**) overall performance across all datasets.

We observe that α does not have a linear relation with the performance measures, and this can be attributed to the differences in data characteristics and how classes and data points interfere with each other during learning. Analyzing the results, we observe that, for $\alpha = 0$, the *Base* performance for HAPT and DSADS is at its highest at around 53%, while for Skoda it is lower compared to $0 < \alpha < 0.5$ by a margin of $\sim$7%. As α increases beyond 0.5, *Base* performance drops reaching its lowest at $\alpha = 1.0$ for Skoda and DSADS and at $\alpha = 0.8$ for HAPT. Opportunity experienced an increase in *Base* performance with $\alpha > 0.5$ reaching its highest at 0.8 while also experiencing a decrease in *New* performance. This follows the intuitive hypothesis that putting more emphasis on previously learned embedding space reduces catastrophic forgetting of base classes but increases intransigence when learning new classes. However, it is not always straightforward as it is affected by the faithfulness of the new latent space being learned in representing the different class clusters learned so far. If at a specific time step, the model update caused a drop in its ability to cluster the different classes, adapting the prototypes to this latent space will hinder its recognition ability making subsequent learning of new classes even more challenging. This can be seen in the Opportunity dataset where setting $\alpha = 1.0$ maintained a high *Base* performance while also resulting in an increase in *New* performance for Opportunity and ultimately in the *Overall* performance. For PAMAP2, on the other hand, increasing $\alpha > 0.5$ decreased *Base* performance but increased *New* performance reaching its highest at $\alpha = 1.0$. This shows that further analysis of the learning behavior of the model at every time step is critical to better understand the changes that happen at the model and prototype level and in turn improve the overall continual learning process.

6.2. Data Augmentation

The ability to learn new classes and adapt to changes with only limited data is a key challenge in continual learning. Data augmentation methods have been introduced to expand the training data and increase the generalization ability of the model during training [58]. While data augmentation is common in image recognition, it is only recently being studied for time series data [59,60]. We investigate the effect of data augmentation on the continual learning performance. To do so, with every incoming batch, we apply four data augmentation techniques [60] enlarging the batch size four times by: (1) adding random jitter or noise, (2) scaling by a constant amount, (3) time warping, and (4) magnitude warping. Comparing the performance of *LAPNet-HAR* with and without augmentation (Table 5), we observe an improvement in base, new and overall performance across multiple datasets. HAPT specifically exhibits a significant boost in overall performance, while Opportunity exhibits a slight drop in performance. It is important to keep in mind that data augmentation methods have not always been found to improve performance and can depend on the dataset characteristics and model [59]. Additional investigation of such methods while also targeting continual learning paradigms is needed.

Table 5. Effect of data augmentation with LAPNet-HAR.

(a) Base Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
w/o augmentation	69.66 ± 9.11	48.73 ± 10.87	35.47 ± 11.46	**51.84 ± 22.27**	**74.28 ± 4.36**
w/ augmentation	**74.87 ± 9.59**	**58.66 ± 8.77**	**36.79 ± 7.99**	51.05 ± 9.20	70.26 ± 14.47
(b) New Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
w/o augmentation	**71.70 ± 6.61**	28.28 ± 6.53	**32.94 ± 4.66**	45.63 ± 12.22	82.14 ± 15.46
w/ augmentation	69.64 ± 6.38	**36.59 ± 2.80**	29.26 ± 4.42	**51.96 ± 5.99**	**88.80 ± 9.28**
(c) Overall Performance					
Models	**PAMAP2**	**HAPT**	**Opportunity**	**DSADS**	**Skoda**
w/o augmentation	65.92 ± 5.16	32.17 ± 5.33	**28.41 ± 3.64**	43.57 ± 10.26	74.78 ± 8.98
w/ augmentation	**66.61 ± 4.79**	**41.33 ± 2.44**	26.26 ± 2.87	**47.49 ± 5.26**	**75.27 ± 5.23**

6.3. Effect of Data and Class Sequence

Continual learning has proven to be a challenging problem where a model has to learn to deal with several challenges such as catastrophic forgetting and concept drift. Thus, researchers tackling this problem often simplified it to an incremental learning of classes divided into a well-defined, equally divided sequence of tasks [13]. While it may sound artificial, this allowed researchers to develop methods that tackle one problem at a time as well as enabled having a defined experimental protocol for comparing different approaches. However, the influence of class and data orderings in the evaluation on such incremental learning systems has received very little attention.

Inter-class similarity is a key challenge in HAR datasets where some activities can have very similar sensor patterns. For example, DSADS includes the activity classes *lying on right* and *lying on back* that can lead to confusion during the model's learning process. Even during offline training, the model wrongly classifies one class with the other. Continual learning is even more challenging since training is done online with only a subset of classes and limited data samples encountered at every time step. Beyond class orderings, data ordering will also have an effect on the learning process depending on whether a gradual or abrupt distribution shift is observed. This is especially true in an online task-free learning setting. Thus, studying the effect of continual learning compared to online learning when a gradual or sudden change in data distribution is observed would give a deeper understanding about the benefits and effectiveness of such systems.

Prior work on HAR-based continual learning have typically dealt with the randomness of class or task orderings by conducting several experimental runs and averaging the results.

Following the same procedure, we observed how performance varied with different class and data orderings. In some cases, performance exhibited a drop which can indicate that more confusion was encountered leading to a more difficult classification task. While analyzing results on multiple class orderings provides a fairer comparison, we aim to investigate, in future work, the impact of class and data orderings in a task-free continual learning setting which can enable better understanding of the continual learning process and the development of more robust systems [61].

6.4. Comparison to Prior Work

In this section, we discuss how our approach compares to other methods, more specifically to other CL analysis applied to HAR [16]. As previously mentioned, task-incremental learning have dominated prior work. Methods developed in the task-based sequential learning setup all often depend on knowing the task boundaries, which give indications of when to distill knowledge after learning a task. Moreover, only data for the current task is observed before moving to another task which guarantees shuffled i.i.d data and convergence to the current task. On the other hand, in a task-free continual learning setting, small non-i.i.d batches are streamed that include data from old as well as new classes. This posed a challenge when comparing our framework to other CL methods in HAR.

In terms of prototype-based classification, the closest method that uses prototypes in class-incremental continual learning was proposed in iCARL [12]. However, the main difference is that it depends on task information during training. More specifically, iCARL stores new task prototypes or exemplars to represent the class mean in feature space. These, in turn, are used for nearest mean prediction and knowledge distillation. Prediction is only possible after fully learning the task at hand. Thus, task boundaries information is needed to apply knowledge distillation when moving to a new task and for storing class exemplars of previous tasks. Moreover, iCARL requires recalculating all prototypes every time for prediction which is computationally expensive. On the other hand, *LAPNet-HAR* performs online evaluation as well as training using the prototypes in an online task-free way. The prototypes are continually refreshed and updated over time eliminating the need to recalculate them every time.

Another branch of work focused on model expansion in a task-free setting, such as CURL [62] and CN-DPM [63], while not explored and benchmarked in HAR. CURL enables unsupervised task-free representation learning using Variational Auto-Encoder with generative replay. CN-DPM follows a Dirichlet process to train a set of experts for subsets of the data. Its key contribution is the unbound dynamic expansion of the learner. *LAPNet-HAR* evades unbound allocation of resources by keeping the network capacity static but instead incorporates prototypes for new classes which are d-dimensional and fixed in size.

7. Lifelong Learning Application Scenarios

Lifelong learning opens up new possibilities in a wide range of applications. In real-world scenarios, systems built with adaptive models can successfully respond to dynamic changes in data and environments, greatly improving their resilience and resulting in rich user experiences. In this section, we provide examples that demonstrate the benefits of continuously adaptive models in human-activity recognition applications.

7.1. A Lifelong Exercise Coach

Wearables that can track physiological measures have become popular in the last few years. Some of these devices, e.g., smartwatches, are now capable of passively recognizing and quantifying a limited set of physical activities such as walking, running, swimming and biking. Consider a future version of these devices that serves as a lifelong exercise guide spanning several decades. The devices might employ pre-trained models for common activities but it is crucial that they learn to recognize new forms of exercises and adapt.

A runner might start with short races at first but become more sophisticated over the years and add strength training to her weekly regimen in order to gain speed, avoid injuries, and tackle longer races. Years into running, the same runner might decide to enroll in a yoga class for recovery after long-runs while also adding a weekly session of plyometrics to gain lower-body power. Later in life, due to a career change, move to a different country, or health issue, the individual might abandon running in favor of a completely different sport such as badminton or golf. Ideally, all of these activities should be monitored by a continuously adaptive model powering a recommendation engine that dispenses feedback and guidance about training, form, nutrition, sleep and other relevant lifestyle factors.

7.2. Long-Term Health Monitoring for Independent Living

A long-term goal of the field of ubiquitous computing has been to develop new sensing, communication, and modeling technologies that enable older adults to live independently in their homes. This vision often requires systems that can track and assess activities of daily living (ADL) in real-world environments and for long periods of time. While many approaches have been successful in recognizing human activities from samples of activity classes, they operate under the prevailing assumption that human behaviors, activity routines, and environmental settings are static and do not change over time. Needless to say, this assumption is misaligned with the dynamics of real life. As individuals grow older, they often develop disabilities that cause them to perform activities differently (e.g., change in walking gait). New hobbies (e.g., gardening) and wellness activities (e.g., resistance exercises to combat osteoporosis) might become part of the everyday routine at the same time that mobility patterns change (e.g., less driving). In light of these anticipated changes, having to retrain an activity recognition and monitoring model from scratch on new collected data is not desirable, if not altogether practical.

7.3. Smart Home and Accessibility

Advances in machine perception combined with lifelong learning have the potential to enable new applications in accessibility. Imagine a smart home that can aid mobility-impaired individuals by identifying cues that they are about to perform primary tasks (e.g., take a shower) and assisting them with secondary tasks (e.g., automatically turn on bathroom lights, adjust water temperature, close bedroom curtains). Similarly, consider a system for blind individuals that provides discreet auditory feedback about what others are doing around them, giving them improved awareness of their surroundings. It is common for visual, hearing, mobility and cognitive impairments to worsen over time. Therefore, offering appropriate and adaptive levels of support and scaffolding are key to realizing these scenarios in a sensible and successful manner. Adaptive and lifelong modeling is key. An additional challenge in the context of smart homes is that physical environments differ significantly, whether due to layout differences or background noises (e.g., ambient hum of air conditioners, urban sounds of city living). Sometimes, environmental characteristics might change unexpectedly and non-deterministically, such as due to construction or a move. Thus, having a mechanism to continuously adapt and personalize to the target environment over the long term is a crucial component to provide a smooth and continuous experience.

8. Limitations and Future Work

In this paper, we presented *LAPNet-HAR*, a lifelong adaptive learning system that is capable of continually learning from streaming data in a task-free setting. We regard our framework as the first step towards truly continual learning in HAR in naturalistic settings. However, lifelong learning has a wider scope and includes several challenges that remain untackled. In this section, we discuss several ways we plan to extend our framework to ultimately develop an end-to-end continual learning system for HAR.

As a first step in investigating the effectiveness of our *LAPNet-HAR*, we made assumptions regarding the ability to discover new activities from streaming data as well as the

availability of high-quality ground truth activity labels. When new activities emerge over time, an obvious challenge is being able to determine when a new activity is encountered. Leveraging our framework's clustering approach with class prototypes, a threshold-based activity discovery mechanism over the metric space [41] can be incorporated. This can further prevent corrupting previously learned prototypes. Moreover, with forming new class prototypes, analyzing measures such as cluster's density, weight, and temporal characteristics can help eliminate unwanted outliers. Assuming activity discovery is enabled, another challenge is acquiring the ground truth labels. In reality, acquiring annotations has been well acknowledged as time- and effort-consuming with several work being devoted to acquiring data labels while reducing user burden [64,65]. Furthermore, acquiring annotations is not only about *how* to annotate but also about *what* data to annotate and *when*. All three questions ultimately alleviate the labeling effort. Active learning directly aims to answer these questions by selecting the most informative data samples to annotate [66]. Adaimi and Thomaz [66] demonstrated how querying and annotating the most uncertain data samples to use for training reduces the labeling effort while maintaining model accuracy.

As the performance for continual learning on HAR still has room for improvement, in the future, we will look into techniques to further mitigate catastrophic forgetting and also reduce the interference between different classes. One way to do so is by exploring other frameworks that better capture long-term signal information. Specifically, Transformers have been recently shown to be successful in extracting the semantic correlations among elements in a long sequence, which is beneficial for time-series forecasting [67]. We plan to study the effectiveness of Transformers in continual learning, studying its encoding ability to represent the longitudinal long-term stream of data. To do so, we will look into another important question that deals with the idea of whether and when model adaptation is really needed. Previous work have more often assumed that, at every time step, the model is updated with a new batch of data. However, in reality, not all new data encountered is adding valuable information, and thus can effectively be ignored. This would further reduce catastrophic forgetting since with every incremental learning step, the model is more prone to forgetting due to the increasing number of intermediate model updates. Using the prototypical framework proposed, we will analyze when updating the model weights is needed. This can be done by studying the rate of change in class prototypes as well as monitoring distribution shifts. Last but not least, the prototype memory stores prototypes for every class seen so far. However, in some cases, an activity class might not be encountered again. For example, a student stops riding a bike after buying a car, or an athlete stops playing football after an injury. Thus, it makes sense for the model to forget an old activity that will never be encountered again and make room for a new one. Therefore, we plan to investigate an aging process where prototypes of outdated classes are disposed of.

In this work, we aimed to demonstrate the promising performance of our framework on several widely used datasets, with activities ranging from 10 up to 19 classes. However, a more longitudinal real-world analysis would provide insights to additional challenges and open research questions, especially in HAR. In some cases, continual learning tends to slow down the adaptation to newly seen data, as opposed to pure online learning, as it tries to preserve previously learned knowledge and stabilize the learning process. More specifically, when the new data is much more informative or representative than the old, continual learning can have a negative effect on the training. For example, imagine a user went on vacation to a new environment. Pure online learning would adapt fast to this new environment as opposed to continual learning that would exhibit a slower adaptation. Essentially, the benefit of the stabilizing effect of continual learning depends on the time scale of the changes in the data. Thus, a large-scale longitudinal analysis would push this field further towards an end-to-end sustainable real-world systems. In the future, we aim to collect a longitudinal dataset, from which we can define a data streaming protocol more

representative of the real-world that researchers can refer to. Ultimately, this would enable researchers to extend the current continual learning field for HAR.

9. Conclusions

This paper proposes *LAPNet-HAR* to support task-free continual learning from data streams in human activity recognition with prototypical networks. *LAPNet-HAR* enables online learning of unlimited number of activities by relying on prototype-based classification. Catastrophic forgetting and model intransigence are addressed by combining experience replay and dynamic prototype adaptation. Inter-class separation is enforced during continual learning using a margin-based contrastive loss. With extensive experiments on five HAR datasets, *LAPNet-HAR* has demonstrated promising results for supporting task-free continual activity recognition. We empirically investigated the efficacy of our prototypical approach on five HAR datasets in task-free data incremental scenarios. The novelty of our work lies in extending the learning paradigm in wearable sensor-based human activity recognition towards lifelong continual learning. Our proposed framework, *LAPNet-HAR*, specifically offers increased real-world practicality and applicability by applying an online task-free incremental learning. Additionally, our extensive analysis and results demonstrate high potentials, not only in human activity recognition but also in various applications such as wearable devices and healthcare. We hope this work would encourage research in this direction, focusing on applications beyond classification or task-incremental learning. Ultimately, this work is considered as a natural next step in moving beyond offline learning and towards lifelong learning which will enable the large-scale and long-term deployment of mobile recognition systems.

References

1. Chun, K.S.; Sanders, A.B.; Adaimi, R.; Streeper, N.; Conroy, D.E.; Thomaz, E. Towards a Generalizable Method for Detecting Fluid Intake with Wrist-Mounted Sensors and Adaptive Segmentation. In Proceedings of the 24th International Conference on Intelligent User Interfaces, Marina del Ray, CA, USA, 17–20 March 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 80–85. [CrossRef]
2. San Chun, K.; Jeong, H.; Adaimi, R.; Thomaz, E. Eating episode detection with jawbone-mounted inertial sensing. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 4361–4364.
3. Cakmak, A.S.; Thigpen, N.; Honke, G.; Alday, E.P.; Rad, A.B.; Adaimi, R.; Chang, C.J.; Li, Q.; Gupta, P.; Neylan, T.; et al. Using Convolutional Variational Autoencoders to Predict Post-Trauma Health Outcomes from Actigraphy Data. In Proceedings of the Machine Learning for Mobile Health Workshop at NeurIPS, Virtual, 6–12 December 2020.
4. Skocir, P.; Krivic, P.; Tomeljak, M.; Kusek, M.; Jezic, G. Activity Detection in Smart Home Environment. *Procedia Comput. Sci.* **2016**, *96*, 672–681. [CrossRef]
5. Adaimi, R.; Yong, H.; Thomaz, E. Ok Google, What Am I Doing? Acoustic Activity Recognition Bounded by Conversational Assistant Interactions. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–24. [CrossRef]
6. Thomaz, E.; Essa, I.; Abowd, G.D. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Osaka, Japan, 7–11 September 2015; pp. 1029–1040.

7. King, W.C.; Chen, J.Y.; Bond, D.S.; Belle, S.H.; Courcoulas, A.P.; Patterson, E.J.; Mitchell, J.E.; Inabnet, W.B.; Dakin, G.F.; Flum, D.R.; et al. Objective assessment of changes in physical activity and sedentary behavior: Pre-through 3 years post-bariatric surgery. *Obesity* **2015**, *23*, 1143–1150. [CrossRef]

8. Chen, Z.; Liu, B. Lifelong machine learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2018**, *12*, 1–207.

9. Polikar, R.; Upda, L.; Upda, S.S.; Honavar, V. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. Syst. Man, Cybern. Part C Appl. Rev.* **2001**, *31*, 497–508. [CrossRef]

10. Kemker, R.; McClure, M.; Abitino, A.; Hayes, T.; Kanan, C. Measuring catastrophic forgetting in neural networks. *AAAI Conf. Artif. Intell.* **2018**, *32*, 11651. [CrossRef]

11. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [CrossRef]

12. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 2001–2010.

13. Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3366–3385. [CrossRef] [PubMed]

14. Lesort, T.; Lomonaco, V.; Stoian, A.; Maltoni, D.; Filliat, D.; Díaz-Rodríguez, N. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Inf. Fusion* **2020**, *58*, 52–68. [CrossRef]

15. Jha, S.; Schiemer, M.; Ye, J. Continual learning in human activity recognition: An empirical analysis of regularization. In Proceedings of the Thirty-Seventh International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020; pp. 1–4.

16. Jha, S.; Schiemer, M.; Zambonelli, F.; Ye, J. Continual Learning in Sensor-based Human Activity Recognition: An Empirical Benchmark Analysis. *Inf. Sci.* **2021**, *575*, 1–21. [CrossRef]

17. Ren, M.; Iuzzolino, M.L.; Mozer, M.C.; Zemel, R. Wandering Within a World: Online Contextualized Few-Shot Learning. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4 May 2021.

18. Daskalos, A.C.; Theodoropoulos, P.; Spandonidis, C.; Vordos, N. Wearable Device for Observation of Physical Activity with the Purpose of Patient Monitoring Due to COVID-19. *Signals* **2022**, *3*, 11–28. [CrossRef]

19. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [CrossRef]

20. Zhou, Y.; Wang, F.; Tang, J.; Nussinov, R.; Cheng, F. Artificial intelligence in COVID-19 drug repurposing. *Lancet Digit. Health* **2020**, *2*, e667–e676. [CrossRef]

21. Amrollahi, F.; Shashikumar, S.P.; Holder, A.L.; Nemati, S. Leveraging clinical data across healthcare institutions for continual learning of predictive risk models. *Sci. Rep.* **2022**, *12*, 8380. [CrossRef] [PubMed]

22. Liu, B.; Xiao, X.; Stone, P. A lifelong learning approach to mobile robot navigation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1090–1096. [CrossRef]

23. Hemati, H.; Borth, D. Continual speaker adaptation for text-to-speech synthesis. *arXiv* **2021**, arXiv:2103.14512.

24. Pancholi, S.; Joshi, A.M. Novel Time Domain Based Upper-Limb Prosthesis Control using Incremental Learning Approach. *arXiv* **2021**, arXiv:2109.04194.

25. Wang, Z.; Jiang, M.; Hu, Y.; Li, H. An incremental learning method based on probabilistic neural networks and adjustable fuzzy clustering for human activity recognition by using wearable sensors. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 691–699. [CrossRef]

26. Mazankiewicz, A.; Böhm, K.; Berges, M. Incremental Real-Time Personalization in Human Activity Recognition Using Domain Adaptive Batch Normalization. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2020**, *4*, 144. [CrossRef]

27. Ntalampiras, S.; Roveri, M. An incremental learning mechanism for human activity recognition. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 6–9 December 2016; pp. 1–6.

28. Siirtola, P.; Röning, J. Context-aware incremental learning-based method for personalized human activity recognition. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 10499–10513. [CrossRef]

29. Mo, L.; Feng, Z.; Qian, J. Human daily activity recognition with wearable sensors based on incremental learning. In Proceedings of the 2016 10th International Conference on Sensing Technology (ICST), Nanjing, China, 11–13 November 2016; pp. 1–5.

30. Gjoreski, H.; Roggen, D. Unsupervised Online Activity Discovery Using Temporal Behaviour Assumption. In Proceedings of the 2017 ACM International Symposium on Wearable Computers, Maui, HI, USA, 11–15 September 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 42–49. [CrossRef]

31. Wu, J.; Harrison, C.; Bigham, J.P.; Laput, G. Automated Class Discovery and One-Shot Interactions for Acoustic Activity Recognition. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1–14. [CrossRef]

32. Fang, L.; Ye, J.; Dobson, S. Discovery and Recognition of Emerging Human Activities Using a Hierarchical Mixture of Directional Statistical Models. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 1304–1316. [CrossRef]

33. Cheng, H.T.; Sun, F.T.; Griss, M.; Davis, P.; Li, J.; You, D. NuActiv: Recognizing Unseen New Activities Using Semantic Attribute-Based Learning. In Proceedings of the 11th Annual International Conference on Mobile Systems, Applications, and Services, Taipei, Taiwan, 25–28 June 2013; Association for Computing Machinery: New York, NY, USA, 2013; pp. 361–374. [CrossRef]

34. Ye, J.; Callus, E. Evolving models for incrementally learning emerging activities. *J. Ambient Intell. Smart Environ.* **2020**, *12*, 1–13. [CrossRef]

35. Chen, T.; Goodfellow, I.; Shlens, J. Net2Net: Accelerating Learning via Knowledge Transfer. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.

36. Lopez-Paz, D.; Ranzato, M.A. Gradient Episodic Memory for Continual Learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.

37. Ye, J.; Nakwijit, P.; Schiemer, M.; Jha, S.; Zambonelli, F. Continual Activity Recognition with Generative Adversarial Networks. *ACM Trans. Internet Things* **2021**, *2*, 1–25. [CrossRef]

38. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 6–11 July 2015.

39. Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; Wierstra, D. Matching Networks for One Shot Learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; Curran Associates Inc.: Red Hook, NY, USA, 2016; pp. 3637–3645.

40. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 4077–4087.

41. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.B.; Larochelle, H.; Zemel, R.S. Meta-Learning for Semi-Supervised Few-Shot Classification. In Proceedings of the 6th International Conference on Learning Representations ICLR, Vancouver, BC, Canada, 30 April–3 May 2018 .

42. Gidaris, S.; Komodakis, N. Dynamic few-shot visual learning without forgetting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 Jun 2018; pp. 4367–4375.

43. Ren, M.; Liao, R.; Fetaya, E.; Zemel, R. Incremental Few-Shot Learning with Attention Attractor Networks. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, CO, Canada, 8–14 December 2019; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Vancouver, CO, Canada 2019; Volume 32.

44. De Lange, M.; Tuytelaars, T. Continual prototype evolution: Learning online from non-stationary data streams. *arXiv* **2020**, arXiv:2009.00919.

45. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality reduction by learning an invariant mapping. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY , USA, 17–22 June 2006; pp. 1735–1742.

46. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; pp. 539–546.

47. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.

48. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 18661–18673.

49. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 1597–1607.

50. Chavarriaga, R.; Sagha, H.; Calatroni, A.; Digumarti, S.T.; Tröster, G.; Millán, J.D.R.; Roggen, D. The Opportunity Challenge: A Benchmark Database for On-body Sensor-based Activity Recognition. *Pattern Recogn. Lett.* **2013**, *34*, 2033–2042. [CrossRef]

51. Reiss, A.; Stricker, D. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. In Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments, Crete, Greece, 6–9 June 2012; ACM: New York, NY, USA, 2012; pp. 40:1–40:8. [CrossRef]

52. Altun, K.; Barshan, B.; Tunçel, O. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognit.* **2010**, *43*, 3605–3620. [CrossRef]

53. Zappi, P.; Roggen, D.; Farella, E.; Tröster, G.; Benini, L. Network-Level Power-Performance Trade-Off in Wearable Activity Recognition: A Dynamic Sensor Selection Approach. *ACM Trans. Embed. Comput. Syst.* **2012**, *11*, 68. [CrossRef]

54. Reyes-Ortiz, J.L.; Oneto, L.; Ghio, A.; Samá, A.; Anguita, D.; Parra, X. Human activity recognition on smartphones with awareness of basic activities and postural transitions. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 177–184.

55. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A survey on concept drift adaptation. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–37. [CrossRef]

56. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [CrossRef]

57. Chaudhry, A.; Dokania, P.K.; Ajanthan, T.; Torr, P.H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 532–547.

58. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]

59. Iwana, B.K.; Uchida, S. An empirical survey of data augmentation for time series classification with neural networks. *PLoS ONE* **2021**, *16*, e0254841. [CrossRef] [PubMed]

60. Um, T.T.; Pfister, F.M.; Pichler, D.; Endo, S.; Lang, M.; Hirche, S.; Fietzek, U.; Kulić, D. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, 13–17 November 2017; pp. 216–220.

61. Masana, M.; Twardowski, B.; Van de Weijer, J. On class orderings for incremental learning. *arXiv* **2020**, arXiv:2007.02145.

62. Rao, D.; Visin, F.; Rusu, A.; Pascanu, R.; Teh, Y.W.; Hadsell, R. Continual Unsupervised Representation Learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 7647–7657.

63. Lee, S.; Ha, J.; Zhang, D.; Kim, G. A Neural Dirichlet Process Mixture Model for Task-Free Continual Learning. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.

64. Intille, S.; Haynes, C.; Maniar, D.; Ponnada, A.; Manjourides, J. μEMA: Microinteraction-based ecological momentary assessment (EMA) using a smartwatch. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 1124–1128.

65. Adaimi, R.; Ho, K.T.; Thomaz, E. Usability of a Hands-Free Voice Input Interface for Ecological Momentary Assessment. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Austin, TX, USA, 23–27 March 2020; pp. 1–5. [CrossRef]

66. Adaimi, R.; Thomaz, E. Leveraging Active Learning and Conditional Mutual Information to Minimize Data Annotation in Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2019**, *3*, 70. [CrossRef]

67. Dirgová Luptáková, I.; Kubovčík, M.; Pospíchal, J. Wearable sensor-based human activity recognition with transformer model. *Sensors* **2022**, *22*, 1911. [CrossRef]