2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Continual Learning for Autonomous Robots: A Prototype-based Approach

Elvin Hajizada[1,2], Balachandran Swaminathan[3], Yulia Sandamirskaya[4]

Abstract— **Humans and animals learn throughout their lives from limited amounts of sensed data, both with and without supervision. Autonomous, intelligent robots of the future are often expected to do the same. The existing continual learning (CL) methods are usually not directly applicable to robotic settings: they typically require buffering and a balanced replay of training data. A few-shot online continual learning (FS-OCL) setting has been proposed to address more realistic scenarios where robots must learn from a non-repeated sparse data stream. To enable truly autonomous life-long learning, an additional challenge of detecting novelties and learning new items without supervision needs to be addressed. We address this challenge with our new prototype-based approach called Continually Learning Prototypes (CLP). In addition to being capable of FS-OCL learning, CLP also detects novel objects and learns from them without supervision. To mitigate forgetting, CLP utilizes a novel metaplasticity mechanism that adapts the learning rate individually per prototype. CLP is rehearsal-free, hence does not require a memory buffer, and is compatible with neuromorphic hardware, characterized by ultra-low power consumption, real-time processing abilities, and on-chip learning. Indeed, we have open-sourced both the PyTorch implementation of CLP and a simpler version in the neuromorphic software framework Lava, targetting Intel's neuromorphic chip Loihi 2. We evaluate CLP on a robotic vision dataset, OpenLORIS. In a low-instance FS-OCL scenario, CLP shows state-of-the-art results. In the open world, CLP detects novelties with superior precision and recall and learns features of the detected novel classes without supervision, achieving a strong baseline of** 99% **base class and** $65\%/76\%$ **(5-shot/10-shot) novel class accuracy.**

I. Introduction

Autonomous, interactive, and lifelong learning are features of human intelligence distinguishing it from the machine intelligence of the modern age. Current machine learning methods outperform both humans and hand-crafted algorithms on a given static data set but fail spectacularly when the key assumptions of the neural network training scheme, e.g., of identically and independently distributed (i.i.d.) data, are violated [1]. To address the limitations of static data distribution, Continual Learning (CL) is an emerging topic in AI. The main issue CL aims to address is catastrophic forgetting, the phenomenon that reflects the trade-off between attaining new knowledge and retaining the old knowledge, also known as the plasticity-stability dilemma [2]. Replay, regularization, parameter isolation, and network expansion methods have been some of the most common techniques in the CL literature [1], [3].

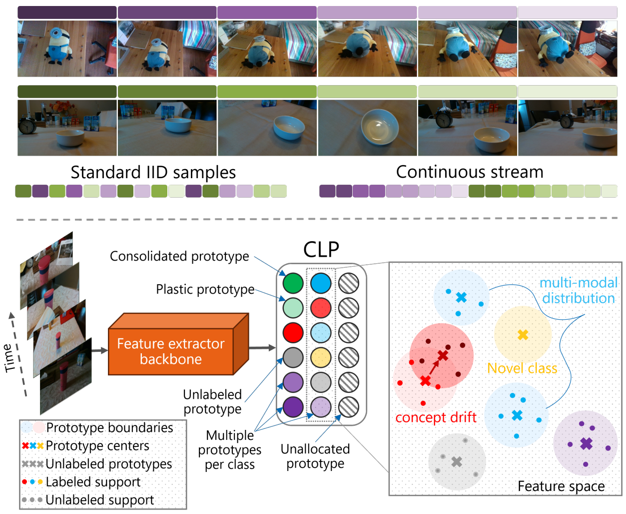


Fig. 1: **(Top)** Online continual learning from a continuous stream of non-i.i.d. data is key for the autonomous learning of robots (the images are from the OpenLORIS dataset). **(Bottom)** Continually Learning Prototypes: architecture overview.

However, the problem of catastrophic forgetting is not the only challenge that must be addressed to close the gap between today's training of deep neural networks and the more natural learning processes we know from humans and animals. Recently, learning objects from a few labeled samples provided through a non-repeating stream of input has gained attention [4], [5] (Fig. 1). This setting, formally called few-shot online continual learning (FS-OCL), is a step towards realistic learning for robots.

Yet, FS-OCL is still far from real-world learning scenarios. For instance, human concept learning involves not only a small amount of direct instruction (e.g., parental labeling) but also large amounts of unlabeled experience (e.g., observation of objects without naming them). This unsupervised learning is continual, autonomous, and interactive. A strong driver of this learning process is novelty detection – the ability to recognize an instance as something not seen before [6], [7]. On the contrary, a common close-world assumption in deep learning is that all the test instances are from the learned classes. Detecting novel instances alone is not enough, however, as such a system should also integrate these novelties into its knowledge, even without supervision [8]. Therefore, we extend FS-OCL to include open-world and semi-supervised learning to achieve the most natural contin-

*This work was supported by Intel
[1]Neuromorphic Computing Lab, Intel Labs, Munich, Germany
[2]School of Computation, Information and Technology, Technical University of Munich, Germany elvin.hajizada@tum.de
[3]Pennsylvania State University, bus44@psu.edu
[4]Institute of Computational Life Sciences (ICLS), Zurich University of Applied Sciences (ZHAW), Schloss 1, 8820 Wädenswil, Switzerland
†Code available at: https://github.com/elvinhajizada/CLP.

979-8-3503-7769-9/24/$31.00 ©2024 IEEE

ual object learning setting for robots, which we shortly call **O**pen **W**orld **C**ontinual **L**earning (OWCL).

We propose ***Continually Learning Prototypes (CLP)*** (Fig. 1), a comprehensive learning algorithm designed to address the challenges of Open World Continual Learning (OWCL). CLP is capable of online continual learning from few-shot, labeled, and unlabeled data in dynamic environments with unknowns. In these non-stationary settings, we store representative examples or "prototypes" of observed classes to capture the prototypical representation of the feature space for each class. CLP can also learn multiple prototypes per class, allowing the system to adapt to new information through novelty detection. These prototypes can dynamically adjust in a semi-supervised manner from a continuous input stream. Furthermore, we introduce a novel mechanism for dynamically adapting the learning rates of individual prototype neurons, inspired by metaplasticity [9]. This mechanism helps address the stability-plasticity dilemma, thereby mitigating catastrophic forgetting.

Crucially, CLP is rehearsal-free and does not maintain a memory buffer, as it targets robotics platforms, which generally have compute, memory, and energy constraints. Additionally, we implemented a simpler version of CLP for neuromorphic chip Loihi 2 and open-sourced it as part of the Lava software framework[1]. Note that the details and results of the neuromorphic implementation are beyond the scope of this paper. Our contributions are summarized as follows:

- We tackle a novel learning scenario called open-world continual learning (OWCL) to evaluate robotic object learning in the most realistic way. This scenario assumes data becomes available sample-by-sample in open-world, where novel classes may appear spontaneously with or without labels. These instances need to be detected and learned, possibly with few shots, all the while avoiding catastrophic forgetting.
- Each initialized prototype functions as a classifier, using cosine similarity to the input feature vector to predict the output. Based on the label, prototypes receive positive or negative updates, refining their classification accuracy. In the absence of labels, CLP operates as a novelty detection-assisted clustering algorithm, capable of identifying and learning new instances while adapting to gradual concept drifts.
- We dynamically adjust multiple prototypes per class in a semi-supervised manner, allowing them to adapt to streaming input effectively.

II. Related work

Continual Learning. Continual Learning (CL) research focuses on overcoming the challenges of learning from a sequence of tasks without experiencing catastrophic forgetting. Common approaches to achieving CL include replay methods, regularization techniques, parameter isolation, and network architecture-based strategies. Replay methods typically involve maintaining a memory buffer that stores representative examples from previously learned data distributions [10], [11] or generating synthetic memories using a generative model to replay to the main model [12]. The objective is to mimic i.i.d. training by enabling simultaneous optimization of both old and new tasks. Regularization techniques [13] utilize stored samples to constrain updates for new tasks within the network, often by estimating the importance of each learned weight and using this information to restrict changes to important weights, thereby preserving prior knowledge [14], [15], [16]. Parameter isolation involves isolating task-specific parameters, which remain unchanged while learning new tasks [17], [18]. While these techniques collectively address catastrophic forgetting, some make strong i.i.d. assumptions or rely on the availability of labeled samples arriving in batches, making them less suitable for autonomous and open-world online continual learning scenarios, which we consider in this work. Furthermore, parameter isolation and regularization-based methods often address incremental learning of tasks [17] by dedicating frozen parts of the network to each task [19], [18], leading to network parameter saturation. Network architecture-based approaches tackle the limitation of network capacity by dynamically expanding the model's size to accommodate new tasks [20], [21]. However, freezing previously learned network parameters and adding new ones for novel tasks can result in scalability issues and network size explosion. On the other hand, CLP dynamically adjusts old task parameters to accommodate concept drift and only adds new network parameters when necessary.

Prototype-based Learning. For memory and compute-constrained embedded devices like robots, storing representative examples or "prototypes" of seen classes has proven beneficial for learning in non-stationary environments. Prototype-based learning methods like self-organizing map (SOM) and learning vector quantization (LVQ), initially introduced by Kohonen [22], laid the foundation for extensive adaption in incremental learning literature of the pre-deep learning era [23], [24]. Recently, deep learning-based CL methods began incorporating prototypes in the network's final layer for inference and learning [25], [11], [26], or as memory buffers for pattern replay [27]. Additionally, these techniques have been applied in few-shot class incremental learning (FSCIL), where networks learn incrementally from small batches of labeled samples [28], [29]. However, the idea that new classes are presented in regular, iterable batches is impractical, and the common practice of using a single prototype per class overlooks the complexity of real-world class distributions [30], [31], which may require multiple prototypes. More recently, CBCL [32] proposed novelty detection-assisted clustering that produces multiple prototypes per class. This method presupposes labeled samples and calculates prototypes as the running average of cluster samples, employing a linearly decaying learning rate without error-based prototype updates. Conversely, original prototype-based supervised approaches [33], [34], [24] dynamically adjust prototypes via stochastic gradient descent. We expand upon these previous works by intro-

[1]Visit CLP in Lava.

ducing a dynamic and independent plasticity mechanism (metaplasticity), akin to a learning rate for each prototype, to address the stability-plasticity dilemma. Furthermore, we introduce the capability of semi-supervised learning while still allowing more than one prototype per class. Inspired by LVQ methods [23], [33], [34], we update prototypes also when they produce errors. These enhancements allow us to effectively manage the evolving demands of open-world continual learning scenarios.

Open World Learning. One such demand arises when the robot operates in an open world where novel objects appear all the time. Therefore, novelty detection is also necessary for realistic continual learning. There is a significant body of work on open-set and open-world recognition [6], [35], [8] that demonstrates continuous detection of novel classes. Subsequently, OWCL, as a realistic and natural learning setting for an autonomous robot, encompasses all of 1) online continual learning, 2) few-shot learning, 3) open-set recognition, and 4) semi-supervised learning of new and old categories. To the best of our knowledge, a comprehensive algorithm that combines all such learning scenarios faced by autonomous agents is under-explored in the literature [8]. This work aims to build a powerful algorithm for continual online learning adaptable to semi-supervised open-world contexts using the prototypes based approach.

III. Problem Setting

Few-shot Online Continual Learning (FS-OCL). In the FS-OCL setting [5], a model sequentially processes and learns a dataset $\mathcal{D} = (\boldsymbol{x_t}, y_t)_{t=1}^{\infty}$ one instance at a time, where $(\boldsymbol{x_t}, y_t)$ is an example-label pair received at time t, with $\boldsymbol{x_t} \in \mathbb{R}^D$. We assume that only a few such labeled samples can be provided, since for the robot users it would be time-consuming and tedious. The model can be decomposed into a pretrained feature extractor and an online classifier: $f(\boldsymbol{x_t}) = F(G(\boldsymbol{x_t}))$, where $G(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}^d$ is the feature extractor backbone, $F(\cdot)$ is the classifier, and d is the dimension of the feature vector. An autonomous learning agent would not have control over the order of data it receives, meaning there is no guarantee of i.i.d. in relation to previous samples. In addition, the model assumes that each sample is seen only once and cannot be stored. The agent must act on the newly learned knowledge without delay and seamlessly arbitrate between learning and inference. Besides, no task labels or boundaries are available to the model.

Open-world Continual Learning. We extend the FS-OCL protocol to include other crucial aspects of real-world autonomous learning. First, at any time t, we consider the set of known object classes $\mathcal{K} = 1, 2, .., C \in \mathbb{N}_+$ where $\mathbb{N}_+$ denotes the set of positive integers. We embrace the "open-world" assumption in our simulation of the natural learning environment for autonomous agents, acknowledging the existence of an unknown set of classes $\mathcal{U} = C + 1, ...,$ that might be encountered later. The known object classes $\mathcal{K}$ are regarded as base classes and labeled in the dataset, ensuring sufficient samples exist for each class to facilitate effective learning. Conversely, unknown classes emerge as the open-world scenario progresses, presenting a limited number of unlabeled samples. At this stage, the established model $\mathcal{M}_C$, trained on C classes, is tasked with identifying and learning these new instances in a few-shot manner, operating without supervision. The model aims to acquire new knowledge while preserving existing foundational knowledge. Consistent with the principles of semi-supervised continual learning [33], [36], it is assumed that real labels will intermittently be provided for the unsupervised gained knowledge.

IV. Continually Learning Prototypes

To tackle the challenges of Open-World Continual Learning (OWCL) for autonomous agents, we propose a novel method called *Continually Learning Prototype* (CLP), an online open-world continual learning method. This method is designed for online learning in dynamic real-world environments, offering several key capabilities: 1) the ability to learn from a continuous stream of data without experiencing catastrophic forgetting, 2) learning in situations with limited data (few-shot learning), 3) detecting and adapting to novel information, and 4) learning without the need for supervision. CLP is built as a prototype-based algorithm, utilizing prototypes as representatives for clusters of instances (Fig. 1), and it seamlessly enables online, few-shot learning (Sec. IV-A). Crucially, it introduces a novel mechanism for adapting the learning rate of individual prototypes (Sec. IV-B), addressing the plasticity-stability dilemma and hence catastrophic forgetting, inspired by the metaplasticity observed in biological neurons [9], [37]. Furthermore, CLP pursues the approach of on-demand allocation of new prototypes and incrementally constructing a multi-modal (multi-cluster) representation for each class, efficiently capturing both simple and complex classes (Sec. IV-D). The allocation process is triggered by CLP's novelty detection mechanism, crucial in tackling open-world scenarios (Sec. IV-C).

A. Prototype-based Learning

At its core, our classifier method (CLP) employs the concept of prototypes. It determines the winner prototype that is most similar to the input feature vector and updates it towards or away from this vector, based on whether it made a correct or incorrect prediction, respectively. Most prototype-based methods use Euclidean distance to find the best matching prototype [38]. On the other hand, CLP uses dot-product similarity, together with normalized feature and prototype vectors, as a proxy to achieve cosine similarity, which is shown to be a more powerful similarity measure for large dimensional vectors [39], [29]. The decision to avoid using cosine similarity directly stems from the increased complexity it introduces to the update rule for prototypes. In contrast, the dot product is chosen due to its widespread availability and well-optimized computations in neural accelerators such as GPUs and neuromorphic chips [40], [41]. Therefore, the similarity measure used by CLP can be expressed as follows:

$$s(\boldsymbol{\mu}, \boldsymbol{x}) = \boldsymbol{\mu} \cdot \boldsymbol{x}, \quad (1)$$

where $s(\boldsymbol{\mu}, \boldsymbol{x})$ is the similarity between a prototype and input sample $\boldsymbol{\mu}, \boldsymbol{x} \in \mathbb{R}^d$ and $\|\boldsymbol{\mu}\| = \|\boldsymbol{x}\| = 1$.

a) Online Learning with Prototypes: Utilizing the defined similarity measure as the cornerstone, CLP adopts a learning rule rooted in stochastic gradient descent, making it adaptable to online learning settings. This characteristic allows CLP to efficiently handle dynamic learning scenarios. First, let's define a prototype population as $P = \{(\boldsymbol{\mu^i}, l^i)\}_{i=1}^n$, where $\boldsymbol{\mu^i}, l^i$ are the center and label of a prototype p_i. The learning rule to update the winner prototype's mean $\boldsymbol{\mu^*}$ when learning input pair $(x, \hat{y})$ can then be written using the gradient of the similarity measure described in (1):

$$\boldsymbol{\mu^*} \leftarrow \boldsymbol{\mu^*} + \alpha \Psi(y^*, \hat{y}) \nabla_{\mu} s(\boldsymbol{\mu^*}, \boldsymbol{x}).$$

We can simplify this equation by using the equality $\nabla_{\mu} s(\boldsymbol{\mu}, \boldsymbol{x}) = \nabla_{\mu} s(\boldsymbol{\mu} \cdot \boldsymbol{x}) = \boldsymbol{x}$:

$$\boldsymbol{\mu^*} \leftarrow \boldsymbol{\mu^*} + \alpha \Psi(y^*, \hat{y}) \boldsymbol{x}, \tag{2}$$

$$\text{where } \boldsymbol{\mu^*} = \underset{i \in [1,n]}{\operatorname{argmax}} \; s(\boldsymbol{\mu^i}, \boldsymbol{x}), \qquad y^* \leftarrow l^*,$$

$$\Psi(y^*, \hat{y}) = \begin{cases} +1 & \text{if } y^* = \hat{y}, \\ -1 & \text{if } y^* \neq \hat{y}, \\ \beta & \text{if unlabeled sample.} \end{cases} \tag{3}$$

α is the learning rate, y^* is the model prediction, which is the label (l^*) of the winner prototype, and $\Psi(y^*, \hat{y})$ is the evaluator of the prediction, in which $\beta \in [0, 1)$ is a positive scalar for unsupervised learning case (Sec. IV-C).

b) Few-shot Learning with Prototypes: In addition to learning online, CLP can also swiftly assimilate novel examples into its knowledge without catastrophic forgetting. This is particularly significant in the context of few-shot learning, where the primary challenge is to acquire meaningful insights from limited data without succumbing to overfitting. CLP tackles this challenge by posing a straightforward inductive bias: there exists an embedding (feature) space where samples group into clusters, which are learnable even from scarce data via prototypes [28]. This allows CLP to perform in different learning scenarios, not only when data is abundant and well-structured but also in one-shot or few-shot settings.

B. Metaplasticity for Continual Learning

While the aforementioned capabilities of CLP make a good start towards our goal of autonomous continual learning, the key contribution of CLP lies in how it manages continual learning. We approach continual learning and catastrophic forgetting from the perspective of the plasticity-stability dilemma. We argue that different parts of knowledge should have different levels of stability, which can be derived from their performance history [42], [9]. For each piece of continually learned knowledge, i.e., for every prototype, CLP maintains an individual plasticity level or learning rate. The performance history is tracked in a variable called "goodness". As a prototype evolves and makes correct predictions, its "goodness" level increases; thus, its learning rate decreases. This is a type of consolidation process that protects knowledge from interference and forgetting during the subsequent learning sessions. However, if a prototype frequently leads to erroneous decisions, the opposite happens, and the learning rate starts to increase. Then, the increased plasticity allows this prototype to adjust and fix the faulty behavior. CLP's adaptive learning rate mechanism (metaplasticity) addresses the plasticity-stability dilemma by selectively consolidating useful knowledge while justifiably forgetting outdated or erroneous parts. Hence, we extend the definition of the prototype population to $P = \{(\boldsymbol{\mu^i}, \alpha^i, g^i, l^i)\}_{i=1}^n$, where $\boldsymbol{\mu^i}, \alpha_i, g_i, l_i$ are the center, learning rate, goodness score and label of the prototype p_i. Mathematically, we modified (2) to make the learning rate individual to each prototype and a function of the goodness as described below:

$$\boldsymbol{\mu^*_{t+1}} \leftarrow \boldsymbol{\mu^*_t} + \alpha^*_t(g^*_t) \Psi(y^*, \hat{y}_t) \boldsymbol{x_t}; \qquad \alpha^*_t = \frac{1}{g^*_t}, \tag{4}$$

$$g^*_{t+1} \leftarrow \max(1, g^*_t + \Psi(y^*, \hat{y}_t)). \tag{5}$$

Specifically, $g^*(t)$ is the goodness score of the winner neuron at time t. Note that $\forall i \in [1, n], g^i_{t=0} = 1$ and hence $\alpha^i_{t=0} = 1$. This means each prototype is assigned an initial learning rate of 1, ensuring it memorizes the initial input feature vector that led to its allocation, i.e., using this vector as the initial center. This becomes crucial when we discuss novelty detection in Sec. IV-C, as encountered novel instances trigger the allocation of new prototypes.

C. Open-world Recognition and Semi-supervised Learning

CLP's other key contribution is its novelty detection mechanism, which opens CLP to the possibility of encountering unknown instances. In a real-world setting, the recognition tasks are almost always open-set, i.e., the learning system will encounter classes that it has not been trained on before [36], [43]. CLP performs open-set learning by detecting unknown instances. As each prototype has a recognition boundary defined by a similarity cutoff c, there will be regions in feature space, called "open space", that are not covered by any of the learned prototypes [6]. CLP detects any sample falling into this space as a novel/unknown instance. This can be described by the CLP's novelty detector function $\nu(\boldsymbol{x}) : \mathbb{R}^d \rightarrow [0, 1]$ and if such novelty is detected, the model predicts label 0:

$$\nu(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \forall p_i \in P, s(\boldsymbol{\mu_i}, \boldsymbol{x}) < c \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

$$y^* = \begin{cases} 0 & \text{if } \nu(\boldsymbol{x}) = 1 \\ l^* & \text{otherwise,} \end{cases}$$

where l^* is the label of the prototype that is most similar to the input x. Crucially, checking if all prototypes' response is below c is a fast operation on parallel or neuromorphic hardware, while rather slow on CPU, where latency would grow linearly with the number of prototypes. After novelty detection, an autonomous learning system should also learn these novel instances on the fly without supervision. CLP learns each such instance by allocating a prototype and memorizing this sample as its center. If, subsequently, some similar samples are encountered, this novel prototype learns the center of the cluster that it situates without supervision, through the unlabeled case of the equation (3). This optimistically assumes the model's prediction is correct but slows down learning to account for the uncertainty of this optimism by a factor of $\beta < 1$. In the beginning, CLP assigns to such a prototype a unique pseudo-label $l_u \in \mathbb{N}_-$, where $\mathbb{N}_-$ is the set of negative integers. When a labeled sample (x_t, y_t) is recognized by the prototype p^j at time t, CLP updates the label of this prototype to the actual label: $l^j \leftarrow y_t$. In a real-world scenario, this can be implemented by allowing the agent to store the raw samples that triggered new prototype allocations and later ask a human user or multi-modal LLM to provide labels for the stored samples, which in turn can be assigned to the corresponding prototypes. Thus, by combining novelty detection, unsupervised continual adaptation, and subsequent labeling, CLP achieves semi-supervised continual learning in an open-set setting.

Another implication of the unsupervised adaptation is the compensation for the possible concept drifts. The non-stationary nature of the real world means that the statistical properties of input data can gradually change over time, a phenomenon known as concept drift [44]. To manage such cases, CLP keeps a small baseline plasticity level even for the most consolidated prototypes, allowing them to compensate for the gradual concept drift without supervision.

D. Learning Multi-modal Representations

Gradually drifting classes are not the only challenge in the online open-set continual learning setting. There can also be abrupt changes or unfamiliar instances of a known class. Most incremental learning methods assume the uni-modality of classes in embedding space; however, that is not guaranteed. As we will see in Sec. V, this assumption does not hold for classes encountered in natural settings that an autonomous agent is exposed to. Therefore, we designed CLP to learn each class as a set of clusters. Thanks to its novelty detection mechanism, CLP can learn simple (uni-modal) and complex (multi-modal) classes by assigning the prototypes to the classes on demand. This way, CLP can learn more complex decision boundaries than other non-parametric approaches [28], [6], which only learn a single prototype per class. In CLP's case, each class may have a different number of prototypes, allocated progressively as the learning continues. Only the maximum size of the prototype population, shared among all classes, must be specified in the beginning. For some classes, one prototype may be enough, while other more complex ones may require the allocation of more prototypes. The flexibility of this on-demand prototype allocation also means the resources are used more efficiently.

V. Experiments

A. General Experimental Setting

To evaluate our method, we use OpenLORIS dataset [45], collected in a real-world setting by a camera attached to a robot. The dataset includes 121 object instances divided into 40 unique classes. Every class has a varying number of object instances between 1 and 9. Each object instance is recorded considering four different environmental factors: clutter of the scene, illumination, occlusion, and pixel size of objects as shown in Fig. 1. For each variation, three levels of difficulty are considered. In addition, the objects are recorded in three different contexts (home, office, and mall). Hence, $4 \times 3 \times 3 = 36$ videos are recorded for each of the 121 objects. We run three experiments that share the following characteristics: first, the learning is always online, i.e., the network learns one sample at a time, and second, we go through data only once, as in stream learning. Finally, we leverage the off-the-shelf ImageNet-trained EfficientNet-B0 model as the static feature extractor (backbone) $G(\cdot)$. This architecture is specifically designed for edge devices. Note that we chose the backbone based on the findings from Hayes and Kanan's online continual learning experiments [4]. Although CLP could seamlessly work with any backbone, experimenting with different backbones is beyond the scope of our work.

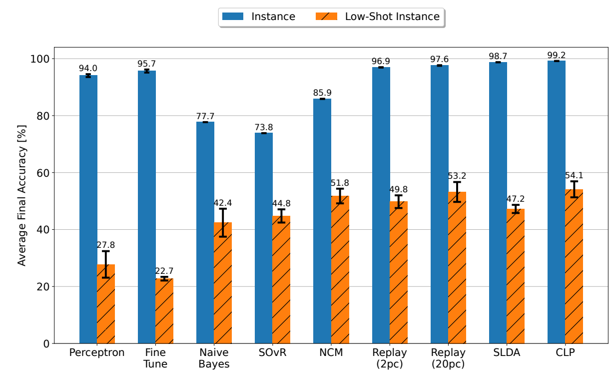


Fig. 2: Fully supervised online continual learning of all classes in the OpenLORIS dataset. The error bars are generated from three random experiments.

B. Fully Supervised Online Continual Learning

In our first experiment, we use a protocol employed by Hayes and Kanan [4], i.e., fully supervised online continual learning of all classes in a single pass through the dataset. We differentiate two cases where training can be done with one or all training videos per class, respectively called "low-shot instance" and "instance" ordering [4]. Using the same experimental setting, we compare CLP to the competing single-layer continual learning methods described in [4]. We measure CLP's performance and reproduce the results of the other methods using the author's open-source code[2]. We ran the experiment three times and compared the results of the CLP to the other methods, as shown in Fig. 2. CLP surpasses all previous methods, both on instance and low-shot instance ordering, setting the state-of-the-art results for online continual learning on the OpenLORIS dataset.

C. Open-set Recognition with Novelty Detection

In the next experiment, we evaluate the CLP's open-set learning capability described in Sec. IV-C. We divide our

[2] https://github.com/tyler-hayes/Embedded-CL

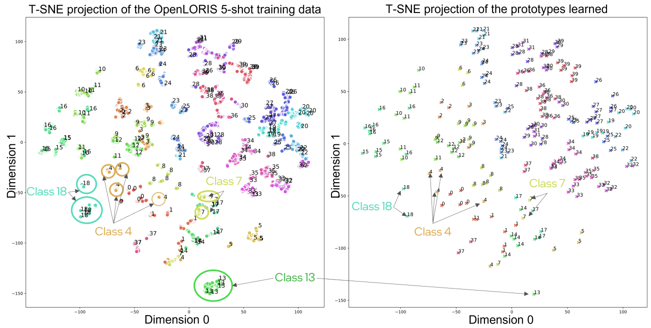
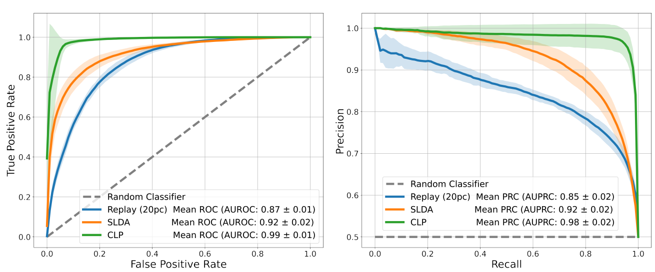


Fig. 3: T-SNE visualization of the OpenLORIS features extracted by pre-trained EfficientNet-B0 backbone and learned prototypes. We randomly chose ten videos (60 frames each) for all 40 classes. The videos from each category may include different object instances but also variations of the same instances. As pointed out in the figure, some classes are represented with a single cluster and hence a single prototype (e.g., class 13), while others are clustered into varying numbers of clusters (e.g., class 4, 7, and 18) and accurately represented by multiple prototypes. This demonstrates that the methods that learn each class with a single prototype are inadequate. Conversely, CLP has adequate representational power thanks to its per-class, on-demand, multi-prototype learning mechanism.



(a) (b)

Fig. 4: Performance analysis of CLP's novelty detection mechanism: (a) ROC curve, (b) precision-recall curve. Shadings indicate standard deviation across the 3 runs.

training set into base and novel class groups, each with 20 non-overlapping object categories. Initially, CLP learns the base classes using all the training samples as in the first experiment. Then, we present 8,000 samples from both the base and novel class test sets to CLP and record the output of the novelty detection function $\nu(\boldsymbol{x})$. The true labels of the novelty detection are zeros for the base and ones for the novel class samples. For each sample, we record the similarity to the most similar prototype as the measure of known class detection. Because novelty detection is regarded as a binary classifier, we can use the receiver operating characteristic curve (ROC) to illustrate its performance as the discrimination threshold is varied [46]. Moreover, the area under ROC (AUROC) is a commonly used metric for measuring open-set detection capabilities [47], [48]. In this experiment, CLP was trained using a prototype similarity cutoff at $c = 0.70$. It is important to note that this similarity cutoff was determined experimentally by selecting the best-performing value for this application. However, considering it as a hyperparameter for future optimization is a potential direction for future work. We compared the CLP performance to the second and third highest-ranking models shown in Fig. 2. In related work, SLDA is recognized as the top method for full-shot instance detection, while output layer fine-tuning with a replay buffer (using 20 samples per class) is best for low-shot instance detection (SLDA and Replay(20pc) in Fig. 2). To produce known/unknown class detection probabilities for each sample, we used the smallest Mahalanobis [7], [49] distance to the sample for SLDA and the top Softmax probability for the replay method. The comparative ROC curves can be seen in Fig. 4a, with the mean AUROC values displayed in the legends. CLP demonstrates superior novelty detection performance, with a mean AUROC of 0.99, surpassing both compared methods. To further evaluate novelty detection, we calculated precision and recall for varying discrimination thresholds between known and unknown classes, as illustrated in Fig. 4b. Similar to AUROC, the area under the precision-recall curve (AUPRC) indicates the effectiveness of the novelty detector, with higher scores reflecting low false positive rates (high precision) and successful detection of most novel instances (high recall). These results show that CLP not only excels in supervised closed-set settings but also outperforms its nearest competitors (SLDA & Replay(20pc)) in detecting novelty in an open-set recognition scenario.

D. *Semi-supervised Few-shot Continual Learning*

In a real-world setting, the intuitive next step after detecting novel instances is to learn them. Thus, we designed the next scenario as the continuation of the previous experiment, where CLP continually learned 20 base classes in a streaming fashion with all available data and labels, building the base model. After the successful validation of CLP's novelty detection (Sec. V-C), this base model can now learn the other 20 novel classes with k-shots ($k = [1, 5, 10, 25]$) per class where each shot is a short video rather than a frame. Crucially, CLP learns these on top of its existing knowledge without supervision. Every time CLP detects a novel instance, it will allocate a new prototype and memorize this sample as its center (Sec. IV-C). Later, CLP continually update unlabeled prototypes without supervision as they detect more samples within their detection boundaries and recognize them using pseudo-labels. Hence, CLP operates as a novelty detection-assisted clustering algorithm when the labels are not available. In line with the semi-supervised continual learning paradigms [33], [36], [50], we assume that the autonomous agent will take a snapshot of each object that triggered novelty detection, and this will be associated with the corresponding newly allocated prototype. Once in a while, a user may provide an actual label for each new prototype allocated during the autonomous, unsupervised learning phase. Of course, this requires significantly less labeling than labeling of an equivalent dataset. To summarize, the experiment simulates a real-world autonomous learning scenario with three phases: (1) base training with

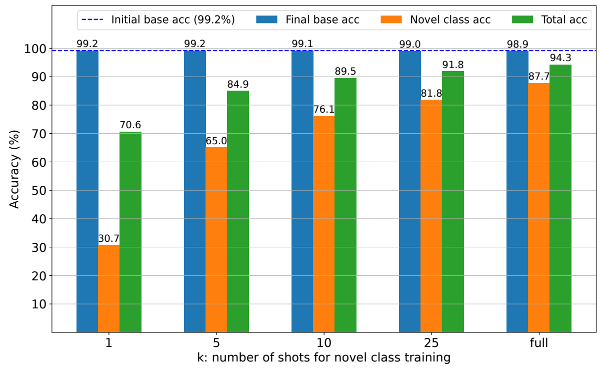


Fig. 5: Few-shot semi-supervised continual learning. CLP retains base class accuracy while learning novel classes without supervision. As the number of videos provided per novel class increases, the accuracy improves significantly.

supervision, (2) open-world few-shot training of novel classes without supervision, and (3) labeling of the novel prototypes. Note that in both phases 1 and 2, CLP learns online through a single pass over the data. Once all the new prototypes are labeled, we test on all base and novel classes.

Guided by earlier experiment outcomes (Sec. V-C) and hyperparameter tuning on the validation set, we adopt a novelty detection threshold of $c = 0.7$. Newly learned prototypes inherit the labels of their allocating instances. Post-initial training, CLP lacks exposure to base class instances, so it cannot rehearse. We record the accuracy on base classes as the baseline to later quantify forgetting. After phase 3 of learning, we measure accuracy both on base and novel classes as presented in Fig. 5. Notably, CLP maintains exceptional recognition of the base classes, exhibiting only marginal decreases in accuracy (up to a maximum of 0.3% when employing the entire training data for novel classes). CLP also achieves unsupervised learning even with one shot. As anticipated in the context of unsupervised learning, the accuracy of novel classes is naturally lower than its supervised counterpart, e.g., in the case of 1-shot learning, we get 30.7% accuracy compared to 54.1% in a supervised setting (Fig. 2). However, this accuracy improves significantly as we increase the number of shots. Notably, in the instance of the full-shot scenario, achieving a novel class accuracy of 87.7% closely aligns with the novelty detection recall (85%) at $c = 0.7$. In addition, CLP allocated 1692 prototypes for base classes and, respectively, 54, 193, 292, and 602 prototypes for 1, 5, 10, and full-shot unsupervised training of novel classes. To summarize, we demonstrate that CLP reliably learns novel classes in a few shots and without supervision, thanks to its superior novelty detection, while still remembering the base classes with minimal loss of accuracy. We believe that the CLP can serve as a strong OWCL baseline for future research into methods addressing similar challenges.

VI. Conclusion & Outlook

In this work, we tackle the challenge of enabling autonomous robots to learn continuously from a non-repeated, sparsely labeled data stream, a critical requirement for achieving true life-long learning. We introduced Continually Learning Prototypes (CLP), a novel approach designed to overcome the limitations of existing continual learning (CL) methods that are often unsuitable for realistic robotic applications. By simulating an autonomous agent that first learns a set of base classes in a supervised environment and then deploying CLP in an open-world setting, we demonstrated its ability to acquire new knowledge without forgetting previously learned information. CLP excels in both supervised and open-world scenarios, achieving state-of-the-art accuracy, with 99.2% when training with all video instances and 54.1% when trained with minimal data in the low-shot instance setting. In open-world scenarios, CLP detects novel objects with superior precision and recall, and it learns new categories without supervision, achieving 99% accuracy for base classes and 76% accuracy for novel classes in a 10-shot scenario. This robust performance underscores CLP's adaptability as a learning algorithm capable of handling complex, real-world environments. Furthermore, CLP's rehearsal-free nature, and compatibility with neuromorphic hardware highlight its potential as a powerful tool for the next generation of autonomous, intelligent robots.

Nevertheless, our current work has some limitations. The hyperparameters like similarity cutoff c and unsupervised learning scaling factor β, are determined experimentally rather than through a more systematic approach. Additionally, there is no mechanism for removing outdated or unuseful prototypes. We also recognize the need for further analysis using different data orderings (e.g. class-incremental setting). It would also be beneficial to demonstrate CLP's effectiveness on other tasks, such as action recognition, and with different data modalities, including audio and odor. Moving forward, we aim to address these limitations by pursuing the following research directions: 1) enabling CLP to learn the similarity cutoff individually for each prototype, 2) implementing a forgetting mechanism for unuseful prototypes, 3) integrating object detection with CLP to develop a continual object detection and learning system, and 4) designing a fully neuromorphic version of the algorithm to run on Intel's Neuromorphic Research Chip, Loihi 2 [40].

References

[1] R. Hadsell, D. Rao, A. A. Rusu, and R. Pascanu, "Embracing change: Continual learning in deep neural networks," *Trends in Cognitive Sciences*, vol. 24, pp. 1028–1040, 12 2020.

[2] S. Yan, J. Xie, and X. He, "Der: Dynamically expandable representation for class incremental learning," *CVPR*, pp. 3013–3022, 3 2021.

[3] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[4] T. L. Hayes and C. Kanan, "Online continual learning for embedded devices," *arXiv preprint arXiv:2203.10681*, 2022.

[5] U. Michieli and M. Ozay, "Online continual learning for robust indoor object recognition," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 3849–3856.

[6] A. Bendale and T. Boult, "Towards open world recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1893–1902.

[7] R. Roady, T. L. Hayes, R. Kemker, A. Gonzales, and C. Kanan, "Are open set classification methods effective on large-scale datasets?" *Plos one*, vol. 15, no. 9, p. e0238302, 2020.

[8] M. Jafarzadeh, A. R. Dhamija, S. Cruz, C. Li, T. Ahmad, and T. E. Boult, "A review of open-world learning and steps toward open-world learning without labels," *arXiv preprint arXiv:2011.12906*, 2020.

[9] P. J. et al, "Contributions by metaplasticity to solving the catastrophic forgetting problem," *Trends in Neurosciences*, vol. 45, no. 9, pp. 656–666, 2022.

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[11] M. De Lange and T. Tuytelaars, "Continual prototype evolution: Learning online from non-stationary data streams," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8250–8259.

[12] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[13] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-gem," *arXiv preprint arXiv:1812.00420*, 2018.

[14] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[15] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 139–154.

[16] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1131–1140.

[17] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.

[18] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[19] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[20] C.-Y. Hung, C.-H. Tu, C.-E. Wu, C.-H. Chen, Y.-M. Chan, and C.-S. Chen, "Compacting, picking and growing for unforgetting continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[21] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, and M. Nabi, "Learning to remember: A synaptic plasticity driven framework for continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 321–11 329.

[22] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[23] V. Losing, B. Hammer, and H. Wersing, "Incremental on-line learning: A review and comparison of state of the art algorithms," *Neurocomputing*, vol. 275, pp. 1261–1274, 2018.

[24] Y. Xu, F. Shen, and J. Zhao, "An incremental learning vector quantization algorithm for pattern classification," *Neural Computing and Applications*, vol. 21, pp. 1205–1215, 2012.

[25] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5871–5880.

[26] T. L. Hayes and C. Kanan, "Lifelong machine learning with deep streaming linear discriminant analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 220–221.

[27] T. L. Hayes, N. D. Cahill, and C. Kanan, "Memory efficient experience replay for streaming learning," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9769–9776.

[28] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[29] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 9057–9067.

[30] A. Ayub and A. R. Wagner, "Tell me what this is: Few-shot incremental object learning by a robot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8344–8350.

[31] K. R. A. et al, "Infinite mixture prototypes for few-shot learning," *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 348–357, 2 2019.

[32] A. Ayub and A. R. Wagner, "Cognitively-inspired model for incremental learning using a few examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 222–223.

[33] Y. Y. Shen, Y. M. Zhang, X. Y. Zhang, and C. L. Liu, "Online semi-supervised learning with learning vector quantization," *Neurocomputing*, vol. 399, pp. 467–478, 7 2020.

[34] O. Beyer and P. Cimiano, "Dyng: Dynamic online growing neural gas for stream data classification." in *ESANN*, 2013.

[35] M. Ren, M. L. Iuzzolino, M. C. Mozer, and R. S. Zemel, "Wandering within a world: Online contextualized few-shot learning," *arXiv preprint arXiv:2007.04546*, 2020.

[36] K. Joseph, S. Khan, F. S. Khan, and V. N. Balasubramanian, "Towards open world object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5830–5840.

[37] D. Kudithipudi, M. Aguilar-Simon, J. Babb, M. Bazhenov, D. Blackiston, J. Bongard, A. P. Brna, S. Chakravarthi Raja, N. Cheney, J. Clune *et al.*, "Biological underpinnings for lifelong learning machines," *Nature Machine Intelligence*, vol. 4, no. 3, pp. 196–210, 2022.

[38] M. Biehl, B. Hammer, and T. Villmann, "Prototype-based models in machine learning," *Wiley Interdisciplinary Reviews: Cognitive Science*, vol. 7, pp. 92–111, 3 2016.

[39] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 831–839.

[40] M. Davies, N. Srinivasa, T. H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, pp. 82–99, 2018.

[41] E. Hajizada, P. Berggold, M. Iacono, A. Glover, and Y. Sandamirskaya, "Interactive continual learning for robots: a neuromorphic approach," in *Proceedings of the International Conference on Neuromorphic Systems 2022*, 2022, pp. 1–10.

[42] F. H. Hamker, "Life-long learning cell structures—continuously learning without catastrophic interference," *Neural Networks*, vol. 14, no. 4-5, pp. 551–573, 2001.

[43] D. Bohus, S. Andrist, A. Feniello, N. Saw, and E. Horvitz, "Continual learning about objects in the wild: An interactive approach," in *Proceedings of the 2022 International Conference on Multimodal Interaction*, 2022, pp. 476–486.

[44] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[45] Q. She, F. Feng, X. Hao, Q. Yang, C. Lan, V. Lomonaco, X. Shi, Z. Wang, Y. Guo, Y. Zhang *et al.*, "Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 4767–4773.

[46] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[47] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[48] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[49] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[50] Y. Shu, Y. Shi, Y. Wang, T. Huang, and Y. Tian, "P-odn: Prototype-based open deep network for open set recognition," *Scientific Reports 2020 10:1*, vol. 10, pp. 1–13, 4 2020.