Label-Efficient Online Continual Object Detection in Streaming Video

Jay Zhangjie Wu[1] David Junhao Zhang[1] Wynne Hsu[2] Mengmi Zhang[3,4] Mike Zheng Shou[1]*

[1]Show Lab, [2]National University of Singapore
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[4]CFAR and I2R, Agency for Science, Technology and Research, Singapore

Abstract

Humans can watch a continuous video stream and effortlessly perform continual acquisition and transfer of new knowledge with minimal supervision yet retaining previously learnt experiences. In contrast, existing continual learning (CL) methods require fully annotated labels to effectively learn from individual frames in a video stream. Here, we examine a more realistic and challenging problem—Label-Efficient Online Continual Object Detection (LEOCOD) in streaming video. We propose a plug-and-play module, Efficient-CLS, that can be easily inserted into and consistently improve existing CL algorithms for object detection in video streams with reduced data annotation costs and model retraining time. We show that our method has achieved significant improvement with minimal forgetting across all supervision levels on two challenging CL benchmarks for streaming real-world videos. Remarkably, with only 25% annotated video frames, our proposed method still outperforms the state-of-the-art CL models trained with 100% annotations on all video frames. The data and source code will be publicly available at https://github.com/showlab/Efficient-CLS.

1. Introduction

Humans have the ability to continuously learn from an ever-changing environment, while retaining previously learnt experiences. In contrast to human learning, prior works [3, 2, 11, 34, 7] show that deep neural networks are prone to catastrophic forgetting. To address the forgetting problem, existing works in continual learning (CL) primarily focus on class-incremental image classification or object detection. Their experiment settings are often idealistic and simplified, where *i.i.d. static images* are usually grouped by class and incrementally presented to computational models in sequence. To learn a particular task containing specific classes, an agent can go through all the data of current task *over multiple epochs*. After that, the learned classes in current task become unavailable, *i.e*., *no overlaps* between the sets of learned classes and unseen classes.

However, these experiment designs deviate from the online continual learning (OCL) setting in the real world, where an agent learns from *temporally correlated non-i.i.d. video streams* in *one single pass*. Given context regularities in natural environments, an agent is likely to encounter cases when objects of previously learnt classes *co-occur* with unknown objects from unseen classes, *e.g*., a computer mouse and a computer monitor often co-occur. Taking these considerations, [34] introduces OCL on object detection in real-world video streams. They evaluate existing CL approaches on this setting and report a huge performance gap compared with offline training.

Based on the setting in [34], we take a significant step further and introduce a novel problem setting called Label-Efficient Online Continual Object Detection (LEOCOD), which highlights two unique challenges. **First**, the setting in [34] is such that the CL algorithms are trained with every mini-batch over multiple passes. We tighten the training recipe in LEOCOD to strictly online, where data is allowed to have one single pass and models are trained on the entire video dataset for only one epoch. **Second**, existing CL models require fully supervised training where box-level ground truth labels of every object on every video frame have to be obtained from human annotators. Unlike static images, acquiring human annotations for object detection on videos can be expensive and daunting. Thus, in LEOCOD, the video frames per mini-batch are sparsely annotated to alleviate the burdens of extensive human labeling, making LEOCOD one step closer to real world application.

Cognitive science works [35, 19] show that humans are efficient at continuously learning from very few annotated data samples. We get inspirations from the theory of Complementary Learning Systems (CLS) in human brains [18], and propose a plug-and-play module for the LEOCOD task, dubbed as Efficient-CLS. In Efficient-CLS, we introduce

*Corresponding Author.

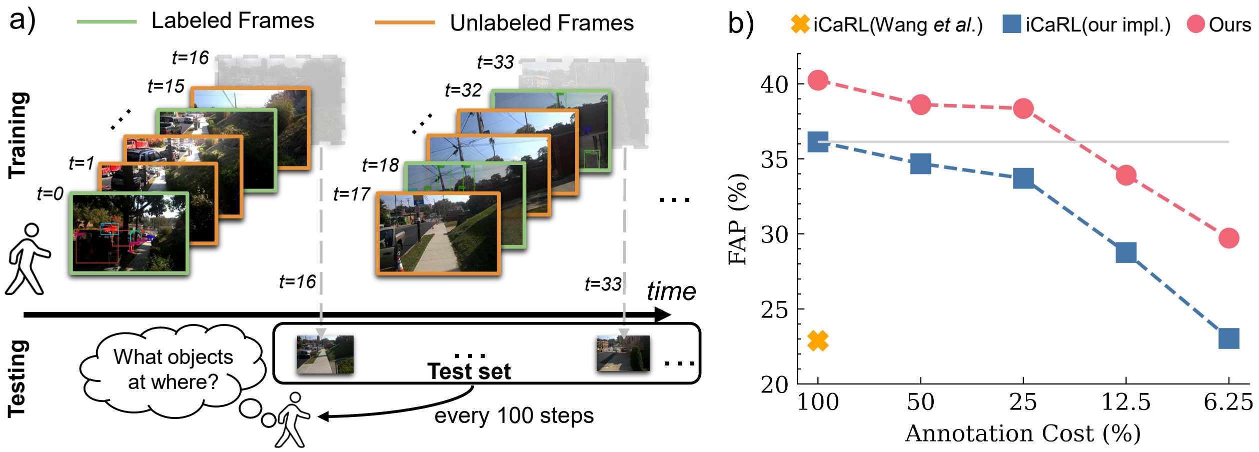

Figure 1. **(a) Problem introduction**: An agent continuously learns from a never-ending online video stream over time. In each training step, out of a mini-batch containing 16 consecutive video frames, only a fixed proportion of frames are labeled (green boundary), while the rest of the frames are unlabeled (orange boundary). Following [34], the video frame after every training mini-batch (transparent) is held out for testing. After every 100 training steps, the agent is evaluated on all the video frames from the test set for object detection. **(b) Key results**: Our proposed method (red) consistently outperforms the best competitive baseline (blue) by a margin of 5%. Remarkably, our model, trained at 25% annotation cost, surpasses the best baseline trained at 100% (grey line). The orange cross denotes the performance of the state-of-the-art model, which is 15% lower than our method.

two feed-forward neural networks as slow and fast learners. In the fast learner, memory is rapidly adapted to the current task. The weights of the slow learner change a little on each reinstatement, and are maintained by taking the exponential moving average (EMA) of the fast learner's weights over time. Though a few continual learning models in previous works [4, 24] also use a similar source of inspiration, they miss the effect of reciprocal connections from slow learners to fast learners, which we intend to address. Inspired by the bidirectional interaction in CLS [14], we reactivate the weights of the slow learners to predict meaningful pseudo labels from the unlabeled video frames and use these pseudo labels to guide the training of the fast learner, closing the loop between the two systems.

We demonstrate the *versatility* and *effectiveness* of our Efficient-CLS on two standard real-world video datasets, OAK [34] and EgoObjects [1]. Our proposed method can be easily integrated into existing CL models and consistently improve their performance by a large margin in LEOCOD. It is worth noting that, with only 25% labeled data, our method surpasses the comparative baselines trained with full supervision (Figure 1(b)).

To summarize, we make the following key contributions:

- We introduce a new, challenging and important problem of label-efficient online continual object detection (LEOCOD) in video streams. Solving this problem would greatly benefit real-world applications in reducing annotation cost and model retraining time.
- We propose Efficient-CLS, a plug-and-play module inspired from the Complementary Learning Systems (CLS) theory, which can be integrated into existing CL models and learn efficiently and effectively with less supervision and minimal forgetting.
- We benchmark existing CL methods on the task of LEOCOD and demonstrate the state-of-the-art performance of our method through extensive experiments.

2. Related Work

2.1. Continual Learning

To alleviate catastrophic forgetting, many continual learning methods exploit an external buffer where a limited number of old samples are stored and used for replay when adapting to a new task. iCaRL [26] stores the representative exemplars in past tasks for knowledge distillation and prototype rehearsal. Gradient Episodic Memory (GEM) [21] formulates optimization constraints on the exemplars in memory. Averaged GEM (A-GEM) [8] is an improved version of GEM that achieves faster training and less memory consumption. GDumb [25] greedily stores samples in memory as they come and trains a model using samples only in the memory. Dark Experience Replay++ (DER++) [6] combines replay with knowledge distillation and regularization, and samples logits along the entire optimization trajectory.

In contrast to classical continual learning where data are separated by task boundaries and models are trained with multiple iterations in every task, we examine a more realistic and challenging problem where data are provided in tiny batches and models are trained on these batches only once. Recently, online continual learning (OCL) has gained increasing interests in computer vision [3, 7, 29, 9, 34]. Many OCL methods rely on representative memory replays to prevent forgetting. [3] utilizes gradients of network parameters to select replay samples of maximum diversity. Subsequent works [2, 29] propose to use losses and scoring functions as criteria for selecting the most representative samples for replay. However, these approaches tackle image classification problem in an artificial setting, where new classes appear in a specific order. Their performance in real-world vision tasks remains unclear.

Lately, [34] benchmarks CL methods in the real-world online setting with full supervision. As the video streams arrive endlessly in a real-time manner, assigning annotations to all the video frames for training computational models is laborious and time-consuming. It becomes even more daunting in object detection tasks where class labels and bounding boxes of all objects on a video frame have to be provided. Reducing burdensome costs of labeling remains an under-explored and challenging problem in online continual object detection. We propose a self-sustaining Efficient-CLS, which is capable of exploiting the unlabeled video frames by pseudo-labeling when the number of labeled frames is limited.

2.2. Complementary Learning Systems

The essence of fast and slow learning in Complementary Learning Systems (CLS) has benefited several continual learning algorithms in image recognition [24, 23, 28, 4, 15]. However, these methods either require the task boundaries, which are not applicable in our online video setting, or they require to train fast and slow learning systems with replay samples from the same replay buffer, which could easily lead to overfitting problem when the replay buffer has limited capacity. To eliminate overfitting problem, [28] and [15] utilize generative replay models to couple sequential tasks in a latent embedding space. While generative approaches have succeeded in artificial and simple datasets, they often fail in complex vision tasks, *e.g.*, object detection. DualNet [23] also employs a slow-fast learning architecture. However, its update process for the slow learner is more computationally demanding because it leans on both self-supervised and supervised learning objectives. The self-supervised phase demands additional training iterations on extensive batches to yield satisfactory results, thereby delaying the training of the fast learner. In contrast, our method updates the slow learner in real-time using the fast learner's weights, obviating the need for distinct slow learner training. Our approach is optimized for online CL scenarios where computational resources are constrained.

2.3. Semi-Supervised Learning

The goal of semi-supervised learning (SSL) [33, 5, 37, 36] is to reduce the demand of labeled data and harness unlabeled data for performance improvement. [31] first introduce SSL to the context of CL, and propose a strategy that combines pseudo-labeling, consistency regularization, out-of-distribution detection, and knowledge distillation to solve the problem of class-incremental image classification. However, to distill knowledge from previous tasks, their model relies heavily on the task boundary that identifies the change of training classes, which is not available in our LEOCOD setting. To reduce annotation costs in object detection, several methods [13, 32, 20] capitalize the teacher-student networks. In general, a teacher model predicts pseudo labels or enforces a consistency loss to guide the student networks. However, these previous works on semi-supervised object detection only consider the offline setting on static image datasets, while none of them has been extended to online continual learning on dynamic video streams. We discovered new insights that pseudo-labeling from slow learner to fast learner can not only reduce annotation overhead, but also alleviate forgetting, which is critical to the design of efficient continual learners in real world.

3. Method

3.1. Problem Setting

We advance the online continual object detection in [34] to a label-efficient and computationally-efficient setting—Label-Efficient Online Continual Object Detection (LEOCOD). In contrast to the setting in [34] where video frames are *extensively annotated* and trained with *multiple epochs*, an agent in LEOCOD continuously learns from a *sparsely annotated* video stream in *a single pass* over time (see Figure 1(a)).

Formally, we consider the online continual object detection on a continuum of video streams $\mathcal{D} = \{D_1, \cdots, D_T\}$ where at time step t, a learning agent receives a mini-batch of continuous video frames D_t from current environment for online training (one single pass). To perform label-efficient object detection, within the batch D_t, only a subset of video frames $D_t^s = (X_t^s, Y_t^s)$ are labeled, while the remaining video frames $D_t^u = (X_t^u)$ are unlabeled. For each labeled data sample, its annotation contains the bounding box locations and their corresponding class labels.

3.2. Efficient Complementary Learning Systems

We propose a plug-and-play module dubbed as Efficient-CLS. Specifically, it consists of two feed-forward networks: (i) the fast learner is designed to quickly encode new knowledge from current data stream and then consolidate it to the slow learner; and (ii) the slow learner accumulates the acquired knowledge from fast learner over time and guides the fast learner with meaningful pseudo labels, when full supervision is not available. Same as [26, 8, 25, 6, 34], we maintain an external episodic memory, as a replay buffer, to store exemplars that can be retrieved for replays alongside ongoing video stream. As the fast and slow learners are

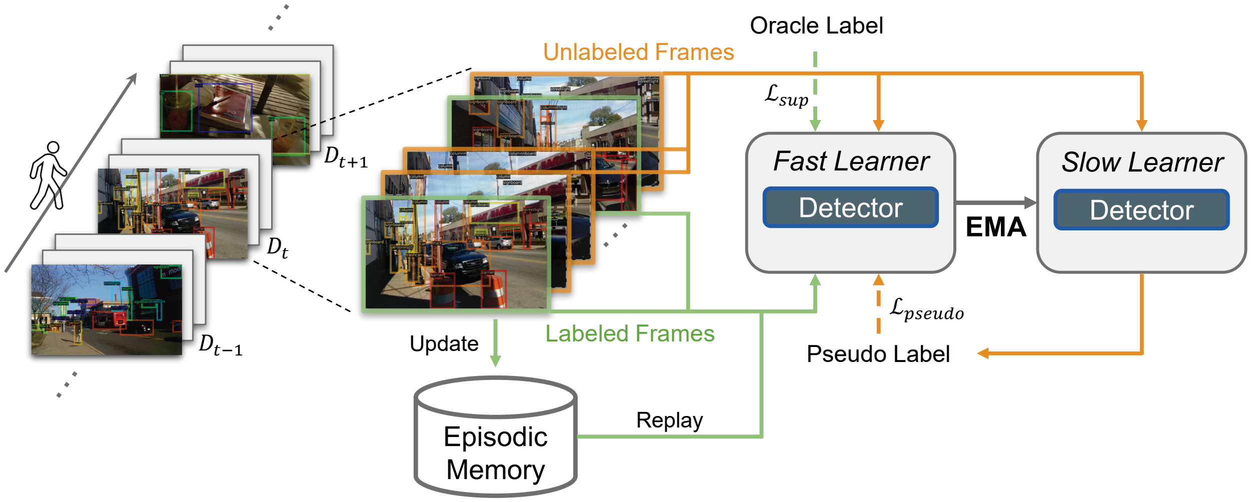

Figure 2. **The framework of Efficient-CLS**. At each learning step, the system receives a batch of temporally continuous data D_t, including labeled (green) and unlabeled (orange) frames. The fast learner trains the labeled frames alongside a small subset of labeled exemplars retrieved from episodic memory with the supervised loss $\mathcal{L}_{sup}$. Meanwhile, the fast learner leverages the pseudo labels generated by the slow learner to optimize a pseudo loss $\mathcal{L}_{pseudo}$. To reinstate memory of the slow learner, the weights of the slow learner are updated by taking the Exponential Moving Average (EMA) of the fast learner's weights. The fast and slow learners are complementary to each other, forming a positive feedback loop.

model agnostic, our Efficient-CLS can be easily integrated into existing CL models, which leads to less supervision and minimal forgetting.

Learning with Labeled Frames. The fast learner and slow learner use the same standard Faster-RCNN [27] detector f. Despite the same architecture, the weights of the fast and slow learners are not shared. We use θ_F and θ_S to denote the network parameters for fast and slow learners respectively. As shown in Figure 2, at each training step t, we use the labeled video frames $D^s_t = (X^s_t, Y^s_t)$ to optimize the fast learner θ_F with the standard supervised loss $\mathcal{L}_{sup}$ in Faster-RCNN [27]. It consists of four losses: Region Proposal Network (RPN) classification loss $\mathcal{L}^{rpn}_{cls}$, RPN regression loss $\mathcal{L}^{rpn}_{reg}$, Region of Interest (ROI) classification loss $\mathcal{L}^{roi}_{cls}$, and ROI regression loss $\mathcal{L}^{roi}_{reg}$. We define $\mathcal{L}_{sup}$ as:

$$\begin{aligned}\mathcal{L}_{sup} = \mathcal{L}^{rpn}_{cls}(X^s_t, Y^s_t) + \mathcal{L}^{rpn}_{reg}(X^s_t, Y^s_t) + \\ \mathcal{L}^{roi}_{cls}(X^s_t, Y^s_t) + \mathcal{L}^{roi}_{reg}(X^s_t, Y^s_t).\end{aligned} \tag{1}$$

Learning with Unlabeled Frames. We introduce a pseudo-labeling paradigm to capitalize the information from unlabeled video frames $D^u_t = (X^u_t)$ for training. In our early exploration, we intuitively use the fast learner for pseudo-labeling as it quickly adapts the knowledge of nearby frames. However, we observe that using the pseudo labels generated by the fast learner for self-replay exhibits biases towards recently seen objects, which is less effective in preventing forgetting. This has also been verified in our ablation study (Section 5.3). In contrast, the slow learner preserves the semantic knowledge over a longer time span which generates pseudo labels with fewer biases. This encourages the fast learner to capture more generic scene representations, hence, in turn, contributing to reinstatement of memory in the slow learner (Section 3.2), resulting in a positive feedback loop.

Given all these design considerations, the slow learner takes the unlabeled video frames D^u_t as inputs to estimate the possible objects of interest and their corresponding bounding box locations. For brevity, we refer these "pseudo bounding boxes and their corresponding class labels" as "pseudo labels" in the paper. To get rid of false positives, we apply a threshold τ to filter out bounding boxes with predicted low confidence scores. Moreover, there also exist repetitive boxes which negatively impact the quality of pseudo-labeling. To address this issue, we use the technique of class-wise non-maximum suppression (NMS) [27] to remove the overlapped boxes and get the high-quality pseudo labels. Formally, the procedure of pseudo label generation is summarized below:

$$Y^u_t = \text{NMS}([f(X^u_t; \theta_S)]_{>\tau}), \tag{2}$$

where $[\cdot]_{>\tau}$ denotes the bounding box selection with confidence score larger than τ.

Given that the video streams are captured from the egocentric perspective in the real world, head and body motions may lead to undesired motion blur effects on some video frames. To enforce our module to learn invariant object representations from these video frames, same as the previous work [38], we apply data augmentation techniques on the pseudo-labeled frames, including 2D image crops, rotations, and flipping. Note that different from image classification, the predicted bounding box locations also need to be updated accordingly after image augmentations. We denote these pseudo-labeled video frames and their re-adjusted pseudo labels after data augmentations as $(\tilde{X}_t^u, \tilde{Y}_t^u)$. We can then use these pseudo pairs $(\tilde{X}_t^u, \tilde{Y}_t^u)$ to train the fast learner by optimizing the pseudo loss $\mathcal{L}_{pseudo} := \mathcal{L}_{cls}^{roi}(\tilde{X}_t^u, \tilde{Y}_t^u) + \mathcal{L}_{reg}^{roi}(\tilde{X}_t^u, \tilde{Y}_t^u)$.

Overall, our Efficient-CLS is jointly trained with the following losses: $\mathcal{L}_{total} = \mathcal{L}_{sup} + \lambda_{pseudo}\mathcal{L}_{pseudo}$, where λ_{pseudo} is the weight of $\mathcal{L}_{pseudo}$.

Synapses Consolidation via EMA. To alleviate forgetting of obtained knowledge, we apply Exponential Moving Average (EMA) to gradually update the slow learner with the fast learner's weights. The evolving weight changes in the slow learner are functionally correlated with the memory consolidation mechanism in the hippocampus and the neocortex [4]. Formally, we define EMA process as:

$$\theta_S = \alpha\theta_S + (1-\alpha)\theta_F, \tag{3}$$

where the $\alpha \in [0, 1]$ is EMA rate. According to the stability-plasticity dilemma, a smaller α means faster adaption but less memorization. Empirically, we set $\alpha = 0.99$, which leads to best performance.

4. Experimental details

4.1. Datasets

We evaluate the continual learning methods on two realistic and challenging video datasets, OAK [34] and EgoObjects [1], for the task of online continual object detection.

OAK [34] is a large egocentric video stream dataset spanning nine months of a graduate student's life, consisting of 7.6 million frames of 460 video clips with a total length of 70.2 hours [30]. The dataset contains 103 object categories. We follow [34] in the ordering of training and testing data splits. One frame every 16 consecutive video frames lasting for 30 seconds is held out to construct a test set and the remaining frames are used for training.

EgoObjects [1] is one of the largest object-centric datasets focusing on object detection task. The dataset contains around 100k video frames with ∼250k annotations, which correspond to 277 categories and 1110 main objects [22]. Additionally, the dataset follows a long-tailed distribution which makes the task more challenging and real-world oriented. We adopt the same rule as OAK dataset to split the training and testing set.

4.2. Baselines

We compare our model against the following baselines: 1) **Vanilla training**: *Incremental Training* is a naive baseline trained sequentially over the entire video stream without any measures to avoid catastrophic forgetting; *Offline Training* is an upper bound that can access all the data throughout training; 2) **Continual learning algorithms**: *EWC* [16], *iCaRL* [26], *A-GEM* [8], *GDumb* [25], *DER++* [6] and *iOD* [17].

The iCaRL model implemented by [34] stands as the SOTA method in online continual object detection. We reproduce their results using the released code[1]. When calculating RPN and ROI losses for replay samples, their iCaRL model neglects the losses of background proposals and penalizes the foreground losses according to the proportion of the current samples and replay samples. We empirically found that this trick hinders the model from effective episodic replay, thus resulting in severe forgetting. Therefore, we re-implement the iCaRL by discarding the reweighting trick and reverting back to the standard RPN and ROI losses. We name these two different implementations as *iCaRL(Wang* et al.*)* and *iCaRL(our impl.)*, respectively.

4.3. Evaluation

Protocols. First, we define the annotation cost as the proportion of number of labeled frames versus the total 16 frames within a mini-batch D_t. For example, if 2 out of 16 consecutive frames within D_t get labeled, the annotation cost is $2/16 = 12.5\%$. The frames to be labeled are randomly selected within each mini-batch D_t. Considering that different choices of labeled frames might influence the model performance, for fair comparisons between models, we fix the choice of randomly selected labeled frames and use the same labeled and unlabeled frames for training all models. Based on the various annotation costs, we introduce two training protocols: *fully supervised protocol* (100% annotation cost) and *sparse annotation protocol* (where the annotation cost is less than 100%). In sparse annotation protocol, we further split the training experiments based on 50%/25%/12.5%/6.25% annotation costs.

Metrics. We evaluate the models with three standard metrics: continual average precision (CAP), final average precision (FAP) and forgetfulness (F) [34]. We employ an Average Precision (AP) metric at an Intersection over Union (IoU) threshold of 0.5, commonly referred to as AP50.

CAP shows the average performance over the entire video stream. That is,

$$\text{CAP} = \frac{1}{N}\sum_{i=0}^{N}\text{CAP}_{t_i} = \frac{1}{NC}\sum_{i=0}^{N}\sum_{c=0}^{C}\text{CAP}_{t_i}^{c}, \tag{4}$$

where $\text{CAP}_{t_i}^{c}$ is the AP of class c on test set at the i-th evaluation step, N is the total number of evaluation steps.

FAP is the final performance of a model after seeing the entire video. That is, $\text{FAP} = \text{CAP}_{t_N}$, where t_N denotes the last evaluation step.

[1] https://github.com/oakdata/benchmark

	Annotation Cost	OAK			EgoObjects		
		FAP (↑)	CAP (↑)	F (↓)	FAP (↑)	CAP (↑)	F (↓)
Incremental	100%	8.38	7.72	0.03	10.21	3.55	1.48
Offline Training	100%	48.28	35.23	-	86.18	59.81	-
EWC	100%	7.73	7.02	-0.12	5.15	1.60	0.57
iOD	100%	7.92	7.14	0.98	8.80	2.64	0.00
iCaRL(Wang *et al.*)	100%	22.89	16.60	-2.95	37.61	21.71	2.79
iCaRL(our impl.)	100%	36.14	26.26	-4.89	60.80	36.41	-0.60
w/ Efficient-CLS	25%	38.36(+2.22)	26.64(+0.38)	-8.20(-3.31)	61.26(+0.46)	39.58(+3.17)	-3.48(-2.88)
	100%	40.24(+4.10)	28.18(+1.92)	-8.10(-3.21)	67.05(+6.25)	40.36(+3.95)	-3.67(-3.07)
A-GEM	100%	36.94	26.19	-5.54	58.79	35.88	-8.38
w/ Efficient-CLS	25%	37.06(+0.12)	26.36(+0.17)	-7.76(-2.22)	63.06(+4.27)	39.46(+3.58)	-7.49(+0.89)
	100%	39.87(+2.93)	27.97(+1.78)	-7.17(-1.63)	66.94(+8.15)	39.57(+3.69)	-11.68(-3.30)
GDumb	100%	35.27	25.29	-6.59	58.85	36.38	-5.21
w/ Efficient-CLS	25%	37.67(+2.40)	25.59(+0.30)	-9.30(-2.71)	62.70(+3.85)	38.78(+2.40)	-8.86(-3.65)
	100%	38.61(+3.34)	26.04(+0.75)	-9.14(-2.55)	63.55(+4.70)	38.98(+2.60)	-7.50(-2.29)
DER++	100%	37.79	25.24	-2.87	55.82	30.84	-6.08
w/ Efficient-CLS	25%	37.93(+0.14)	25.64(+0.4)	-8.90(-6.03)	59.70(+3.88)	34.15(+3.31)	-11.21(-5.13)
	100%	39.61(+1.82)	26.73(+1.49)	-8.30(-5.43)	62.01(+6.19)	33.09(+2.25)	-11.05(-4.97)

Table 1. **Overall performance of existing algorithms and Efficient-CLS on OAK and EgoObjects**. iCaRL(Wang *et al.*) denotes the SOTA model presented in [34], and iCaRL(our impl.) is the same method by our implementation.

F estimates the forgetfulness of the model due to the sequential training. It takes into account the time interval between the presence of an object category and its subsequent presence. For a class c, we sort the $\text{CAP}^c_{t_i}$ according to the time interval k between evaluation time t_i and the last time $t_i - k$ the model is trained on c. After $\text{CAP}^c_{t_i}$ is sorted, all $\text{CAP}^c_{t_i}(i = 0, \cdots, T)$ are divided into K bins $B_{kmin}, \cdots, B_{kmax}$ according to the time interval k. The average CAP (aCAP_k) of each bin B_k is defined as the model's performance for detecting class c after the model has not been trained on c for k time steps. The forgetfulness (F) of the class c is defined as the weighted sum of the performance decrease at each time:

$$\text{F}^c = \sum_{k=kmin}^{kmax} \frac{k - kmin}{\sum_{k=kmin}^{kmax} k - kmin} \times (\text{aCAP}_{kmin} - \text{aCAP}_k). \tag{5}$$

The overall forgetfulness is: $\text{F} = \frac{1}{C}\sum_{c=0}^{C} \text{F}^c$.

4.4. Implementation Details

For a fair comparison, we followed the prior work [34] to use Faster-RCNN [27] with ResNet-50 backbone [12] as our object detection network, which is initialized by the weights pre-trained on PASCAL VOC [10]. We used Adam optimizer with a constant learning rate of 0.0001, and the batch size was set to 16 frames. Same as [34], we maintained a replay buffer with 5 samples per class. At each time step t, we first randomly retrieved 16 video frames from the replay buffer for joint training. We used confidence thresh $\tau = 0.7$ to generate pseudo-labels for unlabeled frames, and applied EMA rate $\alpha = 0.99$ to update the slow learner. The weight of pseudo loss λ_{pseudo} was set to 1.0. All the experiments were conducted on 2 NVIDIA RTX 3090 GPUs.

5. Results

5.1. Fully Supervised Protocol

As the previous work [34] focuses on online continual object detection (OCOD) in video streams, we first evaluated model performance in fully supervised setting (*i.e.*, 100% annotation cost), where all video frames are paired with human labels. The reported results were measured by standard metrics (CAP, FAP, and F, Section 4.3) in Table 1.

Performance of CL baselines. [34] benchmarked Incremental, EWC, iCaRL(Wang *et al.*), and Offline Training on OAK dataset. They found the replay-based method (*i.e.* iCaRL(Wang *et al.*)) outperforms regularization-based method (*i.e.* EWC) by 10% in FAP, while iCaRL(Wang *et al.*) has a huge gap of 30% compared with Offline Training. Similar observations were made in Table 1, but the performance of Incremental and EWC were 4% lower than that in [34], as in our setting we only trained each mini-batch of video frames once (they trained each mini-batch 10 times). As mentioned in Section 4.2, we introduce several variations to the original design of iCaRL(Wang *et al.*). Compared with iCaRL(Wang *et al.*), we observed a huge performance boost in FAP from 22.89% to 36.14% on OAK and from 37.61% to 60.80% on EgoObjects, abridging the gap between the baseline and Offline Training.

We further adapted other standard CL baselines, including iOD, A-GEM, GDumb, DER++, to the OCOD setting for comparisons. iOD is the SOTA method in offline class-incremental object detection. Though it performs well in prior setting, iOD collapses when adapted to online video streams. One explanation is that iOD requires explicit task boundary to trigger the reshape of model gradients that opti-

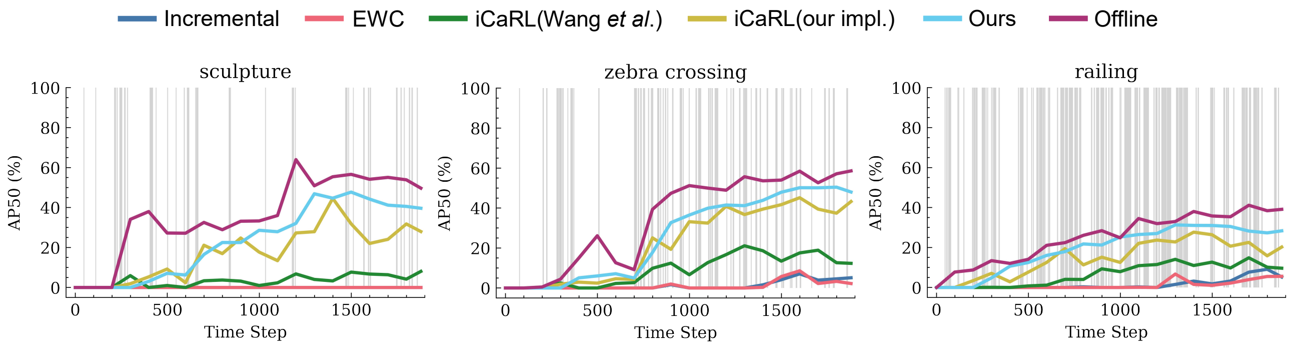

Figure 3. **The changes of $\mathrm{CAP}^c_{t_i}$ with sampled categories on OAK dataset.** The x-axis denotes time step across the entire video stream. The y-axis denotes the AP50 of the category at specific time step (*i.e.*, $\mathrm{CAP}^c_{t_i}$). The grey line indicates the existence of the category.

mizes knowledge sharing between adjacent tasks. However, in online video streams, the task boundaries by classes are no longer available and the change of tasks is hard to identify, resulting in the failure to prevent forgetting. In contrast, replay-based methods (*i.e*., A-GEM, GDumb and DER++) generalize much better to the real-world streaming video.

Performance w/ Efficient-CLS. Inheriting from the benefit of fast and slow learning with EMA, our Efficient-CLS consistently improves all the state-of-the-art CL methods (*i.e*. iCaRL(our impl.), A-GEM, GDumb and DER++) by a significant margin. Taking iCaRL(our impl.) for example, with Efficient-CLS, we observed superior performance and minimal forgetting compared to baseline methods, even when categories appeared infrequently (*e.g*., *sculpture* in Figure 3). Since semantic contextual information is more important in indoor environments on EgoObjects compared to the outdoor environments in OAK, we noticed that the improvement brought by our method is even greater on EgoObjects with an increase of 6.25% in FAP, 3.95% in CAP % and 3.07% in F. Consistent performance gains are also noticeable for A-GEM, GDumb and DER++, demonstrating the effectiveness and flexibility of our method.

5.2. Sparse Annotation Protocol

Performance of CL baselines. The sparse annotation protocol is more challenging than the previous fully supervised protocol as shown by the performance differences when number of annotated video frames decreases (compare the performance of each colored bar along the x-axis within each subplot in Figure 4). We noted that GDumb is more resilient against the reduce of supervision. Specifically, from Table 2 at the lowest annotation cost of 6.25% on EgoObjects dataset, GDumb achieves the highest performance of 38.74%, 22.69% and -4.53% in FAP, CAP and F, which surpasses other baselines by a considerable margin. Same observations can be made on OAK dataset. One possible explanation is that GDumb only trains the data stored in the balanced replay buffer, which makes it less vulnerable to class imbalance problem brought by the reduce of labeled samples in the training set.

	OAK			EgoObjects		
	FAP (↑)	CAP (↑)	F (↓)	FAP (↑)	CAP (↑)	F (↓)
iCaRL	23.04	17.75	-3.31	32.91	19.15	-1.36
w/ Efficient-CLS	**29.72**	**20.31**	**-5.36**	**40.16**	**23.95**	**-2.01**
A-GEM	23.59	16.15	-2.99	21.84	12.28	0.82
w/ Efficient-CLS	**30.18**	**20.59**	**-5.44**	**38.96**	**23.79**	**-5.64**
GDumb	27.37	19.64	-4.25	38.74	22.69	-4.53
w/ Efficient-CLS	**29.07**	**19.99**	**-6.01**	**40.09**	**23.67**	**-5.16**
DER++	24.21	15.93	-3.79	16.95	8.48	2.03
w/ Efficient-CLS	**28.63**	**19.64**	**-4.60**	**35.78**	**20.74**	**-4.69**

Table 2. **Performance of Efficient-CLS at low annotation cost of 6.25%**. The best results are **bold-faced**.

Performance w/ Efficient-CLS. Our proposed Efficient-CLS is a plug-and-play module that can be easily inserted into and enhance existing continual learning algorithms with the ability to use unlabeled video frames effectively. In both OAK and EgoObjects dataset, Efficient-CLS consistently improves the comparative SOTAs in all three evaluation metrics regardless of various degrees of annotation cost. As shown in Table 2, at a lower annotation cost of 6.25%, Efficient-CLS doubles the performance of DER++ and A-GEM in terms of FAP and CAP, and achieves an even larger improvement in preventing forgetting. Thanks to the useful information from pseudo labels predicted by the slow learner in Efficient-CLS, our method is more robust to various annotation costs, compared with SOTAs (compare the rate of change of blue bars *vs*. red bars over different degrees of annotation cost). Most remarkably, Efficient-CLS with 25% annotation cost has already outperformed comparative SOTAs with 100% annotation cost (see Table 1).

5.3. Ablation Study

We assessed the importance of our key design choices. The Complementary Learning Systems (CLS) in Efficient-CLS is the key for rapidly adapting to learn new tasks, meanwhile, retaining previously learnt knowledge. It constitutes of two memory reinstatement mechanisms: one is

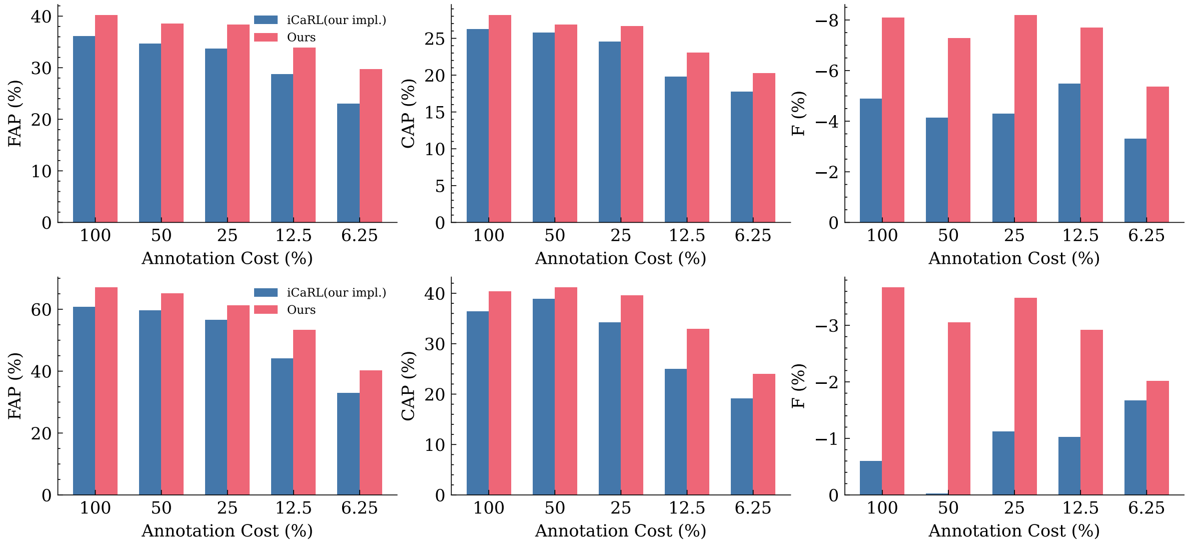

Figure 4. **Evaluation of online continual object detection in video streams with three metrics (FAP, CAP and F, Section 4.3) on OAK dataset (first row) and EgoObjects dataset (second row)**. The higher the bars are, the better. The x-axis denotes the percentage of video frames that are labeled in the video stream. It ranges from 6.25% to 100% (full supervision). The y-axis indcates the performance using different evaluation metrics. Ours (iCaRL(our impl.) w/ Efficient-CLS, red) consistently beats the comparative SOTA (iCaRL(our impl.), blue) in all evaluation metrics.

		50%			25%			12.5%			6.25%		
EMA	PL	FAP (↑)	CAP (↑)	F (↓)	FAP (↑)	CAP (↑)	F (↓)	FAP (↑)	CAP (↑)	F (↓)	FAP (↑)	CAP (↑)	F (↓)
✗	✗	34.68	25.78	-4.15	33.70	24.57	-4.30	28.76	19.80	-5.48	23.04	17.75	-3.31
✓	✗	35.74	25.77	-4.82	34.79	25.62	-4.35	31.72	21.16	-7.24	27.84	20.03	-3.96
✗	✓	35.61	25.56	-3.76	34.95	25.65	-3.65	31.60	22.44	-4.83	26.39	19.50	-1.99
✓	✓	**38.61**	**26.90**	**-7.29**	**38.36**	**26.64**	**-8.20**	**33.92**	**23.04**	**-7.71**	**29.72**	**20.31**	**-5.36**

Table 3. **Effectiveness of Exponential Moving Average (EMA) and Pseudo-labeling (PL) for iCaRL(our impl.) on OAK dataset at annotation cost 50%, 25%, 12.5% and 6.25%**. The best results are **bold-faced**.

synaptic weight transfer from fast to slow learner via exponential moving average (EMA); and the other is reciprocal replay from slow learner to fast learner with pseudo-labeling (PL). Here we studied their effects individually.

Ablation: Exponential Moving Average (EMA). We removed EMA by setting the α in Equation 3 to 1, where the model weights of the fast learner and slow learner are now shared throughout the learning process. Note that in the fully supervised protocol, the pseudo-labeling is turned off and the Efficient-CLS equals to the EMA alone. From Table 1 at 100% annotation cost, we observed that removing EMA leads to significant performance drops ranging from 2% to 8%, for all the comparative baselines on both OAK and EgoObjects. This shows that the slow learner can effectively consolidate the knowledge from the fast learner, and constructively alleviate catastrophic forgetting by synapses consolidation over time. Similar observations were made in Table 3 in the sparse annotation protocol (compare Row 2 *vs*. Row 1, Row 4 *vs*. Row 3). It is worth noting that, the performance difference between naive model (Row 1) and its variant with EMA (Row 2) is slightly larger in lower supervision (*i.e*. 12.5%, 6.25%) than higher supervision (*i.e*. 50%, 25%). One possible reason is that, compared with higher supervision, the fast learner suffers more forgetting in lower supervision; hence, the effect of removing EMA becomes stronger in lower supervision, again highlighting the importance of EMA.

Ablation: Pseudo-labeling (PL). We ablate our model by removing the PL of the slow learner across varying annotation costs and reported the results in Table 3. The removal of PL (Row 2) leads to a performance drop of around 2% in FAP, 1% in CAP and 0.5-4% in F, compared with our full Efficient-CLS (Row 4). It implies that the slow learner captures useful semantic information from unlabeled video frames and these predicted pseudo labels are helpful in training the fast learner.

To investigate whether pseudo labels predicted by the fast learner itself could help stream learning, we conducted another ablation experiment where we performed PL without EMA (Row 3). Compared with the naive model (Row 1), we observed a performance increase from 28.76% to 31.60% in FAP and 19.80% to 22.44% in CAP. It indicates that, due to the temporal correlation in video stream, pseudo labels predicted by the fast learner can serve as an informative supervision for the training of the fast learner itself.

However, replaying the self-predicted pseudo labels on the fast learner fails to prevent forgetting, as indicated by the drop from -5.48% to -4.83% in F. It is possible that the pseudo labels generated by the fast learner only bias towards the classes which have already been learnt very well and fail to reinforce the fast learner to improve on the poorly-learnt classes. Different from the fast learner, the slow learner integrates semantic information over time. The predicted pseudo labels carry more semantic information, which is useful for fast learner to capture more generic object representations during pseudo label replays. Again, this emphasizes that the reciprocal replay from the slow learner to the fast learner is critical for memory reinstatement, which has been missing in the computational modeling literature of CLS.

6. Conclusion

To imitate what humans see and learn in the real world, we introduce a more realistic and challenging problem of label-efficient online continual object detection (LEOCOD) in video streams. Addressing this problem would greatly benefit real-world applications by reducing model retraining time and data labeling costs. Inspired by the Complementary Learning Systems (CLS) theory, we propose a plug-and-play module, namely Efficient-CLS, that can be easily integrated into and improve existing continual leaning algorithms. We evaluate Efficient-CLS on two challenging real-world video datasets, where our method achieves the state-of-the-art performance in preventing catastrophic forgetting, all while requiring minimal annotation effort.

Acknowledgements This research is funded by the National Research Foundation, Singapore under its NRFF Award NRF-NRFF13-2021-0008 & NRF-NRFF15-2023-0001, AI Singapore Programme (AISG Award No: AISG-GC-2019-001-2A & AISG2-RP-2021-025), Mike Zheng Shou's Start-Up Grant from NUS, Mengmi Zhang's Startup Grant from Agency for Science, Technology, and Research (A*STAR), and Early Career Investigatorship from Center for Frontier AI Research (CFAR), A*STAR. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore.

References

[1] Egoobjects dataset. https://ai.facebook.com/datasets/egoobjects-dataset/, May 2022. 2, 5

[2] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. *Advances in neural information processing systems*, 32, 2019. 1, 3

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019. 1, 2, 3

[4] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. *arXiv preprint arXiv:2201.12604*, 2022. 2, 3, 5

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 3

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 2, 3, 5

[7] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2203.03798*, 2022. 1, 2

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018. 2, 3, 5

[9] Hung-Jen Chen, An-Chieh Cheng, Da-Cheng Juan, Wei Wei, and Min Sun. Mitigating forgetting in online continual learning via instance-aware parameterization. *Advances in Neural Information Processing Systems*, 33:17466–17477, 2020. 2

[10] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 6

[11] Enrico Fini, Stéphane Lathuiliere, Enver Sangineto, Moin Nabi, and Elisa Ricci. Online continual learning under extreme memory constraints. In *European Conference on Computer Vision*, pages 720–735. Springer, 2020. 1

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[13] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32, 2019. 3

[14] Daoyun Ji and Matthew A Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107, 2007. 2

[15] Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017. 3

[16] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 5

[17] Joseph Kj, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Incremental object detection via meta-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 5

[18] Dharshan Kumaran, Demis Hassabis, and James L McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7):512–534, 2016. 1

[19] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017. 1

[20] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021. 3

[21] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 2

[22] Lorenzo Pellegrini, Chenchen Zhu, Fanyi Xiao, Zhicheng Yan, Antonio Carta, Matthias De Lange, Vincenzo Lomonaco, Roshan Sumbaly, Pau Rodriguez, and David Vazquez. 3rd continual learning workshop challenge on egocentric category and instance level object understanding. *arXiv preprint arXiv:2212.06833*, 2022. 5

[23] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34, 2021. 3

[24] Quang Pham, Chenghao Liu, Doyen Sahoo, and HOI Steven. Contextual transformation networks for online continual learning. In *International Conference on Learning Representations*, 2020. 2, 3

[25] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European conference on computer vision*, pages 524–540. Springer, 2020. 2, 3, 5

[26] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 3, 5

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 4, 6

[28] Mohammad Rostami, Soheil Kolouri, and Praveen K Pilly. Complementary learning for overcoming catastrophic forgetting using experience replay. *arXiv preprint arXiv:1903.04566*, 2019. 3

[29] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9630–9638, 2021. 2, 3

[30] Krishna Kumar Singh, Kayvon Fatahalian, and Alexei A Efros. Krishnacam: Using a longitudinal, single-person, egocentric dataset for scene understanding tasks. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 5

[31] James Smith, Jonathan Balloch, Yen-Chang Hsu, and Zsolt Kira. Memory-efficient semi-supervised continual learning: The world is its own replay buffer. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 3

[32] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 3

[33] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 3

[34] Jianren Wang, Xin Wang, Yue Shang-Guan, and Abhinav Gupta. Wanderlust: Online continual object detection in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10829–10838, 2021. 1, 2, 3, 5, 6

[35] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, 2020. 1

[36] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-aware contrastive semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14421–14430, 2022. 3

[37] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 3

[38] Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. In *European conference on computer vision*, pages 566–583. Springer, 2020. 4