Comparative Study of Natural Replay and Experience Replay in Online Object Detection

Baptiste Wagner Denis Pellerin Sylvain Huet
Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab
38000 Grenoble, France
baptiste.wagner@gipsa-lab.grenoble-inp.fr

Abstract

Online Object Detection (OOD) algorithms play a crucial role in dynamic and real-world computer vision applications. In these scenarios, models are trained on a data stream where old class samples are revisited, a phenomenon known as Natural Replay (NR). During training, NR occurs unevenly across object categories, leading to evaluation metrics biased towards the most frequently revisited classes. Existing benchmarks lack proper quantification of NR and depict short-term training scenarios on a single domain. As a result, evaluating generalization capabilities and forgetting rates of models become challenging in OOD. In this paper, we address the challenges surrounding the evaluation of OOD models by proposing two key contributions. Firstly, we define a metric to quantify NR in an OOD scenario and show how NR is related to class specific forgetting. Secondly, we introduce a novel benchmark, EgOAK, which introduces a long-term training scenario that involves frequent domain shifts. It allows the evaluation of models' generalization capabilities and forgetting of knowledge on past domains. Our results in this OOD setting reveal that Experience Replay, a memory-based method, is particularly effective for better generalization to new domains and for preserving past knowledge. Leveraging replay from memory helps to address the low natural replay rate for rarely revisited classes, resulting in improved adaptability and reliability of models in dynamic environments.

1. Introduction

Online object detection (**OOD**) is a critical task in computer vision, particularly in real-time applications such as robotics [11], autonomous driving [24, 25] or recognition for VR/AR headsets [2]. The goal of OOD algorithms is to continuously adapt to new tasks while retaining knowledge from previously learned ones. This phenomenon in which a model gradually loses its past knowledge as it learns new tasks is commonly stated as catastrophic forgetting [7].

In OOD scenarios, Natural Replay (**NR**) occurs as a result of the continuous streaming of data, mimicking the real-world experience of a dynamic agent navigating in changing environments [28]. NR is the process of naturally revisiting instances of old object classes throughout the data stream during training [5, 8, 12]. However, we provide evidence in this paper that NR varies for each class, leading to uneven exposure to previously seen data. Consequently, classes that are rarely replayed in the data stream exhibit more forgetting than frequently revisited classes.

Evaluating the forgetting rate of an algorithm on past knowledge becomes challenging due to the presence of NR, which acts as an important parameter determining the classes that will be forgotten in a given scenario. Existing benchmarks for OOD do not explicitly define or quantify NR [28, 29], and this lack of consideration hinders a precise assessment of forgetting.

Moreover, accurately measuring two key aspects of model performance, namely forgetting and generalization [13], is crucial for ensuring their reliability in the long term [12]. Generalization reflects the model's ability to perform well on already known classes, but on unseen data from new domains [10]. While existing OOD scenarios offer valuable insights for short-term evaluations in a single domain, a deeper investigation of model performance across longer-term horizons and with domain shifts should be conducted.

In this paper, we aim to address these challenges with two contributions and provide a comprehensive evaluation framework for OOD, especially the generalization capabilities and forgetting rates of models in the presence of NR.

First of all, we conduct an extensive experimental study on the EgoObjects dataset [19], which are thoughtfully designed to exhibit varying degrees of NR across different object classes. Our investigations aim to reveal the influence of NR on model evaluation, particularly its effect on the forgetting rate. We introduce a novel metric to quantify NR in OOD scenarios and demonstrate its correlation with forgetting.

Finally, understanding how models behave in extended scenarios and their ability to generalize to other domains holds significant importance for real-world applications. In regard to these considerations, we introduce a new benchmark, EgOAK, to address the challenges in model evaluation and provide a comprehensive evaluation framework for OOD models.

EgOAK has been specifically designed to tackle these inquiries by offering a comprehensive evaluation framework that considers the dynamics of NR and domain changes. The core idea behind EgOAK is the alternating training on tasks from two datasets, EgoObjects [19] and OAK [28]. It ensures that the model is exposed to tasks of different domains in a controlled manner. EgOAK provides valuable insights into the models' adaptability to new tasks from different domains while retaining knowledge from previously learned ones.

In the following sections of this paper, we delve deeper into the topic of OOD evaluation in the presence of NR. After reviewing related works, we investigate class forgetting and its relation to NR, providing insights into the impact of NR on model adaptability and forgetting rates. Building on these findings, we introduce a novel benchmark, EgOAK, designed to address the limitations of current evaluation methodologies and enable comprehensive assessments of online object detection algorithms. Finally, we conclude by summarizing our key findings and discussing their implications for the development of more adaptive and robust OOD models.

2. Related Work

Continual Object Detection Object detection has been extensively studied in the field of computer vision [30, 32]. More recently, the development of object detection models for Continual Object Detection, where models need to adapt to new tasks while retaining knowledge from previous ones, has become an emerging research area [15, 18, 20, 23, 31]. Existing works in Continual Object Detection often evaluate their models on widely used benchmarks such as COCO [14] and Pascal VOC [6]. However, benchmarks based on these datasets are artificially built and introduce the issue of ”background shift” [18], wherein the model is trained on all images from the dataset, but only a portion of category annotations is provided across all categories at each task. In our work, we focus on online object detection, which presents a more natural and dynamic training scenario. Our approach involves a data stream mimicking an agent navigating in a natural dynamic environment.

Datasets for OOD OAK [28] stands as the pioneering benchmark for OOD, offering ego-centric video snippets captured from a student's perspective at Krishna campus. Notably, OAK has been used in the context of semi-supervised OOD [29] as well as the EgoObjects dataset. EgoObjects was introduced during the CLVision CVPR22 workshop [19]. It depicts ego-centric videos taken indoor environments focusing on everyday objects. Despite these efforts, the inherent natural replay present in OOD training scenarios has not been adequately accounted for in these datasets. The absence of natural replay consideration hinders a comprehensive evaluation of model performance in dynamic environments.

Natural replay Unlike OOD scenarios, NR is not integrated into benchmarks for classification tasks in continual learning. However, some studies in classification tasks manually incorporate NR into training scenarios to simulate the real-world experience of encountering previously seen objects in dynamic environments [5, 8]. Additionally, classification models have been compared in the context of long-term training scenarios with natural replay in [12]. In this work, we study NR in the context of OOD. In particular, we propose a metric to quantify NR and show its impact on forgetting rates of models.

3. Natural Replay and its Relation to Specific Class Forgetting

Accurately measuring forgetting in OOD algorithms is a crucial aspect to assess their performance in the long term [12]. However, quantifying forgetting in the presence of NR during training poses significant challenges, making it difficult to evaluate and compare object detectors accurately.

In OOD scenarios [28], NR occurs as a result of the continuous streaming of data designed to mimic the real-world experience of a dynamic agent navigating in changing environments. As the model is exposed to the stream of incoming data, it encounters instances of old classes multiple times over the course of training.

However, the extent of this NR varies, and certain classes may be more frequently revisited than others, leading to uneven exposure to previously seen data. This discrepancy poses a challenge in accurately evaluating a model's performance, particularly when standard evaluation metrics aggregate results across all classes.

In this section, we delve deeper into the concept of NR by proposing a metric to quantify its occurrence. Additionally, we conduct experiments to establish a correlation between NR and class forgetting, aiming to provide a more detailed understanding of their relationship and implications for model evaluation in OOD scenarios.

3.1. Quantifying Natural Replay

In order to quantify NR on OOD scenarios, we propose the introduction of two new metrics, denoted as NRR (Natural Replay Rate) and NRS (Natural Replay Score).

The first metric NRR quantifies the extent to which a class y is replayed throughout the data stream. It is com-

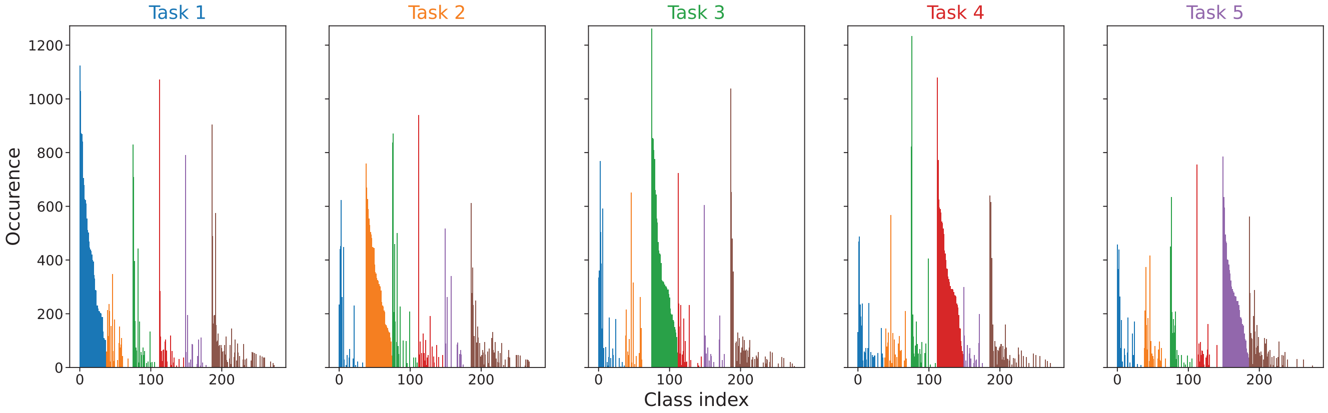

Figure 1: Occurrence of classes in EgoObjects [19]. Tasks are composed of video frames focusing on disjoint object categories. In each frame, all objects, including the focused one and those in the background, are annotated, resulting in Natural Replay ($NRS = 0.51$). Class indexes are sorted by class occurrence in their respective tasks. Classes in brown are additional classes that are not depicted as focused objects in the dataset.

puted as the index of dispersion [27] over the class occurrences in each task. It is computed using the following formula:

$$NRR(y) = \frac{T((\sum_{i=1}^{T} occ_i(y))^2 - \sum_{i=1}^{T} occ_i(y)^2)}{(T-1)(\sum_{i=1}^{T} occ_i(y))^2} \quad (1)$$

with T the number of tasks in the scenario and $occ_i(y)$ the number of occurrences of class y in task i,

If a class y is present in only one task, its associated NRR is 0. In contrast, when the occurrences of a class are dispersed across multiple tasks uniformly, the NRR is 1.

Finally, the second proposed metric NRS (Natural Replay Score) is designed to quantify the level of NR in a given OOD scenario. This score is computed by averaging the Natural Replay Rates (NRR) across all C classes of the dataset:

$$NRS = \frac{1}{C}\sum_{y=0}^{C-1} NRR(y) \quad (2)$$

3.2. Experiments on EgoObjects

To investigate the impact of NR on training and model performance in OOD scenarios, we conducted an experiment on the EgoObjects dataset [19].

The EgoObjects dataset was initially introduced in the CLVision challenge at CVPR 2022 [19] to create a benchmark for evaluating continual object detectors. In this challenge scenario, the EgoObjects videos were divided into five distinct tasks based on the labels of the focused objects within each video. This approach ensured that focused object classes were predominantly represented in their respective tasks. However, the presence of background objects, which are also annotated in each frame, introduces the possibility of NR. A class object might appear as the main focused object in its respective task, while also appearing as a background object in other tasks.

To illustrate the distribution of class occurrences, Figure 1 displays the occurrence of classes across tasks. While classes of focused objects are more prevalent in their designated tasks, NR manifests as peaks in class occurrences in other tasks. In this scenario, the resulting NRS is 0.51 indicating a consistent presence of NR in the data stream.

For each experiment, we trained a Faster-RCNN [22] architecture using Stochastic Gradient Descent with a learning rate of 2, momentum of 0.9 and weight decay of 0.0001. As in [21], we found that using a high learning rate enables the model to quickly adapt to the new data present in the stream. Specifically, during each training iteration on the data stream, only the FPN (Feature Pyramid Network), RPN (Region Proposal Network), and box classifier were fine-tuned on the data stream. We used a frozen Mobile-Net [9] encoder, pre-trained on COCO [14] for feature extraction as it allows fast image processing with online constraints.

The training scenario on EgoObjects follows the scheme suggested by the challenge [19]. In this approach, the model is trained sequentially on the five tasks, following the construction methodology described previously based on the focused object class label. As in the original scenario, images within each task are shuffled to maintain the assumption of independent and identically distributed (iid) data within each task. However, in contrast to the training scheme proposed by the challenge, we train the model for only one epoch on each task to accommodate the online context. This ensures that only a single pass is made over the data stream.

Two strategies were studied: naive and the replay-based

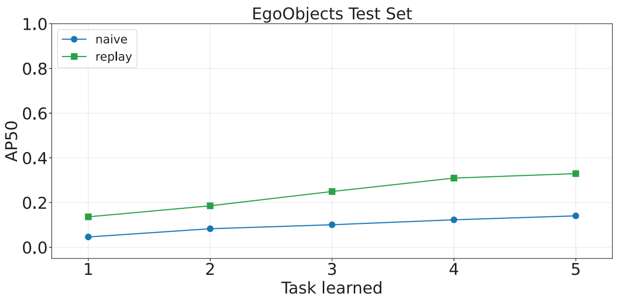

Figure 2: AP50 performance (Average Precision with an IOU of 0.5) on the EgoObjects dataset. The Faster-RCNN model was trained using two strategies: the naive approach and the replay-based method ER [4].

method ER [4]. In the naive approach, we trained the model without accounting for the potential effects of catastrophic forgetting. This strategy generally leads to high rates of forgetting.

ER [4] is a common strategy in online continual learning for classification tasks to mitigate catastrophic forgetting [1, 3, 4, 16, 17]. In this method, an external memory buffer stores samples from previous tasks. When receiving a new batch from the data stream, a batch of the same size is randomly sampled from memory. The model is then trained on the combined batch, which includes data from the current and previous tasks. This allows the model to reinforce knowledge from previous tasks while learning the new task. For our experiments, we implement the memory buffer using a Reservoir Buffer [26] strategy with a size of 1000 images.

3.3. Results

Figure 2 illustrates the performance evolution of the naive approach and the ER method in the presence of NR. Both strategies exhibit an increasing trend, suggesting overall progress in model performance. However, a closer examination reveals that certain classes are forgotten over time due to the uneven exposure to NR.

In Figure 3, we give the AP50 performance (Average Precision with an IOU of 0.5) of four classes with different NRR on the experiment on EgoObjects. For the Naive method, the AP50 evolution follows the NR of each class. This indicates that the forgetting potential of classes is related to NR. This is expected as the model, trained in a naive way, is subject to catastrophic forgetting: it forgets knowledge about past classes when trained on new ones. However, performance on classes with a high NRR, like *plate* tends to generalize better across time.

The replay-based method ER demonstrated the capability to mitigate forgetting even for classes with limited NR. These findings suggest that incorporating replay-based strategies in OOD algorithms can significantly contribute to knowledge retention across various classes, enhancing the model's adaptability and robustness over time.

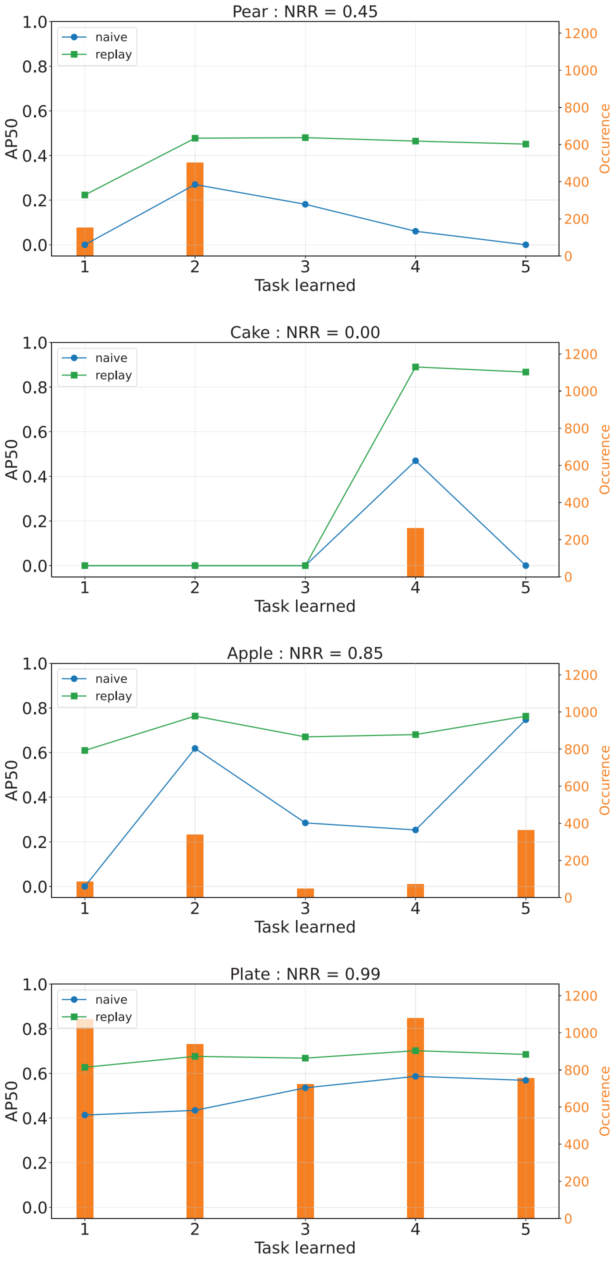

Figure 3: AP50 performance (Average Precision with an IOU of 0.5) and occurrence per task of four classes (*pear*, *cake*, *apple*, and *plate*) with different Natural Replay Rates (NRR) on the EgoObjects dataset. The Faster-RCNN model was trained using two strategies: the naive approach and the replay-based method ER [4].

3.4. Discussion

Our investigation into quantifying NR [5, 8, 12] highlights its significant impact on model evaluation in OOD scenarios [19,28]. We emphasize the need for caution when assessing models on scenarios exhibiting varying levels of NR across different classes, as this can lead to conclusions.

The issue with Natural Replay (NR) lies in how it affects forgetting of past classes. On average across all classes, the performance may seem to steadily improve, giving the impression of overall progress. Both the naive approach and the ER method demonstrate this trend of increasing performance as depicted in Figure 2. However, The uneven occurrence of NR among classes can result in the forgetting of certain specific classes. This forgetting effect can be obscured when only considering the overall performance, masking the fact that some classes are being forgotten over time.

As the model continually encounters new tasks and domains, the risk of catastrophic forgetting increases, potentially leading to a significant loss of knowledge for specific classes. Therefore, understanding how models behave in the long term, especially in dynamic environments with emerging classes and domains, is crucial for ensuring their reliability and adaptability.

In these considerations, our findings reveal that objects experiencing limited NR, *i.e.* a low NRR (Eq. 1), require an external memory to mitigate forgetting. Memory-based approaches, like the ER method, emerge as promising solutions to address these long-term challenges and improve model robustness in OOD scenarios.

To address these questions, we present a new benchmark EgOAK in the following sections of this paper. This benchmark enables comprehensive evaluations for OOD within a long-term scenario where domain shifts occur.

4. EgOAK: An Evaluation Benchmark for OOD with Domain Changes

In the subsequent sections of the paper, we introduce EgOAK, a novel benchmark designed to facilitate a more robust evaluation of online object detection algorithms in the presence of NR.

The training scenario of EgOAK is created by alternating tasks between the two datasets, EgoObjects [19] and OAK [28]. This approach enables a more reliable assessment of model adaptability in dynamic environments. At each task transition, data from a new domain becomes available and new categories emerge in both the indoor environments of EgoObjects and the outdoor environments of OAK.

Through this new benchmark EgOAK, we aim to contribute to the advancement and enhancement of OOD algorithms by enabling the development of more adaptive and robust models tailored for long-term online training in real-world scenarios.

4.1. Scenario Task Composition

The proposed training scenario involves alternating between the two datasets, EgoObjects and OAK, which enables a more precise evaluation of forgetting and model generalization.

It consists of T tasks, alternated between EgoObjects and OAK datasets. Each dataset is split into $T/2$ tasks as follows:

EgoObjects Dataset [19]: We use the decomposition proposed by the challenge, utilizing the focused object from each video to sort and separate the videos into $T/2$ tasks. To ensure an equal number of images between OAK and EgoObjects, only one frame out of every two is used from the EgoObjects dataset. For constructing the test set, we follow the same strategy as OAK [28] , selecting one frame out of every 16 from the original videos.

OAK Dataset [28]: We adopt the scenario proposed by [28] and concatenate the videos of OAK. To create $T/2$ distinct tasks, we divide the OAK training stream into $T/2$ segments. The provided test set is used. It is constructed by taking one frame out of every 16 from the original videos, while the remaining 15 frames are associated with the training set.

In this study, we use a total of T = 6 tasks, with 3 tasks assigned to each dataset. Table 1 provides a summary of the number of images for each task in this particular configuration.

Train Tasks	$T1_{ego}$	$T2_{oak}$	$T3_{ego}$	$T4_{oak}$	$T5_{ego}$	$T6_{oak}$
# Images	11364	10646	12157	10646	11349	10646
Test Sets	$Test_{ego}$			$Test_{oak}$		
# Images	2325			1996		
Total	$Total_{ego}$			$Total_{oak}$		
# Images	37195			33934		

Table 1: Scenario Composition and Image Count for Each Task in the EgOAK Benchmark for $T = 6$. The table presents the number of images in each task from datasets EgoObjects (EGO) and OAK. Additionally, the total image count for each dataset and the number of images in the test sets are included.

When considering the original scenarios on EgoObjects [19] and OAK [28] datasets separately, they exhibit NRS scores of 0.51 and 0.92 respectively. In comparison, our proposed scenario achieves an NRS score of 0.42. This indicates that our scenario significantly reduces NR compared to using only one dataset. Moreover, employing both datasets enables a more accurate measurement of models' generalization capabilities, as it involves testing across two distinct domains. Consequently, it enables a more accurate

	Continual Average Precision **(CAP)**				Final Average Precision **(FAP)**			
	$\mathcal{C}$	$\mathcal{C}_{com}$	$\mathcal{C}_{\text{ego–only}}$	$\mathcal{C}_{\text{oak–only}}$	$\mathcal{C}$	$\mathcal{C}_{com}$	$\mathcal{C}_{\text{ego–only}}$	$\mathcal{C}_{\text{oak–only}}$
Naive	12.6	21.1	11.5	11.4	18.8	28.4	13.6	20.1
ER	25.5	33.6	32.2	16.8	36.7	43.0	46.2	24.4

Table 2: Evaluation Results on the EgOAK benchmark on different class sets: all classes in both datasets $\mathcal{C}$, classes in common between EgoObjects and OAK $\mathcal{C}_{com}$, classes that exclusively belong to the EgoObjects dataset $\mathcal{C}_{ego\text{–}only}$, classes that exclusively belong to the OAK dataset $\mathcal{C}_{oak\text{–}only}$.

evaluation of online object detectors, which we describe in the following section.

4.2. Class-Set Specific Evaluation

In our proposed scenario, we ensure that the model is exposed to distinct visual characteristics and new object categories in a controlled manner with dataset task transitions during training. This controlled exposure allows us to assess the model's adaptability to dynamic environments more accurately.

During evaluation, the model is tested on the constructed test sets from each dataset. This evaluation setup enables us to measure the extent to which the model forgets its knowledge when transitioning between datasets or if it can generalize its learning to new data effectively.

Given the two sets of classes from both datasets $\mathcal{C}_{ego}$ and $\mathcal{C}_{oak}$, we propose a more fine-grained evaluation on three different class sets:

- $\mathcal{C}_{com} = \mathcal{C}_{ego} \bigcap \mathcal{C}_{oak}, |\mathcal{C}_{com}| = 29$, classes in common between EgoObjects and OAK.
- $\mathcal{C}_{ego\text{–}only} = \mathcal{C}_{ego} \setminus \mathcal{C}_{com}, |\mathcal{C}_{ego\text{–}only}| = 248$, classes that exclusively belong to the EgoObjects dataset.
- $\mathcal{C}_{oak\text{–}only} = \mathcal{C}_{oak} \setminus \mathcal{C}_{com}, |\mathcal{C}_{oak\text{–}only}| = 56$, classes that exclusively belong to the OAK dataset.

Each class set serves a distinct purpose in the evaluation process. Firstly, the set of common classes enables the assessment of the model's generalization capabilities, as these classes are present in both datasets, spanning two different domains. Secondly, each exclusive dataset classes set allows us to measure forgetting. For instance, when training the model on the second task using the OAK dataset, we can evaluate the model's performance on the class set exclusive to EgoObjects. This evaluation helps to determine if the model has forgotten knowledge of class-specific objects in EgoObjects while learning other classes in the OAK dataset.

4.3. Evaluation metrics

We employ several evaluation metrics [28] to assess the performance of online object detection models in the presence of NR. These metrics enable a comprehensive analysis of model adaptability and forgetting rates across different class sets.

Continual Average Precision (CAP): CAP measures the continual learning performance of the model throughout the training process. It is computed as the average of the Average Precision (AP) values across all evaluation steps in time for each class set. For class set $\mathcal{C}$, CAP can be expressed as follows:

$$CAP(\mathcal{C}) = \frac{1}{T} \sum_{t=1}^{T} AP_t(\mathcal{C}) \tag{3}$$

where T is the total number of tasks, and $AP_t(\mathcal{C})$ represents the AP for class set $\mathcal{C}$ at time step t.

Final Average Precision (FAP): FAP measures the model's overall performance at the end of training. It is computed as the AP value for each class set at the last time step (T). The FAP for class set $\mathcal{C}$ can be represented as:

$$FAP(\mathcal{C}) = AP_T(\mathcal{C}) \tag{4}$$

4.4. Results

In all our experiments, we used the same training setup as in our previous experiments in section 3.2. Specifically, we compared the Naive and ER strategies on EgOAK. Table 2 shows the CAP and FAP for each training strategy and class set: all classes $\mathcal{C}$, $\mathcal{C}_{com}$, $\mathcal{C}_{ego-only}$ and $\mathcal{C}_{oak-only}$.

For the Naive strategy, the CAP values are significantly lower compared to the ER strategy for all class sets. The CAP values for all classes is 12.6, indicating that the Naive strategy performs poorly in terms of overall AP. This suggests that without considering the potential effects of catastrophic forgetting, the model's performance on both common and specific classes for both EgoObjects and OAK domains is limited.

On the other hand, the ER strategy exhibits much higher CAP values, with 25.5 for all classes. By mitigating catastrophic forgetting through the use of a memory buffer, the model's performance is significantly enhanced, particularly in terms of classifying common and specific classes for both EgoObjects and OAK domains.

The FAP values also demonstrate the superiority of the ER strategy. The Naive strategy shows lower FAP values

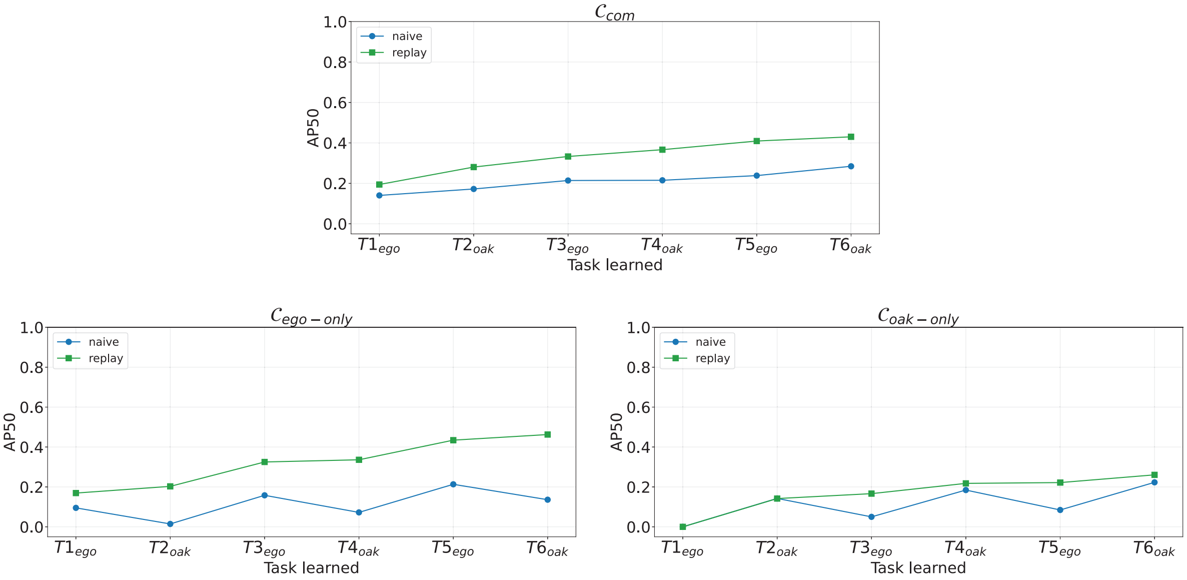

Figure 4: Performance Evolution of the Naive strategy and the replay-based method ER [4] on EgOAK. Each graph shows the evolution of a subset of classes from both datasets: classes in common between EgoObjects and OAK ($\mathcal{C}_{com}$), classes that exclusively belong to the EgoObjects dataset ($\mathcal{C}_{ego\text{–}only}$), and classes that exclusively belong to the OAK dataset ($\mathcal{C}_{oak\text{–}only}$).

for all classes (18.8), while the ER strategy yields higher FAP values (36.7). This indicates that the ER strategy not only performs better overall but also provides improved AP in detecting focused classes, which are crucial for online object detection scenarios.

In Figure 4, we present a qualitative analysis of the model's performance on three class sets as the training progresses, comparing the Naive and ER strategies. The results clearly demonstrate that the ER strategy consistently outperforms the Naive strategy across all class sets.

Regarding the evaluation of generalization capabilities, both methods, Naive and ER, gradually increase their performance on the common class set $\mathcal{C}_{com}$ as the training progresses. However, the ER strategy exhibits better generalization, consistently achieving higher AP compared to the Naive strategy. This suggests that the ER strategy allows the model to generalize more effectively on new domains.

Next, we examine the performance on the dataset-specific class sets, $\mathcal{C}_{ego-only}$ and $\mathcal{C}_{oak-only}$. During training on a task from a specific dataset, the Naive method shows a significant drop in average precision for specific classes that belong to the other dataset. This indicates that the Naive method suffers from catastrophic forgetting at each task transition. In contrast, the ER models exhibit a more stable performance and are capable of limiting the AP drop on one dataset when trained on the other. This highlights the effectiveness of the ER strategy in mitigating catastrophic forgetting and retaining knowledge across both datasets.

Overall, the qualitative analysis supports the quantitative results, showing that the ER strategy consistently outperforms the Naive strategy in terms of generalization and forgetting capabilities. The ER strategy demonstrates a more robust performance, effectively adapting to new tasks and minimizing knowledge loss during task transitions. These findings underscore the importance of memory-based strategies, like ER, for developing adaptive and reliable online object detection models capable of handling dynamic long-term scenarios.

4.5. Discussion

The EgOAK benchmark is designed to provide a more robust and comprehensive evaluation of online object detection models. It aims to address the limitations of existing benchmarks by introducing a controlled training scenario that reflects the challenges faced in long-term scenarios with domain shifts. By alternating tasks between datasets EgoObjects [19] and OAK [28], EgOAK ensures exposure to distinct visual characteristics and object categories while minimizing NR.

Our experimental investigations revealed the limitations of the Naive approach when confronted with scenarios involving NR and domain shifts. Specifically, the Naive model displayed significant drops in average precision on domain shift, indicating its susceptibility to catastrophic forgetting and its limited adaptability in dynamically changing environments.

In contrast, replay-based methods, particularly ER [4], emerged as a promising solution to address these challenges. ER consistently outperformed the Naive approach, showcasing its generalization capabilities and its effectiveness in mitigating catastrophic forgetting.

These findings emphasize the crucial role played by replay-based methods in the context of NR and domain shifts. Implementing strategies like ER offers a robust and adaptive approach for online object detection scenarios, ensuring sustained model performance and enhancing the long-term adaptability of models.

5. Conclusion

In this paper, we addressed the challenges surrounding the evaluation of Online Object Detection (OOD) algorithms in the presence of Natural Replay (NR).

The lack of proper quantification of NR in existing scenarios makes it difficult to accurately assess model performance, particularly concerning their forgetting rate. As different object classes experience varying levels of NR, the evaluation of model performance becomes biased to more frequently replayed classes.

Furthermore, current benchmarks with NR primarily focus on short-term scenarios with only one domain. To overcome these limitations, we introduced the EgOAK benchmark, which enables a more comprehensive evaluation for OOD on the generalization capabilities and forgetting rates of models when trained in dynamic and changing environments.

Memory-based methods emerge as crucial components for long-term OOD in the presence of NR. By storing and replaying less frequently encountered class samples, these methods effectively counteract the uneven class exposure to NR, enhancing model performance. Moreover, memory-based approaches contribute to improved long-term performance by boosting generalization capabilities and mitigating forgetting when learning from new domains.

The utilization of memory-based methods proves to be a promising strategy to address the challenges posed by NR in OOD, making models more adaptive and reliable over extended periods of training.

References

[1] 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.

[2] Dan Bohus, Sean Andrist, Ashley Feniello, Nick Saw, and Eric Horvitz. Continual learning about objects in the wild: An interactive approach. In *Proceedings of the 2022 International Conference on Multimodal Interaction*, pages 476–486, 2022.

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

[4] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

[5] Andrea Cossu, Gabriele Graffieti, Lorenzo Pellegrini, Davide Maltoni, Davide Bacciu, Antonio Carta, and Vincenzo Lomonaco. Is class-incremental enough for continual learning? *Frontiers in Artificial Intelligence*, 5:829842, 2022.

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

[7] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

[8] Hamed Hemati, Andrea Cossu, Antonio Carta, Julio Hurtado, Lorenzo Pellegrini, Davide Bacciu, Vincenzo Lomonaco, and Damian Borth. Class-incremental learning with repetition. *arXiv preprint arXiv:2301.11396*, 2023.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 1(8), 2017.

[11] Timothée Lesort. *Apprentissage continu : S'attaquer à l'oubli foudroyant des réseaux de neurones profonds grâce aux méthodes à rejeu de données*. Thèse, Institut Polytechnique de Paris, June 2020.

[12] Timothée Lesort, Oleksiy Ostapenko, Diganta Misra, Md Rifat Arefin, Pau Rodríguez, Laurent Charlin, and Irina Rish. Scaling the number of tasks in continual learning. *arXiv preprint arXiv:2207.04543*, 2022.

[13] Sen Lin, Peizhong Ju, Yingbin Liang, and Ness Shroff. Theory on forgetting and generalization of continual learning. *arXiv preprint arXiv:2302.05836*, 2023.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[15] Xialei Liu, Hao Yang, Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Multi-task incremental learning for object detection. *arXiv preprint arXiv:2002.05347*, 2020.

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

[17] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

[18] Angelo G Menezes, Gustavo de Moura, Cézanne Alves, and André CPLF de Carvalho. Continual object detection: A review of definitions, strategies, and challenges. *Neural Networks*, 2023.

[19] 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.

[20] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern recognition letters*, 140:109–115, 2020.

[21] 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.

[22] 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.

[23] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017.

[24] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21371–21382, 2022.

[25] Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023.

[26] Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

[27] Jeffery T Walker. *Statistics in criminal justice: Analysis and interpretation*. Jones & Bartlett Learning, 1999.

[28] 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.

[29] Jay Zhangjie Wu, David Junhao Zhang, Wynne Hsu, Mengmi Zhang, and Mike Zheng Shou. Label-efficient online continual object detection in streaming video. *arXiv preprint arXiv:2206.00309*, 2022.

[30] Syed Sahil Abbas Zaidi, Mohammad Samar Ansari, Asra Aslam, Nadia Kanwal, Mamoona Asghar, and Brian Lee. A survey of modern deep learning based object detection models. *Digital Signal Processing*, 126:103514, 2022.

[31] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020.

[32] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023.