Relaxing the Forget Constraints in Open World Recognition

Relaxing the Forget Constraints in Open World Recognition

Dario Fontanel, Fabio Cermelli, Antonino Geraci,
Mauro Musarra, Matteo Tarantino, and Barbara Caputo

Politecnico di Torino, Turin, Italy
{dario.fontanel, fabio.cermelli, barbara.caputo}@polito.it,
{antonino.geraci, mauro.musarra, matteo.tarantino98}@studenti.polito.it

Abstract. In the last few years deep neural networks has significantly improved the state-of-the-art of robotic vision. However, they are mainly trained to recognize only the categories provided in the training set (closed world assumption), being ill equipped to operate in the real world, where new unknown objects may appear over time. In this work, we investigate the open world recognition (OWR) problem that presents two challenges: (i) learn new concepts over time (incremental learning) and (ii) discern between known and unknown categories (open set recognition). Current state-of-the-art OWR methods address incremental learning by employing a knowledge distillation loss. It forces the model to keep the same predictions across training steps, in order to maintain the acquired knowledge. This behaviour may induce the model in mimicking uncertain predictions, preventing it from reaching an optimal representation on the new classes. To overcome this limitation, we propose the Poly loss that penalizes less the changes in the predictions for uncertain samples, while forcing the same output on confident ones. Moreover, we introduce a forget constraint relaxation strategy that allows the model to obtain a better representation of new classes by randomly zeroing the contribution of some old classes from the distillation loss. Finally, while current methods rely on metric learning to detect unknown samples, we propose a new rejection strategy that sidesteps it and directly uses the model classifier to estimate if a sample is known or not. Experiments on three datasets demonstrate that our method outperforms the state of the art.

Keywords: Open world recognition · Robot vision · Deep learning.

1 Introduction

Over the last few years, the emergence of deep neural networks has brought significant improvements in the robotic vision, being used in multiple tasks such as grasping [9], tool selection [37], depth prediction [29], and autonomous driving [21]. However, modern deep architectures are still trained under the *closed world assumption* (CWA) which assumes that every category the model will need to recognize is fixed and known a priori during the training phase. Clearly, this is a significant limitation since the real-world is continuously changing and the model will likely encounters new classes while operating in new environments. Recognizing the necessity of breaking the CWA, [2] proposed the the *open world recognition* (OWR) problem. It consists of two sub-challenges: (i) incremental learning [36,4,3,43], which requires models to extend their knowledge over time without forgetting already learned concepts (*i.e.* incurring into catastrophic forgetting [30]) and (ii) open set recognition [38,14], which requires models to distinguish already seen concepts from unknown ones.

Standard OWR approaches [2,8,28,13] addressed the two challenges separately. To deal with catastrophic forgetting, the state-of-the-art methods [28,13] employ a knowledge distillation loss [18] that prevents changes of the classification outputs for old classes. The model is forced to maintain consistent prediction also when it is not confident, resulting in an overly-regularized training and preventing the model to correctly adapt the feature space when learning novel classes. To overcome this limitation, in this paper we propose a novel distillation loss, *i.e.* the *Poly loss*. It has been designed following two criteria: (i) maintaining the output unchanged when the model is certain, and (ii) letting the model be free to change when the prediction is uncertain. This formulation allows to effectively updates the network to represent novel classes, while also preventing forgetting of the old knowledge. Despite the advantages of the Poly loss, preserving the model unchanged may prevent it from achieving optimal representation on new classes. To this end, we propose the *forget constraint relaxation* strategy. It relaxes the constrain imposed by the distillation loss by randomly removing the contribution from the loss computation of some old classes at each iteration. To address the second challenge of OWR, *i.e.* distinguish between known and unknown samples, the standard approach is to rely on metric learning. In particular, state-of-the-art methods [2,28,13] couple the nearest class mean (NCM) classification strategy [8] with a rejection threshold to categorize a sample into the set of known categories or predict it as unknown. Despite its effectiveness, this approach has two drawbacks: (i) it considers all the features as equally important and (ii) it suffers the curse of dimensionality. In this work, we abandon the metric learning approach in favor of a rejection strategy based on a linear classifier that computes a score for each class as the dot product between the feature representations and a set of class specific learnable weights, implicitly weighting each feature by its importance. As [13], we learn class-specific thresholds on an held-out set.

Following previous works, we benchmark our contributions on Core50 [27], RGB-D Object Dataset [39] and CIFAR-100 [20] datasets, demonstrating the benefits of our new components and outperforming the state of the art.

Contributions. To summarize, in this paper we tackle the challenges of OWR scenario. In particular, we introduce the Poly loss, a novel distillation loss that allows changes in network output when the old model is not confident about the prediction. We propose a forget-constraint relaxation strategy that allows the network to reach an optimal representation of novel classes and a new rejection strategy, abandoning the metric learning approach in favor of a linear classifier. We benchmark our approach on three datasets, showing that it outperforms the previous state of the art.

2 Related work

Open world recognition. The necessity of breaking the CWA for robot vision systems [41] has prompted numerous research efforts aiming at equipping models with the capability of both automatically detecting unknown concepts and incorporating them during subsequent learning phases. To that purpose, [2] introduced open world recognition (OWR) as a realistic benchmark for developing agents able to act in the real world. [2] empowers the Nearest Class Mean (NCM) classifier [31,16] with the ability of detecting unknowns, proposing the Nearest Non-Outlier algorithm (NNO). NNO uses a fixed rejection threshold to categorize a test sample as belonging to a known or unknown class. To tackle the OWR scenario, [8] develops the Nearest Ball Classifier which exploits the confidence of the prediction to compute the rejection threshold. [28] extends the NNO method of [2] by incorporating a dynamic updating strategy for the rejection threshold and using a deep neural network as feature extractor. Recently, [13] improves the performances of NCM based classifier introducing two clustering losses and proposing to explicitly learn a specific threshold for each category. Recently, [12] proposed an OWR benchmark considering different visual conditions, showing that current methods struggle in discriminating between unknown and known samples belonging to a different visual domains. In this paper, we go beyond the NCM-based metric learning approach, proposing a simpler but effective rejection strategy that takes advantage of the network outputs confidence.

Knowledge Distillation-based Incremental learning. Knowledge distillation has been first proposed by [18] as a technique to transfer knowledge from a teacher (cumbersome) model to a student (simple) one. The idea has been then adapted by [24] in incremental learning to alleviate catastrophic forgetting [30]. They considered as teacher the model frozen after the previous learning step and as student the model trained on the new incoming data, and they forced the student to keep its predictions consistent with the teacher. In the following, multiple works proposed different variations this idea in the context of classification [36,11,1,44,5,26,45,42,25,19,32] and only recently in semantic segmentation [6,33,10] and object detection [40,34,35]. Please refer to [7,23] for an extensive survey of incremental learning methods.

While previous works in the OWR setting [28,13] adopted the knowledge distillation strategy presented in [36], we develop a new distillation loss that considers the model uncertainty to prevent forgetting while learning new classes.

3 Method

3.1 Problem formulation

The open world recognition (OWR) setting is composed of multiple training steps. In the first step, the system is provided with an initial training set $\mathcal{T}_0$ composed by N_0 samples, *i.e.* $\mathcal{T}_0 = (x_i, y_i)_{i=1^{N_0}}$, where x_i indicate an image and $y_i \in \mathcal{Y}_0$ is the relative class label. In any following step T, the system is provided a new training set $\mathcal{T}_T$, containing samples belonging to a set of novel classes $\mathcal{Y}_T$, where $\mathcal{Y}_T \bigcap \mathcal{Y}_t = \emptyset \ \forall t \in [0, T-1]$. The goal of OWR is to find a model f that maps an image x to the respective class, if it is known at the step T, or to the unknown class u, *i.e.* $f : X \to \mathcal{K}_t \cup \text{u}$, with $\mathcal{K}^T = \bigcup_{t=0}^{T} \mathcal{Y}_t$ indicates the set of known classes at step T. The model f must be incrementally updated at every training step T to predict new classes but it must still be able to detect unknown concepts. Without loss of generality, we consider a model f made of two components: a feature extractor ω mapping images into a feature space $\mathcal{Z}$ ($\omega : \mathcal{X} \to \mathcal{Z}$), and a scoring function ϕ mapping features in $\mathcal{Z}$ to class probabilities ($\phi : \mathcal{Z} \to [0,1]^{|\mathcal{K}^T|}$). We note that, as in [28,13], we consider binary class probabilities obtained from a sigmoid function.

OWR then presents two challenges: (i) learning new classes without forgetting the old ones and (ii) recognizing whether new data falls into previously learned categories or not [2]. In the next section we focus on the former challenge, while in section 3.4 we discuss the latter.

3.2 Learning Without Forgetting

Preliminaries. While learning novel categories without accessing the old data, the model is prone to catastrophic forgetting [15,30], *i.e.* it gradually forgets the classes it has learned in previous step. To alleviate the catastrophic forgetting issue, previous works [36,13,28] regularize the model using knowledge distillation [18] which forces the current model $\mathcal{M}_T$ to behave like the model of the previous training step $\mathcal{M}_{T-1}$. Practically, this is accomplished by interpreting the outputs of the previous model $\mathcal{M}_{T-1}$ as pseudo-targets within a loss function, so that a sample x can be identified by the the current model $\mathcal{M}_T$ as belonging to previously observed classes with a certain probability.

During the training step T, the model is then trained using sum of two different terms, *i.e.* the *classification loss* and the *distillation loss*. State-of-the-art methods [28,13,36] employ the binary cross-entropy (BCE) loss for both terms. Formally, the loss is defined as:

$$L = -\frac{1}{|\mathcal{T}_T|} \sum_{(x_i, y_i \in \mathcal{T}_T)} L_C(x_i, y_i) + L_{D_{BCE}}(x_i), \tag{1}$$

with

$$L_C(x_i, y_i) = \sum_{c \in \mathcal{Y}^T} \delta_{c=y_i} \log(\phi(x_i)) + \delta_{c \neq y_i} \log(1 - \phi(x_i),$$

$$L_{D_{BCE}}(x_i) = \sum_{c \in \mathcal{K}^{T-1}} q_i^c \log(\phi(x_i)) + (1 - q_i^c) \log(1 - \phi(x_i)),$$

where x_i is a sample drawn from the training set $\mathcal{T}_T$, y_i is its ground truth label, $\phi(x_i)$ is the model prediction, q_i^c is the old model probability for class c, with *i.e.* $q_i = \phi^{T-1}(x_i)$, $\mathcal{K}^{T-1}$ is the set of old classes, and $\mathcal{Y}^T$ the set of new ones.

The distillation loss $L_{D_{BCE}}$ prevents any changes in the model's output, forcing the probability to be equal to the one obtained by the previous model. Indeed, this loss is highly beneficial when the outputs of the previous network are close to a value of maximum certainty (either 0 or 1), preventing any change that may cause the novel network to lose its ability to predict the old classes. However, when the old network is uncertain (outputs values around 0.5), as often occurs when seeing novel classes samples, the network is forced to maintain its uncertainty, preventing to reach an optimal configuration for new classes.

Poly loss. We therefore aim at finding a distillation loss that prevents the network from modifying the outputs with the maximum certainty, favoring instead the modification of those closer to 0.5, *i.e.* the ones with the highest uncertainty. To formulate a new classification and distillation combination, two major criteria must be satisfied:

1. The total loss L must be globally continuous, differentiable and strictly convex.
2. The distillation loss must have its only minimum in $\phi_T^c = \phi_{T-1}^c$ which means that the first derivative of L must be equal to 0 only when for a certain class c the outputs of both the current and the previous models are equal, i.e. $\phi_T^c = \phi_{T-1}^c$.

To satisfy these criteria, we propose to exploit a *polynomial function* as it is the simplest and most versatile function that can approximate as nearly as needed every continuous function defined on a closed interval.

To prevent the network from changing the outputs with the highest certainty, encouraging instead the modification of those closer to maximum uncertainty status, we formulate Poly loss as follows:

$$\begin{aligned} L_{D_{POLY}}(x_i) = \sum_{y=1}^{s-1} \frac{1}{4}\Big(&(2\phi_T^i(x_i) - 1)^4 - \\ &4(2\phi_{T-1}^i(x_i) - 1)^3(2\phi_T^i(x_i) - 1) + 3\Big), \end{aligned} \tag{2}$$

where ϕ_{T-1}^i indicates the outputs of the previous model $\mathcal{M}_{T-1}$ interpreted as pseudo-target of class c and ϕ_T^i indicates the outputs of the current model $\mathcal{M}_T$.

Overall, for training the network, we replace $L_{D_{BCE}}$ with the Poly loss $L_{D_{POLY}}$, obtaining the following cost function:

$$L = -\frac{1}{|\mathcal{T}_T|} \sum_{(x_i, y_i \in \mathcal{T}_T)} L_C(x_i, y_i) + L_{D_{POLY}}(x_i). \tag{3}$$

3.3 Forget-Constraint Relaxation

Despite the advantages of the Poly loss, the constraints it imposes may still be too binding. If, on the one hand, forcing the network at step T to not change too much in comparison to the network at step $T-1$ helps to prevent forgetting, on the other hand, this behavior may prevent the network at step T from reaching the *best* possible configuration, which may be very different from what it was at step $T-1$. Indeed, a further improvement would be to relax such constraints, increasing the degrees of freedom of the network. The goal this time is to allow updates in the model configuration not only when the targets are close to 0.5,

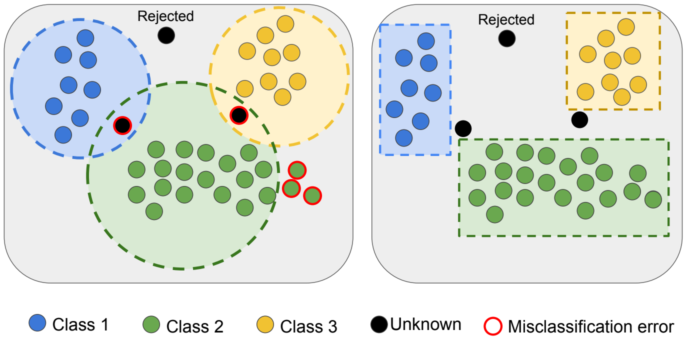


Fig. 1: The figure illustrates the metric learning-based (left) and our rejection (right) strategies. Metric learning weights each feature equally, resulting in a sub-optimal rejection strategy. Differently, our strategy considers each feature independently, modeling better the classes distributions.

as obtained using the Poly loss, but also when they are close to 0 or 1. For this reason, we propose the *forget-constraint relaxation (FCR)* strategy that randomly removes some of the old classes from the loss computation, by simply setting their contribution in the distillation loss L_D to 0. More formally,

$$\widetilde{L}_D = \sum_{c \in C^{t-1}} \frac{R^c}{p} L_D(x_i), \tag{4}$$

where

$$R^c = \begin{cases} 1 & \text{with } \textit{probability} \quad p, \\ 0 & \text{with } \textit{probability} \quad 1-p; \end{cases} \tag{5}$$

and L_D can be any distillation loss (*e.g.*, $L_{D_{POLY}}$).

When an old class c is removed from the loss computation, *i.e.* $R^c = 0$, the output of the new model ϕ_T^i can take any value for c to minimize the total loss L, allowing it to properly learn the novel class. However, we are not letting the model to forget that class: in the next iterations, it is likely that the class c is again considered in the equation, *i.e.* $R^c = 1$, and the distillation loss will prevent the catastrophic forgetting phenomenon. We remark that this could not be obtained using simple strategies such as multiplying the distillation loss by a positive weight below 1 since, while the network would learn new classes more easily, it would also quickly forget the old ones.

3.4 Rejection strategy

In this section we will analyze the second challenge of OWR, namely the models capability of categorizing as unknown data that does not belong to the set of previously learned classes. The standard approach for detecting unknown samples is to employ a metric learning approach [8,2,28,13]. In particular, previous works assume that the feature extractor ω projects samples in an embedding space where samples of the same class are closer than samples of any other classes.

Following this assumption, they compute for each class a centroid, *i.e.* the mean of the feature representation of the samples of that class, and they consider a sample as unknown if its representation is more distant than a threshold η from all the class centroids.

Despite the improvements introduced in B-DOC [13], *i.e.* using class-specific thresholds and learning them rather than computing them, we still identify two important limitations with the metric learning approach:

1. All the features are treated as equally informative to compute per-class thresholds;
2. Computing feature distances on a large scale suffers the curse of dimensionality.

As illustrated in Fig.1, we argue that considering all the feature as equally important (issue 1) is sub-optimal, since not all the features are meaningful to identify a certain class. Consider, for example, a model having a feature identifying whether or not a *wheel* is present. This feature would be hugely important to classify the *car* class, but it is totally meaningless to classify the *dog* class. Thus, we need to properly consider each feature, weighting its contribution depending on how important it is for a certain class.

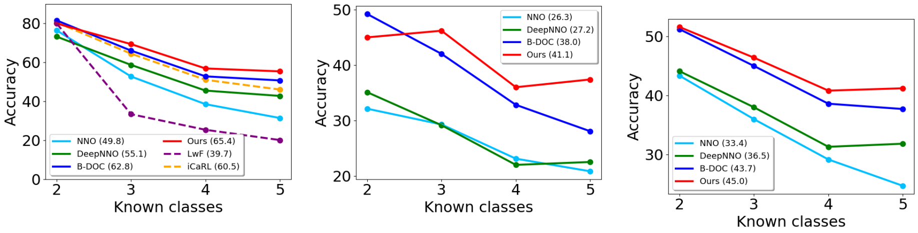
To deal with both issues, we propose an approach that completely abandons the metric learning approach. To take into account the different importance of each dimension of the feature vector, we propose to directly use the network classifier weights, that implicitly provide the features importance for each class. The classifier computes the dot product between the sample feature representation and the weights of a certain class, producing a scalar value for each class, *i.e.* the classification score, addressing also the issue 2. Intuitively, the classification score is a value indicating the confidence for a sample x to belong to class c. Thus, as in B-DOC [13], we define a threshold for each class c, η_c, and given an image x_i, we implement the following rejection policy:

$$\begin{cases} \text{accept}, & \text{if } \exists c \in \mathcal{C} : (< \omega(x_i), \mathbf{w}_c >) > \eta_c; \\ \text{reject}, & \text{otherwise}. \end{cases} \tag{6}$$

Chiefly, following B-DOC [13], our training strategy consists of two steps: in the first one, we train the feature extractor on the training set while minimizing eq. 3, and in the second one, we learn the thresholds η on a set of samples that we excluded from the training set. Keeping frozen all the network parameters, we learn η_c minimizing the following cost function:

$$\mathcal{L}_{GR}(x, c) = \sum_{c \in \mathcal{C}} \max(0, k \cdot (\eta_c - (< \omega(x), \mathbf{w}_c >))), \tag{7}$$

where k is equal to 1 if $k = y_i$, and -1 otherwise. Intuitively, if the sample belonging to class c has a lower score than η_c, the threshold η_c will decrease. On the other hand, if a sample not belonging to class c obtains a higher score than η_c, it will increase.

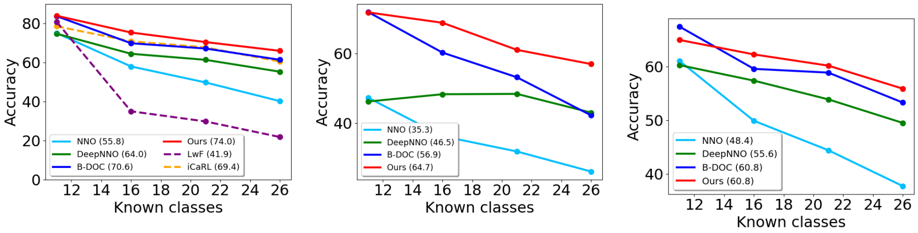


(a) Closed World Without Rejection

(b) Closed World With Rejection

(c) OWR Harmonic Mean

Fig. 2: Comparison of LwF [24], iCaRL [36], NNO [2], DeepNNO [28], B-DOC [13], and our method on Core50 dataset [27]. The parenthesis denote the average accuracy among the different incremental steps.



(a) Closed World Without Rejection

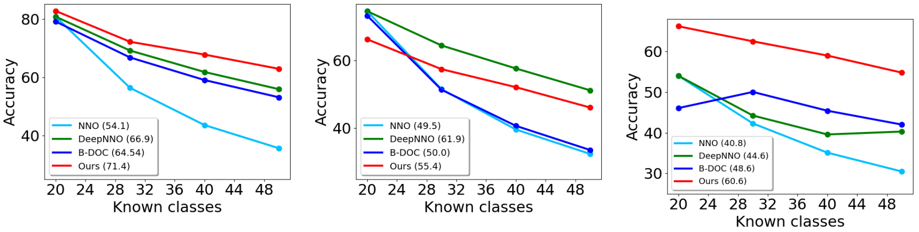
(b) Closed World With Rejection

(c) OWR Harmonic Mean

Fig. 3: Comparison of iCaRL [36], NNO [2], DeepNNO [28], B-DOC [13] and our method on RGB-D Object dataset [22]. The parenthesis denote the average accuracy among the different incremental steps.

4 Experiments

Datasets. Following the same evaluation protocol of [13], we evaluate the performance of our model on three datasets: Core50 [27], RGB-D Object [22] and CIFAR-100 [20]. Core50 dataset [27] represents a very challenging benchmark with 50 different objects grouped into 10 semantic categories and captured in 11 distinct sequences under shifting conditions. Following [13], we divide the 10 categories into two splits: 5 are considered as known classes and the remaining 5 as unknown. We use the first 2 known classes as the initial training set and we incrementally add the other classes one by one. The RGB-D Object dataset [22] contains 51 different semantic categories of daily-life objects collected in a controlled scenario. Following previous works [28,13], we divided its categories into two split: the first 26 categories are considered as known classes, while the remaining 25 are considered as unknown ones. Among the 26 categories, the first 11 ones constitute the initial training set and the remaining ones are added incrementally in 4 steps of 5 classes each. CIFAR-100 [20] is a largely adopted benchmark to compare incremental class learning algorithms [36]. It consists of 100 semantic categories with 500 training images and 100 testing images per class. Follow previous works [28,13] we divide the dataset into 50 known and 50 unknown categories. As for Core50 [27] and RGB-D Object dataset [22], we



(a) Closed World Without Rejection

(b) Closed World With Rejection

(c) OWR Harmonic Mean

Fig. 4: Comparison of NNO [2], DeepNNO [28], B-DOC [13] and our method on CIFAR-100 dataset [20]. The parenthesis denote the average accuracy.

identify an initial training set, which in this case corresponds to 20 classes chosen among the known set. We then incrementally add the remaining ones in steps of 10 classes each.

Networks architectures and training protocols. Following previous works [13] we employ a ResNet-18 architecture [17] as backbone. For each dataset we start by training the network from scratch. For Core50 dataset, we use 5 epochs for the initial training classes and 20 epochs for the incremental steps. For RGB-D Object dataset, instead, we train the network on the initial classes for 4 epochs and then incrementally for 10 epochs. Finally, for CIFAR-100 we set to 70 both the epochs for the initial learning stage and the following incremental steps. We set the learning rate to 0.02 for both the RGB-D Object and Core50 datasets, while we use 0.2 for CIFAR-100. We adapt Stochastic Gradient Descent (SGD) with momentum 0.9 and a weight decay of 10^{-3} for the RGB-D Object and 10^{-4} for both Core50 and CIFAR-100. To learn η_c on the held-out set of samples, we use 20 epochs for the three datasets. We use a learning rate of 0.001 for Core50, 0.07 for the RGB-D Object dataset and 0.01 for CIFAR-100. We also employ the same strategy for memory management of [13], which set the maximum storable samples up to 2000. 40% of the instances from memory are then drawn to construct each training batch. 20% of the stored samples, instead, are not used to directly train the model but only to learn the class-specific thresholds.

Metrics. Following previous works [13,28,2], we use three standard metrics for comparing the performances of OWR methods. In the closed world *without rejection* setting, the models is evaluated only on the known set of classes, with no possibility of considering any sample as unknown. In the closed world *with rejection* scenario, instead, the model may either categorize a sample into one of the known classes or classify it as unknown. This scenario is much more difficult than the preceding one because samples from the known set of classes may be misclassified as unknowns. Overall, for open world evaluations, we use the standard harmonic mean (OWR-H) metric defined in [13].

To compute the method performance, we randomly picked 5 distinct sets of known categories for each dataset and we repeated each experiments 3 times. The final performance is obtained averaging the results of each run and order.

Results. In the following, we report the comparison of our method with the state-of-the-art of OWR (NNO [2], DeepNNO [28], and B-DOC [13]) and incremental learning (IL) (iCaRL [36] and LwF [24]). Fig. 2 reports the results on Core50 dataset. On the closed world without rejection (Fig. 2a), our method outperforms the OWR state of the art, surpassing B-DOC [13] by 2.6% on average, and even IL methods, surpassing iCaRL by 4.9% and LwF by 25.7%. This result indicates that the adoption of $L_{D_{POLY}}$ is beneficial for learning, obtaining a model more robust on predictions over the old classes. Considering the closed world with rejection (Fig. 2b), our method rejects less known classes, obtaining higher performances than B-DOC, on average, by 3.1%. It outperforms previous methods especially in incremental steps, indicating that introducing new classes does not reduce the confidence on previous classes. Finally, considering both known and unknown samples, our method is superior to previous works, outperforming B-DOC by 1.6% on the OWR-H (Fig. 2c).

Fig. 3 reports the results on RGB-D Object dataset. Similarly to Core50 dataset, our method outperforms IL methods, surpassing iCaRL by 14.1% in the last step and by 4.6% on average (Fig. 3a). It also surpasses OWR methods by a large margin when considering rejection (Fig. 3b), achieving an average accuracy of 64.7%, more than 7% w.r.t. B-DOC and DeepNNO. The effectiveness of our method is also confirmed by the OWR-H metric (Fig. 3c), where it archives performance comparable to B-DOC and outperforms DeepNNO and NNO.

Finally, we report in Fig. 4 the results on CIFAR-100 dataset. As for Core50 and RGB-D datasets, our method outperforms OWR state-of-the-art by a large margin. In particular, in the closed world without rejection, it surpasses DeepNNO by 4.5% and B-DOC by 6.8% on average (Fig. 4a). In closed world with rejection (Fig. 4b) DeepNNO achieves slightly higher performance, reaching up to 52.5%. The reason is that DeepNNO classifies most of the samples into known classes, failing in rejecting them as unknown. This behaviour is confirmed by the OWR-H metric (Fig. 4c) in which our method achieves much higher performance than DeepNNO (56.1% vs 42.8%), benefiting from the rejection strategy based on features importance.

Ablations. Due to lack of space, we report the ablation studies in the supplementary material.

5 Conclusion

In this work, we studied the open world recognition problem in robot vision. We first proposed to relax the forget-constraint imposed by previous methods to prevent catastrophic forgetting. In particular, we proposed a new distillation function, the Poly loss, that enabled changes in the model's output when it was uncertain about the old class prediction. Moreover, we introduced the forget-constraint relaxation strategy to further relax the distillation constraint on certain samples, enabling the network to reach an optimal representation for novel classes without forgetting previous classes. Second, we abandon the metric-learning strategy to detect unknown samples and we propose to directly use the model's classifier. We demonstrate the benefits of our contributions on Core50, RGB-D Object, and CIFAR-100 datasets outperforming the state of the art.

References

1. Belouadah, E., Popescu, A.: Il2m: Class incremental learning with dual memory. In: ICCV-19
2. Bendale, A., Boult, T.: Towards open world recognition. In: CVPR-15
3. Camoriano, R., Pasquale, G., Ciliberto, C., Natale, L., Rosasco, L., Metta, G.: Incremental robot learning of new objects with fixed update time. In: ICRA-17
4. Camoriano, R., Traversaro, S., Rosasco, L., Metta, G., Nori, F.: Incremental semi-parametric inverse dynamics learning. In: ICRA-16
5. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: ECCV-18
6. Cermelli, F., Mancini, M., Bulò, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. CVPR-20
7. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: Continual learning: A comparative study on how to defy forgetting in classification tasks. arXiv preprint arXiv:1909.08383 **2**(6) (2019)
8. De Rosa, R., Mensink, T., Caputo, B.: Online open world recognition. arXiv:1604.02275 (2016)
9. Della Santina, C., Arapi, V., Averta, G., Damiani, F., Fiore, G., Settimi, A., Catalano, M., Bacciu, D., Bicchi, A., Bianchi, M.: Learning from humans how to grasp: A data-driven architecture for autonomous grasping with anthropomorphic soft hands. RA-L-19
10. Douillard, A., Chen, Y., Dapogny, A., Cord, M.: Plop: Learning without forgetting for continual semantic segmentation. In: CVPR-21
11. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: ECCV-20
12. Fontanel, D., Cermelli, F., Mancini, M., Caputo, B.: On the challenges of open world recognition under shifting visual domains. RA-L-20 **6**(2)
13. Fontanel, D., Cermelli, F., Mancini, M., Rota Buló, S., Ricci, E., Caputo, B.: Boosting deep open world recognition by clustering. RA-L **5**(4), 5985–5992 (2020)
14. Fragoso, V., Sen, P., Rodriguez, S., Turk, M.: Evsac: accelerating hypotheses generation by modeling matching scores with extreme value theory. In: ICCV-13
15. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in cognitive sciences **3**(4) (1999)
16. Guerriero, S., Caputo, B., Mensink, T.: Deep nearest class mean classifiers. In: ICLR-WS-18
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR-16
18. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv 1503.02531 (2015)
19. Hu, X., Tang, K., Miao, C., Hua, X.S., Zhang, H.: Distilling causal effect of data in class-incremental learning. In: CVPR-21
20. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., Technical report, University of Toronto. (2009)
21. Kumar, V.R., Yogamani, S., Rashed, H., Sitsu, G., Witt, C., Leang, I., Milz, S., Mäder, P.: Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving. RA-L-21 **6**(2)
22. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: ICRA-11
23. Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Díaz-Rodríguez, N.: Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. Information Fusion **58**, 52–68 (2020)

24. Li, Z., Hoiem, D.: Learning without forgetting. T-PAMI-17
25. Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A.D., Jui, S., de Weijer, J.v.: Generative feature replay for class-incremental learning. In: CVPR-20
26. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: CVPR-20
27. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: CoRL-17
28. Mancini, M., Karaoguz, H., Ricci, E., Jensfelt, P., Caputo, B.: Knowledge is never enough: Towards web aided deep open world recognition. In: ICRA-19
29. Mancini, M., Costante, G., Valigi, P., Ciarfuglia, T.A., Delmerico, J., Scaramuzza, D.: Toward domain independence for learning-based monocular depth estimation. RA-L-17 **2**(3)
30. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
31. Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In: ECCV-12
32. Michieli, U., Zanuttigh, P.: Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In: CVPR-21
33. Michieli, U., Zanuttigh, P.: Knowledge distillation for incremental learning in semantic segmentation. CVIU-21 **205**
34. Peng, C., Zhao, K., Lovell, B.C.: Faster ilod: Incremental learning for object detectors based on faster rcnn. Pattern Recognition Letters **140** (2020)
35. Perez-Rua, J.M., Zhu, X., Hospedales, T.M., Xiang, T.: Incremental few-shot object detection. In: CVPR-20
36. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: CVPR-17
37. Saito, N., Ogata, T., Funabashi, S., Mori, H., Sugano, S.: How to select and use tools?: Active perception of target objects using multimodal deep learning. RA-L-21 **6**(2)
38. Scheirer, W.J., de Rezende Rocha, A., Sapkota, A., Boult, T.E.: Toward open set recognition. T-PAMI-12 **35**(7)
39. Schwarz, M., Milan, A., Periyasamy, A.S., Behnke, S.: Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter. IJRR-18 **37**(4-5)
40. Shmelkov, K., Schmid, C., Alahari, K.: Incremental learning of object detectors without catastrophic forgetting. In: ICCV-17
41. Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J., Upcroft, B., Abbeel, P., Burgard, W., Milford, M., et al.: The limits and potentials of deep learning for robotics. IJRR-18 **37**(4-5)
42. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: CVPR-20
43. Valipour, S., Perez, C., Jagersand, M.: Incremental learning for robot perception through hri. In: IROS-17
44. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: CVPR-19
45. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: CVPR-20