Tackling Online One-Class Incremental Learning by Removing Negative Contrasts

Nader Asadi[*1, 2] **Sudhir Mudur**[1] **Eugene Belilovsky**[*1, 2]

[1] Concordia University, Montreal, Canada
[2] Mila, Montreal, Canada

Abstract

Recent work studies the supervised online continual learning setting where a learner receives a stream of data whose class distribution changes over time. Distinct from other continual learning settings the learner is presented new samples only once and must distinguish between all seen classes. A number of successful methods in this setting focus on storing and replaying a subset of samples alongside incoming data in a computationally efficient manner. One recent proposal ER-AML achieved strong performance in this setting by applying an asymmetric loss based on contrastive learning to the incoming data and replayed data. However, a key ingredient of the proposed method is avoiding contrasts between incoming data and stored data, which makes it impractical for the setting where only one new class is introduced in each phase of the stream. In this work we adapt a recently proposed approach (*BYOL*) from self-supervised learning to the supervised learning setting, unlocking the constraint on contrasts. We then show that supplementing this with additional regularization on class prototypes yields a new method that achieves strong performance in the one-class incremental learning setting and is competitive with the top performing methods in the multi-class incremental setting.

1 Introduction

Continual learning is a paradigm that aims to allow deep learning algorithms which will have the ability to learn online from a non-stationary and never-ending stream of data. Such systems must become capable of acquiring new knowledge, while avoiding catastrophic forgetting of previously seen data, a problem commonly known and suffered by gradient-based neural networks [11]. A number of common continual learning scenarios exist in the classification setting, each with their own set of related but also different challenges. These are often characterized along a number of axes. The first is which information is available to the learner at training and test time. In our setting we focus on the single-head setting where a learner is unaware at test time which task the data belongs to. In other words, when new classes are presented the learner must learn to distinguish them from all previously observed classes. Another common distinction is the online and offline setting. In the offline setting the learner receives the full set of data for each task and can perform unlimited training on this before moving to the next. On the other hand in the online setting the learner receives one or a small number of samples from a stream and must process and/or store these samples under a computational and memory budget. In this work we focus on the latter.

An extreme continual classification scenario involves the learner observing individual classes at each changing point in the data stream. Until now this scenario has been studied to a limited degree in the offline setting [13, 10], but has not been considered in the online setting. This is due to the inherent challenging nature of the online setting which has only recently started obtaining results

[*]Correspondence to: {nader.asadi, eugene.belilovsky}@concordia.ca

Workshop on Distribution Shifts, 35th Conference on Neural Information Processing Systems (NeurIPS 2021).

competitive with iid baselines [5, 4]. Indeed, in [10] they use an expensive regularization approach that is impractical under the constraints of online continual learning. Furthermore, the method proposed by [10] uses a pre-trained model, performing an initial stage of offline iid pretraining on half of the classes in the dataset. Where as, our aim is to tackle the problem without any assumption of pre-trained models.

In the online continual learning setting the best performing methods rely on various forms of rehearsal, where old samples are stored in a finite memory and reused at later points in the stream. A common strong baseline is experience replay[8, 2]. Recently, Caccia et al. [4] showed that in online continual learning settings, after each distribution shift (task boundaries), the model observes a significant drift in the representation of previously learned classes. They hypothesize that this is fundamentally due to: (i) new class samples representations lying close to older classes and (ii) the loss structure of the standard cross entropy applied on a mix of seen and unseen classes. To mitigate this, they proposed a method to allow fine-grained control over which samples will be pushed away from other samples given an incoming batch. The main downside of methods proposed by Caccia et al. [4], is the critical need for negative samples in the incoming batch to learn the representation of incoming data while avoiding catastrophic forgetting. The result is the inability of their methods to be applied to the more challenging setting where the model observes one class at a time in an online data stream.

In this work, we apply recent ideas from [12], which break the dependence on negative samples in self-supervised contrastive learning to the supervised continual learning setting. This allows us to maintain an asymmetric loss structure between replay samples and incoming data as in [4], while providing learning on new incoming class data without the need to contrast to old classes or their representations. Augmenting this approach with a regularization term that constrains class prototypes, further yields a new method which far exceeds performance of strong online CL benchmarks in the one class incremental setting and is competitive with the top performing methods in the multi-class incremental setting.

2 Methods

Similar to [4], given a model $f_\theta(x)$ representing a neural network architecture with parameters θ, we want to minimize the classification loss $\mathcal{L}$ on the newly arriving data batch while not negatively impacting previous learning of other classes, and have the ability to be applied to one-class incremental settings. We opt for a specific loss structure on the incoming batch that would enable the model to learn the representation of each class independently and in isolation from all other classes, either in the incoming batch or in buffered samples. This removes the need for negative samples and enables the model to learn useful representations, even in one-class incremental settings. We first present our supervised modification of BYOL and apply it in the asymmetric setting of [4], then propose our new method.

2.1 Supervised BYOL (SupBYOL)

In order to remove the need for negative samples to learn the representation of incoming batch, we apply a supervised modification of BYOL [12] on the incoming data. BYOL uses a twin architecture with online and target networks. The online network is comprised of three stages: an encoder f_θ, a projector g_θ, and a predictor q_θ. The target network has the same architecture but with different parameters ξ, and is the exponential moving average of the online network: $\xi \leftarrow \tau\xi + (1-\tau)\theta$. We use the following loss, a modification of BYOL loss [12], on the incoming data $\mathbf{X}^{in}$.

$$\mathcal{L}_1^{byol} = -\frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}_{in}} \frac{1}{|P(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in P(\mathbf{x}_i)} \texttt{sim}\big(q_\theta(z_\theta), z_\xi\big) \tag{1}$$

where $\texttt{sim}(a,b) = \frac{a^T b}{\tau \|a\| \|b\|}$, and $z_\theta = g_\theta(f_\theta(x_i))$ and $z_\xi = g_\xi(f_\xi(x_i))$ are the projections outputted by online and target networks respectively. We denote the incoming N data points by $\mathbf{X}^{in}$, data replayed from the buffer by $\mathbf{X}^{bf}$, and the set of positive samples with respect to $\mathbf{x}_i$ by P. Following [4], for the rehearsal step, we apply a modified cross-entropy objective as per [14].

$$\mathcal{L}_2(\mathbf{X}^{bf}) = -\sum_{x_i \in \mathbf{X}_{bf}} \log \frac{\exp\big(\texttt{sim}\big(\mathbf{c}_{y(x_i)}, f_\theta \mathbf{z_i}\big)\big)}{\sum_{y \in \mathcal{Y}_{all}} \exp\big(\texttt{sim}\big(\mathbf{c}_y, \mathbf{z_i}\big)\big)} \tag{2}$$

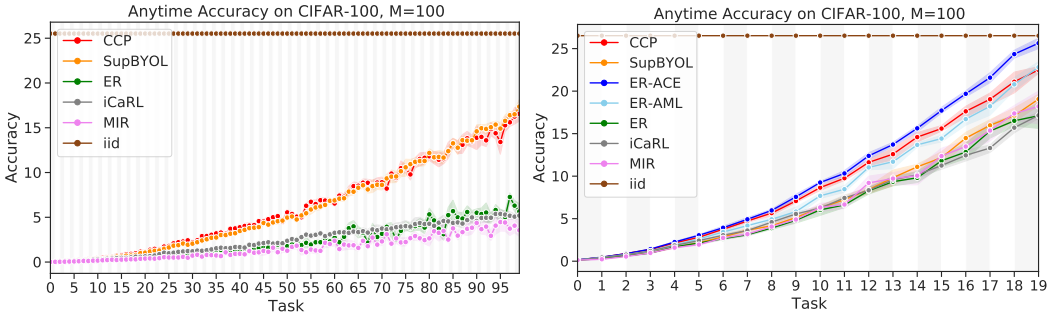


Figure 1: Split CIFAR-100 anytime evaluation results in one-class(left) and multi-class(right) incremental settings with $M = 100$. SupBOYL and CCP greatly outperform the existing methods in one-class setting while being competitive with the top performing methods in multi-class setting. Note that ER-ACE and ER-AML cannot be applied in one-class setting since they require more than one class in the incoming data.

where Y_{all} is the set of all observed classes, and $z_i = g_\theta(f_\theta(x_i))$.

2.2 CCP: Continual Contrast of Class Prototypes

Although BYOL can learn the representation of each class independently from other classes, in case of supervised learning, specially where we have multiple classes in the incoming data, it is tricky to evade collapsed representations. In some cases, we observed representation collapse between new classes in the incoming data or old classes in the memory buffer, which results in forgetting of the corresponding classes (drop in accuracy). Also, due to the twin architecture design, BYOL has a larger compute footprint than ER-AML, which makes it less suitable for online continual learning.

Inspired by [6], we use randomly initialized prototypes as class cluster representatives to reduce intra-class variance while enforcing inter-class variance. We represent each observed class $y \in \mathcal{Y}$ by a prototype $\mathbf{c}_y$ in prototypes memory $\mathcal{C}$. The network parameters θ and class prototypes $\mathcal{C}$ are jointly optimized to project an instance $\mathbf{x}_i \in \mathbf{X}^{in}$ of class y close to its corresponding cluster prototype $\mathbf{c}_y$ as well as other samples of class y in the batch, i.e. positive samples $P(\mathbf{x}_i)$. In order to evade collapsed representations, we use a contrast term between class prototypes which enforces inter-class variance. So, as the prototypes act like a representative for the whole corresponding class samples, we can enforce inter-class variance without any need for negative samples, and also provide stable training without BYOL's optimization tricks such as twin architecture and predictor head. We formulate the objective as follows:

$$\mathcal{L}_1^{ccp} = -\frac{1}{N} \sum_{\mathbf{x}_i \in \mathbf{X}_{in}} \left(\mathtt{sim}(\mathbf{z}_i, \mathbf{c}_{y(x_i)}) + \frac{1}{|P(\mathbf{x}_i)|} \sum_{\mathbf{x}_p \in \mathbf{P}(\mathbf{x}_i)} \mathtt{sim}\big(\mathbf{z}_i, \mathbf{z}_p\big) \right) + \frac{1}{|\mathcal{C}|} \sum_{\mathbf{c}_i \in \mathcal{C}} \sum_{\mathbf{c}_j \in \mathcal{C}/i} \mathtt{sim}\big(\mathbf{c}_i, \mathbf{c}_j\big) \tag{3}$$

where $\mathtt{sim}(a, b) = \frac{a^T b}{\tau \|a\| \|b\|}$, $z_i = g_\theta(f_\theta(x_i))$, and $y(x)$ denotes the class label of x. As in the case of SupBYOL, we combine this with the same $\mathcal{L}_2$ term applied to the buffered samples. In order to avoid contrast between new and old classes introduced by the last term, we do not directly update old class prototypes using gradients. In other words, for the incoming data, we perform stochastic optimization to minimize $\mathcal{L}_1 + \mathcal{L}_2$ with respect to θ and $c \in \mathcal{C}^{in}$, where $\mathcal{C}^{in}$ is the class labels of the incoming data. However, to update the old class prototypes, we follow [9] and use the replayed samples to obtain a momentum update after each training step to stabilize the model against drastic changes in the representation of learned classes: $c_y \leftarrow \alpha c_y + (1 - \alpha)\bar{c}_y$ where $y \in \mathcal{Y}_{bf}$ denotes the label of old classes stored in the buffer and $\bar{c}_y$ is the updated prototype for class y.

3 Experiments

Our method enables the model to learn the representation of each class in isolation from other classes without the need to have more than one class in the incoming data. Following [7, 4], we use a reduced

	Accuracy (↑ is better)				Forgetting (↓ is better)		
	$M = 5$	$M = 20$	$M = 50$	$M = 100$	$M = 20$	$M = 50$	$M = 100$
iid online	63.4 ± 0.6	63.4 ± 0.6	63.4 ± 0.6	63.4 ± 0.6	N/A	N/A	N/A
fine-tuning	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	10.0 ± 0.0	100.0± 0.0	100.0± 0.0	100.0± 0.0
ER [8]	15.7 ± 1.7	18.2 ± 2.6	19.6 ± 3.1	21.2 ± 2.3	54.7± 2.5	51.1± 2.8	45.7± 2.6
iCarl [15]	18.8 ± 1.6	20.5 ± 1.5	23.3 ± 1.2	24.4 ± 0.9	46.1± 1.7	42.8± 1.7	40.1± 1.4
MIR [1]	17.8 ± 1.8	19.3 ± 2.7	20.2 ± 2.2	22.0 ± 2.4	52.0± 2.9	47.4± 2.6	41.9± 2.8
DER++ [3]	20.1 ± 1.4	21.2 ± 2.4	23.2 ± 2.4	24.6 ± 1.5	58.9± 2.2	53.3± 2.2	49.8± 1.8
SupBYOL (ours)	**24.4** ± 1.2	30.6 ± 1.4	33.3 ± 2.1	36.0 ± 2.3	28.3± 1.6	25.6± 2.1	23.1± 2.4
CCP (ours)	**24.2** ± 0.9	**34.3** ± 1.1	**36.0** ± 1.3	**39.1**± 1.1	**27.2**± 1.0	**24.2**± 1.1	**21.9**± 1.0

Table 1: Accuracy and Forgetting results on Split CIFAR-10 in **one-class** incremental setting(10 tasks) with augmentations and different buffer sizes. Averages and standard deviations are computed over five runs. SupBYOL and CCP outperform other methods with a considerable margin in both Accuracy and Forgetting.

	Accuracy (↑ is better)				Forgetting (↓ is better)		
	$M = 5$	$M = 20$	$M = 50$	$M = 100$	$M = 20$	$M = 50$	$M = 100$
iid online	63.4 ± 0.6	63.4 ± 0.6	63.4 ± 0.6	63.4 ± 0.6	N/A	N/A	N/A
fine-tuning	17.9 ± 0.2	17.9 ± 0.2	17.9 ± 0.2	17.9 ± 0.2	80.9± 0.1	80.9± 0.1	80.9± 0.1
iCarl [15]	33.4± 1.0	39.2± 0.8	41.6± 0.9	42.3± 0.8	31.3± 0.8	30.8± 1.2	29.4± 1.6
ER [8]	28.4± 1.0	40.3± 0.6	42.8± 1.2	49.4± 1.3	26.8± 0.8	24.4± 1.2	22.8± 1.6
MIR [1]	29.8± 1.0	41.8± 0.6	45.6± 0.7	49.3± 0.6	38.2± 1.2	21.6± 0.9	15.8± 1.1
DER++ [3]	31.8 ± 0.9	39.3± 1.0	46.7± 1.1	52.3± 1.1	29.7± 1.1	24.5± 1.0	19.0± 1.1
ER-AML [4]	**36.4**± 1.4	**47.7**± 0.7	**52.6** ± 1.1	**55.7**± 1.3	19.8± 0.4	16.4± 0.4	15.9± 0.3
ER-ACE [4]	35.1± 0.9	43.4± 1.6	49.3 ± 1.2	53.7± 1.1	**18.3**± 0.6	**15.2**± 0.8	**14.6**± 0.8
SupBYOL (ours)	25.4 ± 1.2	36.6 ± 1.3	41.4 ± 1.2	43.6 ± 1.8	20.7± 1.1	18.9± 1.2	17.4± 1.2
CCP (ours)	34.2 ± 0.9	42.0 ± 1.1	47.6 ± 1.0	51.2 ± 0.9	19.7± 0.8	16.3± 1.0	**14.4**± 0.8

Table 2: Accuracy and Forgetting results on Split CIFAR-10 in **multi-class** incremental setting (5 tasks) with augmentations and different buffer sizes. Averages and standard deviations are computed over five runs. Our proposed method, CCP, is competitive with top performing methods in this setting.

Resnet-18 with *batch size* and the *rehearsal batch size* of *10* samples. All of the models are trained in single head setting, so the task id is not revealed to the model at test time. Similar to [4], we find data augmentation to be useful in most of the settings, specially in simple datasets like CIFAR-10 with a small buffer size where the model might overfit on the buffer samples. Our data augmentation pipline consists of a simple random crop followed by random horizontal flip. We use SGD for optimization with a learning rate of 0.1 as in [1]. We now present the experiments on 10 and 100 task settings for Split CIFAR-10 and Split CIFAR100.

Split CIFAR-10 typically partitions the dataset into 5 disjoint tasks containing two classes each ([1, 16]). In this work we also consider partitioning into 10 disjoint sets (1 class each). When applying the 10 class split we will indicate $S = 1$, while the case of 2 class tasks will be denoted $S = 2$.

We show in Table 1 results for split CIFAR10 on $S = 1$ and for $S = 2$ in Table 2 showing the overall accuracy and forgetting at the end of the task sequence for a variety of memory settings. Observe that in the one class setting ER-ACE and ER-AML cannot be applied as they require other classes in the incoming data. SupBOYL and CCP greatly outperform the existing methods in this setting, with CCP obtaining top performance in all categories. For the multi-class setting of $S = 2$, SupBYOL performs poorly, but CCP greatly improves upon SupBYOL and achieves performance close to ER-AML and ER-ACE in both accuracy and forgetting categories.

Split CIFAR-100 Consists of 100 classes typically split into 20 tasks, each containing a disjoint set of 5 labels ($S = 5$). In our one class incremental work we will also consider the case of splitting into 100 distinct task switches ($S = 1$). All CIFAR experiments process 32×32 images.

In Figure 1, we show the results on $S = 1$ and $S = 5$. In the $S = 1$ setting especially as the number of classes grows, we observe increasing margins over existing methods for both SupBYOL and CCP. On the other hand, in the $S = 5$ setting, SupBYOL does not perform well as in the CIFAR-10, while CCP matches the performance of ER-AML and is close to the performance of ER-ACE (which nearly matches the i.i.d performance).

4 Conclusion

Our major contribution is a new method to handle online one-class incremental learning without the need for negative contrasts. We demonstrated that this method can outperform strong baselines, and is also applicable and highly competitive in traditional online continual learning settings. Furthermore we have shown that recent advances in self-supervised learning without contrasts can be adapted to supervised settings, particularly in continual classification. Future work can consider if our approach can be applied in the offline one class incremental setting.

Acknowledgements

We would like to thank Concordia University and Mila for the provided computational resources and research environments. We also would like to acknowledge the support from Compute Canada. Lastly, we thank Amir Sarfi for helpful discussions and suggestions.

References

[1] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *arXiv preprint arXiv:1908.04742*, 2019.

[2] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *arXiv preprint arXiv:1903.08671*, 2019.

[3] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *arXiv preprint arXiv:2004.07211*, 2020.

[4] Lucas Caccia, Rahaf Aljundi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. Reducing representation drift in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021.

[5] Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned continual compression with adaptive quantization modules. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

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

[8] 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.

[9] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*, 2019.

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020.

[11] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[13] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

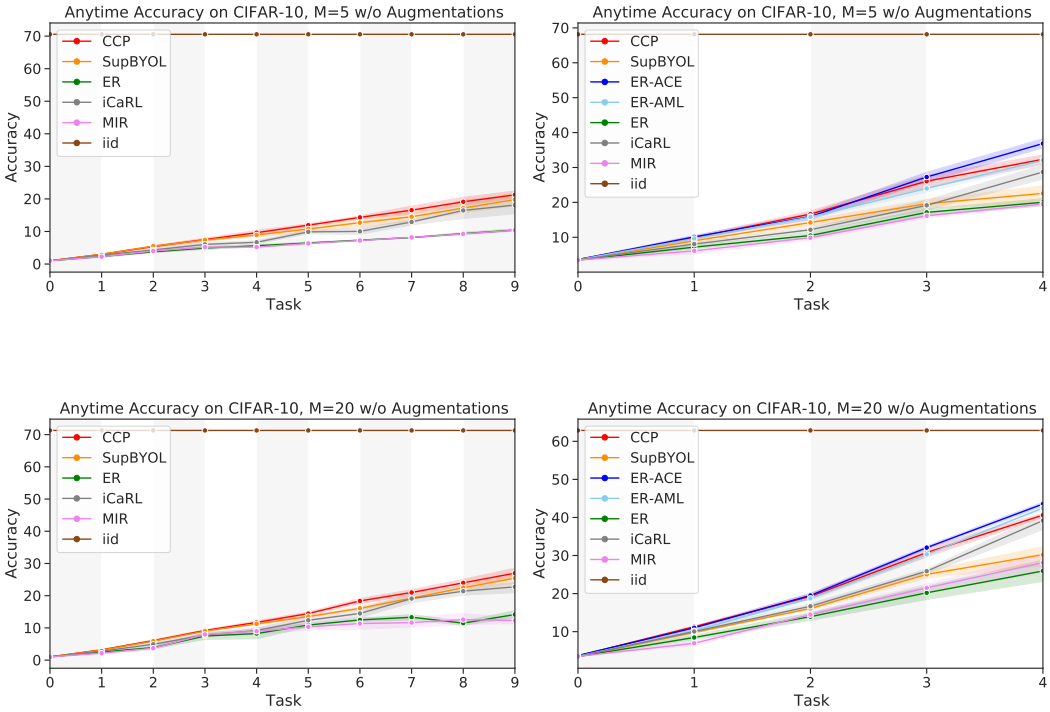
[14] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5822–5830, 2018.

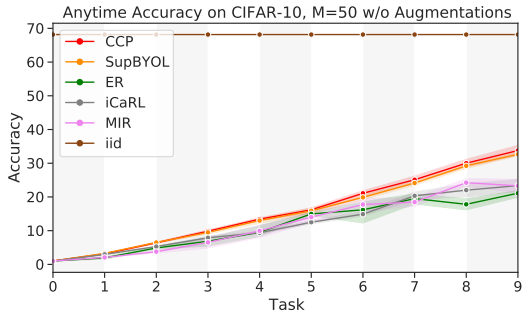
[15] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

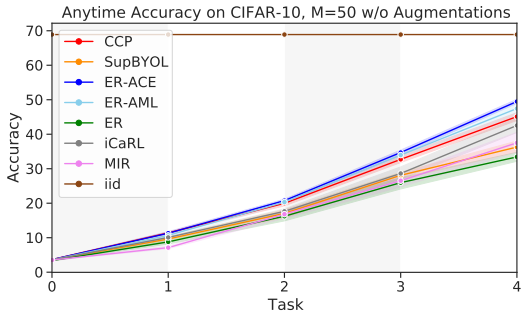
[16] Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. *arXiv e-prints*, pages arXiv–2009, 2020.

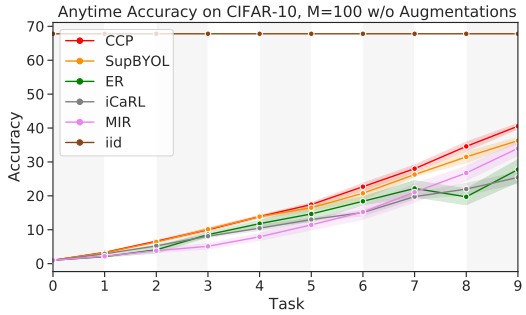
A Appendix

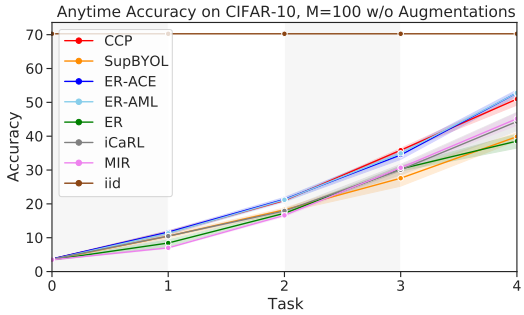
In this section, we provide some extra anytime evaluation results for various memory sizes on the considered datasets , i.e. Split CIFAR-10 and Split CIFAR-100, in both one-class(left plots) and multi-class(right plots) incremental learning. The results complement the ones presented in the main text.

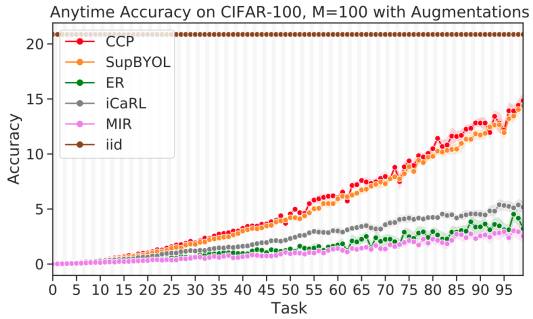


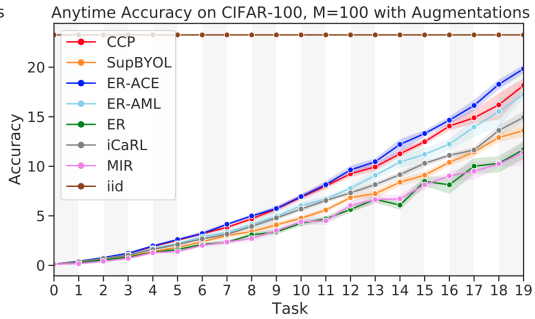

Anytime Accuracy on CIFAR-10, M=50 w/o Augmentations
CCP
SupBYOL
ER
iCaRL
MIR
iid
Accuracy
Task


Anytime Accuracy on CIFAR-10, M=50 w/o Augmentations
CCP
SupBYOL
ER-ACE
ER-AML
ER
iCaRL
MIR
iid
Accuracy
Task


Anytime Accuracy on CIFAR-10, M=100 w/o Augmentations
CCP
SupBYOL
ER
iCaRL
MIR
iid
Accuracy
Task


Anytime Accuracy on CIFAR-10, M=100 w/o Augmentations
CCP
SupBYOL
ER-ACE
ER-AML
ER
iCaRL
MIR
iid
Accuracy
Task


Anytime Accuracy on CIFAR-100, M=100 with Augmentations
CCP
SupBYOL
ER
iCaRL
MIR
iid
Accuracy
Task


Anytime Accuracy on CIFAR-100, M=100 with Augmentations
CCP
SupBYOL
ER-ACE
ER-AML
ER
iCaRL
MIR
iid
Accuracy
Task