Incremental Meta-Learning via Episodic Replay Distillation for Few-Shot Image Recognition

Kai Wang
Computer Vision Center
Barcelona, Spain
kwang@cvc.uab.es

Xialei Liu
Nankai University
Tianjin, China
xialei@nankai.edu.cn

Andy Bagdanov
MICC, University of Florence,
Florence, Italy
andrew.bagdanov@unifi.it

Luis Herranz
Computer Vision Center
Barcelona, Spain
lherranz@cvc.uab.es

Shangling Jui
Huawei Kirin Solution
Shanghai, China
jui.shangling@huawei.com

Joost van de Weijer
Computer Vision Center
Barcelona, Spain
joost@cvc.uab.es

Abstract

In this paper we consider the problem of incremental meta-learning in which classes are presented incrementally in discrete tasks. We propose Episodic Replay Distillation (ERD), that mixes classes from the current task with exemplars from previous tasks when sampling episodes for meta-learning. To allow the training to benefit from a large as possible variety of classes, which leads to more generalizable feature representations, we propose the cross-task meta loss. Furthermore, we propose episodic replay distillation that also exploits exemplars for improved knowledge distillation. Experiments on four datasets demonstrate that ERD surpasses the state-of-the-art. In particular, on the more challenging one-shot, long task sequence scenarios, we reduce the gap between Incremental Meta-Learning and the joint-training upper bound from 3.5% / 10.1% / 13.4% / 11.7% with the current state-of-the-art to 2.6% / 2.9% / 5.0% / 0.2% with our method on Tiered-ImageNet / Mini-ImageNet / CIFAR100 / CUB, respectively.

1. Introduction

Meta-learning, also commonly referred to as "learning to learn", is a learning paradigm in which a model gains experience over a sequence of learning episodes.[1] This experience is optimized so as to improve the model's future learning performance on unseen tasks [1]. Meta-learning is one of the most promising techniques to learning models that can flexibly generalize, like humans, to new tasks and environments not seen during training. This capability is generally considered to be crucial for future AI systems. Few-shot learning has emerged as the paradigm-of-choice to test and evaluate meta-learning algorithms. It aims to learn from very limited numbers of samples (as few as just one), and meta-learning applied to few-shot image recognition in particular has attracted increased attention recently [2–5].

However, most few-shot learning methods are limited in their learning modes: they must train with a large number of classes, with a large number of samples per class, and then generalize and recognize new classes from few samples. This can lead to poor performance in practical in-

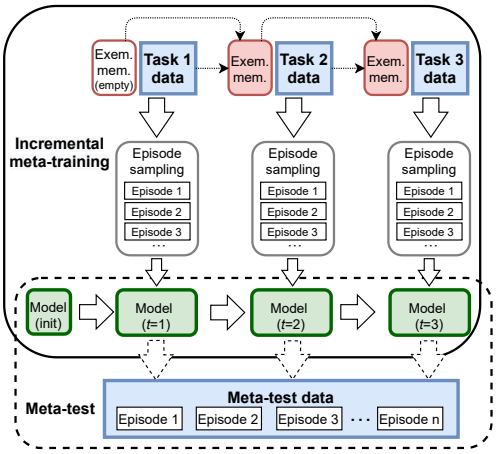


Figure 1. Incremental meta-learning with optional exemplar memories [6]. Data from the previous tasks, unless in the exemplar memory, is unavailable in successive ones. Conventional meta-learning assumes a large number of base classes available for episodic training, while *incremental* meta-learning requires that the meta-learner updates incrementally when a new set of classes (a new *task*) arrives.

[1] To avoid ambiguities, we use the term *episode* in the sense used in meta-learning rather than how it is used in continual learning. We use *task* in the sense of continual learning to refer to a disjoint group of new classes.

cremental learning situations where the training tasks arrive continually and there are insufficient categories at any given time to learn a performant and general meta-model. The study of learning from data that arrives in such a sequential manner is called *incremental* learning [7, 8]. Catastrophic forgetting is the main challenge of incremental learning systems [9]. To address both the challenges of incremental and meta learning, Incremental Meta-Learning (IML, illustrated in Fig. 1) was proposed as a way of applying few-shot learning in such incremental learning scenarios [6].

To address the IML problem, [6] propose the Indirect Discriminant Alignment (IDA) method. In this method, class centers from previous tasks are represented by *anchors* which are used to align (by means of a distillation process) the old and new discriminants. They show that this greatly reduces forgetting for short sequences of tasks. They also extended IDA with exemplars[2] (EIML) from old tasks, but surprisingly results showed that this fails to outperform IDA without exemplars. This seems counter-intuitive, since exemplars usually boost performance in incremental learning. We identify the following drawbacks of IDA and EIML: (i) in IDA the anchors are fixed after obtaining them from their corresponding tasks, while semantic drift will gradually make prediction worse with successive tasks; (ii) in EIML exemplars are used only for distillation and computing class anchors, while they are not mixed with current tasks to make the training more robust; and (iii) evaluation is only performed on short sequences (maximally 3 tasks).

In this paper we propose Episodic Replay Distillation (ERD) to better exploit saved exemplars and achieve significant improvement in IML. ERD first divides episode construction into two parts: the *exemplar sub-episodes* containing only exemplars from past tasks, and the *cross-task sub-episodes* containing a mixture of previous task exemplars and current task data. Exemplar sub-episodes are then used to produce episode-level classifiers for distillation over the query set. Cross-task sub-episodes combine previous task exemplars with current task samples using a sampling probability P. Since the current task contains more samples, and thus higher diversity, a lower P makes better use of the previous and the current samples – an interesting departure from conventional continual learning in which we would typically desire *more* replay from past tasks.

The main contributions of this paper are: 1) **Cross-task meta-learning**: we apply a *cross-task meta loss* which explicitly uses the exemplars during the meta-learning. This loss results in higher quality feature representations and better generalization to new few-shot recognition problems. 2) **Episodic replay distillation**: we exploit the exemplars to efficiently transfer the knowledge from the previous to the current model. Since the exemplars are closer to the class prototypes of previous tasks, this results in more efficient knowledge distillation. We are the first to show how exemplar replay can be used for incremental meta-learning, as the previous attempt at this only showed marginal improvements with exemplars [6]. 3) **Experimental evaluation**: we are the first to evaluate incremental meta-learning on long task sequences (evaluation is increased from just 3 tasks in [6] to 16 tasks in our work). Our method significantly outperforms the state-of-the-art using both Prototypical Networks and Relation Networks.

[2] *Exemplars* refer to a small buffer of samples from previous tasks that can be used during the training of new ones. Note that the rest of samples are discarded and can not be accessed anymore.

2. Related work

In this section we briefly review the work from the literature most related to our proposed approach.

2.1. Few-shot learning

Few-shot learning can be categorized into three main classes of approaches according to which aspect is enhanced using prior knowledge: data augmentation, model enhancement and algorithm-based methods [10]. Among them, few-shot learning based on metrics or optimization-based approaches are the main streams in current research.

Metric-based methods. These approaches use embeddings learned from other tasks as prior knowledge to constrain the hypothesis space. Since samples are projected into an embedding subspace, the similar and dissimilar samples can be easily discriminated. Among these techniques, ProtoNets [11], RelationNets [12], MatchNets [13] and TADAM [14] are the most popular.

Optimization-based methods. These use prior knowledge to search for the model parameters which best approximate the hypothesis in search space, and use prior knowledge to alter the search strategy by providing good initialization or guiding optimization steps. Representative methods are MAML++ [15], Reptile [16] and MetaOptNet [17].

2.2. Continual learning

Continual learning methods can be divided into three main categories [7]: replay-based, regularization-based and parameter-isolation methods. Since parameter-isolation methods are restricted to the task-aware settings [7], we only discuss the first two categories which are relevant.

Replay methods. These prevent forgetting by including data (real or synthetic) from previous tasks, stored either in an episodic memory or via a generative model. There are two main strategies: exemplar replay [18–21] and pseudo-replay [22, 23]. The classification model in continual learning is a joint classifier, the exemplars are used to correct the bias [21] or regularize the gradients [24]. However, in incremental meta-learning there is no joint classifier (only a temporary classifier for each episode). Thus, those exemplar-

based methods need adaptation to the incremental meta-learning. Replay for Incremental Meta-Learning should be at the episode level instead of the image level.

Regularization-based methods. These approaches add a regularization term to the loss function which impedes changes to the parameters deemed relevant to previous tasks. The difference depends on how to estimate relevance, and these methods can be further divided into data-focused [25] and prior-focused [26]. Data-focused methods use knowledge distillation from previously-learned models. Prior-focused methods estimate the importance of model parameters as a prior for the new model.

Distillation methods in continual learning are trying either to align the outputs at the feature level [23, 27] or the predicted probabilities after a softmax layer [25]. However, aligning at the feature level has been observed to be not effective [6] and the lack of a unified classifier makes it impossible to align in probabilities level. Thus, we also must adapt distillation to the Incremental Meta-Learning setting.

2.3. Meta-learning for continual learning

In addition to the Incremental Meta-Learning setting, there are a few works on continual learning that exploit meta-learning, such as La-MAML [28], iTAML [29] and OSAKA [30]. These methods focus on improving model performance on task-agnostic incremental classification. There is also some work focusing on dynamic, few-shot visual recognition systems [31–33], which aim to learn novel categories from only a few training samples while at the same time not forgetting the base categories. Another related setting is FSCIL [34–37], where they constrain the continual learning tasks using a few labeled samples (excluding the base task, which has many classes and abundant images to enable learning a strong pretrained model).

Different from these variants, the incremental meta-learning setting adopts the original objective of few-shot learning: to make the model generalize to unseen tasks even when training over incremental tasks, where each task contains a significant amount of data (and is therefore more similar to standard class-incremental learning) on which we train our meta-learner. Since the seen classes are increasing, the model should gain more generalization ability instead of over-fitting to the current task. The meta-learning can then perform few-shot classification on unseen classes – something which is not considered in few-shot class-incremental learning (FSCIL). And since conventional continual learning methods are not suitable for incremental meta-learning, we propose our approach specifically for this setting.

3. Methodology

Future learning systems will aim to continually integrate new tasks without requiring joint training over all previously seen data [26, 38]. Specifically, the combination of incremental learning with meta-learning is relevant, since at test-time new problems with unseen classes are evaluated. Therefore, it is important to develop incremental learning theory on how this new information can be absorbed by the learner to further improve its performance on future tasks. Furthermore, incremental learning does not require the learner to be trained from scratch every time new data arrives (which is also more sustainable), and it can be applied in settings where it is prohibited to retain all past data due to privacy concerns or governmental legislation. A practical example of incremental meta-learning is a robot which must continue to function – with minimal labelling effort – in new scenarios where it must manipulate previously unseen objects. At the same time, it should incrementally improve its model to increase performance in future scenarios. Other scenarios include Lifelong Person Re-identification [39], and few-shot drug discovery [40].

In this section, we start by defining the standard few-shot learning formulation and then introduce the *incremental meta-learning* setup. Then in sections 3.2 and 3.3 we describe our approach to Incremental Meta-Learning and its application to few-shot image recognition.

3.1. Few-shot and meta-learning

We first introduce the standard formulation of few-shot learning, then describe the incremental meta-learning approach as applied to few-shot classification.

Conventional few-shot learning. An approach to standard, non-incremental classification is to learn a parametric approximation $p(y|x;\theta)$ of the posterior distribution of the class y given the input x. Such models are trained by minimizing a loss function over a dataset D (e.g. the empirical risk). Few-shot learning, however, presents extra difficulties since the number of samples available for each class y is very small (as few as one). In the meta-learning paradigm, training is divided into two phases: meta-training, in which the model learns how to learn few-shot recognition, and meta-testing where the meta-trained model is evaluated on unseen few-shot recognition tasks.

Meta-training for few-shot learning consists of H *episodes* (meta-training task in few-shot learning terminology), where each episode D^τ is drawn from the train split. Few-shot recognition problems consisting of N classes with K training samples per class are referred to N-way, K-shot recognition problems. Each episode is divided into support set S and query set Q: $D^\tau = (S, Q)$, where $S = \{(x_i, y_i)\}_{i=1}^{NK}$ consists of N training classes each with K images, and $Q = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{NK^Q}$ is a set of K^Q images for each of the N selected classes in the episode.

More specifically, we formulate our method based on *ProtoNets* in this section, and discuss its extension to Relation Networks later. ProtoNets consist of an embedding module f_θ and a classifier module g. First, the support set

S is fed into the embedding module f_θ to obtain class prototypes $\mathbf{c}_k$:

$$\mathbf{c}_k = \frac{1}{K} \sum_{(x_i,k)\in S} f_\theta(x_i). \quad (1)$$

Then, an episode-specific classifier is applied to the query set, where the prediction for class k of query image $\hat{x}$ is:

$$g_k(f_\theta(S), f_\theta(\hat{x})) = p(y=k|\hat{x};\theta) = \frac{\exp(-d(f_\theta(\hat{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_\theta(\hat{x}), \mathbf{c}_{k'}))} \quad (2)$$

where the summation in the denominator is over all classes k' in the support set and d is the Euclidean distance as in ProtoNets. Then the meta-loss for updating θ is:

$$L_{meta}(\theta; S, Q) = -\sum_{(\hat{x},\hat{y})\in Q} [\log g_{\hat{y}}(f_\theta(S), f_\theta(\hat{x}))]. \quad (3)$$

Incremental Meta-Learning. When performing incremental meta-learning, data arrives as a sequence of disjoint tasks: $X_1, ..., X_t, ..., X_T$, where T denotes the number of tasks , and t the current training session. The aim of Incremental Meta-Learning is to incrementally learn the parameters θ_t for task t from the disjoint tasks:

$$\theta_t^* = \arg\min_{\theta_t} L(\theta_t; \theta_{t-1}, S_t, Q_t), \quad (4)$$

Depending on whether we store exemplars from previous tasks, the support set S_t and query set Q_t can be constructed differently using samples in the current task and exemplars from previous tasks. These query and support sets are described in detail in the next section.

3.2. Cross-task episodic training

Keeping exemplars from previous tasks is a successful approach to avoid catastrophic forgetting in conventional incremental learning [18, 41, 42]. However, it is not obvious how to use exemplars for incremental *meta*-learning. We propose a novel way of using exemplars for this setup.

To fully exploit exemplars E_t from previous tasks, for each episode during meta-learning we construct two sets of support and query images (see Figure 2-(a)). Each episode is broken down into two sets of few-shot problems:

- In each episode, we construct a *cross-task sub-episode* by sampling N classes from the current task with probability $1-P$ and from previous tasks with probability P. It means for each of the N classes in the episode, a Bernoulli trial with probability P determines if the class is drawn from a past task. Thus, we have on average $N \times (1-P)$ classes from the current task and $N \times P$ from the past. Then, for each class, we randomly sample K images as support set S^m and K_Q images as query set Q^m (m denotes that we *mix* the exemplars with current task samples here).

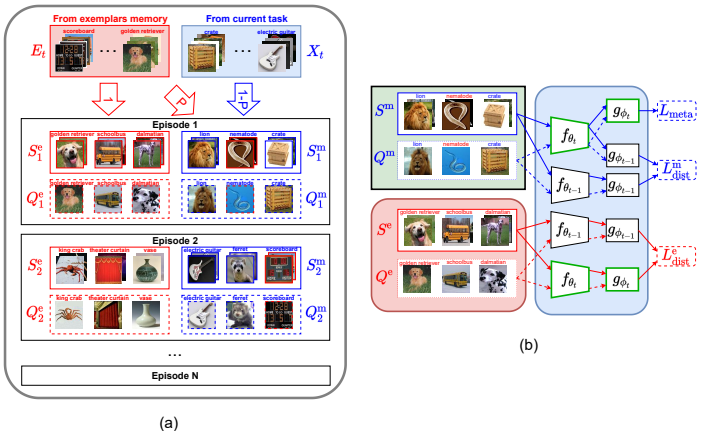


Figure 2. (a) Proposed episode sampling. During episodic meta-learning we build two sets of few-shot problems: *exemplar sub-episodes* based on only exemplars from previous tasks (S_i^e and Q_i^e) and *cross-task sub-episodes* with a mix of exemplars from previous tasks and samples from the current task (S_i^m and Q_i^m). (b) Proposed Episodic Replay Distillation framework. Modules in green are the current embedding model, which are updated with both cross-task and exemplar sub-episodes. Red lines and blue lines are data flows for exemplar sub-episode and cross-task sub-episode, respectively. Solid lines and dotted lines indicate the data flows from support set and query set respectively. When computing loss for ProtoNets, g is a parametric-free operation, while for Relation Networks, g consists of a set of parameters ϕ.

- We also construct an *exemplar sub-episode* by sampling N classes from only the exemplars from previous tasks, each with $K+K_Q$ images to form a support set S^e and query set Q^e. Note that this episode is only composed of *exemplars* from previous tasks.

The reason we sample cross-task sub-episodes with probability P is that exemplars are normally much fewer than samples in the current task, and thus the exemplars are not expected to be as varied as the samples from the current task. With a probability P, we can control the balance between current and previous classes in the cross-task sub-episode. And it doesn't influence the update of the memory.

Given S^m, Q^m, the cross-task meta-training loss is defined as:

$$L_{\text{meta}}(\theta_t; S^m, Q^m) = -\sum_{(\hat{x},\hat{y})\in Q^m} \log g_{\hat{y}}(f_{\theta_t}(S^m), f_{\theta_t}(\hat{x})). \quad (5)$$

This loss is only computed over S^m and Q^m since in S^m we have samples from the previous and current tasks.

The intra-task meta-loss used in [6] only performs the meta-learning on the data of the current task. The quality of the meta-learner is expected to improve with when learned on wide variety of classes [2–4]. Only considering the classes within the current task is therefore expected to

limit its generalization. In conclusion, we propose to also exploit the replay memory during the meta-learning by performing the meta-learning on both the cross-task and exemplar sub-episodes.

For saving exemplar to E_t, we consider two widely used strategies. The first strategy stores N_{ex} exemplars for each class of each previous task, which is standard in replay-based continual learning methods (UCIR, PODNet, etc.). In this case, the buffer is linearly increased by training sessions. The second strategy fixes the maximum buffer size to M exemplars. We apply both settings in the ablation study and will use the increasing buffer strategy as default.

3.3. Episodic Replay Distillation (ERD)

In addition to cross-task episodic training, multiple distillation losses are applied to avoid forgetting when we update the current model (see Figure 2-(b)). We first explore distillation using *exemplar sub-episodes*. It is computed as:

$$L^e_{dist}(\theta_t;\theta_{t-1},S^e,Q^e)=\sum_{\hat{x}\in Q^e}\{KL[g(f_{\theta_{t-1}}(S^e),f_{\theta_{t-1}}(\hat{x}))\,||\,g(f_{\theta_t}(S^e),f_{\theta_t}(\hat{x}))]\} \tag{6}$$

where $f_{\theta_{t-1}}$ is the embedding network from the previous task with parameters θ_{t-1}. During training, only the current model f_{θ_t} is updated and $f_{\theta_{t-1}}$ is frozen.

Next, similar to Eq. 6, we also propose a distillation loss using *cross-task sub-episodes*. It is computed according to:

$$L^m_{dist}(\theta_t;\theta_{t-1},S^m,Q^m)=\sum_{\hat{x}\in Q^m}\{KL[g(f_{\theta_{t-1}}(S^m),f_{\theta_{t-1}}(\hat{x}))\,||\,g(f_{\theta_t}(S^m),f_{\theta_t}(\hat{x}))]\} \tag{7}$$

The only difference between this distillation loss function and Eq. 6 is the inputs.

Finally, θ_t is updated by minimizing:

$$L(\theta_t;\theta_{t-1},S^e,Q^e,S^m,Q^m)=L_{\text{meta}}+\lambda_m L^m_{\text{dist}}+\lambda_e L^e_{dist}, \tag{8}$$

where λ_m and λ_e are trade-off parameters.

3.4. Extension to Relation Networks

Episodic Replay Distillation is not limited to ProtoNets. It can also be extended to Relation Networks [12], which consist of a relation module with parameters ϕ. Losses introduced in previous sections are adapted as:

$$L_{meta}(\theta_t,\phi_t;S^m,Q^m)=\sum_{(x,y)\in S^m,(\hat{x},\hat{y})\in Q^m}[g_{\phi_t}(\mathcal{C}(f_{\theta_t}(x),f_{\theta_t}(\hat{x})))-\mathbf{1}(y=\hat{y})]^2, \tag{9}$$

where $\mathcal{C}$ is the concatenation of support and query set embeddings, and $\mathbf{1}$ a Boolean function returning 1 when its argument is true and 0 otherwise. Distillation losses are updated as:

$$\begin{aligned}L^m_{dist}(\theta_t,\phi_t;\theta_{t-1},S^m,Q^m)=&\\ \sum_{x\in S^m,\hat{x}\in Q^m}&[g_{\phi_{t-1}}(\mathcal{C}(f_{\theta_{t-1}}(x),f_{\theta_{t-1}}(\hat{x})))\\ &-g_{\phi_{t-1}}(\mathcal{C}(f_{\theta_t}(x),f_{\theta_t}(\hat{x})))]^2\\ L^e_{dist}(\theta_t,\phi_t;\theta_{t-1},\phi_{t-1},S^e,Q^e)=&\\ \sum_{x\in S^e,\hat{x}\in Q^e}&[g_{\phi_{t-1}}(\mathcal{C}(f_{\theta_{t-1}}(x),f_{\theta_{t-1}}(\hat{x})))\\ &-g_{\phi_t}(\mathcal{C}(f_{\theta_t}(x),f_{\theta_t}(\hat{x})))]^2\end{aligned} \tag{10}$$

Although Relation Networks and ProtoNets adopt different ways to calculate the prediction probabilities for given query images, they share similar network architectures with embedding and classification modules. This type of architecture is widely used in metric-based few-shot learning, and we believe that our method can be easily adapted to other methods with similar architectures.

4. Experiments

Here we report on a range of experiments to quantify the contribution of each element of the proposed approach and to compare our performance against the state-of-the-art in continual few-shot image classification. More experimental results are reported in the supplementary material.

4.1. Experimental setup

Here we describe the datasets and experimental protocols used in our experiments.

Datasets. We evaluate performance on four datasets: Mini-ImageNet [13], CIFAR100 [43], CUB-200-2011 [44] and Tiered-ImageNet [45]. Mini-ImageNet consists of 600 84×84 images from 100 classes. We propose a split with 20 of these classes as meta-test set unseen during training sessions. The other 80 classes are used to form the incremental meta-training set which is split into 16 tasks with equal numbers of classes for incremental meta-learning. Each class in each task is then divided into a meta-training split with 500 images, from which support and query sets are sampled for each episode, and a test split with 100 images that is set aside for task-specific evaluation. We select $N_{ex}=20$ exemplars per class before learning the next task.

CIFAR100 also contains 100 classes, each with 600 images, so we use the same splitting criteria as for Mini-ImageNet. The CUB dataset contains 11,788 images of 200 birds species. We split 160 classes into an incremental meta-training set and the other 40 are kept as a meta-test set of unseen classes. We divide the 160 classes into 16 equal incremental meta-learning tasks. Since there are fewer images per class, we choose $N_{ex}=10$ images per class as

exemplars for each previous task and 20 images as test split for each class in each task. On Tiered-ImageNet, We keep the same test split (8 categories, 160 classes) as in the original setup, then split the training and validation classes (26 categories, 448 classes) into 16 equally-sized tasks. We select $N_{ex} = 20$ exemplars for each class and 300 images per class as the test split.

Implementation details. We use ProtoNets as our main meta-learner, but also validate ERD using Relation Networks. We evaluate both the 4-Conv [11] and ResNet-12 [46] backbones as feature extractors. We sample $H = 50$ episodes per task in each training epoch. We train each meta-learning task for 200 epochs using Adam [47] with a learning rate as 0.001.

We evaluate on two widely used few-shot learning scenarios: 1-shot/5-way and 5-shot/5-way. We include results on both incremental training tasks and the unseen meta-test set. For each task (including the unseen set), we randomly construct N_{ep} episodes to obtain the final performance of the meta-learner, which is computed as the mean classification accuracy across the N_{ep} episodes. N_{ep} is set as 1000. For exemplar selection for ProtoNets, we use the Nearest-To-Center (NTC) criterion to select samples closest to the class mean (see Supplementary for a comparison of rehearsal selection methods). If the exemplar memory of size M is full, we iteratively remove exemplars from the class with the most exemplars until there remain only M total exemplars. For Relation Networks, since the image embeddings are feature maps instead of feature vectors, we cannot obtain class prototypes and therefore use random selection. By default we set $\lambda_m = \lambda_e = 0.5$ and $P = 0.2$. All reported results are an average of three runs under one fixed, randomly-generated class order for each dataset.

Compared methods. We compare our method with a finetuning baseline (FT), IDA [6], and a variant of IDA with N_{ex} exemplars per class (EIML). The meta-test upper bounds are obtained by jointly training on all training tasks and testing on the unseen meta-test split (i.e., the standard setting in non-incremental, few-shot learning). We evaluate on two sets of tasks separately for comparison. At each training session, we evaluate on previously *seen* classes as a way of measuring forgetting. This we call *mean accuracy on seen classes*. Performance on the *meta-test set* (all unseen classes) instead measures the generalization ability to new few-shot recognition problems.

4.2. Experimental results.

In this section, we report on experiments performed on 16-task incremental few-shot learning scenarios.

Comparison with the state-of-the-art. Here we report results on 16-task 1-shot/5-shot 5-way incremental meta-learning on all four datasets. The first and third rows in Figure 3 show mean accuracy on previous tasks. It is clear that we achieve significantly less forgetting compared to other methods. In the second and fourth rows, the meta-test accuracy for ERD increases with more meta-training tasks due to seeing more diverse classes, while for IDA and EIML the performance drops significantly in some training sessions. This might be due to forgetting on previous tasks and overfitting to the current one. Notably, for our method, the meta-test accuracy after the last task is much closer to the joint training upper bound.

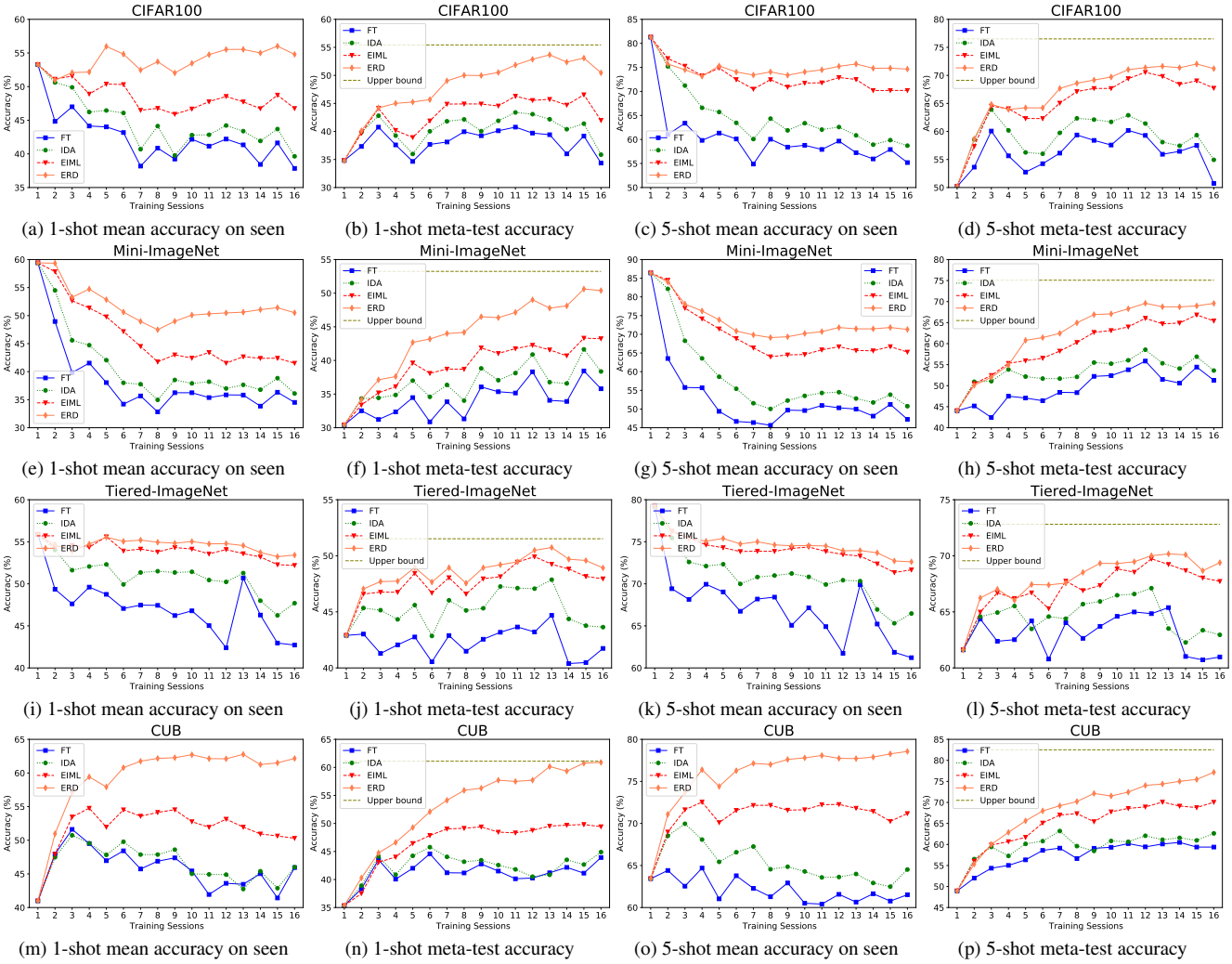
Note also that EIML works much better than IDA after around 4 tasks. In the original IDA paper, the authors report similar results for both IDA and EIML, which might simply be due to only evaluating on very short sequences of two or three tasks. This is likely caused by anchor drift in IDA and the fact that in EIML exemplars could be used to recalibrate them. In general. All methods work better in the 5-shot evaluation. The underlying reason for this is that 1-shot recognition is more complex than 5-shot.

For Tiered-ImageNet, the trends are similar to CIFAR100 and Mini-ImageNet, but the performance difference between EIML and ERD is smaller since there are more classes in each task on Tiered-ImageNet. For CUB, we generally see similar trends as in the other three datasets. However, since CUB is a fine-grained dataset, the forgetting in *mean accuracy on seen classes* is not as serious as for the other three coarse-grained datasets. Instead, we observe increasing *mean accuracy on seen classes*, which could be because the new tasks benefit from the accumulated knowledge from the old ones.

Finally, in Figures 4a and 4b we report the average over 10 random orders on CIFAR100 to show the robustness of our model to changing task order.

4.3. Comparison with standard CL methods

Our main comparison was presented in the previous section comparing to the state-of-the-art IDA/EIML method especially designed for incremental meta learning. Here, in Figure 5, we also compare our method with three state-of-the-art CL methods: iCaRL [18], PODNet [48] and UCIR [20]. Note that these methods were not designed for incremental meta-learning and cannot be directly applied to this scenario. To adapt these methods to incremental meta-learning, we use them to continually learn representations and then evaluate them with a nearest-centroid classifier for few-shot learning. For the evaluation on seen classes, we follow the same protocol as IDA where the average classification accuracy is calculated over N_{ep} episodes. Observe how on seen classes UCIR works better than iCaRL and PODNet, however under meta-test evaluation, iCaRL works the best among the standard CL methods. PODNet performs similarly to the FT baseline in both cases. Our method, that is especially tailored for incremental meta-learning, outperforms the standard CL methods by



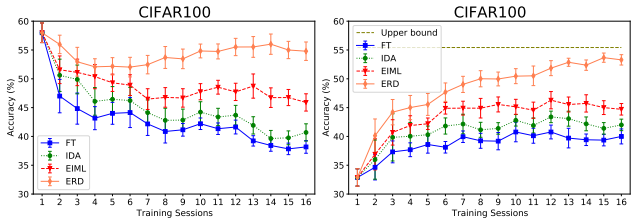
(a) 1-shot mean accuracy on seen (b) 1-shot meta-test accuracy (c) 5-shot mean accuracy on seen (d) 5-shot meta-test accuracy

(e) 1-shot mean accuracy on seen (f) 1-shot meta-test accuracy (g) 5-shot mean accuracy on seen (h) 5-shot meta-test accuracy

(i) 1-shot mean accuracy on seen (j) 1-shot meta-test accuracy (k) 5-shot mean accuracy on seen (l) 5-shot meta-test accuracy

(m) 1-shot mean accuracy on seen (n) 1-shot meta-test accuracy (o) 5-shot mean accuracy on seen (p) 5-shot meta-test accuracy

Figure 3. Results on the 1- and 5-shot, 5-way 16-task setup with a 4-Conv backbone and ProtoNets meta-learner. Evaluations are on CIFAR100, Mini-ImageNet and Tiered-ImageNet datasets.



(a) 1-shot mean Acc. on seen (b) 1-shot meta-test Acc.

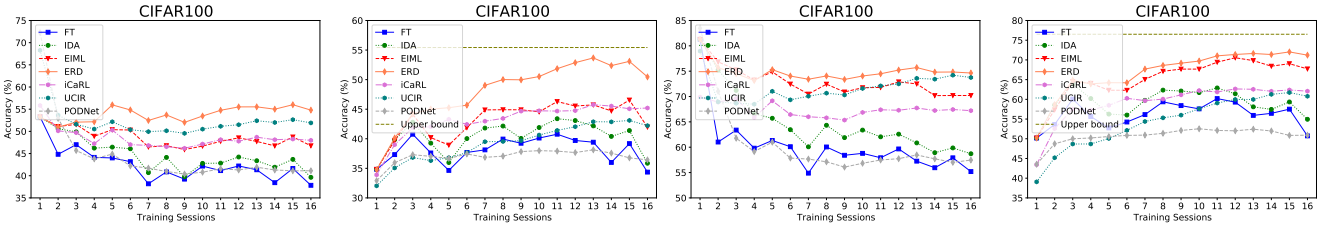
Figure 4. Experimental results with 10 task orderings.

a large margin – especially for few-shot evaluation on unseen classes, where our 1-shot meta-test accuracy outperforms iCaRL by around 8.5% after 16 tasks.

4.4. Ablation studies

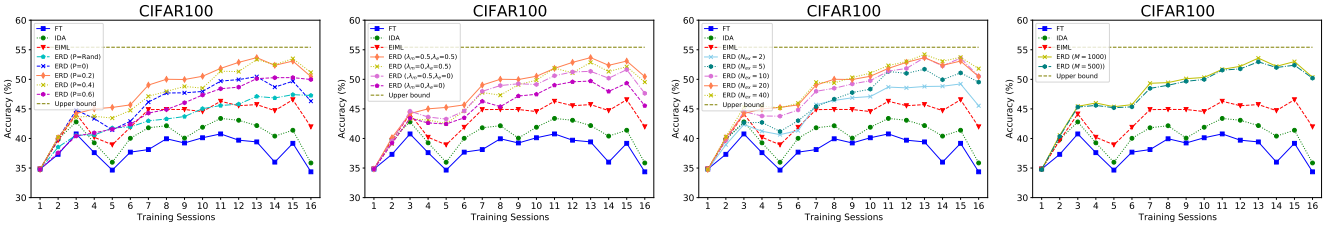
Here we show ablation studies on CIFAR100 in the 16-task 1-shot/5-way scenario with 4-Conv as the backbone. We report *meta-test accuracy* to compare among variants.

Ablation on P with $\lambda_m = \lambda_e = 0.5$. As shown in Figure 6a, ERD obtains the best performance with $P = 0.2$. This is what we use by default for all previous experiments. When $P = 0$, it means there are no previous classes in the *cross-task sub-episode*, which performs worse than our variants with higher probabilities, especially with $P = 0.2$ and $P = 0.4$. As P decreases from 0.6 to 0.2, the performance consistently improves. The reason is that lower P results in more current samples, which can ensure the diversity of the training samples. This phenomenon is different from the conventional use of exemplars in incremental learning, where more balanced exemplar sampling is preferable. We use the notation $P = \text{Rand}$ to identify that P is not fixed, but that classes in each cross-task sub-episode are randomly selected from all encountered classes up to now and P is increasing with successive tasks. This achieves worse results because there are more and more previous classes



(a) 1-shot mean accuracy on seen (b) 1-shot meta-test accuracy (c) 5-shot mean accuracy on seen (d) 5-shot meta-test accuracy

Figure 5. Comparison with CL methods on 1-shot 5-way 16-task setting with a 4-Conv backbone and ProtoNets meta-learner on CIFAR-100. (Left) Mean accuracy on seen classes. (Right) Meta-test accuracy on the unseen meta-test set.



(a) Ablation on P (b) Ablation on λ_e, λ_m (c) Ablation on N_{ex} (d) Ablation on M

Figure 6. Ablation study on 16-task 1-shot/5-way setup on CIFAR100 with 4-Conv. We plot the meta-test accuracy to compare.

with less diverse samples. Most of our variants outperform EIML by a large margin. We keep $P \geq 0.2$ to ensure that at least one previous class occurs in each episode for 5-way few-shot learning.

Ablation on λ_m and λ_e with $P = 0.2$. To understand the role of each distillation component in Eq. (8), we ablate the distillation loss terms. As shown in Figure 6b, our method achieves the best results with $\lambda_m = 0.5$ and $\lambda_e = 0.5$, which indicates that both distillation terms play a crucial role in overcoming forgetting and generalizing to unseen tasks. ERD with $\lambda_m = 0.5$, $\lambda_e = 0$ works similarly to ERD with $\lambda_m = 0$, $\lambda_e = 0.5$. They both achieve much better performance than without using distillations.

Ablation on memory buffer with $P = 0.2, \lambda_m = \lambda_e = 0.5$. In this experiment, we fix other hyper-parameters to show how different numbers of exemplars affect incremental learning performance. We provide results for various N_{ex} and also with a bounded buffer size M which are both commonly used for exemplar replay. From Figure 6c we see that increasing N_{ex} leads to a noticeable increase in performance going from 2 to 20 exemplars (note that in EIML increasing the number of exemplars does not influence performance). However, also for ERD the gain is marginal beyond 20 exemplars per class. From Figure 6d, we observe that with a smaller bounded buffer with only $M = 500$ exemplars, ERD is still close to the joint training upper bound, showing the importance of proposed sub-episodes.

4.5. Extension to Relation Networks

Since in Relation Networks there is no embedding to exploit for computing prototypes as in ProtoNets, IDA and EIML cannot be directly applied. Therefore, we only compare with FT in this experiment. As the experimental results shown in Table 1, our model not only surpasses the FT baseline significantly, but also gets close to the joint training upper bounds after the last task, especially on CUB dataset.

Learner:	Relation Networks											
Datasets:	Mini-ImageNet				CIFAR100				CUB			
Backbone:	4-Conv											
	1-shot/5-way ***16-task*** setting											
	Upper bound: 52.0				Upper bound: 59.2				Upper bound: 51.6			
Sessions:	2	4	8	16	2	4	8	16	2	4	8	16
FT	24.4	28.5	25.5	28.7	31.1	30.0	35.2	26.8	37.1	38.1	37.0	34.0
ERD	**27.7**	**29.9**	**34.5**	**30.1**	**35.6**	**39.3**	**45.7**	**35.9**	**37.3**	**42.9**	**47.9**	**42.5**

Table 1. Meta-test accuracy by training sessions on the 16-task settings. We evaluate 1-shot/5-way few-shot recognition on Mini-ImageNet, CIFAR-100 and CUB.

5. Conclusions

In this paper we proposed Episodic Replay Distillation, an approach to incremental few-shot recognition. We are the first to show how this successful tool can be used for incremental meta-learning. We exploit the exemplars to perform cross-task meta-learning which improves the discriminative power of the learned representations. In addition, we also use exemplars to perform our proposed episodic replay distillation. Both contributions are shown to considerably improve performance. Experiments on multiple few-shot learning datasets demonstrate the effectiveness of ERD.

Acknowledgements

We acknowledge the support from Huawei Kirin Solution, the Spanish Gouvernement funded project PID2019-104174GB-I00/ AEI / 10.13039/501100011033 and Ramón y Cajal grant RYC2019-027020-I (MICINN, Spain).

References

[1] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, Meta-learning in neural networks: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). 1

[2] J.-C. Su, S. Maji, B. Hariharan, When does self-supervision improve few-shot learning?, in: European Conference on Computer Vision, Springer, 2020, pp. 645–666. 1, 4

[3] P. Bateni, R. Goyal, V. Masrani, F. Wood, L. Sigal, Improved few-shot visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 14493–14502. 1, 4

[4] K. Li, Y. Zhang, K. Li, Y. Fu, Adversarial feature hallucination networks for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 13470–13479. 1, 4

[5] S. Yang, L. Liu, M. Xu, Free lunch for few-shot learning: Distribution calibration, International Conference on Learning Representations (2021). 1

[6] Q. Liu, O. Majumder, A. Achille, A. Ravichandran, R. Bhotika, S. Soatto, Incremental few-shot meta-learning via indirect discriminant alignment, in: European Conference on Computer Vision, Springer, 2020, pp. 685–701. 1, 2, 3, 4, 6

[7] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). 2

[8] J. Schwarz, W. Czarnecki, J. Luketina, A. Grabska-Barwinska, Y. W. Teh, R. Pascanu, R. Hadsell, Progress & compress: A scalable framework for continual learning, in: International Conference on Machine Learning, PMLR, 2018, pp. 4528–4537. 2

[9] M. McCloskey, N. J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of learning and motivation, Vol. 24, Elsevier, 1989, pp. 109–165. 2

[10] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM Computing Surveys (CSUR) 53 (3) (2020) 1–34. 2

[11] J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, Advances in Neural Information Processing Systems (2017). 2, 6

[12] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, T. M. Hospedales, Learning to compare: Relation network for few-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208. 2, 5

[13] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, Advances in Neural Information Processing Systems (2016). 2, 5

[14] B. N. Oreshkin, P. Rodriguez, A. Lacoste, Tadam: Task dependent adaptive metric for improved few-shot learning, Advances in Neural Information Processing Systems (2018). 2

[15] A. Antoniou, H. Edwards, A. Storkey, How to train your maml, International Conference on Learning Representations (2019). 2

[16] A. Nichol, J. Achiam, J. Schulman, On first-order meta-learning algorithms, arXiv preprint arXiv:1803.02999 (2018). 2

[17] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 10657–10665. 2

[18] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010. 2, 4, 6

[19] A. Chaudhry, M. Ranzato, M. Rohrbach, M. Elhoseiny, Efficient lifelong learning with a-gem, International Conference on Learning Representations (2019). 2

[20] S. Hou, X. Pan, C. C. Loy, Z. Wang, D. Lin, Learning a unified classifier incrementally via rebalancing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 831–839. 2, 6

[21] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, Y. Fu, Large scale incremental learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 374–382. 2

[22] H. Shin, J. K. Lee, J. Kim, J. Kim, Continual learning with deep generative replay, in: Advances in Neural Information Processing Systems, 2017, pp. 2994–3003. 2

[23] C. Wu, L. Herranz, X. Liu, J. van de Weijer, B. Raducanu, et al., Memory replay gans: Learning to generate new categories without forgetting, Advances in Neural Information Processing Systems 31 (2018) 5962–5972. 2, 3

[24] D. Lopez-Paz, M. Ranzato, Gradient episodic memory for continual learning, Advances in Neural Information Processing Systems (2017). 2

[25] Z. Li, D. Hoiem, Learning without forgetting, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (12) (2017) 2935–2947. 3

[26] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proceedings of the national academy of sciences 114 (13) (2017) 3521–3526. 3

[27] X. Liu, C. Wu, M. Menta, L. Herranz, B. Raducanu, A. D. Bagdanov, S. Jui, J. van de Weijer, Generative feature replay for class-incremental learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 226–227. 3

[28] G. Gupta, K. Yadav, L. Paull, La-maml: Look-ahead meta learning for continual learning, in: Advances in Neural Information Processing Systems, 2020, pp. 11588–11598. 3

[29] J. Rajasegaran, S. Khan, M. Hayat, F. S. Khan, M. Shah, itaml: An incremental task-agnostic meta-learning approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 13588–13597. 3

[30] M. Caccia, P. Rodriguez, O. Ostapenko, F. Normandin, M. Lin, L. Caccia, I. Laradji, I. Rish, A. Lacoste, D. Vazquez, et al., Online fast adaptation and knowledge accumulation: a new approach to continual learning, in: Advances in Neural Information Processing Systems, 2020, pp. 16532–16545. 3

[31] S. Gidaris, N. Komodakis, Dynamic few-shot visual learning without forgetting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4367–4375. 3

[32] M. Ren, R. Liao, E. Fetaya, R. S. Zemel, Incremental few-shot learning with attention attractor networks, Advances in Neural Information Processing Systems (2019). 3

[33] S. W. Yoon, D.-Y. Kim, J. Seo, J. Moon, Xtarnet: Learning to extract task-adaptive representation for incremental few-shot learning, in: International Conference on Machine Learning, PMLR, 2020, pp. 10852–10860. 3

[34] I. Achituve, A. Navon, Y. Yemini, G. Chechik, E. Fetaya, Gp-tree: A gaussian process classifier for few-shot incremental learning, International Conference on Machine Learning (2021). 3

[35] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, Few-shot class-incremental learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12183–12192. 3

[36] P. Mazumder, P. Singh, P. Rai, Few-shot lifelong learning, Proceedings of the Conference on Artificial Intelligence (2021). 3

[37] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, Y. Xu, Few-shot incremental learning with continually evolved classifiers, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 12455–12464. 3

[38] S. Thrun, A lifelong learning perspective for mobile robot control, in: Intelligent robots and systems, Elsevier, 1995, pp. 201–214. 3

[39] N. Pu, W. Chen, Y. Liu, E. M. Bakker, M. S. Lew, Lifelong person re-identification via adaptive knowledge accumulation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 7901–7910. 3

[40] H. Altae-Tran, B. Ramsundar, A. S. Pappu, V. Pande, Low data drug discovery with one-shot learning, ACS central science 3 (4) (2017) 283–293. 3

[41] E. Belouadah, A. Popescu, I. Kanellos, A comprehensive study of class incremental learning algorithms for visual tasks, Neural Networks 135 (2021) 38–54. 4

[42] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, J. van de Weijer, Class-incremental learning: survey and performance evaluation on image classification, arXiv preprint arXiv:2010.15277 (2020). 4

[43] A. Krizhevsky, Learning multiple layers of features from tiny images (2009). 5

[44] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Tech. Rep. CNS-TR-2011-001, California Institute of Technology (2011). 5

[45] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, R. S. Zemel, Meta-learning for semi-supervised few-shot classification, International Conference on Learning Representations (2018). 5

[46] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. 6

[47] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (2015). 6

[48] A. Douillard, M. Cord, C. Ollion, T. Robert, E. Valle, Podnet: Pooled outputs distillation for small-tasks incremental learning, in: European Conference on Computer Vision, Springer, 2020, pp. 86–102. 6