Entropy-Guided Self-Regulated Learning Without Forgetting for Distribution-Shift Continual Learning with blurred task boundaries

Rui YANG [1 2] **Matthieu Grard** [3] **Emmanuel Dellandrea** [1 2] **Liming CHEN** [1 2]

Abstract

Continual Learning (CL) aims to endow machines with the human-like ability to continuously acquire novel knowledge while retaining previously learned experiences. Recent research on CL has focused on Domain-Incremental Learning (DIL) or Class-Incremental Learning (CIL) with well-defined task boundaries. However, for real-life applications, e.g., waste sorting, robotic grasping, etc., the model needs to be constantly updated to fit new data. Additionally, there is usually an overlap between new and old data. Thus, task boundaries may not be well defined, and a more smooth scenario is needed. In this paper, we propose a more general scenario, namely Distribution-Shift Incremental Learning (DS-IL), which enables soft task boundaries with possible mixtures of data distributions over tasks and thereby subsumes the two previous CL scenarios: DIL and CIL are simply DS-IL. Moreover, given the increasingly greater importance of data privacy in real-life applications and, incidentally, data storage efficiency, we further introduce an entropy-guided self-regulated distillation process **without memory**, which leverages data similarities between tasks with soft-boundaries. Experimented on a variety of datasets, our proposed method outperforms or matches state-of-the-art continual learning methods.

1. Introduction

In recent years, intelligent systems have been required to possess Continual Learning (CL) ability, *i.e.*, the capability to learn novel knowledge from a new task while preserving previously learned knowledge and experience. However, current intelligent systems usually suffer from a notorious problem named catastrophic forgetting, *i.e.*, they quickly forget previously learned knowledge when learning new tasks.

After years of research, a variety of continual learning scenarios have been proposed. Depending upon the assumptions made for CL, three different scenarios, namely Task-Incremental Learning (TIL), Domain-Incremental Learning (DIL), and Class-Incremental Learning (CIL) (see details in Figure 1, Figure 3 and Figure 2), were first proposed in (van de Ven & Tolias, 2019)(Hsu et al., 2019) to evaluate the performance of different CL methods. TIL assumes that task identity is provided for both training and inference, whereas both DIL and CIL assume that task identity is only available during training and remains *unknown* during testing. By definition, it is assumed that novel data of unknown classes occur in CIL over tasks, while novel data of only known classes can appear in DIL. However, the previous three CL settings, as defined in (van de Ven & Tolias, 2019)(Hsu et al., 2019), assume that during training, task boundaries between tasks are clear and well-defined. For example, a new task must only have new unknown classes in CIL, while a new task must have new images(variety in illumination, background, texture, *etc.*) of known classes in DIL. However, for real-life applications, *e.g.*, waste sorting, data (garbage) are collected from different regions and on different days. If the model updates every day, the garbage collected on one day can be considered to be one single task. In this case, a new task may or may not have new unknown classes or new images from different domains, *i.e.*, there may be an overlap of classes and images between tasks. Thus, there are no well-defined task boundaries, and neither DIL nor CIL can describe this situation.

In (Lomonaco & Maltoni, 2017), a new scenario named New Instances and Classes (NIC) was proposed for their dataset CORe50(Lomonaco & Maltoni, 2017), where new instances and new classes can occur in a new task. However, it did not propose a clear method for generating NIC on other datasets. In (Zeno et al., 2021), it proposed the Continuous Task Agnostic scenario, which considers the continuous linear transition of task distribution. However, it

[*]Equal contribution [1]Liris, Ecole Centrale De Lyon, Ecully, French [2]Ecole Centrale de Lyon, Ecully, French [3]Séliane, saint-étienne, French. Correspondence to: Rui YANG <rui.yang1@ec-lyon.fr>, Matthieu Grard <m.grard@sileane.com>, Emmanuel Dellandrea <emmanuel.dellandrea@ec-lyon.fr>, Liming CHEN <liming.chen@ec-lyon.fr>.

only studies the linear transition between two tasks, meaning that application of this scenario is limited. Considering the generality of different scenarios, DIL and CIL have attracted much more attention in the literature. Current research, *e.g.*, (Rebuffi et al., 2017), (Douillard et al., 2020), (Yu et al., 2020), (Verma et al., 2021), mainly considers DIL and CIL to be two sub-questions of continual learning that are studied separately. However, both DIL and CIL involve data distribution shifts (see details in Section 3) . In this paper, we propose a novel CL perspective, namely Distribution-Shift Incremental Learning (DS-IL). More general and based on data distribution shifts, it can subsume the previous CL sub-scenarios and enables mixtures of data distributions over tasks with blurred boundaries.

A general assumption in CL is that past data cannot be entirely stored for privacy or memory constraint reasons. However, the previously learned model can be used for the next training. For a classification problem, the learning model is generally a discriminative model, so that for a given input, the softmax output is the probability distribution of the class labels. However, the information on data distribution is lost after training, meaning that the previous objective function cannot be easily reconstructed by only using a discriminative model. **Regularization-based methods**, *e.g.*, EWC(Kirkpatrick et al., 2017), SI(Zenke et al., 2017), Riemannian Walk(Chaudhry et al., 2018). These use a second-order approximation (Fisher information, Hessian matrix, etc.) to replace the previous objective function. These methods work well for Task-Incremental Learning, *i.e.*, when the system is informed of the task ID from which the data derive, and struggle against other more challenging scenarios (van de Ven & Tolias, 2019). **Memory-based methods**, *e.g.*, ER(Rolnick et al., 2019), LWF(Li & Hoiem, 2018), iCaRL (Rebuffi et al., 2017), make use of the actual data, whereas others, *e.g.*, DGR(Shin et al., 2017), DeepInv(Yin et al., 2020), rely upon generated data. They use memory or pseudo-memory to replace the previous dataset and approximate the previous objective function. They are better equipped to overcome forgetting than regularization-based methods but usually need an extra memory buffer and more computation. **Parameter isolation methods**, *e.g.*, PNN(Rusu et al., 2016), EFT(Verma et al., 2021), RKR(Singh et al., 2021), dedicate different subsets of the model parameters to different tasks. These methods only work when task boundaries are well-defined.

In order to better adapt to real applications, a method that needs less memory budget, more computation efficiency, and more flexibility is required. We observe that the task boundary is not always rigid between tasks, and that there is usually an overlap of data distributions between different tasks for many continual learning scenarios. In this case, these overlapping data can help us reconstruct the previous objective function, thus retaining knowledge. The remaining challenge is how to find these data. In this paper, we propose to make use of the previous model's entropy information on the current data to identify data similarity. By using this information, we can adjust distillation loss more precisely, based on the degree of familiarity of the learned model with the current data.

The contributions of this paper can be summarized as follows:

- We propose a novel continual learning scenario, namely Domain-Shift Incremental Learning (DS-IL), which enables mixtures of data distributions over tasks with soft task boundaries and thereby subsumes both DIL and CIL;
- In order to deal with mixtures of data distributions between tasks under DS-IL, we introduce an entropy-guided self-regulated knowledge distillation loss which, without memory, preserves previously acquired knowledge based on the degree of familiarity of the learned model with the current task's data;
- Using PACS and CIFAR-100 datasets, we define two novel experimental benchmarks in order to evaluate continual models *w.r.t.* the DS-IL scenario;
- Using the proposed novel DS-IL benchmarks on PACS and CIFAR-100, we show the effectiveness of the proposed ER-LwF (Entropy-Guided Self-Regulated Learning without Forgetting).

2. Related Work

2.1. Domain-Incremental Learning

(van de Ven & Tolias, 2019)(Hsu et al., 2019) first introduce the concept of Domain-Incremental Learning(DIL), in which there is no new class, but rather novel instances of classes from previous tasks. The objective of DIL is to learn a unified classifier that can classify all samples encountered so far. The core issue in DIL is how to deal with the shift in input space distribution, *e.g.*, changes in background, texture, pose, *etc.* For example, autonomous driving systems trained with the data collected in good weather conditions may encounter extreme weather conditions. DIL needs to learn from the data in extreme weather conditions while preserving the knowledge learned in good weather conditions. ER(Rolnick et al., 2019) shows that random selection of exemplars after training and random sampling of a set of memories during training with the new data can already achieve satisfying performances. Furthermore, RM(Bang et al., 2021) proposes to use diverse data augmentation techniques. MDR(Volpi et al., 2021) introduces the meta-update combined with rich data augmentation. However, DIL still assumes that well-defined task boundaries are known during

training. The proposed DS-IL setting extends DIL by enabling mixtures of data distributions over tasks and thereby enabling soft switching over tasks where task boundaries are not well-defined.

2.2. Class-Incremental Learning

Class-Incremental Learning (CIL) is currently the most studied CL scenario, where new classes occur sequentially. Similar to DIL, memory-based methods achieve the best performance. iCaRL(Rebuffi et al., 2017) first proposes a solution for CIL. It contains 1) an episodic memory selected by the herding rule; 2) a distillation loss of the feature's output; 3) a *nearest-mean-of-exemplars* classifier using the mean feature of the Episodic Memory to classify. PodNet(Douillard et al., 2020) proposes to distill not only the feature's outputs but also the intermediate layers' outputs. SDC(Yu et al., 2020) proposes to approximate the drift of the prototype's feature by using the current data. BiC(Wu et al., 2019) finds that the last fully connected layer has a strong bias towards the new classes and proposes to use a linear model to correct the bias. GEM(Lopez-Paz & Ranzato, 2017) proposes to use memory to limit the current gradient. In this paper, we propose a new memory-free approach that can also cooperate with memory-based methods.

2.3. Task Agnostic Continual Learning

Task Agnostic Continual Learning(Zeno et al., 2021) is the scenario in which task boundaries are unknown or not well-defined. It mainly creates the linear transition between two successive tasks by adjusting the proportion of the data of the two tasks. However, since the transition in a real application is not just linear between two tasks, it is still limited to a real application.

2.4. Knowledge Distillation

The notion of knowledge distillation was first proposed in (Hinton et al., 2015) for model compression. It uses the output of a large model to replace the original label for given data. This new label can help a small model learn faster. LWF(Li & Hoiem, 2018) first introduced this notion into CL. It considers the previously learned model as the teacher model and replays the memory and the teacher model's output. As aforementioned, many CL methods make use of knowledge distillation as a basic technique. PodNet(Douillard et al., 2020) distills the intermediate layer output as well. RRR(Ebrahimi et al., 2021) distills the attention map of the memory exemplar. UD(Kurmi et al., 2021) distills the previous model's uncertainty on the memory exemplar. In this paper, we propose to calculate distillation loss not from memory but from current data, based on the familiarity of the previous model with the current data using entropy in a self-regulated process. We will discuss this in detail in Section 4.

3. Distribution-Shift Incremental Learning (DS-IL)

For simplicity without loss of generality, let us consider only two tasks T_1 and T_2 with their data D_1 for T_1 : $\left\{x_i^1, y_i^1\right\}_{i\in 1:n_1}$ drawn from a joint distribution $P_1(x, y)$, and D_2 for T_2 : $\left\{x_i^2, y_i^2\right\}_{i\in 1:n_2}$ drawn from a joint distribution $P_2(x, y)$. Let l be the loss function (cross-entropy for classification). Let ϕ be the output of the model, and θ its parameters. Then, we can construct the objectives for T_1 and T_2, respectively:

$$T_1 : L_1 = \sum_{1}^{n_1} l(\phi(y|\theta, x_i^1), y_i^1)) \tag{1}$$

$$T_2 : L_2 = \sum_{1}^{n_2} l(\phi(y|\theta, x_i^2), y_i^2)) \tag{2}$$

Gradient descent, such as Stochastic Gradient Descent (SGD), can be used to update the parameters θ to minimize the objective function. If $P_1(x, y)$ equals $P_2(x, y)$, L_1 approximates L_2. Thus, when training only with D_2, L_1 will not increase, and the model will not forget T_1. However, if $P_1(x, y)$ is very different from $P_2(x, y)$, the gradient direction of L_1 may violate the gradient direction of L_2 when updating the parameters only with the gradients of L_2. Therefore, the objective function L_1 will increase, and the phenomenon of *catastrophic forgetting* will occur.

From a Bayesian perspective, joint distribution $P(x, y)$ can be decomposed as follows:

$$P(x, y) = P(x|y)P(y) \tag{3}$$

$$P(x, y) = P(y|x)P(x) \tag{4}$$

$P(x)$ and $P(y)$ denote the probability density of the input and output, respectively. $P(x|y)$ denotes the conditional probability of an output given an input. $P(x|y)$ denotes the conditional probability of an input given an output. For continual learning, the shift of these four terms is primarily involved, *e.g.*, $P(x)$ mainly changes in Domain-Incremental Learning (DIL), whereas $P(y)$ mainly changes in Class-Incremental Learning (CIL). We can create different scenarios by adjusting the proportion of these four terms.

From this data distribution shift perspective, it is easy to interpret DIL or CIL as DS-IL and extend them for Smooth-DIL or CIL when task transitions are blurred and data distributions mixed up. Thus, DS-IL is a continual learning

scenario in which models are to learn knowledge from the successive shifting of $P(x)$, $P(y)$, $P(x|y)$, and $P(y|x)$ over tasks.

3.1. From CIL to DS-IL

For a classic CIL, each new task only contains a few new classes. Thus, the density probability of the previous classes equals zero, and some new output variables y appear. The CIL setting can be considered to be an extreme case of the DS-IL setting where data labels are only available for the novel classes of the second task T_2. Thus, $P(y)$ changes rapidly. By smoothing the change of $P(y)$, we can increase the similarity between tasks and generate a smoother and more flexible version of CIL, as shown in Figure 2.

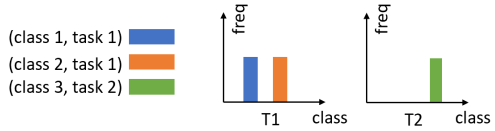

Figure 1. Illustration of TIL in which the data have both class id and task id. Moreover, task id is provided for both training and testing.

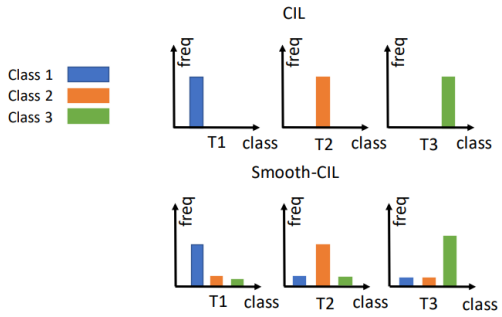

Figure 2. Illustration of CIL and Smooth-CIL using three tasks with three classes. In the original CIL setting, each task only contains instances of one class. The Smooth-CIL breaks this limit, and each task may have multiple classes. Thus, the shift of $P(y)$ in our scenario is smoother.

3.2. From DIL to DS-IL

The classic DIL defines each domain as a single task with well-defined task boundaries. This means that the domain shift is discontinuous. Thus, we can quickly transform a DIL into a CIL by labeling the data with domain and class labels as shown in Figure 3, thereby interpreting DIL as DS-IL. However, in many real-life applications, there are no clear and well-defined task boundaries with evident domain labels. Task transitions can be fuzzy, and data distributions of different tasks can overlap and mix up, resulting in a Smooth-DIL as shown in Figure 3, which is enabled by DS-IL.

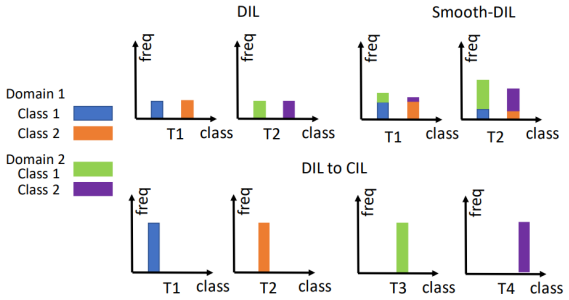

Figure 3. Assume there are two domains, and each domain contains the same class labels. The classic DIL considers each domain as a single task. In Smooth-DIL, each task can contain two domains, but the distribution of different domains is different for each task. Therefore, from DIL to CIL, we can label data by using both the domain label and the class label. In this figure, there are four different labels, corresponding to four tasks.

4. Entropy-Guided Self-Regulated Learning without Forgetting

To ensure simplicity without loss of generality, we consider the same hypotheses as in Section 3. There are only two tasks. From Equation (1) and Equation (2), we can derive the joint objective for both task 1 and task 2 as:

$$L_{total} = L_1 + L_2 \tag{5}$$

$$= \sum_{D_1} l(\phi(y|\theta, x_i^1), y_i^1)) + \sum_{D_2} l(\phi(y|\theta, x_i^2), y_i^2)) \tag{6}$$

When training on T_2, if we could store all the data from T_1, then we can construct the objective L_1. However, for CL, we can only use a small part or none of D_1 (dataset for T_1). Thus, the core problem is how to approximate L_1.

4.1. LWF (Learning without Forgetting)

LWF(Li & Hoiem, 2018) proposes to use the previous model to generate a soft label and replaces the cross-entropy loss with a distillation loss. In this work, to fit our scenario, we only use the concept of knowledge distillation to represent LWF. θ^* represents the parameters learned from T_1; l_{lwf} corresponds to the distillation loss; M is the memory set which is from D_1 and D_2. Then, the new objective is as follows:

$$\hat{L_{total}} = \hat{L_1} + L_2 \tag{7}$$

$$\hat{L_1} = \sum_{M} l_{lwf}(\phi(y|\theta, x_i^1), \phi(y|\theta^*, x_i^1))) \tag{8}$$

LWF uses distillation loss to approximate the previous objective function L_1.

4.2. The proposed ER-LWF approach

As we discussed in Section 3, the distribution shift between tasks is not always rigid and discontinuous. Thus, some current data may possibly be similar to previous data. Here, we use $D_1 \cap D_2$ to represent the set of similar data. It then becomes crucial to find this intersection without access to D_1. We propose to use the output of the previously learned model to verify if the data are similar. Let EI denote the entropy. Then, EI for given data is defined as follows:

$$EI(x, y) = EI(P(y|\theta^*, x)) = EI(softmax(\phi(y|\theta^*, x))) \tag{9}$$

The closer EI is to zero, the more familiar and confident the model is about the input data, and the more corresponding data is valuable for reconstructing the L_1. On the contrary, if EI is large, output distribution is close to discrete uniform distribution. This means that the model hardly classifies the given input data.

Rather than looking for similar data in $D_1 \cap D_2$, we propose to use EI to modulate distillation loss. Given an N-class classifier, the upper-bound of EI over its outputs corresponds to discrete uniform distribution:

$$EI_{upper} = -\sum_{1}^{N}(\frac{1}{N} log(\frac{1}{N})) = log(N) \tag{10}$$

$$0 \leqslant EI(x, y) \leqslant log(N) \tag{11}$$

Thus, we can define our $Entropy - Guided$ weight (EG-weight) for given data as follows:

$$w_{eg}(x, y) = 1 - \frac{EI(x, y)}{log(N)} \tag{12}$$

Then, the previous objective function could be approximated as:

$$\hat{L_1} = \sum_{D_2} w_{eg}(x_i^2, y_i^2) l_{lwf}(\phi(y|\theta, x_i^2), \phi(y|\theta^*, x_i^2))) \tag{13}$$

From Equation (13), we only use the current data D_2 and our EG-weight to approximate the previous loss function. The benefits of this formulation are threefold:

- Memory is not necessary;
- $w_{eg}(x, y)$ can verify if a model is familiar with data from D_2 or not. If the model is familiar with data, then $w_{eg}(x, y)$ will be close to 1, else it will be close to 0;
- It can be used for online updates.

The corresponding algorithm, ER-LWF, is detailed in Algorithm 1.

Algorithm 1 ER-LWF for continual learning

```
Train(θ, D, n, T)
D: # dataset; θ: # model parameters;
T: # of tasks; N: # of batches
for t = 1:T do
  for n = 1:N do
    sample data (x_n, y_n) from D_t;
    # CE: Cross-Entropy loss;
    l_current = CE(φ(y|x_n, θ), y_n);
    if t >1 then
      # KD: knowledge distillation loss;
      # θ*: previous model parameters;
      l_lwf = KD(φ(y|x_n, θ), φ(y|x_n, θ*))
      # w_eg(x_n, y_n) : EG-weight;
      l_ER−LWF = w_eg(x_n, y_n) * l_lwf
    else
      l_ER−LWF = 0
    end if
    l_total = l_current + l_ER−LWF
    Update(l_total, θ)
  end for
end for
```

5. Experiments

In this section, we first present our experimental protocols under DS-IL in Section 5.1 and then introduce the methods we use in Section 5.2. Finally, the results are discussed in Section 5.3.

5.1. Experimental protocols

As we discussed in Section 3, we can generate a Smooth-CIL and a Smooth-DIL for classic CIL and DIL, respectively. In this section, we continue this idea and conduct experiments on two datasets: PACS and CIFAR-100.

PACS is an image dataset originally for data generalization(Li et al., 2017). It consists of four domains: photo, art painting, cartoon, and sketch. Each domain contains seven classes. For simplicity, each image is resized to 64×64 for all scenarios, and 20% of each domain is used to construct our test set. We repeated each experiment three times.

First, for DIL, the dataset is divided into four tasks. Each task has one domain. The learning order arbitrarily chosen is: $art - painting \rightarrow cartoon \rightarrow photo \rightarrow sketch$.

We then generate six tasks containing all seven categories for Smooth-DIL. However, the four domains are mixed up with different proportions over the six tasks, and also the domain ratio is randomly generated. To compensate for this randomness, we generate three experiments with varying domain ratios.

In our CIL under the DS-IL perspective, each task contains all the categories. However, there is only one dominant class, and others have few exemplars. PACS has seven classes. Thus, we consider seven tasks, where each task contains only one dominant class.

Similar to Smooth-DIL, Smooth-CIL has six tasks for which the category ratio changes. The category ratio is randomly generated. We generate three experiments with varying proportions of the class.

CIFAR-100 (Krizhevsky, 2009) has 20 superclasses, where each superclass contains five classes. Each class comprises 600 images, of which 500 are used for training and 100 for testing. To use the dataset in our experiments, we need to create a notion of the domain in CIFAR-100. For example, there is a superclass *fish* that corresponds to five classes: *aquarium fish*, *flatfish*, *ray*, *shark*, *trout*. Each can be seen as a domain of *fish*. Thus, we use the superclass label as our new class label and the original class label as the domain label. Finally, the original CIFAR-100 is transformed into a dataset containing 20 classes and five domains. For simplicity, we keep the same training set and test set as the original version of CIFAR-100(Krizhevsky, 2009). We use the original image size that is 32*32*3. Each experiment is repeated three times.

Then, the rest is similar to the PACS dataset. For DIL, each superclass has five domains (subclasses). Thus, we generate five tasks for DIL. Then, we generate eight tasks containing different domain distributions for Smooth-DIL. We created ten tasks for CIL, where each task has two dominant classes. Finally, for Smooth-CIL, we produced eight tasks containing different class label distributions. We provide the details in the supplementary material to ensure completeness.

5.2. Training Methods

We use the standard PyTorch(Paszke et al., 2019) implementation of ResNet-18(He et al., 2016) in both protocols. Because our method does not use memory, we mainly compare our approach to other regularization-based methods (online-EWC, SI, LWF without memory). We also test ER and LWF as the baselines of memory-based methods. FOO-VB(Zeno et al., 2021), which targets the task agnostic CL scenario, is also tested for comparison.

We use **Cumulative** and **Fine-tuning** as the upper bound and lower bound, respectively. **Cumulative** means that all of the data seen so far are available for each training. **Fine-tuning** means that the previous model parameters are used as the initial parameters for the next task. Furthermore, the memory budget is set to 400 for PACS and to 1000 for CIFAR-100.

To ensure a fair comparison, we turn off the data augmentation for all experiments. We rely on the Adam optimizer(Kingma & Ba, 2015), which is re-initialized for each task. All hyperparameters are selected by a grid search on the validation set. Our network is trained on a single RTX 3070 8GB GPU and an i7-10700 8-core CPU.

5.3. Results

5.3.1. PACS RESULTS

Table 1 and Figure 4 show the results on the PACS dataset(Li et al., 2017). The results on the Smooth-CIL and Smooth-DIL show that our ER-LWF outperforms all other methods, including even the memory-based ones. In the smooth version, the similarities between different tasks are magnified. Thus, current data are more likely to reconstruct the previous loss function. That is why our method performs well and can even reach the upper bound.

From the DIL column in Table 1, we find that all regularization-based methods fail in this scenario, and that their performance is close to the lower bound (Fine-tuning), including our ER-LWF approach. In contrast, memory-based approaches perform better.

For the results on the PACS CIL, our ER-LWF performs best compared with other regularization-based methods, and its performance is very close to that of LWF with memory. However, there is still a big gap between the upper bound and other methods. This means that only replaying the previous data is not enough, and that the loss function or the gradient requires more constraints.

To conclude, our method does not work on DIL but performs well on the others. Furthermore, it consistently performs better than the LWF without memory.

5.3.2. CIFAR-100 RESULTS

The results on the CIFAR-100 dataset are presented in Table 2 and Figure 5. As for Smooth-DIL, CIL, and Smooth-CIL, similar phenomena to the PACS experiments can be observed. Our proposed ER-LWF outperforms all other regularization-based methods, and its performance is even better than memory-based methods for smooth scenarios.

Table 1. Results on DIL, Smooth-DIL, CIL, and Smooth for the PACS dataset(Li et al., 2017). **Acc** corresponds to Accuracy, while **BF** stands for Backward Transfer defined in (Lopez-Paz & Ranzato, 2017).For both of them, the bigger the better. The budget of memory-based methods is set at 400 images in total. The value here is the percentage and is the average accuracy of the model over all experiments after training on all tasks. We display the best without-memory method in **bold** font. Methods with an asterisk * use memory.

Methods	DIL		Smooth-DIL		CIL		Smooth-CIL	
	Acc(%)	BF(%)	Acc(%)	BF(%)	Acc(%)	BF(%)	Acc(%)	BF(%)
Cumulative(upper)	54.2	0.8	59.5	6.3	58	14.7	58.91	7.8
Fine-tuning (lower)	30	−38.9	54.2	4.5	30	4.7	50.6	2.4
Online-EWC(Chaudhry et al., 2018)	29.6	−39.7	54.8	4.6	30.7	5.4	50.8	2.3
SI(Zenke et al., 2017)	**31.6**	−38.5	55.8	**5.6**	30	4.3	51.1	2.7
FOO-VB(Zeno et al., 2021)	29.6	−21.3	58.5	5.3	38.5	7.5	55.4	3.8
LWF(Li & Hoiem, 2018)	27.7	**−26.8**	57.4	5.5	29.5	8.2	55	5.4
ER*(Rolnick et al., 2019)	44.3	−12.1	54.4	4.1	37.8	4.1	52	2.8
LWF(memory)*(Li & Hoiem, 2018)	47.7	−7.1	57.9	5.9	39.8	4.2	56	5.3
ER-LWF (ours)	30.2	−32.5	**58.9**	**5.6**	**38.8**	**10.7**	**56.1**	**5.5**

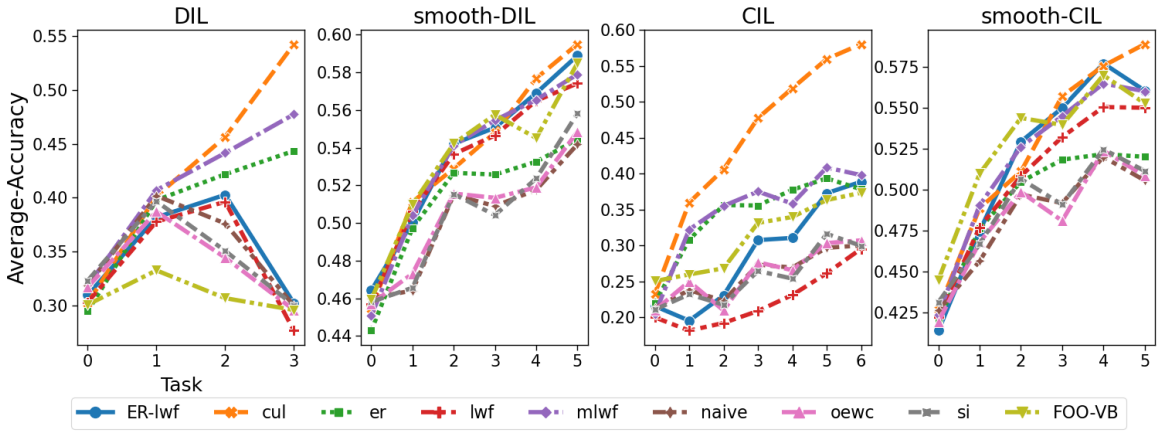

Figure 4. Results related to the PACS dataset(Li et al., 2017). Testing accuracy (average on three runs) is provided after training on each task for different methods and different scenarios.

For the DIL setting with CIFAR-100, we do not observe the same result as for PACS. The LWF family's methods all perform well. This can be explained by the way we define the DIL scenario with CIFAR-100. As discussed in Section 5.1, CIFAR-100 has 20 superclasses, and each superclass contains five subclasses. In the proposed DIL protocol, each subclass is considered to be a domain, thereby leading to slight differences between tasks. As a result, we can reconstruct the previous objective more efficiently compared with the PACS DIL.

We also find that FOO-VB(Zeno et al., 2021), which initially targets the task agnostic CL scenario (an extremely smooth scenario), is close to our method for the Smooth-DIL and Smooth-CIL scenarios. However, our method performs much better when data distribution shifts more rapidly, as shown in the DIL and CIL plots of Figure 5.

Compared to LWF, we designed EG-weight Equation (12) to self-regulate the original LWF loss. This can give more weight to similar data and less weight to unrelated data. Thus, the model can have more plasticity when the current data are different and more stability when the current data are similar. This is confirmed by the results obtained on the CIFAR-100 and PACS, from which we can observe that our method consistently outperforms or matches the performance of LWF without memory.

6. Limitations

In this work, to better fit real-life applications, we make an assumption that task boundaries are blurred, in other words their data distributions overlap or at least have a certain degree of similarity. Smooth-CL setting, *i.e.*, Smooth-DIL and CIL, and regular CL setting, *i.e.* DIL and CIL, correspond to high-level and low-level similarity, respectively. However, if the task boundaries are well-defined with no overlapping of domains, then our method may fail since it makes use of the current data and the previous model output

Table 2. Results on the CIFAR-100 dataset(Krizhevsky, 2009). The budget for memory-based methods is set at 1000. It uses the same setting as Table 1.

Methods	DIL		Smooth-DIL		CIL		Smooth-CIL	
	Acc(%)	BF(%)	Acc(%)	BF(%)	Acc(%)	BF(%)	Acc(%)	BF(%)
Cumulative(upper)	38.8	−1.4	41.7	8.8	42.9	14.5	41.4	9.2
Fine-tuning (lower)	28.4	−26.5	35.3	4.7	15.5	1.7	33.1	3.7
Online-EWC(Chaudhry et al., 2018)	30.3	−30.7	36.1	4.6	15.6	2.2	34.4	4.2
SI(Zenke et al., 2017)	31.0	−30.5	37.7	6.7	15.5	1.4	33.1	3.2
FOO-VB(Zeno et al., 2021)	31.4	−29.9	**41.4**	7.5	18.5	3.4	40.2	6.2
LWF(Li & Hoiem, 2018)	36.9	−19.6	39.8	5.9	23.9	5.9	38	5.2
ER*(Rolnick et al., 2019)	31.3	−23.0	36.2	4.4	24.1	3.4	36.2	5.5
LWF(memory)*(Li & Hoiem, 2018)	36.7	−4.8	39.8	5.9	26.0	7.5	38.3	6.9
ER-LWF (ours)	**37.0**	**−12.6**	**41.4**	**7.9**	**25.8**	**8.2**	**40.3**	**7.9**

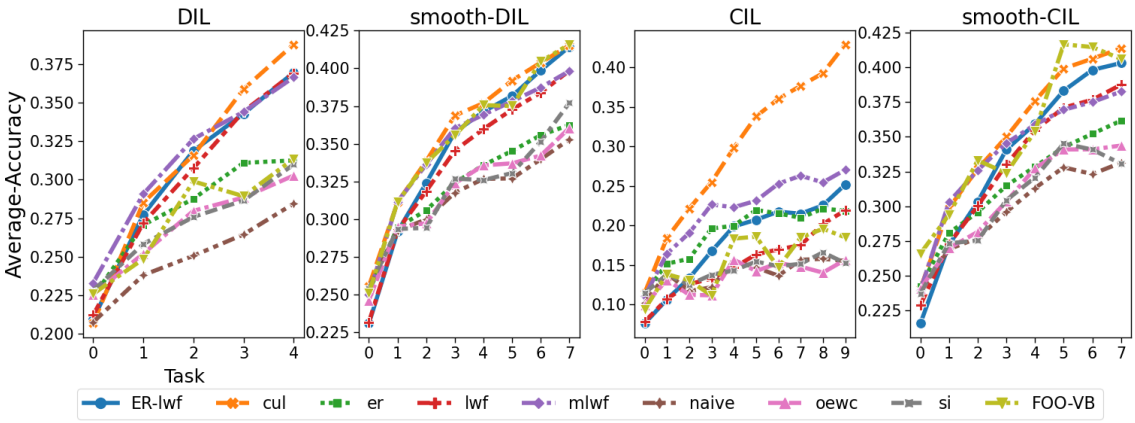

Figure 5. Results related to the CIFAR-100 dataset(Krizhevsky, 2009).Testing accuracy (average over three runs) is provided after training on each task for different methods and different scenarios.

to approximate the previous loss function. If the previous model cannot recognize or provide a meaningful output, the EG-weight Equation (12) will be close to zero. Then, our method degenerates to **Fine-tuning**. In our experiments, the most difficult CL scenario is the DIL setting on the PACS dataset. Since we assign each domain to a single task in DIL, the similarity between the data of different domains may be very slight when the gap between different domains is too large. In this situation, we cannot find similar data from the current task, meaning that the previous loss function cannot be reconstructed. From the DIL plot of Figure 4, we can see that model performance degenerates rapidly in the final task. In our experiments, the last task of the DIL corresponds to *sketch*, which is the domain that is most different. We think that this is what accounted for this poor performance. Figure 4.

7. Conclusion

In this paper, we propose a new continual learning paradigm, namely Distribution-Shift Incremental Learning (DS-IL), which, by considering soft task boundaries as encountered in real-world applications, subsumes traditional Domain-Incremental Learning (DIL) and Class-Incremental Learning (CIL) scenarios. Furthermore, we propose ER-LWF, a novel CL method which extends LWF to deal with CL under the DS-IL setting where there are no well-defined task boundaries. It uses the entropy information of the previously learned model's output on the current data to self-regulate the original knowledge distillation loss. This EG-weight can provide the model with more stability when the current data are similar to the previous data, or with more plasticity otherwise. Based on the results of the PACS and CIFAR-100 datasets, we show that our proposed CL method, without memory, consistently outperforms the classic LWF (without memory) and can reach the upper bound for the Smooth-CL scenarios.

In this work, only the previously learned model's output is used when learning a new task. However, there is still a lot of information stored in the weights of the previous model parameters. Therefore, learning how to use this information to overcome forgetting is our next research direction.

References

Bang, J., Kim, H., Yoo, Y., Ha, J.-W., and Choi, J. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8218–8227, June 2021.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. S. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision – ECCV 2018*, pp. 556–572, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01252-6.

Douillard, A., Cord, M., Ollion, C., Robert, T., and Valle, E. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, 2020.

Ebrahimi, S., Petryk, S., Gokul, A., Gan, W., Gonzalez, J. E., Rohrbach, M., and trevor darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tHgJoMfy6nI.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *NIPS*, abs/1503.02531, 2015.

Hsu, Y.-C., Liu, Y.-C., Ramasamy, A., and Kira, Z. Re-evaluating continual learning scenarios: A categorization and case for strong baselines, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, 2017. ISSN 0027-8424. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/content/114/13/3521.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009.

Kurmi, V. K., Patro, B. N., Subramanian, V. K., and Namboodiri, V. P. Do not forget to attend to uncertainty while mitigating catastrophic forgetting. *WACV*, 2021.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, D., Yang, Y., Song, Y., and Hospedales, T. M. Deeper, broader and artier domain generalization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5543–5551, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society. doi: 10.1109/ICCV.2017.591. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.591.

Li, Z. and Hoiem, D. Learning without forgetting. *PAMI*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081.

Lomonaco, V. and Maltoni, D. Core50: a new dataset and benchmark for continuous object recognition. *CoRR*, abs/1705.03550, 2017. URL http://arxiv.org/abs/1705.03550.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. In *NIPS*, 2017.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. *CVPR*, pp. 5533–5542, 2017.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., and Wayne, G. Experience replay for continual learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Paper.pdf.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *CoRR*, abs/1606.04671, 2016. URL http://arxiv.org/abs/1606.04671.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In *NIPS*, 2017.

Singh, P., Mazumder, P., Rai, P., and Namboodiri, V. P. Rectification-based knowledge retention for continual learning. *CVPR*, abs/2103.16597, 2021. URL https://arxiv.org/abs/2103.16597.

van de Ven, G. M. and Tolias, A. S. Three scenarios for continual learning. *CoRR*, abs/1904.07734, 2019. URL http://arxiv.org/abs/1904.07734.

Verma, V. K., Liang, K. J., Mehta, N., Rai, P., and Carin, L. Efficient feature transformations for discriminative and generative continual learning. *CVPR*, abs/2103.13558, 2021. URL https://arxiv.org/abs/2103.13558.

Volpi, R., Larlus, D., and Rogez, G. Continual adaptation of visual representations via domain randomization and meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

Yin, H., Molchanov, P., Alvarez, J. M., Li, Z., Mallya, A., Hoiem, D., Jha, N. K., and Kautz, J. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *The IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., mei Cheng, Y., Jui, S., and van de Weijer, J. Semantic drift compensation for class-incremental learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6980–6989, 2020.

Zenke, F., Poole, B., and Ganguli, S. Continual learning through synaptic intelligence. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/zenke17a.html.

Zeno, C., Golan, I., Hoffer, E., and Soudry, D. Task-Agnostic Continual Learning Using Online Variational Bayes With Fixed-Point Updates. *Neural Computation*, 33(11):3139–3177, 10 2021. ISSN 0899-7667. doi: 10.1162/neco_a_01430. URL https://doi.org/10.1162/neco_a_01430.