ELASTIC MEAN-TEACHER DISTILLATION MITIGATES THE CONTINUAL LEARNING STABILITY GAP

Anonymous authors
Paper under double-blind review

ABSTRACT

Nowadays, neural networks are being used to solve a variety of tasks. They are very effective when trained on large datasets. However, in continual learning, they are trained on non-stationary stream of data, which often results in forgetting of the previous knowledge. In the literature, continual learning models are exposed to a sequence of tasks, and must learn each task one by one. They are then evaluated at the end of each learning session. This allows to measure the average accuracy over all tasks encountered so far. Recently De Lange et al. (2022) showed that continual learning methods suffer from the Stability Gap, encountered when evaluating the model continually. Even when the performance at the end of training is high, the worst-case performance is low, which could be a problem in applications where the learner needs to always perform greatly on all tasks while learning the new task. In this paper, we propose to apply a refined variant of knowledge distillation, adapted to the class-incremental learning setting, and used in combination with replay, to improve the stability of the continual learning algorithms. We also propose to use a distillation method derived from the Mean teacher distillation training paradigm introduced in semi-supervised learning. We demonstrate empirically that the use of this method enhances the stability in the more challenging setting of online continual learning.

1 INTRODUCTION

It is more and more common to use neural networks in order to solve a variety of tasks. In fact, learning neural networks with backpropagation has proven capable of good generalization properties even when using overparametrized networks (Krizhevsky et al., 2017). However, these good learning properties only occur when the data is provided in an independant and identically distributed manner. When learning on a stream whose distribution varies over time, neural networks suffer from *catastrophic forgetting* (McCloskey & Cohen, 1989; Goodfellow et al., 2014; Kirkpatrick et al., 2017b), they tend to forget knowledge acquired from learning experiences. This is problematic for instance when learning large scale dataset that are acquired from continuously evolving streams of data, since it is very expensive to store all of the data and learn from scratch when we want to incorporate new knowledge into the neural network. This is why the field of *continual learning* is trying to tackle these issues.

A variety of benchmarks have been introduced in continual learning in order to evaluate several aspects of the continual learning agent. *Task-incremental learning* (De Lange et al., 2021; van de Ven & Tolias, 2018), and *Class-incremental learning* (Masana et al., 2020; Belouadah et al., 2021) are two of the most popular continual learning settings. Both of these settings as well as most others separate the learning into distinct tasks that are encountered sequentially by the agent. The goal of the agent is then to accumulate knowledge from each of the new tasks without forgetting the previous ones, under the constraint that it should store only a limited amount of past data. After learning a task, the agent is then evaluated on all previous tasks to determine how much it has forgotten of it.

This is the classical manner of evaluating the agent. However, another way of evaluating coined *continual evaluation* (De Lange et al., 2022) or *anytime inference* (Koh et al., 2022) has been experimented with. When evaluating continually, we want the agent to perform correctly not only at task boundaries, but also at any moment in time whenever learning a task. In (De Lange et al., 2022), the authors noticed that it is often the case that, whenever learning a new task, the performance on pre-

vious tasks decreases drastically before going back to normal. This behaviour is problematic, since for many real-world applications, the agent must be applied for inference while it is learning. In this article, we propose a method to improve the stability of continual learning algorithms, evaluate it using the stability metrics from De Lange et al. (2022), and drastically reduce the stability gap.

Our contributions are the following:

- We identify the task gradient unbalance as probable causes for the stability gap, and find that this imbalance could be due to the growth of the logits norm over training, resulting in bad behaviour of the cross-entropy loss used in classification.
- To solve the latter problem, we propose a knowledge distillation process with special care given to the handling of new class logits, that we name Elastic Distillation. We show empirically that this process, combined with replay, greatly improves the stability of continual learning algorithms. Importantly, our results also hint that the traumatic event of the stability gap can lead to overall performance drop, and that addressing it leads to improved final performance for continual learners.
- We show that Elastic Distillation is applicable to the boundary-free setting when used in the Mean-Teacher distillation scheme introduced in the semi-supervised learning literature, where it outperforms existing methods in that setting, in terms of stability and accuracy.

2 Continual learning and The Stability Gap

In continual classification, the learning agent must learn the parameters $\theta \in \Theta$ of a function $f : (\mathcal{X}, \Theta) \mapsto \mathcal{Y}$ from the image input space $\mathcal{X}$ to the label space $\mathcal{Y}$. It must do so by seeing a stream of data $\mathcal{S} = \{(x^1, y^1), (x^2, y^2), ...(x^n, y^n)\}$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Each data tuple is drawn from a time varying distribution $(x^t, y^t) \sim \mathcal{D}_t$. In classical machine learning the training data distribution does not depend on time, but this is added as a constraint in continual learning. In both cases the goal of the agent is to perform well on new samples drawn from the joint distribution $\mathcal{D}$, which is marginalized over time.

In offline training, the agent's experience is often separated into a training phase, during which it learns the parameters θ by seeing data samples drawn from $\mathcal{D}$, followed by a testing phase, where the agent is evaluated in order to determine how well it has matched the target distribution. In continual learning, however, the training distribution is not entirely accessible from the start, but it is time varying. In practice, continual learning is simplified to allow for easier analysis by studying distributions that come from a discrete set and switch from one distribution to another. Each of such time periods during which the distribution does not change is referred to as a task. Evaluation is then performed at the end of learning each task on a held out test set that is gathered over the course of training whenever encountering a new distribution. To summarize, where offline machine learning exposes the agent to a fixed distribution over time, and a final evaluation stage, most continual learning settings expose the agent to discretely varying distributions, and sparse evaluation.

While all of the above simplifications make sense, they are still far from what the human learning experience is like, and from fitting the requirements of many real-world applications. In comparison to the above, humans experience continuously time-varying distributions and continual evaluation. In order to remedy to one of these points, De Lange et al. (2022) lay the basis and encouraged the study of continual evaluation of neural networks. In continual evaluation, the model is continuously evaluated during, instead of after each task. Indeed, they noticed that the performance on previous tasks often drops at task shift before coming back to a higher value later in training, this is what they refer to as the stability gap.

2.1 Stability Metrics

In this section, we present two of the metrics proposed in De Lange et al. (2022), that we use in this article to assess the stability of the algorithms. We denote $\mathbf{A}(E_i, f_t)$ as the accuracy of f_t (model at iteration t), on the evaluation task E_i. The minimum accuracy for task k, $\min\text{-ACC}_{\mathrm{T_k}}$ (see Eq. 1), records the minimum accuracy for the task k while learning subsequent tasks. It gives a good idea of the worst case performance of the agent on a given task. The worst-case accuracy, $\text{WC-ACC}_{\mathrm{t}}$ (see

Eq. 2), combines information from the minimum accuracy on previous tasks and the accuracy on the current task. WC-$\mathrm{ACC_t}$ can be seen as a trade-off between stability and plasticity, which gives equal importance to each task.

$$\text{min-ACC}_{\mathrm{T_k}} = \frac{1}{k-1}\sum_{i}^{k-1} \min_{n} \mathbf{A}(E_i, f_n), \forall |T_{i-1}| < n \leq t \tag{1}$$

$$\text{WC-ACC}_{\mathrm{t}} = \frac{1}{k} A(E_k, f_t) + (1 - \frac{1}{k}) \text{ min-ACC}_{\mathrm{T_k}} \tag{2}$$

2.2 Online Continual Learning and Boundary free Continual learning

Popular continual learning scenarios often assume that data arrive in large batches of i.i.d data, with sharp distribution shifts happening whenever a new batch becomes available. We call this setting boundary-aware continual learning, due to the additional information provided by the data aggregation. Task-free continual learning (Aljundi et al., 2019b; Rao et al., 2019; Lee et al., 2020; De Lange & Tuytelaars, 2021) removes this form of supervision by assuming a stream of small mini-batches. This has as an effect that the granularity of the data distribution can be refined. In practice, it is even possible to keep meta-tasks that define the granularity of the distribution, while letting the agent assume this distribution could change at any new mini-batch, as done in (Aljundi et al., 2019a; Hu et al., 2021). In this article, we explore this last setting which is particularly challenging for classical continual learning methods. In particular, any method that needs to be aware of the change in data distribution, like LwF (Li & Hoiem, 2017) or EWC (Kirkpatrick et al., 2017a), would suffer in such a setting.

Continual evaluation fits naturally with online continual learning, since this setting assumes the data arrives as a stream of separate mini-batches. In a real world application, we could update the model using each new bit of available data, and then use it for inference while waiting for new data to train on. In that case it is also important that the model exhibits stable training, which we will try to achieve in this article.

3 Understanding the Stability Gap

To understand the root causes behind the stability gap, we propose to study the learning dynamics immediately after a task shift. This is the crucial moment where the worst-case accuracy suddenly drops, resulting in a drastic performance loss. After a few iterations, many continual learning methods recover the original performance (or close to it) (see Fig. 3), which indicates that the stability gap is happening due to some fundamental qualitative changes in the learning dynamics soon after the task shift. A fundamental quantity to study is the gradient of the loss $\nabla\mathcal{L}$ and its norm $||\nabla\mathcal{L}||$.

We split the gradient into two components, as done in De Lange et al. (2022)

$$\nabla\mathcal{L} = \alpha\nabla\mathcal{L}_{stability} + (1-\alpha)\nabla\mathcal{L}_{plasticity}, \tag{3}$$

where α controls the ratio between stability and plasticity. Each of these component is dependent on the specific continual learning method. $\nabla\mathcal{L}_{stability}$ is the stability gradient, which comes from the part of the loss that tries to prevent forgetting of previous tasks, while ($\nabla\mathcal{L}_{plasticity}$) is the plasticity gradient, which comes from part of the loss that tries to learn the task at hand. For example, in experience replay $\nabla\mathcal{L}_{stability}$ is the gradient of the cross-entropy loss on previous task samples, while $\nabla\mathcal{L}_{plasticity}$ is the gradient on new task samples.

For the most popular continual learning methods, $\mathcal{L}_{stability}$ tend to be well-behaved during training. Typically, high values of $\mathcal{L}_{stability}$, for example due to excessively high regularization coefficients, will inhibit learning, while low values result in catastrophic forgetting. Either way, $\mathcal{L}_{stability}$ should not cause the drop in performance shown by the plots in Figure 1 and 3, because if the regularization coefficient is too small the stability gap would not be recovered in the later epochs, as we see in the plot. Therefore, we focus our attention on $\nabla\mathcal{L}_{plasticity}$.

After a task shift, $\nabla\mathcal{L}_{plasticity}$ tend to be very large, while $\nabla\mathcal{L}_{stability}$ tends to be small. The unbalance between $\nabla\mathcal{L}_{plasticity}$ and $\nabla\mathcal{L}_{stability}$ at the start of a new task is problematic because

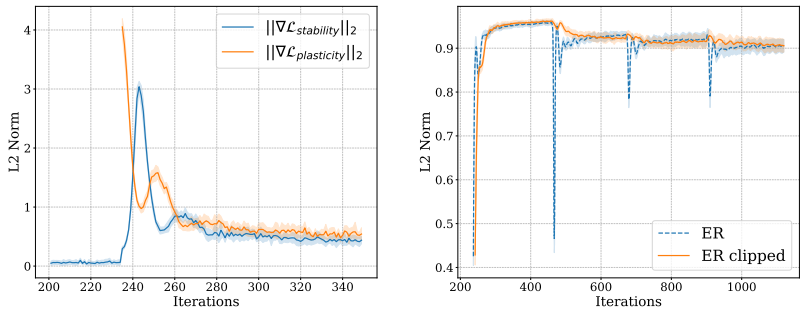

Figure 1: (Left) Close look at the task-specific gradient norm behaviour at task shift between the two first tasks of Split-Mnist. (Right) Task 2 validation accuracy on Split-Mnist (5 tasks) for both experience replay, and replay for which last task gradient is clipped to a low value. Mean and Standard deviation are computed over 10 seeds

it inhibits the effect of $\mathcal{L}_{stability}$ during the first phases of learning. We illustrate the difference between old and new task gradient in Figure 1. We see that at task shift, Task 2 gradient starts with a higher magnitude than Task 1 gradient. This generates a response in Task 1 gradient's magnitude, which in turn gets large.

We verify experimentally whether $\nabla\mathcal{L}_{plasticity}$ is one of the root cause of the stability gap. To balance the two gradient contributions, we clip the norm of $\nabla\mathcal{L}_{plasticity}$ to a low value, high enough to allow some adaptation to the new task but low enough to prevent a large stability gap and to obtain a gradient norm comparable $\nabla\mathcal{L}_{stability}$. We test this simple mitigation on MNIST. The results in 1 indicate that gradient clipping solves the stability gap since the accuracy (orange curve) does not show the drastic drops after a task shift but remains stable throughout training. However, gradient clipping was not found to be sufficient on more complex datasets than MNIST, and we therefore develop several more principled techniques in the following sections.

In the remainder of this section, we show that controlling $\nabla\mathcal{L}_{plasticity}$ in class-incremental scenarios requires regularizing the logits of the units to prevent unbounded growth (Section 3.1) and we show why popular CL methods have a large stability gap (Section 3.2).

3.1 Role of Logits and Recency Bias

Empirically, it is well known that most continual learning methods in class-incremental setting tend to be biased towards recent classes (Rebuffi et al., 2017; Wu et al., 2019; Hou et al., 2019). In practice, the bias is the result of logits for new classes becoming larger on average than logits of past classes. There are two main possibilities that give rise to this property: the average logits for new classes become larger and larger over time, keeping the previous one mostly on the same scale; or it might occur that new logits are on the same scale as previous ones but old logits are pushed towards negative values. Figure 2 shows an histogram of the activations for Replay, Replay and LwF, and Elastic Distillation (our method). Standard methods like Replay and LwF have a strong recency bias, which gets worse over time.

In the previous section, we proposed the new gradient norm $\nabla\mathcal{L}_{plasticity}$ as the main cause of the stability gap. We will investigate now its role in a typical class-incremental setting. Typically, deep neural networks for classification are trained via a cross-entropy loss computed after a softmax computed over the logits. Due to the softmax operation, in order to increase the probability for new units, the new logits must increase while the old logits must decrease. Without any form of mitigation, this interaction can quickly result in catastrophic forgetting.

At the beginning of training, high logits value for old classes will result in small probabilities y^{new} for new classes, close to zero, resulting in a high gradient that will quickly increase the new logits and decrease the old ones. This is a source of instability which results in a high $\nabla\mathcal{L}_{plasticity}$. After some iterations, the probability y^i will grow and the gradient will become smaller. This is shown

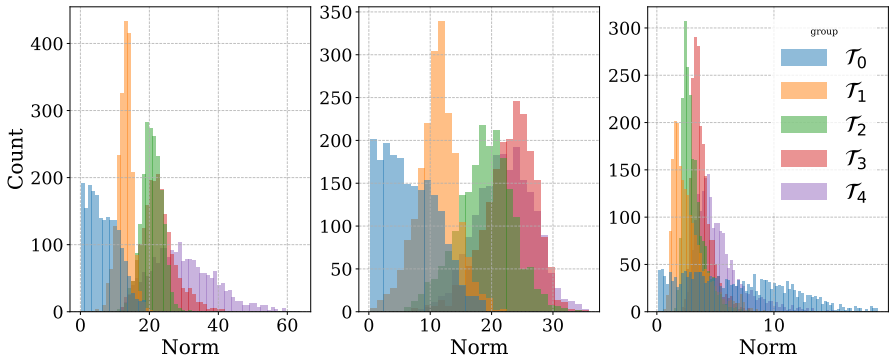

Figure 2: CIFAR10 (5 splits), Histogram of the logits norm of test set images from current task after training each task. Three methods are compared, ER (Left), ER + LwF (Middle) and ER + ED (Right)

in Figure 1, which clearly shows that the instability is fixed after around 50 iterations after the task shift (iteration 300).

Therefore, in order to mitigate the stability gap, a continual learning method must limit the growth of the logits during training. In particular, trying to control the gradient norm becomes more difficult if the old units have large logits.

3.2 Stability gap in Replay and LwF

Replay methods (Chaudhry et al., 2019; Rebuffi et al., 2017; Castro et al., 2018) like experience replay (ER) mitigate forgetting by storing a buffer of samples from previous tasks and rehearsing over time. In a replay method, $\mathcal{L}_{stability}$ is the loss computed over the samples in the buffer, while $\mathcal{L}_{plasticity}$ is the loss for new samples. At the start of a new task, the model will have a high accuracy for buffer samples, making $\mathcal{L}_{stability}$ close to zero, which means that the total loss is dominated by $\mathcal{L}_{plasticity}$. The magnitude of the gradient depends on the new probabilities, which will be small due to the interaction between the new and the old logits due to the softmax, resulting in a bigger gradient. As a result, we have an evaluation gap at the beginning of a new task due to the outsized contribution of the new task. The gap appears to be recovered after more iterations since $\mathcal{L}_{stability}$ will grow and start to stabilize the training.

Knowledge distillation (Hinton et al., 2014) is another popular method in continual learning. For example, Learning without Forgetting (LwF) (Li & Hoiem, 2017) stores the model after learning each task and uses the last copy of the model to perform distillation. LwF distills knowledge from a previous version of the model, effectively enforcing a form of consistency of the model's predictions over time. Assuming the knowledge distillation loss never grows too much, temporal consistency would guarantee the preservation of the old predictions, and therefore mitigate the stability gap.

Regarding the logits growth, in Figure 2 we can clearly see that a combination of Replay and LwF results in a consistent growth of the logits over time. This property will results in a poorer stability over time, with bigger stability gaps due to the increasing gradient norm.

In conclusion, while continual learning methods such as Replay and LwF can obtain a good performance, both suffer from a stability gap because they are unable to ensure stability at the beginning of training. Notice that our analysis can be easily generalized to other continual learning strategies.

4 Methodology

4.1 Elastic distillation

As stated in Sec. 3.2, using the distillation loss on it's own or combining it naively with replay by applying the LwF distillation loss on old logits (ER + LwF) fails to regularize the new logits

that appear at each task in the *class-incremental learning* scenario. For the same reasons than the ones explained in Sec. 3.1, this leads to a decrease in old tasks performance because of a bias in favour of the last task. This motivates the need for a distillation method which also handles the new logits. We propose a new distillation loss that is able to handle the new class logits, and coin it **Elastic-Distillation** (ED). The difference with previous distillation techniques is that we apply the distillation loss on all logits, including the new ones. However, we treat them apart from previous logits by pushing new classes probability towards 0. As a result, this stability loss aims to maintain the accuracy on previous task data, while limiting the accuracy progress from the new task. This is counterbalanced by the cross-entropy loss, which is trying to raise the accuracy on the current task. We find experimentally that this trade-off is beneficial to retaining old task performance, and is still able to learn the new task satisfyingly.

4.2 MEAN-TEACHER DISTILLATION

In the boundary-free setting, it is not possible to know when a distribution shift happens. For this reason, classical distillation would fail to provide a stable teacher model. The default behaviour for a distillation method would then be to set the teacher model to the model learned on previous mini-batch. However, this implies that the teacher model would change every few iterations and in effect not provide a stable training signal anymore. In order to improve the effectiveness of knowledge distillation in that setting, we use a Temporal Ensemble (Samuli & Timo, 2017) as a teacher model. A Temporal ensemble is an ensemble made of models taken at different points on the training trajectory. Since such an ensemble can give only a small weight to individual models, the function it models cannot vary drastically from one training iteration to another, making it a good candidate for a stable teacher model. Since we don't want the memory constraints of the algorithm to increase too much, it seems a reasonable choice to use Mean-Teacher model (Tarvainen & Valpola, 2017), which ensembles an arbitrary amount of models from the training trajectory by averaging the weights of the student models.

Mean teacher distillation has been proposed in Tarvainen & Valpola (2017) to improve results in the semi-supervised learning task, in which only a fraction of training labels are available, but the rest of the data don't have a label. This method uses an additional model which is not trained by gradient descent but rather built as an exponential moving average of the student model, which one is trained by classical gradient descent. The two models interact through a knowledge distillation Hinton et al. (2014) term added to the loss of the student, which ensures that the student's predictions are consistent with the ones of the teacher.

The teacher model with parameters θ_{ema}, is computed as an exponential moving average of the training model at every iteration t following equation 4, where λ is a momentum parameter, θ^t the parameters of the training model at iteration t and θ^{t-1}_{ema} the previous parameters of the teacher model. Distillation is then performed using the output of θ^t_{ema} when computing the gradient at step $t+1$, using the following loss 5, where λ is the weight of the *consistency* cost.

$$\theta^t_{ema} = \lambda\theta^{t-1}_{ema} + (1-\lambda)\theta^t \tag{4}$$

$$\mathcal{L} = \mathcal{L}_{CE}(f(x;\theta), y) + \lambda.\mathcal{L}_{CE}(f(x;\theta), f(x, \theta_{ema})) \tag{5}$$

In Tarvainen & Valpola (2017), the authors noticed that their method offered no improvement when using all of the training labels, but only offered improvement in the semi-supervised setting. This is true in the setting where the data distribution does not change, but we demonstrate in this article that this method can also help to improve the stability of continual learning agent in case of time-varying training distribution. We found that when combined with replay, this method is more efficient than classical distillation in the challenging online continual learning setting.

5 EXPERIMENTS

Scenario. We experiment on the *class-incremental learning* setting since this is the one that has been found to suffer the most from the stability gap, we will refer to this first setting as the *boundary-*

aware scenario. We follow the experimental setup in De Lange et al. (2022) and use experience replay (Chaudhry et al., 2019) for all datasets, with a fixed memory size of 1000 exemplars for CIFAR-10 and MNIST, 2000 exemplars for CIFAR-100, and 10000 exemplars for Mini-Imagenet. Each training mini-batch is formed out of half of exemplars from previous tasks and half from the current task. We also perform experiments in the online, *boundary-free* (task-free) scenario discussed in Sec. 2.2. However, to ensure that the results are comparable to the ones of the previous scenarios, we keep the same data splits and tasks number. In this setting, several training passes on each mini-batch are allowed, we chose to use the same number of passes than the number of epochs in the previous scenarios so that the total number of training iterations matches the ones of the previous scenarios.

Methods. In the boundary-aware scenario, we compare the use of **replay (ER)** with its combination with the proposed elastic distillation scheme (**ER + ED**). In the boudary-free scenario, we present results for the 3 methods, **replay (ER)**, **ER + ED**, and the distillation scheme that we propose to apply in this setting, using the Mean-teacher architecture to perform distillation, **ER + MT-KD**.

Datasets. We perform experiments on 3 datasets. **CIFAR-10** is a 10-class dataset that contains 60000 images of size 32 by 32 and 3 color channels Krizhevsky (2009). **CIFAR-100** has the same image dimensions and number of images but with 100 classes.**Mini-Imagenet** Vinyals et al. (2016) is a 100-class version of **ImageNet** Russakovsky et al. (2015), that contains 60000 images which are rescaled to 84 by 84. We split these datasets into 5, 5, 10 and 20 tasks respectively.

Training and Implementation Details. For the majority of the experiments, we stick to the configuration used in De Lange et al. (2022). For CIFAR-10, and Mini-Imagenet we use a slim version of Resnet-18, while for CIFAR-100 we use a version of Resnet-32. For all datasets we learn for 10 epochs per task using Stochastic Gradient Descent with a momentum of 0.9, and a learning rate of 0.1 for all datasets. With a batch size of 128 for Mini-Imagenet and CIFAR-100, and 256 for CIFAR-10. We run each experiment for 6 seeds and report the mean and standard deviation. We make use of the Avalanche framework (Lomonaco et al., 2021) for all of our experiments. We will make the code available on github upon acceptance.

Continual Evaluation Details. In De Lange et al. (2022), the authors choose to keep a fixed size validation set on previous tasks and evaluate the model every ρ_{eval} iterations. In this article, we choose to evaluate after every training iteration ($\rho_{eval} = 1$), and we keep a validation set of growing size, taking 5% of the incoming data of each task as validation subset.

5.1 Boundary-Aware Scenarios

On **CIFAR10**, $ER + ED$ drastically reduces previous tasks accuracy drop (See Fig. 3). For the first task shift, the accuracy drop using only replay reaches 52%, while it reaches only 18% when using $ER + ED$. Furthermore, for this dataset, the previous task accuracy as well as the final average accuracy is improved when using the method, which we believe is due to a reduction of the task-recency bias. WC-ACC is highly improved, from 18% to 29% at the end of training. Overall average accuracy is also improved from 36% to 51% (See Tab. 1).

On **CIFAR100**, we observe a similar improvement in terms of accuracy drop reduction (See Fig. 3). For every task shift, the initial accuracy drop observed for replay is reduced by the addition of elastic distillation. For this dataset, the first task accuracy even keeps growing after the first task shift, before suffering a much smaller drop than replay at the next task shift (from 23% to 5% drop). For this dataset, the WC-ACC is lower for $ER + ED$ during training of the second task, this means that for this task the distillation imposed too much stability, which could be reduced by better tuning of the regularization parameter. However, overall, the gains in WC-ACC are the most notable for this dataset, with a final difference of 13%, improving 10% to 23%, standing just 10% below the average accuracy by which it is upper-bounded. The average accuracy is also increased from 28.4% to 33.7% 1.

On **Mini-Imagenet**, while the accuracy of $ER + ED$ before task shift remains close from the one of ER (See Fig. 4) , the accuracy drops are much smaller when using elastic distillation. This is the dataset where replay suffers the most from the stability gap. The use of $ER + ED$ more than doubles final WC-ACC, improving it from 5% to 11%.

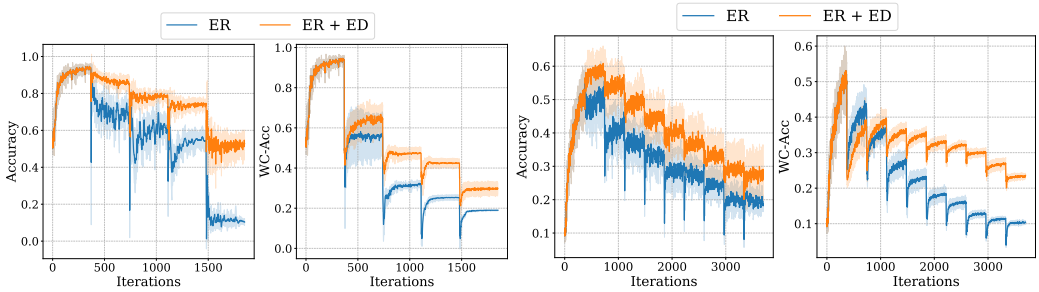

Figure 3: **Boundary-Aware Scenario** (Left) CIFAR10 (5 splits) (Right) CIFAR100 (10 splits), Task 1 accuracy over the course of training, for both normal experience replay (ER), and when applying on top the proposed elastic distillation scheme (ER + ED). On the right of each plot, WC-ACC_t at each training iteration. Mean and Standard deviation are computed over 6 seeds

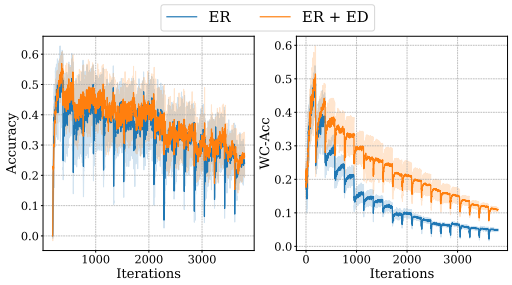

Figure 4: **Boundary-Aware Scenario** Mini-Imagenet (20 splits), Task 2 accuracy over the course of training, for both normal experience replay (ER), and when applying on top the proposed elastic distillation scheme (ER + ED). On the right of each plot, WC-ACC_t at each training iteration. Mean and Standard deviation are computed over 6 seeds

Across all datasets, we notice that WC-ACC is increased by a large margin by the use of the proposed method. Note that while it has been improved, WC-ACC remains under the average accuracy (by which it is upper-bounded). This means that a small stability gap is still present after the application of the method. The remaining part of the gap might be due to the fact that the gradient of the distillation loss is low at the very beginning of training a new task, since teacher model and student model give similar outputs at that moment, this issue has also been mentioned in De Lange et al. (2022).

5.2 Boundary-Free Scenarios

On CIFAR10 , ER+ED provides a slight improvement over ER, and it results in a 2.6% improvement in final average accuracy (See Tab. 1), and overall a better worst-case accuracy (See Fig. 5). $ER + MT$-ED yields a significant improvement in both the accuracy (+ 9.9% compared to ER) and WC-ACC. Notice in particular how the accuracy drop is made smoother by $ER + MT$-ED at the last task shift (Left plot of 5).

On CIFAR100 , $ER + ED$ fails to improve both accuracy and worst-case accuracy as expected given the unstable nature of the teacher model used for distillation, but $ER + MT$-ED improves both accuracy by 5.3% (See Tab. 1) and more than doubles worst-case accuracy (See Fig. 5). This dataset is again the one where the gains in WC-ACC are the most important.

On Mini-Imagenet , $ER + ED$ offers particularly surprising improvements over ER without the use of mean-teacher. We think this might be due to the high number of exemplars used with this dataset that renders the multiple training pass training more efficient. On this dataset, ER+MT-ED does not give significant improvements over the latter.

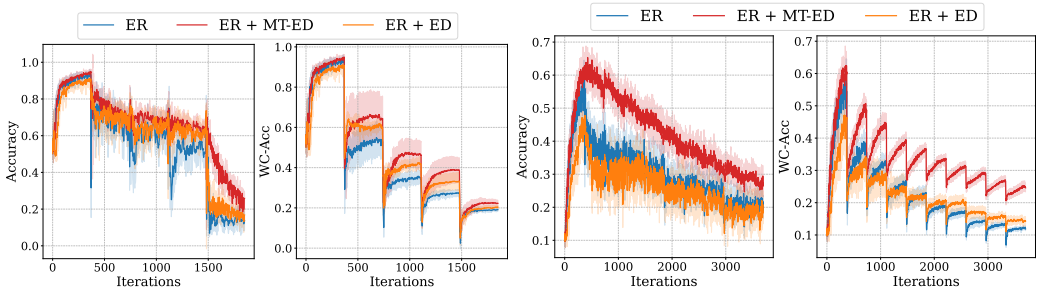

Figure 5: **Boundary-Free Scenario**, (Left) CIFAR10 (5 splits) (Right) CIFAR100 (10 splits), Task 1 accuracy over the course of training, for both replay (ER), when applying on top the proposed elastic distillation scheme (ER+ED), and when additionally using the mean-teacher for distillation (ER + MT-ED). On the right of each plot, WC-ACC$_t$ at each training iteration. Mean and Standard deviation are computed over 6 seeds

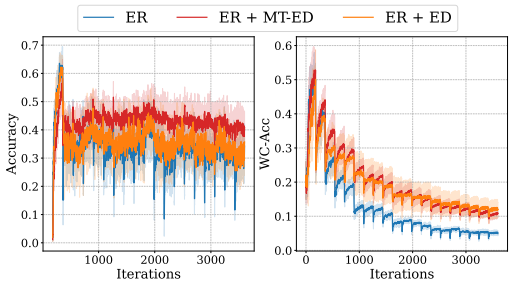

Figure 6: **Boundary-Free Scenario**, Mini-Imagenet (20 splits), Task 2 accuracy over the course of training, for both replay (ER), when applying on top the proposed elastic distillation scheme (ER + ED), and when additionally using the mean-teacher for distillation (ER + MT-ED) . On the right of each plot, WC-ACC$_t$ at each training iteration. Mean and Standard deviation are computed over 6 seeds.

Overall, ER+MT-ED helps palliate the lack of stability obtained with ER+ED in the boundary-free scenario, offering significant gains over this method when it fails to improve the stability on its own. Again in this setting, improvements in stability are often accompanied with improvements in final performance.

Dataset	CIFAR10	CIFAR100	Mini-Imagenet
ER	37.8 ± 1.4	27.2 ± 1.3	25.5 ± 1.7
ER+ED	40.4 ± 2.6	26.2 ± 2.1	**26.4** ± 1.7
ER+MT-ED	**47.7** ± 2.1	**32.5** ± 0.83	26.3 ± 2.8

Dataset	CIFAR10	CIFAR100	Mini-Imagenet
ER	36.7 ± 1.2	28.4 ± 0.7	24.4 ± 2.7
ER+ED	**51.4** ± 1.5	**33.7** ± 1.5	24.4 ± 2.8

Table 1: Final average accuracy for the two proposed methods and the replay baseline in the boundary-free (Left) and boundary-aware (Right) setting

6 Conclusion

We identified the gradient unbalance towards the last task as a cause of the stability gap in continual evaluation. We argued that the continuous growth of logits values is partially responsible for this gradient unbalance. Our proposed elastic distillation method helps to tackle the gradient unbalance by limiting the logits growth for new tasks. Furthermore, we show that applying the Mean-Teacher distillation scheme helps to reduce the stability gap in the challenging boundary-free setting. We hope that the analysis and methods discusses in this paper will help to further motivate the development of new techniques that intend to mitigate the stability gap in online continual learning. We demonstrated that addressing the stability gap can also improve overall performance, and thus encourage further research to not overlook the continual evaluation aspect of learning, even in situations where stability is not needed.

REPRODUCIBILITY STATEMENT

We will release our source code on github upon acceptance. All of our experiments are based on the Avalanche library Lomonaco et al. (2021), a continual learning library based on Pytorch Paszke et al. (2019). Additionnaly, we will provide configuration files and instructions to reproduce every experiment.

REFERENCES

Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper/2019/file/15825aee15eb335cc13f9b559f166ee8-Paper.pdf.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11254–11263, 2019b.

Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021.

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *ICML Workshop: Multi-Task and Lifelong Reinforcement Learning*, 2019.

Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8250–8259, 2021.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Matthias De Lange, Gido van de Ven, and Tinne Tuytelaars. Continual evaluation for lifelong learning: Identifying the stability gap. *arXiv preprint arXiv:2205.13452*, 2022.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *Proc. Int. Conf. Learn. Repres.*, 2014.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. nips 2014 deep learning workshop. *arXiv preprint arXiv:1503.02531*, 2014.

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*, pp. 831–839, 2019.

Huiyi Hu, Ang Li, Daniele Calandriello, and Dilan Gorur. One pass imagenet. *arXiv preprint arXiv:2111.01956*, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017a.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017b.

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nrGGfMbY_qK.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2nd Continual Learning in Computer Vision Workshop, 2021.

Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *arXiv preprint arXiv:2010.15277*, 2020.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. 1989.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, pp. 2001–2010, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Laine Samuli and Aila Timo. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, volume 4, pp. 6, 2017.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. In *NeurIPS Continual Learning Workshop*, 2018.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019.