Clustering-based Domain-Incremental Learning

Christiaan Lamers [1]
NORCE
Grimstad, 4879, Norway
chla@norceresearch.no

René Vidal
Institute for Data Engineering and Science
University of Pennsylvania
Philadelphia, USA
vidalr@seas.upenn.edu

Nabil Belbachir
NORCE
Grimstad, 4879, Norway
nabe@norceresearch.no

Niki van Stein
Leiden Institute of Advanced Computer Science
Leiden, 2333 CA, The Netherlands
n.van.stein@liacs.leidenuniv.nl

Thomas Bäck
Leiden Institute of Advanced Computer Science
Leiden, 2333 CA, The Netherlands
t.h.w.baeck@liacs.leidenuniv.nl

Paris Giampouras
Johns Hopkins University
Baltimore, MD 21218, US
parisg@jhu.edu

Abstract

We consider the problem of learning multiple tasks in a continual learning setting in which data from different tasks is presented to the learner in a streaming fashion. A key challenge in this setting is the so-called "catastrophic forgetting problem", in which the performance of the learner in an "old task" decreases when subsequently trained on a "new task". Existing continual learning methods, such as Averaged Gradient Episodic Memory (A-GEM) and Orthogonal Gradient Descent (OGD), address catastrophic forgetting by minimizing the loss for the current task without increasing the loss for previous tasks. However, these methods assume the learner knows when the task changes, which is unrealistic in practice. In this paper, we alleviate the need to provide the algorithm with information about task changes by using an online clustering-based approach on a dynamically updated finite pool of samples or gradients. We thereby successfully counteract catastrophic forgetting in one of the hardest settings, namely: domain-incremental learning, a setting for which the problem was previously unsolved. We showcase the benefits of our approach by applying these ideas to projection-based methods, such as A-GEM and OGD, which lead to task-agnostic versions of them. Experiments on real datasets demonstrate the effectiveness of the proposed strategy and its promising performance compared to state-of-the-art methods.

This work is supported by the project ULEARN "Unsupervised Lifelong Learning" and co-funded under the grant number 316080 of the Research Council of Norway.

1. Introduction

Continual learning can be described as the ability to continually learn over time by accommodating new knowledge while retaining previously learned experiences [33, 27]. We humans typically have no problem with retaining old experiences while at the same time being able to learn new tasks. For example: when a child learns to ride a bike, she does not forget the previous experience of learning how to walk.

In sharp contrast, standard machine learning algorithms typically assume that independent and identically distributed (i.i.d.) training examples of a task are given and use Empirical Risk Minimization (ERM) to learn a model for the task [36]. While this approach can be naturally extended to the setting in which samples arrive in an online fashion, when the task changes the conditional distribution of the data given the task also changes. As a consequence, the performance of the model on previously learned tasks significantly degrades when trained on new tasks, a phenomenon known as *catastrophic forgetting*.

Existing methods that deal with catastrophic forgetting often assume that the moment the task changes and the identity of the task are known at training time. For in-

[1]Corresponding author.

stance, Averaged Gradient Episodic Memory (A-GEM) [7] and Orthogonal Gradient Descent (OGD) [14] counteract catastrophic forgetting by solving a constrained optimization problem for each task change, which ensures that the loss function: a) decreases on the current task and b) does not increase on previous tasks. The constraints on previous tasks are enforced by storing either *labeled data samples* (A-GEM) or *model gradients* (OGD) from previous tasks as new tasks incrementally arrive. Thus, knowledge of a task change is needed to both solve the constrained optimization problem and update the pool of stored samples or gradients. Moreover, both A-GEM and OGD use pool size that grows with the number of tasks, making memory requirements prohibitive for a large number of tasks. While such memory requirements could be reduced by maintaining a constant and finite memory, this would inevitably lead to catastrophic forgetting as the number of tasks grows.

The aforementioned weaknesses raise two critical questions:

1. *Can we develop a memory and projection-based continual learning algorithm that does not require knowledge of task boundaries?*

2. *Can we address catastrophic forgetting more effectively for a large number of tasks while maintaining a constant and finite amount of memory?*

Paper contributions. In this work, we address these questions by proposing an online clustering-based approach that renders standard projection-based continual learning algorithms task-agnostic. This approach successfully counteracts forgetting in the setting of domain-incremental learning, a setting for which this problem was previously unsolved [35]. The proposed approach is generic and can be applied to different projection-based algorithms. To showcase its merits, we focus on the A-GEM and OGD algorithms and propose two new task-agnostics versions called Task Agnostic Averaged Gradient Episodic Memory (TA-A-GEM) and Task Agnostic Orthogonal Gradient Descent (TA-OGD). These algorithms reduce the amount of forgetting when training on different tasks without the need to know any task boundaries and identities. This is achieved by dynamically updating the pool of *labeled data samples* (A-GEM) or *model gradients* (OGD) each time a new batch becomes available. In addition, unlike A-GEM and OGD, which store a growing number of samples or gradients as the number of tasks increases, leading to prohibitive memory requirements in practical scenarios, the proposed TA-A-GEM and TA-OGD methods have constant and finite memory requirements by keeping a finite number of samples or gradients throughout the training process. To achieve this, TA-A-GEM and TA-OGD leverage the structure of the training data, which are now grouped into clusters of samples or gradients. Specifically, for each new batch, we first uniformly draw samples or gradients from the current batch and use them to initialize a predefined number of clusters using the samples or gradients as the cluster centers. After initialization, new samples or gradients are assigned to the cluster center with minimum ℓ_2 distance. To keep a constant memory, when the maximum cluster size is reached we remove less informative cluster members and update the cluster center with the average of the cluster members.

In short, this paper makes the following contributions:

- We propose a generic clustering-based method for successfully extending projection-based continual learning algorithms to a task-agnostic context. We focus on two state-of-the-art projection-based algorithms i.e., A-GEM and OGD showing that the proposed strategy enjoys the merits of memory and projection-based methods [14, 23, 12] without requiring knowledge of the task identity or task changes.

- By leveraging the structure of the data from previously seen tasks, we can retain the information needed to address catastrophic forgetting, such as training data (A-GEM) or model gradients (OGD), while keeping the memory-size finite via a simple and efficient clustering procedure. We thus depart from the standard approach of OGD and A-GEM, which demand a growing amount of memory as new tasks sequentially arrive, which is impractical in real-world scenarios.

- We provide extensive experimental results for different continual learning settings on various datasets showing the promising performance of the proposed task-agnostic algorithms (TA-A-GEM and TA-OGD) compared to state-of-the-art methods.

2. Related Work

This section starts with an explanation of the three types of incremental learning. It then reviews the stability-plasticity dilemma, which continual learning methods have to face. Moreover, we present the main ideas of memory and projection-based continual learning approaches to which class the proposed TA-A-GEM and TA-OGD method belong and the main advances in task continual learning. Finally, we review the recent works leveraging representation learning for deriving efficient continual learning algorithms.

2.1. Domain-incremental learning

In continual learning, different tasks can arrive in sequence. The learner must therefore learn new tasks incrementally. This is referred to as *incremental learning*. Three types of incremental learning can be specified: *task-incremental learning*, *domain-incremental learning* and

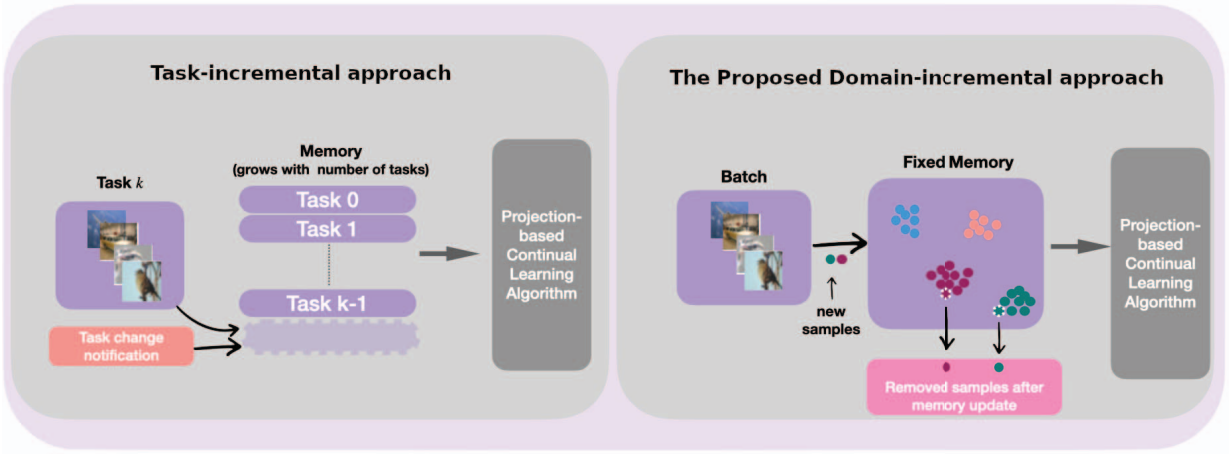

Figure 1: After the task-incremental method is finished with the training on task T_k, the memory (containing either labeled data samples in the case of A-GEM or model gradients in the case of OGD) is updated. This method is made domain-incremental by using an online clustering-based approach for updating the memory while keeping its size fixed.

class-incremental learning [35]. In task-incremental learning, the task identity is known to the learner during the training and testing phase. In domain-incremental learning, the task identity is not known to the learner at both training and testing time. In class-incremental learning, the learner must learn to identify a growing number of classes. Since we focus on a scenario where the number of classes is static and the task identity is not known during training and testing, we focus on the *domain-incremental* setting. Alleviating catastrophic forgetting in such a scenario is an important unsolved challenge [35].

2.2. The Stability-Plasticity Dilemma

The balancing act between being able to gain new knowledge while assuring old knowledge is not lost is referred to as the *stability-plasticity dilemma* [26]. Continual learning approaches can be categorized in three major trends based on how the stability-plasticity dilemma is handled [9, 27]. The first trend is to use the concept of *regularization* of synaptic plasticity, where the plasticity of important weights is constrained in order to retain old skills, like the Memory Aware Synapses used in a continual setting in [3]. Elastic Weight Consolidation (EWC) is a seminal work of this class. When a new task arrives, EWC learns the optimal weights for this task, while penalizing changes of the weights towards values that are far from the optimal ones for the previous task [21]. Several other variants of EWC have appeared in the literature and we refer the readers to [9] for a detailed review. The second trend is *expansion* [31, 2, 25, 13], where a neural network is expanded by allocating new neural resources in order to gain new skills, while leaving old neurons unchanged in order to retain old skills. Finally, according to the third trend, which is *repetition*, old information is repeatedly fed to the network, along with new information. This can be implemented by applying a complementary learning system for integrating old and new skills and applying experience replay, or by simply mixing old and new data in the training step. In the literature, various approaches of the so-called replay-based methods which rely on the principle of repetition have come to the scene. These methods make use of memory resources and vary in the strategy they follow [30, 23, 32, 8, 29, 34, 22, 38].

This paper uses the terms "replay-based" and "memory-based" interchangeably because they represent similar concepts. Still, we tend to favor "replay-based" when a method stores samples from the dataset and "memory-based" when it stores different information. The proposed TA-A-GEM builds on A-GEM [7], which stores samples from the training set, and can thus be considered "replay based". The proposed TA-OGD builds on OGD [14], and thus, in principle, falls into the category of memory-based methods since it stores gradients. At the same time, the proposed TA-A-GEM and TA-OGD use a projected gradient step and, hence, are also a projection-based approach. Note that this projection step implicitly regularizes the weights; therefore, A-GEM and OGD bear similarities with the regularization-based methods. Next, we elaborate on the specific class of memory-based and projection-based continual learning algorithms.

2.3. Memory-based and Projection-based Continual Learning Methods

Over the last few years, several memory-based and projection-based methods have been proposed in the literature, [23, 14]. These make use of memory for storing information from the past, which helps to update the model towards non-forgetting directions. The goal is to address

catastrophic forgetting by means of imposing certain constraints on the weight-updating process. Many different approaches have appeared in the literature over the last few years. In [23], the authors propose to update weights in directions that do not increase the loss function values on samples of previously seen tasks. The resulting algorithm, dubbed Gradient Episodic Memory (GEM), thus stores a predefined number of gradients of the loss function corresponding to old tasks, [7, 23]. These are then used for updating the model by solving a constrained optimization problem. Orthogonal Gradient Descent (OGD) [14] stores a growing number of gradients of the model corresponding to old tasks' samples. In the weight update step, it projects its loss gradient to a direction that is orthogonal to all stored gradients. Specifically, gradients of the loss are projected on the orthogonal basis spanned by the stored gradients. In doing so, directions that increase forgetting of past tasks are excluded when the model learns a new task. This assumes however that the stored gradients remain relevant, even when the weights of the model move during the training process, thus arriving at a different point in the configuration space in which older tasks can have different gradients. Averaged Gradient Episodic Memory (A-GEM) [7] solves this problem by storing labeled data samples instead of gradients. It projects the loss gradient orthogonal to a reference gradient that is calculated at every training step from a subset of the stored labeled data. Though showing promising performance in addressing catastrophic forgetting, memory-based and projection-based methods suffer from two fundamental weaknesses: a) they require the moment of task change to be available in order to know when the memory should be updated, and b) memory cost should either scale with the number of tasks, e.g., in OGD [14], which is infeasible in real-world scenarios, or the stored data per task will decrease as in the case of GEM [23], which also hinders the ability of the algorithm to address forgetting when it encounters a large number of tasks.

2.4. Task Agnostic Continual Learning

Task boundaries and identities are rarely available in practical continual learning applications. In light of this, various task-agnostic continual learning methods have been proposed in the literature. In [16], the authors propose an auxiliary mechanism to detect tasks while counteracting forgetting. The resulting method operates in a task-agnostic environment showing promising empirical performance. Several other approaches have been proposed in the same spirit [5, 18]. Another line of work hinges on online learning ideas completely neglecting task identity or the need to know the moment of task change. In [39], the authors propose Bayesian Gradient Descent (BGD), an online variational Bayes approach in which model parameters with low variance are considered more important for previous tasks and, thus, are less updated. The opposite holds for parameters with high variance (hence high uncertainty). A similar idea for task-free continual learning appeared in [4]. Namely, the authors modified the so-called Memory Aware Synapses (MAS) algorithm in [1], in order to operate in a task-agnostic online learning setup. For, they use an importance weight regularizer which penalizes changes to model parameters which negatively affect model performance on prior tasks. Finally, in [20] the authors propose an online task-agnostic memory-based method. The main idea is to edit the stored-in-memory gradients used for addressing forgetting by solving an optimization problem in an online fashion. Recently, the idea of using self-supervised representations for task-agnostic continual learning was proposed in [28], showing promising empirical performance.

Though the emergence of clustering in episodic memory has been recently acknowledged in the child development literature [19], to the best of our knowledge, the proposed TA-A-GEM and TA-OGD are the first algorithms that use online clustering for dynamically updating the memory of continual learning methods. While we focus on A-GEM and OGD, the adopted strategy could be applied to other memory-based and task-dependent continual learning approaches for allowing them to operate in task-agnostic environments.

2.5. Representation Learning

Representation learning aims to find insightful data representations by exploiting their structure [24]. Recently, learned representations have been at the heart of several continual learning algorithms. In [6], the authors employed low-rank orthogonal subspace representations of the model parameters formulating continual learning as an optimization over the Stiefel manifold problem. The reported results showed promising performance and the ability of the approach to counteract forgetting. In [15], *holistic* representations learned via a mutual information maximization criterion were employed in the continual learning setting. The method can learn feature representations of the current task that are useful for the future tasks, hence leading to models that are more robust to forgetting. In [12], a variant of the projection-based OGD method was proposed. The main idea is to perform principal component analysis on the set of stored gradients of the model and keep only the most informative principal components. However, the work in [12], still assumes that task changes are provided to the algorithms and batch processing is utilized. Hence it is far from our proposed online clustering-based task-agnostic algorithms.

3. Proposed Approach

We assume that the n tasks $\{T_i\}_{i=1}^{n}$ arrive sequentially and that during task T_k the data from tasks T_i for $i < k$ are

not presented to the learner. Each task consists of pairs of data points $(x, y) \in T_k$, where $x \in \mathbb{R}^d$ is the input and y is a label. Here we assume that each task is a classification task and that all classification tasks share the same classes $j = 1, \ldots, c$, where c is the number of classes. Therefore, we can represent $y \in \mathbb{R}^c$ as a one-hot class encoding vector, i.e., $y_j = 1$ when j is the class label and $y_j = 0$ otherwise. We denote the network model as $f(x; w) \in \mathbb{R}^c$, where $w \in \mathbb{R}^p$ denotes the p-dimensional weights (parameters) of the network and $f_j(x; w)$ is the j-th logit corresponding to the j-th class. The model is trained to predict the class label for input x.

The proposed Task Agnostic Averaged Gradient Episodic Memory (TA-A-GEM) and Task Agnostic Orthogonal Gradient Descent (TA-OGD) methods rely on the forgetting counteracting mechanisms of Averaged Gradient Episodic Memory (A-GEM) [7] and Orthogonal Gradient Descent (OGD) [14], respectively. Next, we briefly describe the main ideas behind A-GEM and OGD and refer the reader to the Appendix or [7] and [14] for further details.

Both A-GEM and OGD assume the identity k_t of the task T_{k_t} at time step t is known. The empirical loss, during time step t, with a batch size $|T_{k_t}|$, is given by,

$$L_t(w) = \frac{1}{|T_k|} \sum_{(x,y) \in T_k} L_{(x,y)}(w), \tag{1}$$

where the per sample loss $L_{(x,y)}(w)$ is assumed to be the cross-entropy, which is defined as

$$L_{(x,y)}(w) = -\sum_{j=1}^{c} y_j \log \left(\frac{\exp f_j(x; w)}{\sum_{m=1}^{c} \exp f_m(x; w)} \right). \tag{2}$$

Both A-GEM and OGD use a pool of samples to counteract the catastrophic forgetting. The difference is that OGD stores network gradient, while A-GEM stores training data.

3.1. Clustering-based Task Agnostic A-GEM (TA-A-GEM) and OGD (TA-OGD)

Figure 1 shows our strategy to convert a task-aware task-incremental projection algorithm to a task-agnostic domain-incremental algorithm. Task-incremental projection algorithms like A-GEM and OGD keep a pool of samples from either the training data or model gradients, respectively. This pool of samples is used to mitigate catastrophic forgetting of previous tasks through projection. When the algorithm is finished with training on one task, it stores samples from this task before it starts training on the new task. In this way, it ensures that the samples in the pool are relevant for previous tasks when addressing forgetting. However, this comes at the cost of *requiring to know the moment a task changes*. In our approach, we make this process task-agnostic by updating the pool of samples during the process of training, i.e. *the pool of samples is updated every time the model is trained on a batch*. This removes the need to know the moment the task changes but introduces the problem that the size of the pool now grows more rapidly. However, *our goal is to keep the memory requirements constant in the number of tasks*. Hence, a strategy is necessary to decide which samples should be added to the pool and which ones should be removed during the updating process. Our strategy aims to select stored samples in a way that addresses forgetting all previous tasks in the most efficient way while being constrained by constant and finite pool size. Because we aim for a true task-agnostic setting, all tasks are made to have the same label space, so the task identity can not be inferred from the labels.

Next, we detail the proposed online clustering-based approach that consists of the following four steps:

1) *Initialization:* We first set the number of clusters Q and consider the first Q samples becoming available as the centers $\boldsymbol{\mu}_i, q = 1, 2, \ldots, Q$ of these clusters.

2) *Cluster assignment:* A new sample $\mathbf{z}_p$ (corresponding to a training sample in the case of A-GEM or gradient logit in the case OGD) is assigned to the cluster q^* that minimizes the ℓ_2 norm i.e.,

$$q^* = \text{argmin}_{q \in \{1,2,\ldots,Q\}} \|\mathbf{z}_p - \boldsymbol{\mu}_q\|_2^2 \tag{3}$$

3) *Memory update:* The size of each cluster is predefined, and once the maximum size has been reached, for new samples to that assigned to that cluster an equal number of older samples residing in the cluster should be removed. Note that the process of accepting/rejecting new samples and deciding which "old" samples to delete could be implemented using information-theoretic criteria or rejection sampling-based ideas. Here, in an effort to simplify the approach and make it computationally efficient, we follow a first-in-first-out (FIFO) approach. This dictates that samples that arrived first in the cluster are the first to be removed. Note that the strategy followed ensures that samples corresponding to a task with information distinct from other tasks will not be deleted from the pool. This will occur since these samples will "live" within clusters that will not be updated and thus remain unaffected by the memory updating process.[1]

3) *Update of cluster means:* Once samples are assigned to the clusters and the memory has been updated, the cluster means are re-computed i.e.,

$$\boldsymbol{\mu}_q = \frac{1}{P} \sum_{p=1}^{P} \mathbf{z}_p^q, \qquad \forall i = 1, 2, \ldots, N, \tag{4}$$

where P denotes the size of the clusters and $\mathbf{z}_p^q$ the p_{th} element of cluster q. For the case of the task-agnostic version of A-GEM, i.e., TA-A-GEM, we have $\mathbf{z}_p \equiv \mathbf{x}_p \in \tilde{M}_t$

[1] Empirical findings reported in the Appendix corroborate our hypothesis.

(where t here denotes the batch index) whereas for the task-agnostic OGD algorithm (TA-OGD) $\mathbf{z}_p \equiv \nabla f_j(\mathbf{x}_p, w_i^*)$. Our clustering-based strategy is depicted at Fig. 2, while a pseudo-code of the algorithm is given in the Appendix.

A single or a different pool for each class? A possible complication that can occur is that more similarity exists between samples of the same class that are of a different task than between different classes of the same task. If this happens, *class* information will be well represented in the pool, but *task* information can be easily lost. Since class labels of the samples are available, a way to get around that issue and disentangle the class from task information is to use a different pool for each class. In that case, samples are first assigned to a pool based on their class label. Then, the procedure described above is independently followed for each pool. It is worth noting that this is critically important for the task-agnostic version of A-GEM (TA-A-GEM) since the pool contains training samples of different classes. Samples corresponding to the same class but different tasks, e.g., a digit and its rotated version might be close in the input space. As a result, if a single pool is used, those two samples will be assigned to the same cluster, and hence task information will be lost. This phenomenon is more likely not to be observed in the case of TA-OGD since clustering takes place in the space of model gradients, which are sufficiently separated for different tasks even for samples corresponding to same classes.

The role of hyperparameters: The choice of hyperparameters, such as the number of clusters Q and their size, is important. A large number of clusters Q, allows more task and class diversity to be stored in different clusters in memory. The size of the clusters should be large enough so it can capture the essence of a specific task. However, the size of Q and the cluster size should be kept as small as possible to reduce the memory footprint. A trade-off can be made where Q is large, and the cluster size is small versus using a small Q with a large cluster size. In addition, we follow an adaptive strategy for the learning rate of the projected gradient step. Note that this is a form of task detection that our method does not necessarily need. Our focus is to create a truly task-agnostic method without any task detection. Specifically, the learning rate η^t at iteration t decreases as follows:

$$\eta^t = a\eta^{t-1}, \tag{5}$$

where $a < 0$, when the loss function is *smoothly* increasing for a given number of iterations. This allows the algorithm to update the weights of the model following a non-increasing path for the loss function. Moreover, when a sudden increase is observed, then the learning rate is reset to its initial value (therefore increases), i.e., $\eta^t = \eta_{ini}$. The reasoning behind this rule is that spikes of the loss most likely imply task-change and therefore, a higher learning rate can help to move fast along decreasing directions of the loss corresponding to the new task. Empirical results on the effect of the sampling rate, the number, and the size of clusters on the performance of the proposed method, and more details on the adaptive updating process of learning rate, are provided in Section 4 and Appendix.

4. Experiments

We divide the experiments into two main classes: a) the *disjoint tasks experiment* and b) *the continuous change experiments*. The task-aware methods are notified of the task change, while the task-agnostic methods do not get this information. In the continuous change experiments, discrete tasks still exist, but task boundaries are no longer clearly defined. Details on the experimental setting can be found in the Appendix. Since there is no clear point that a task-aware method can be notified, only task-agnostic methods are included in this experiment. For both methods, all tasks are made to have the same label space, since it should not be possible to infer the task identity from the labels. In cases where the label spaces are disjoint, the labels are cast to the same label space. Since no task identity is provided during training, the method is tested in a domain-incremental setting [35]. Following empirical observations, we use the learning rate scheduler described in Section 3.3 for the case of OGD and the proposed task-agnostic version of it i.e., TA-OGD. The network used for training is a multi-layer perceptron (MLP) with two hidden layers of 200 nodes. To compare the performance of the tested methods, we use three metrics: a) The *validation accuracy*, b) The *average validation accuracy* over all tasks trained on thus far and c) The amount of *forgetting*. For an exact mathematical definition of these quantities , we refer to the Appendix. To create separate tasks from existing datasets, three task generation mechanisms are implemented: a) task permutation, b) task rotation and c) class splitting. For the details of this task generation, we refer to the Appendix.

4.1. Disjoint tasks experiment

Table 1 shows the results of the first class of experiments. It shows the average accuracy over all tasks trained thus far, thereby capturing both the ability to remember old tasks and the ability to learn new tasks. The average accuracy was then averaged over 20 epochs, then over five runs. Plots of these results can be found in the Appendix. Our proposed TA-OGD and TA-A-GEM algorithms significantly outperform the state-of-the-art task-agnostic BGD algorithm, [39], on the MNIST [11], Fashion MNIST [37] and NOT MNIST datasets. Moreover, their performance is comparable to BGD on CIFAR10 and SVHN. Focusing on MNIST, Fashion MNIST and NOT MNIST, we observe that at the *permutation experiments*, no remarkable differences can be seen among the methods. This can be explained by the fact that the baseline SGD method shows little signs of

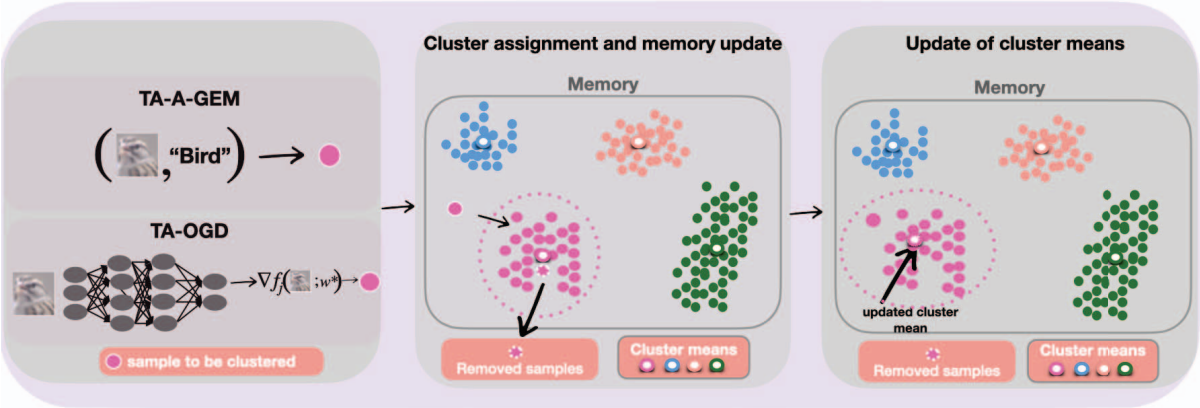

Figure 2: The clustering mechanism to add training set samples / model gradient samples to the memory by matching it to the closest cluster (pink cluster), as used by TA-A-GEM / TA-OGD.

forgetting in the first place. For the *rotation experimen* A-GEM is a clear winner, it is however not task-agnosti On MNIST and NOT MNIST, TA-OGD and TA-A-GE are moderately effective at mitigating forgetting. On Fas ion MNIST however, TA-A-GEM is clearly the best metho among all the tested task-agnostic methods. We attained th most remarkable results on the *class split experiments*. O MNIST, both TA-OGD and TA-A-GEM clearly outperfor the other task-agnostic methods. On Fashion MNIST, TA A-GEM's performance is even on par with A-GEM, whil on NOT MNIST, TA-OGD takes the crown by performin on par with A-GEM, which is a task-aware method.

4.2. Continuous task change experiment

The results of the *continuous change experiments* are extremely similar to the results in the *disjoint tasks experiments*. They can be found in the Appendix. These experiments show that the proposed TA-OGD and TA-A-GEM fare just as well in the challenging setting where task boundaries are blurred.

4.3. Effectiveness of the clustering-based procedure

In order to demonstrate the benefits obtained by the proposed clustering-based approach, we compared the performance of TA-A-GEM with and without clustering. To deactivate clustering we skipped the cluster assignment step and new samples were randomly to allocated clusters. Similarly to our approach, an equal number of old samples of update clusters is removed to keep the memory size constant. For this experiment, a MLP was trained on Fashion MNIST, with the task split segmentation. All settings are the same as in the *disjoint tasks experiments*.

Figure 3 and 4 show the content of each cluster during training time. Each horizontal line corresponds to a cluster. Each task is associated with a unique color, which represents the oldest task information that is present in the

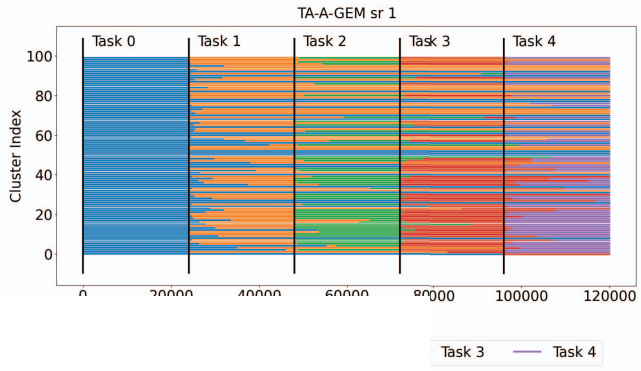

Figure 3: Clustering helps us address forgetting: samples from old tasks remain in the pool even after the of training on Task 4 ends.

cluster. The horizontal line changes color the moment that the last information of the oldest task disappears from the cluster. Then, the second oldest task information becomes the new oldest task information. The moment that a new task starts -not available to the algorithms- is indicated by a black vertical line. As it can be observed in Figs 3 and 4, clustering helps in keeping a greater variety of task information in the gradient pool, with samples from Task 0 or Task 1 still being present in clusters even after the end of training on samples from Task 4. On the other hand, the use of random cluster assignment results in information of old task being almost immediately lost after a task change, thus illustrating the merits of our proposed clustering-based approach.

5. Conclusions and future directions

In an effort to counteract catastrophic forgetting in a task-agnostic setting, we proposed a clustering-based strategy to make task-aware projection methods task-agnostic

	MNIST			Fashion MNIST			NOT MNIST			CIFAR10			SVHN		
	perm	rot	class	perm	rot	class	perm	rot	class	perm	rot	class	perm	rot	class
SGD	0.842	0.657	0.804	0.735	0.468	0.807	**0.850**	0.598	0.864	0.373	0.346	0.724	**0.599**	0.392	0.759
SGD lr adapt	0.842	0.663	0.811	0.736	0.469	0.820	**0.851**	0.598	0.888	0.376	0.347	**0.727**	**0.596**	0.390	0.762
BGD	**0.883**	0.682	0.790	**0.765**	0.507	0.802	**0.856**	**0.633**	0.875	**0.385**	**0.357**	0.718	**0.591**	**0.417**	0.763
TA-OGD	**0.871**	**0.705**	0.857	0.749	0.516	0.893	0.845	0.625	**0.937**	0.328	0.343	**0.731**	0.547	0.393	**0.773**
TA-A-GEM	**0.876**	0.688	**0.878**	0.746	**0.604**	**0.931**	**0.853**	0.602	0.884	0.365	0.343	**0.726**	**0.605**	0.399	**0.772**
OGD	0.865	0.690	0.822	0.757	0.512	0.839	0.846	0.627	0.925	0.360	0.348	0.731	0.587	0.400	0.768
A-GEM	0.884	0.806	0.952	0.761	0.706	0.934	0.854	0.740	0.947	0.360	0.356	0.741	0.552	0.451	0.827

Table 1: **Average validation accuracy**, averaged over all tasks trained thus far, then averaged over all epochs, then averaged over five runs, for the **disjoint tasks experiments** when using a MLP. Per column, the best result for the *task-agnostic* methods are written in bold. In case a task-agnostic method's result is less optimal and not significantly different from the best result, with a confidence of 99%, it is also written in bold. The results for the *task-aware* methods OGD and A-GEM are given for context. Since these algorithms benefit from knowing task identities and changes, we just use them here as baselines for indicating the best performance we can achieve.

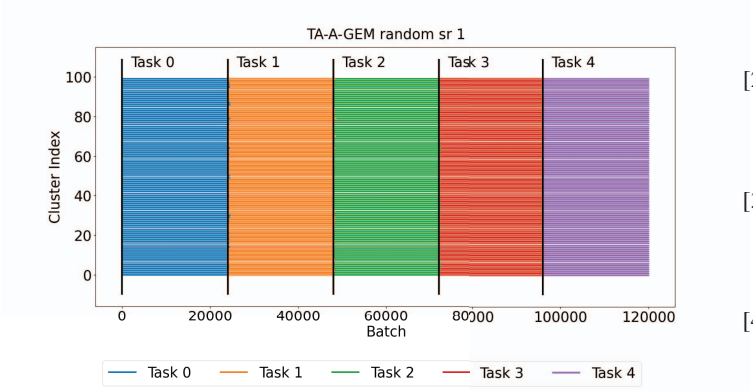

Figure 4: When using random cluster assignment, the information of old tasks is almost immediately lost, once a new tasks starts.

with constant memory requirements. By leveraging the structure in the sampled data (in the case of TA-A-GEM) and model gradients (in the case of TA-OGD), we can effectively counteract catastrophic forgetting without providing knowledge of a task change and the need of a growing amount of memory. Extensive experimental results provided in section 4.3 and the Appendix show the benefits of our clustering-based method. As a future direction, we aspire to explore more sophisticated, yet computationally efficient, methods for the clustering and memory update step. Our goal is also to illustrate the merits of our method on larger networks such as a ResNet [17], or more complicated datasets such as ImageNet [10]. It is worth noting that our proposed method is generic hence we also intend to inquire its application as an off-the-shelf tool to other projection-based methods.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 4

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017. 3

[3] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019. 3

[4] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019. 4

[5] Massimo Caccia, Pau Rodríguez, Oleksiy Ostapenko, Fabrice Normandin, Min Lin, Lucas Caccia, Issam Laradji, Irina Rish, Alexandre Lacoste, David Vazquez, and Laurent Charlin. Online fast adaptation and knowledge accumulation (OSAKA): A new approach to continual learning. *Advances in Neural Information Processing Systems*, 2020-December(NeurIPS), 2020. 4

[6] Arslan Chaudhry, Naeemullah Khan, Puneet Dokania, and Philip Torr. Continual learning in low-rank orthogonal subspaces. *Advances in Neural Information Processing Systems*, 33:9900–9911, 2020. 4

[7] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. *arXiv preprint arXiv:1812.00420*, 2018. 2, 3, 4, 5

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 3

[9] 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. 3

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image

database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 8

[11] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 6

[12] Thang Doan, Mehdi Bennani, Bogdan Mazoure, Guillaume Rabusseau, and Pierre Alquier. A Theoretical Analysis of Catastrophic Forgetting through the NTK Overlap Matrix. 130, 2020. 2, 4

[13] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 3

[14] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, volume 108, pages 3762–3773. PMLR, 26–28 Aug 2020. 2, 3, 4, 5

[15] Yiduo Guo, Bing Liu, and Dongyan Zhao. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, pages 8109–8126. PMLR, 2022. 4

[16] James Harrison, Apoorva Sharma, Chelsea Finn, and Marco Pavone. Continuous meta-learning without tasks. *Advances in Neural Information Processing Systems*, 2020-December(NeurIPS), 2020. 4

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[18] Xu He, Jakub Sygnowski, Alexandre Galashov, Andrei A Rusu, Yee Whye Teh, and Razvan Pascanu. Task agnostic continual learning via meta learning. *arXiv preprint arXiv:1906.05201*, 2019. 4

[19] Sebastian S Horn, Ute J Bayen, and Martha Michalkiewicz. The development of clustering in episodic memory: A cognitive-modeling approach. *Child development*, 92(1):239–257, 2021. 4

[20] Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient Based Memory Editing for Task-Free Continual Learning. (NeurIPS):1–22, 2020. 4

[21] 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, 2017. 3

[22] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. *arXiv preprint arXiv:2110.10031*, 2021. 3

[23] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017. 2, 3, 4

[24] Yi Ma, Doris Tsao, and Heung-Yeung Shum. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23(9):1298–1323, 2022. 4

[25] Nikhil Mehta, Kevin Liang, Vinay Kumar Verma, and Lawrence Carin. Continual learning using a bayesian nonparametric dictionary of weight factors. In *International Conference on Artificial Intelligence and Statistics*, pages 100–108. PMLR, 2021. 3

[26] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013. 3

[27] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1, 3

[28] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021. 4

[29] Eugene Belilovsky Massimo Caccia Min Lin Laurent Charlin Tinne Tuytelaars Rahaf Aljundi, Lucas Caccia. Online continual learning with maximally interfered retrieval. *arXiv:1908.04742*, 2019. 3

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. Icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 3

[31] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[32] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 3

[33] Sebastian Thrun. *Lifelong Learning Algorithms*, pages 181–209. Springer US, Boston, MA, 1998. 1

[34] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020. 3

[35] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022. 2, 3, 6

[36] Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999. 1

[37] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[38] Fei Ye and Adrian G Bors. Task-free continual learning via online discrepancy distance learning. *Advances in Neural Information Processing Systems*, 35:23675–23688, 2022. 3

[39] Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*, 2018. 4, 6