IMBALANCED LIFELONG LEARNING WITH AUC MAXIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Imbalanced data is ubiquitous in machine learning, such as medical or fine-grained image datasets. The existing continual learning methods employ various techniques such as balanced sampling to improve classification accuracy in this setting. However, classification accuracy is not a suitable metric for imbalanced data, and hence these methods may not obtain a good classifier as measured by other metrics (e.g., recall, F1-score, Area under the ROC Curve). In this paper, we propose a solution to enable efficient imbalanced continual learning by designing an algorithm to effectively maximize one widely used metric in an imbalanced data setting: Area Under the ROC Curve (AUC). We find that simply replacing accuracy with AUC will cause *gradient interference problem* due to the imbalanced data distribution. To address this issue, we propose a new algorithm, namely DIANA, which performs a novel synthesis of model DecouplIng ANd Alignment. In particular, the algorithm updates two models simultaneously: one focuses on learning the current knowledge while the other concentrates on reviewing previously-learned knowledge, and the two models gradually align during training. We conduct extensive experiments on datasets across various imbalanced domains, ranging from natural images to medical and satellite images. The results show that DIANA achieves state-of-the-art performance on all the imbalanced datasets compared with several competitive baselines. We further consider standard balanced benchmarks used in lifelong learning to show the effectiveness of DIANA as a general lifelong learning method.

1 INTRODUCTION

Natural data are inherently imbalanced: a few classes contain significantly more samples than other classes. Current research on lifelong learning (Kirkpatrick et al., 2017; Lopez-Paz et al., 2017; Chaudhry et al., 2018b) mostly focuses on balanced datasets, while ignoring the more challenging imbalanced classification problem. This impedes applications of lifelong learning to online advertisement (Hu et al., 2022), satellite imagery (Tasar et al., 2019), or medical image classification (Irvin et al., 2019). Moreover, it is not suitable to use classification accuracy to assess the model performance in these domains due to the data imbalanced issue. One of the popular metrics for measuring the performance of classifiers on the imbalanced task is the Area Under the Curve (AUC) (Hanley & McNeil, 1982; 1983). However, the current lifelong learning methods are limited to maximize the classification accuracy (Kirkpatrick et al., 2017; Lopez-Paz et al., 2017; Chaudhry et al., 2018b) which makes them not suitable for imbalanced lifelong learning. Although one may tackle the imbalanced problem by class balanced sampling (Chrysakis & Moens, 2020; De Lange & Tuytelaars, 2021; Kim et al., 2020) to form balanced training batches, these approaches still may not be able to directly optimize metrics such as AUC.

Memory-based lifelong learning methods (Lopez-Paz et al., 2017; Chaudhry et al., 2018b; Aljundi et al., 2019) achieve competitive performance across commonly used lifelong learning benchmarks. In memory-based lifelong learning methods, a replay buffer is used to store a subset of examples from old tasks for rehearsal. The gradient computed on the replay buffer (Lopez-Paz et al., 2017) is used as a reference to alter the direction of the gradient computed on the current task.

Proceeding from the memory-based lifelong learning methods for maximizing classification accuracy, one may think of applying the existing methods and replacing the metric of classification accuracy with AUC. One possible approach to directly maximize AUC for imbalanced lifelong learning is to employ the minimax reformulation of AUC as in the literature of online AUC maximization (Ying et al., 2016; Liu et al., 2020) . This minimax reformulation introduces a data-dependent decision threshold of the model to decouple the pairwise formulation of the AUC objective which facilitates

model update in an online fashion. However, we find that maximizing AUC with memory-based lifelong learning introduces an issue called *gradient interference*. In particular, when the data stream is imbalanced, the gradients computed on the current task can interfere with the gradients computed on the replay buffer severely.

In this paper, we propose a novel algorithm, called DIANA, to address the *gradient interference problem* when replacing accuracy with AUC in the imbalanced lifelong learning setting. We first formulate the objective as a composite optimization problem as in Lopez-Paz et al. (2017); Chaudhry et al. (2018b); Guo et al. (2020). Similar to existing memory-based lifelong learning methods, we aim to maximize AUC on both the current task and the replay buffer to prevent catastrophic forgetting. Notably, DIANA is designed with two novel techniques to address the *gradient interference problem*: model decoupling and alignment. In particular, DIANA decouples the learning of previous tasks and current task into two models: one focuses on learning the current task while the other reviews previously-learned knowledge. The two models gradually align during training due to an alignment penalty. Since each model computes its own gradients, we can reduce interference between the learning of the current task and the reviewing of old tasks. As we will show, the introduction of the additional model greatly alleviates the gradient interference problem for maximizing AUC with an imbalanced data stream continually, while still being computationally efficient.

Our contributions can be summarized as follows:

- We advance imbalanced lifelong learning through a completely orthogonal approach to the traditional balanced sampling techniques, which enables the lifelong learning algorithm to directly maximize an important metric (AUC) in the imbalanced data setting. We also identify the *gradient interference problem* under imbalanced setting when the existing memory-based lifelong learning methods are simply applied to maximize AUC.
- We design a new algorithm for maximizing AUC in imbalanced lifelong learning, DIANA, which decouples conflicting gradients into two models with an alignment penalty. We show that DIANA can alleviate the *gradient interference problem*.
- We verify the efficacy of DIANA on imbalanced lifelong learning benchmarks across natural images, medical images, and satellite images. We show that DIANA outperforms several state-of-the-art lifelong learning algorithms by a large margin (e.g., 6.5% AUC score on average over five benchmark datasets). In addition, compared with the approaches which purely use balanced sampling, we show that the DIANA method coupled with the balanced sampling techniques can outperform these approaches by an average 6.6% AUC score on five imbalanced datasets.
- We further expand the scope of our algorithm by considering maximizing AUC on balanced multi-class classification problems, which are standard benchmarks in lifelong learning literature.

2 Related Work

2.1 Lifelong Learning

Lifelong learning is an important topic in machine learning (Thrun & Mitchell, 1995) and is extensively studied in recent years (Parisi et al., 2019; Mai et al., 2022; Borsos et al., 2020). The current methods tackle lifelong learning from multiple objectives.

Regularization-based approaches: Regularization-based approaches aim at preserving important weights for old tasks. Representative works include EWC (Kirkpatrick et al., 2017) which adopted Fisher information matrix, PI (Zenke et al., 2017) which introduced *intelligent synapses*, RWALK (Chaudhry et al., 2018a) which utilized a KL-divergence based regularization for preserving knowledge of old tasks, and MAS (Aljundi et al., 2018) in which the importance measure for each parameter was computed based on how sensitive the predicted output function is to a change in this parameter.

Transfer learning-based methods: Transfer learning-based methods aim at leveraging *knowledge* from old tasks for learning new tasks. Representative works include PROG-NN (Rusu et al., 2016) which adds a new "column" with lateral connections to previous hidden layers for each new task and OWN (Zeng et al., 2019) which enables networks to continually learn different mapping rules in a context-dependent way.

Memory-based approaches: Episodic memory based lifelong learning methods (Hayes et al., 2021; Verwimp et al., 2021; Jin et al., 2021) leverage a small episodic memory for storing examples from old tasks. In GEM (Lopez-Paz et al., 2017), A-GEM (Chaudhry et al., 2018b), MEGA (Guo et al., 2020) and OGD (Farajtabar et al., 2019), the direction of the current gradient is modified to overcome forgetting in lifelong learning. In MER (Riemer et al., 2018), meta-learning is employed as a subroutine for mitigating catastrophic forgetting. In iCARL (Rebuffi et al., 2017), *class exemplars* are stored for each class and used for classification in class incremental lifelong learning. In experience replay (ER) based methods (Chaudhry et al., 2019; Aljundi et al., 2019), the model is trained continuously with batch gradient descent by sampling examples from the current task and the episodic memory. CTN (Pham et al., 2020) exploits dual memory to store task-specific features.

Imbalanced lifelong learning: Existing imbalanced lifelong learning approaches mainly focus on maintaining a balanced memory. CBRS (Chrysakis & Moens, 2020) tackles imbalanced lifelong learning by using a Class-Balanced memory population strategy. Similar to CBRS, PRS (Kim et al., 2020), CoPE (De Lange & Tuytelaars, 2021) and Rainbow Memory (Bang et al., 2021) introduces different class balanced sampling strategies.

2.2 AUC Optimization

Online AUC maximization aims to design algorithms to overcome the difficulty of sampling pairwise data due to the definition of AUC. Zhao et al. (2011) addressed this problem by maintaining a buffer and stored representative examples to construct the positive-negative label pair to calculate the gradient. Gao et al. (2013) maintained the mean and the covariance matrix for the streaming data and performed a gradient-based update. Ying et al. (2016) introduced the saddle point reformulation of AUC maximization with the squared loss and developed an algorithm that can update the model once receiving one data to maximize AUC. There are some future extensions of solving the saddle point formulation under different scenarios, including algorithms with fast rate under function growth condition (Liu et al., 2018), deep learning (Liu et al., 2020), proximal gradient methods (Lei & Ying, 2021), variance reduction (Dan & Sahoo, 2021). However, none of them are directly applicable in continual learning setting, since they do not take into account the catastrophic forgetting.

3 Preliminaries

Lifelong Learning. We closely follow the lifelong learning settings in Lopez-Paz et al. (2017); Chaudhry et al. (2018b); Guo et al. (2020). Specifically, we consider *task-incremental lifelong learning* in which case the tasks are arriving sequentially. Suppose we have a total of T tasks: $\{D_1, ..., D_T\}$. For each task D_i, we have a set of training examples $\{\mathbf{x}_j, y_j\}_{j=1}^{K}$. In this paper, we consider imbalanced classification, i.e., each class can have a different number of samples. The given model $f_{\mathbf{w}}$ is trained continuously on the tasks over a *single* pass of the samples. After training, the model is evaluated on the test datasets to assess its performance. The goal of task-incremental lifelong learning is to achieve high performance across all tasks. The crux of task-incremental lifelong learning is the catastrophic forgetting: the model tends to forget previously acquired knowledge while being trained on a new task.

In memory-based lifelong learning methods (Lopez-Paz et al., 2017; Chaudhry et al., 2018b; Guo et al., 2020), a replay buffer is used to store a subset of examples from old tasks. The central idea of (Lopez-Paz et al., 2017; Chaudhry et al., 2018b; Guo et al., 2020) is to utilize the replay buffer for computing gradients which serves as the reference for modifying the direction of the gradient computed on the current task.

Online AUC Optimization. AUC is defined as the probability of the score of the positive sample being larger than the negative example. Denote $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{+1, -1\}$ by feature and label respectively, and denote $\mathbf{z} = (\mathbf{x}, y)$ by the feature-label pair. We assume that $\mathbf{z}$ is sampled from an unknown distribution $\mathbb{P}$. Define $p = \Pr(y = 1) = \mathbb{E}_y \left[\mathbb{I}_{[y=1]}\right]$ as the likelihood of a random data being positive, where $\mathbb{I}(\cdot)$ is the indicator function. AUC for a general scoring function $h : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\mathrm{AUC}(h) = \Pr\left(h(\mathbf{w}; \mathbf{x}) \geq h(\mathbf{w}; \mathbf{x}') | y = 1, y' = -1\right), \tag{1}$$

where $\mathbf{w}$ is the model parameter, $\mathbf{z} = (\mathbf{x}, y)$ and $\mathbf{z}' = (\mathbf{x}', y')$ are drawn independently from $\mathbb{P}$, $h(\mathbf{w}; \mathbf{x})$ is the scoring function parameterized by $\mathbf{w}$. Following Gao et al. (2013); Ying et al. (2016); Liu et al. (2020), we use the squared function as a surrogate to replace the indication function and

end up with the following loss function:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \mathbb{E}_{\mathbf{z},\mathbf{z}'}\left[(1-h(\mathbf{w};\mathbf{x})+h(\mathbf{w};\mathbf{x}'))^2 \big| y=1, y'=-1\right].$$

The above formulation depends on pairwise data with both positive and negative labels, so it is hard to optimize in the online learning setting. It was shown in Ying et al. (2016) that the AUC maximization problem can be formulated as a minimax saddle point problem, and the stochastic gradient descent ascent algorithm can be employed to solve this saddle point problem. The saddle point reformulation is described in Proposition 1.

Proposition 1 *(Ying et al., 2016) The optimization problem (3) is equivalent to*

$$\min_{\mathbf{w}\in\mathbb{R}^d,(a,b)\in\mathbb{R}^2} \max_{\alpha\in\mathbb{R}} f(\mathbf{w},a,b,\alpha) := \mathbb{E}_{\mathbf{z}}\left[F(\mathbf{w},a,b,\alpha;\mathbf{z})\right], \tag{2}$$

where $\mathbf{z}=(\mathbf{x},y)\sim\mathbb{P}$*, and*

$$\begin{aligned} F(\mathbf{w},a,b,\alpha,\mathbf{z}) &= (1-p)\left(h(\mathbf{w};\mathbf{x})-a\right)^2\mathbb{I}_{[y=1]} + p(h(\mathbf{w};\mathbf{x})-b)^2\mathbb{I}_{[y=-1]} - p(1-p)\alpha^2 \\ &+ 2(1+\alpha)\left(ph(\mathbf{w};\mathbf{x})\mathbb{I}_{[y=-1]} - (1-p)h(\mathbf{w};\mathbf{x})\mathbb{I}_{[y=1]}\right). \end{aligned} \tag{3}$$

It is worth mentioning that DIANA is also built upon the saddle point reformulation. The main difference between our work and previous works is that our work focuses on developing memory-based lifelong learning to maximize AUC in the imbalanced continual learning setting (to alleviate catastrophic forgetting), instead of the traditional online learning setting.

4 DIANA

In this section, we introduce a novel algorithmic framework for optimizing AUC in the lifelong learning setting with an imbalanced data stream. The proposed algorithmic framework is built upon the memory-based lifelong learning methods which leverage a replay buffer for rehearsal. The essence of the algorithmic framework is to maximize the AUC score both on the current task and replay buffer as a composite optimization problem. We circumvent the requirement of a pair of samples for computing AUC score based on the literature of online AUC optimization (Ying et al., 2016; Liu et al., 2018). By decoupling conflict gradients into two models and aligning gradually, we effectively overcome the gradient interference problem caused by imbalanced data.

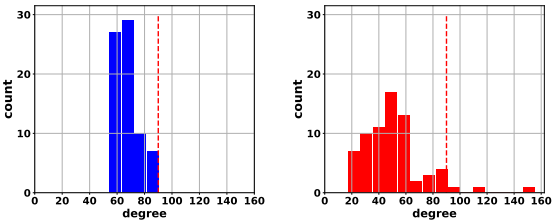
4.1 Imbalanced Lifelong Learning by Maximizing AUC

In lifelong learning, the model $f_{\mathbf{w}}$ is trained sequentially over T tasks. On each task t, the samples are arriving in a batch-wise fashion. Let $\mathbf{w}_t^k$ denote the model parameter on the k-th minibatch of the t-th task. Define $\mathbf{z}_t$ and $\hat{\mathbf{z}}_t$ as random variables following the distribution of the t-th task's data and the replay buffer's data upon t-th task respectively. To balance the current task and the replay buffer, similar to Guo et al. (2020) we define $\lambda_1(\cdot),\lambda_2(\cdot):\mathbb{R}^d\mapsto\mathbb{R}_+$ as real-valued functions which depend on the state of the model. On the k-th minibatch of the t-th task, we aim to solve the following optimization problem:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \lambda_1(\mathbf{w}_k^t)\cdot \mathrm{AUC}_t(\mathbf{w}) + \lambda_2(\mathbf{w}_k^t)\cdot \mathrm{AUC}_{\text{ref}}(\mathbf{w}) := \\ & \lambda_1(\mathbf{w}_k^t)\cdot \mathbb{E}_{\mathbf{z}_+,\mathbf{z}_-}\left[\mathrm{AUC}_t(\mathbf{w};\mathbf{z}_+,\mathbf{z}_-)\right] + \lambda_2(\mathbf{w}_k^t)\cdot \mathbb{E}_{\hat{\mathbf{z}}_+,\hat{\mathbf{z}}_-}\left[\mathrm{AUC}_{\text{ref}}(\mathbf{w};\hat{\mathbf{z}}_+,\hat{\mathbf{z}}_-)\right], \end{aligned} \tag{4}$$

where $\mathbf{w}$ is the model parameter, $\mathrm{AUC}_t(\mathbf{w})$ denotes the population AUC at the t-th task, $\mathrm{AUC}_{\text{ref}}(\mathbf{w})$ denotes the population AUC of the replay buffer, $\mathbf{z}_+$ ($\mathbf{z}_-$) and $\hat{\mathbf{z}}_+$ ($\hat{\mathbf{z}}_-$) denote random samples with positive (negative) labels on current task and replay buffer respectively, $\lambda_1(\mathbf{w}_k^t)$ and $\lambda_2(\mathbf{w}_k^t)$ characterize the scaling factors of the two AUC values on current task and replay buffer respectively on the k-th minibatch at the t-th task. The choice of the scaling factors determines the degree of prioritizing the current task or replay buffer. We choose $\lambda_1(\mathbf{w}_k^t)$ and $\lambda_2(\mathbf{w}_k^t)$ based on the model performance as in Guo et al. (2020).

However, by the pairwise formulation of AUC, to solve the problem (4), one needs to sample a pair of positive and negative examples ($\mathbf{z}_+$ and $\mathbf{z}_-$, $\hat{\mathbf{z}}_+$ and $\hat{\mathbf{z}}_-$) at every iteration, which is not feasible for lifelong learning. A natural idea to address this issue is to employ the minimax reformulation



(a) Balanced ISIC2019 (b) Imbalanced ISIC2019

Figure 1: The distributions of the angles between the current gradient and reference gradient.

(Proposition 1) of AUC (Ying et al., 2016; Liu et al., 2020), which ends up with the following problem:

$$\min_{\mathbf{w}}[\lambda_1(\mathbf{w}_k^t) \min_{(a,b)\in\mathbb{R}^2} \max_{\alpha\in\mathbb{R}} \mathbb{E}_{\mathbf{z}_t} F(\mathbf{w}, a, b, \alpha; \mathbf{z}_t) \quad + \lambda_2(\mathbf{w}_k^t) \min_{(a,b)\in\mathbb{R}^2} \max_{\alpha\in\mathbb{R}} \mathbb{E}_{\hat{\mathbf{z}}_t} F(\mathbf{w}, a, b, \alpha; \hat{\mathbf{z}}_t)], \tag{5}$$

where F is defined in Equation (3). This is equivalent to the following formulation,

$$\min_{\mathbf{w},a_1,a_2,b_1,b_2} \max_{\alpha_1,\alpha_2} \left[\lambda_1(\mathbf{w}_k^t)\mathbb{E}_{\mathbf{z}_t} F(\mathbf{w}, a_1, b_1, \alpha_1; \mathbf{z}_t) + \lambda_2(\mathbf{w}_k^t)\mathbb{E}_{\hat{\mathbf{z}}_t} F(\mathbf{w}, a_2, b_2, \alpha_2; \hat{\mathbf{z}}_t)\right]. \tag{6}$$

We can solve the problem (6) by stochastic gradient descent on variables $\mathbf{w}, a_1, a_2, b_1, b_2$ and stochastic gradient ascent on variables α_1, α_2. The stochastic gradient w.r.t. $\mathbf{w}$ is $\lambda_1(\mathbf{w}_k^t)\nabla F_{\mathbf{w}}(\mathbf{w}, a_1, b_1, \alpha_1; \mathbf{z}_t) + \lambda_2(\mathbf{w}_k^t)\nabla F_{\mathbf{w}}(\mathbf{w}, a_2, b_2, \alpha_2; \hat{\mathbf{z}}_t)$ which consists of the gradients on the current task and the gradients on the replay buffer, we refer to the gradients as the current gradient and reference gradient respectively.

4.2 Gradient Interference Problem

Imbalanced datasets are ubiquitous (Ramyachitra & Manikandan, 2014) but are largely overlooked by current research efforts on lifelong learning. We find that the imbalanced nature of the data stream poses severe challenges for optimization. Specifically, the gradients on the current task and the gradients on the replay buffer may *interfere* with each other during training due to a mismatch of the data distribution of the current task and the replay buffer. In the following, we empirically demonstrate this phenomenon on a real-world dataset.

We consider two settings: imbalanced data distribution and balanced data distribution. We construct imbalanced data from the medical dataset ISIC2019 (Gutman et al., 2016) which consists of 25331 images and 8 classes. These classes are divided into 4 disjoint tasks representing positive and negative samples, respectively. We reduce the number of positive samples to be 5% of the negative ones. The setting of balanced data is to keep the number of samples of different classes equal.

We train the models based on the formulation of Equation (6) on balanced and imbalanced data respectively. We compute the current gradient and reference gradient in each mini-batch. The angle distribution is shown in Figure 1. In the balanced case, the angles between the current gradient and the reference gradient are almost acute angles. In the imbalanced case, there appear to be more obtuse angles. That means the gradients computed on the current task interfere with the gradients computed on the replay buffer, severely affecting the training of the model.

4.3 Model Decoupling and Alignment

To address the gradient interference problem, we introduce our algorithm DIANA. The high-level idea is to use a relaxation of Equation (6) such that we can still learn useful information even if the gradients at the current task and the replay buffer are conflicting. In particular, we first note the equivalent formulation of (6):

$$\begin{aligned} &\min_{\mathbf{w},\mathbf{v},a_1,a_2,b_1,b_2} \max_{\alpha_1,\alpha_2} \left[\lambda_1(\mathbf{w}_k^t)\mathbb{E}_{\mathbf{z}_t} F(\mathbf{w}, a_1, b_1, \alpha_1; \mathbf{z}_t) + \lambda_2(\mathbf{v}_k^t)\mathbb{E}_{\hat{\mathbf{z}}_t} F(\mathbf{v}, a_2, b_2, \alpha_2; \hat{\mathbf{z}}_t)\right], \\ &\text{s.t.} \quad \mathbf{w} = \mathbf{v}. \end{aligned} \tag{7}$$

Algorithm 1 Lifelong AUC Maximization with Model Decoupling and Alignment (DIANA)

Buffer $\leftarrow \{\}$
for $t \leftarrow 1$ to T **do**
  for $k \leftarrow 1$ to $|D_t^{tr}|$ **do**
    if Buffer $\neq \{\}$ **then**
      $\zeta_k^t \leftarrow$ SAMPLE(Buffer)
      Compute $\lambda_1(\mathbf{w}_k^t)$ and $\lambda_2(\mathbf{v}_k^t)$
    else
      Set $\lambda_1(\mathbf{w}_k^t) = 1$ and $\lambda_2(\mathbf{v}_k^t) = 0$.
    end if
    Update $\mathbf{w}, \mathbf{v}, a_1, a_2, b_1, b_2$ by one step of stochastic gradient descent w.r.t. the objective function defined in (8)
    Update α_1, α_2 by one step of stochastic gradient ascent w.r.t. the objective function defined in (8) .
    Buffer $\leftarrow$ Buffer$\bigcup(\xi_k^t)$
    Remove samples if the Buffer is full.
  end for
end for

Since using the same model for data distribution of different tasks would lead to the gradient interference problem, we propose a model decoupling and alignment technique to address the issue. In particular, we propose to solve the following problem (8) to relax the equality constraint in (7),

$$\min_{\mathbf{w},\mathbf{v},a_1,a_2,b_1,b_2} \max_{\alpha_1,\alpha_2} \left[\lambda_1(\mathbf{w}_k^t)\mathbb{E}_{\mathbf{z}_t} F(\mathbf{w}, a_1, b_1, \alpha_1; \mathbf{z}_t) + \lambda_2(\mathbf{v}_k^t)\mathbb{E}_{\hat{\mathbf{z}}_t} F(\mathbf{v}, a_2, b_2, \alpha_2; \hat{\mathbf{z}}_t)\right] + \beta \cdot \text{dist}(\mathbf{w}, \mathbf{v}), \tag{8}$$

where $\text{dist}(\cdot, \cdot)$ denotes a distance function between two models, and $\beta > 0$ is a penalty parameter. The typical choice of dist can be squared loss, distillation loss (Hinton et al., 2015), etc. The term $\beta \cdot \text{dist}(\mathbf{w}, \mathbf{v})$ is referred to as the alignment penalty.

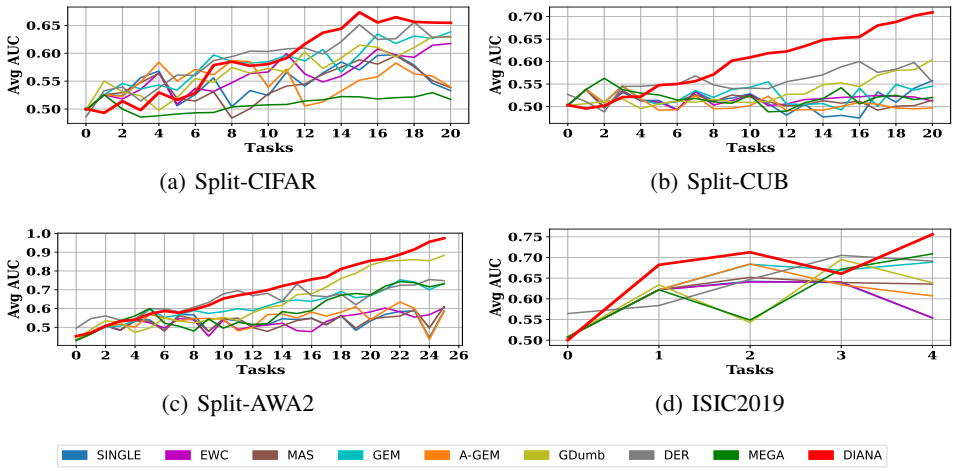
In view of Equation (8), we know that we are decoupling one model as in (7) into two models $\mathbf{w}$ and $\mathbf{v}$ under the coordination of an alignment penalty, which has the following two benefits. First, the formulation is a standard minimax optimization, where stochastic gradient descent ascent suffices to solve it efficiently. Second, it can partially alleviate the gradient interference problem since conflicting gradients are decoupled. For example, in DIANA, the gradient w.r.t. $\mathbf{w}$ is $\lambda_1(\mathbf{w}_k^t)\nabla_{\mathbf{w}} F(\mathbf{w}, a_1, b_1, \alpha_1; \mathbf{z}_t) + \beta\nabla_{\mathbf{w}}\text{dist}(\mathbf{w}, \mathbf{v})$. The gradient w.r.t. $\mathbf{w}$ consists of two terms, the first term represents the current gradient which depends on the current data, while the second term characterizes the gradient of the distance function between two models and it is independent of data. Essentially, we undermine the impact of some noisy gradients on the model update. By pulling the current model close to the model on the replay buffer *in each step*, the current model can essentially *learn* from the model on the replay buffer to retain performance on old tasks.

For implementation, since we cannot solve (8) exactly, we use one step of stochastic gradient descent ascent as an approximate solution. This is consistent with the literature of memory-based lifelong learning (Chaudhry et al., 2018b; Guo et al., 2020). Please refer to Algorithm 1 for details.

5 Experiments

5.1 Experimental Setups

Datasets: We perform experiments on popular lifelong learning benchmarks, Split-CIFAR (Zenke et al., 2017), Split-CUB and Split-AWA2 (Chaudhry et al., 2018b). Moreover, to further explore the application of lifelong AUC Maximization in industry, a medical dataset ISIC2019 (Gutman et al., 2016) and a satellite dataset EuroSat (Helber et al., 2019) are also introduced as new benchmarks. The medical dataset ISIC2019 consists of 25331 medical images. The goal for ISIC2019 is to classify dermoscopic images among 8 different diagnostic categories: Melanoma, Melanocytic nevus, Basal cell carcinoma, Actinic keratosis, Benign keratosis Dermatofibroma, Vascular lesion, Squamous cell carcinoma, and none of them. While satellite dataset EuroSat covers 13 spectral bands and consists of 10 classes with in total of 27,000 labeled and geo-referenced images, including cities, rivers, and forests. We believe it is important to measure performance in real industrial scenarios, especially for the medical scenario, which is naturally imbalanced and prefers AUC as the criterion rather than accuracy. A false positive causes severe consequences, such as fault diagnosis of cancers. Split-CIFAR and Split-CUB consist of 20 tasks, while Split-AWA2 has 25 tasks, ISIC2019 has 4



(a) Split-CIFAR (b) Split-CUB

(c) Split-AWA2 (d) ISIC2019

Figure 2: Evolution of average AUC during the lifelong learning process. For the results on EuroSat, please see Figure 5 in Appendix.

tasks and EuroSat has 5 tasks. For ISIC2019, every two classes constitute a task (4 tasks in total since ISIC2019 has 8 classes). Similar to ISIC2019, we construct one task with data from two classes (5 tasks in total since EuroSat has 10 classes).

Make Imbalanced: Following Yuan et al. (2021), we make training set imbalanced with a predefined imbalanced ratio (**imratio**) and leave validation set and test set unchanged. According to chosen imbalanced ratio, positive samples are randomly discarded, until the ratio of positive samples to all samples equals the imbalanced ratio. Except for multi-class balanced classification and the imbalanced ratio ablation in Appendix, we set imratio=0.05 in all experiments, which means that only 5% data are positive. If a task has multiple classes, half of the classes are regarded as negative and the rest are regarded as positive. For example, Split-CIFAR is separated into 20 disjoint tasks, each task contains 5 classes, classes 0-3 are regarded as negative, and classes 4-5 are positive.

Metrics: Since our purpose is to maximize the AUC score, AUC (AUC_T) is used as the primary metric. AUC_T represents the averaged AUC value when finishing training the T-th task, where T is the total number of tasks. On the other hand, to be consistent with previous works (Kamp et al., 2018), Accuracy (ACC_T) and Forgetting (FGT_T) are also reported. ACC_T is the average accuracy tested on all tasks after finishing training T-th task. FGT_T measures the drop of AUC on past tasks after training on the T-th task. It's defined as $FGT_T = \frac{1}{T-1}\sum_{j=1}^{T-1}(\max_{l\in\{1,\cdots,T-1\}} AUC_{l,j} - AUC_{T,j})$. where $AUC_{l,j}$ is the AUC score tested on the j-th task after training on the l-th task.

Implementation Details: For Split-CIFAR, ISIC2019 and EuroSAT, we use a reduced ResNet18 (Chaudhry et al., 2018b) to handle small input resolution. For Split-CUB and Split-AWA2, we use a standard ResNet18 pre-trained model on ImageNet. We set batch size as 64 for Split-CIFAR, Split CUB, and Split-AWA2, while batch size as 128 for ISIC2019 and EuroSAT. The learning rate is 0.1 across different datasets and methods. As to memory size for each task, it's fixed to 64 for Split-CIFAR, Split CUB, and Split-AWA. Fixed to 128 for ISIC2019 and EuroSAT.

Baselines: We consider EWC (Kirkpatrick et al., 2017) , MAS (Aljundi et al., 2018), GEM (Lopez-Paz et al., 2017), A-GEM (Chaudhry et al., 2018b), MEGA (Guo et al., 2020), DER (Buzzega et al., 2020) and GDumb (Prabhu et al., 2020) as baselines. EWC and MAS are regularization-based approaches. GEM, A-GEM, MEGA, DER, and GDumb are built upon episodic memory. We also consider a simple baseline that uses stochastic gradient descent to train these tasks sequentially without any memory or regularization. It's marked as SINGLE in our experiments. All baselines are task-incremental and only require one-pass, so class-incremental(Shim et al., 2021; Mai et al., 2021) and multiple-passes methods (Ebrahimi et al., 2020; Mallya & Lazebnik, 2018) are not considered.

For a fair comparison, we use the same memory size and regular reservoir sampling (Chaudhry et al., 2019) for all the memory-based methods, including GEM, A-GEM, DER, MEGA, and DIANA. We want to point out that since the benchmark datasets are imbalanced, to be fair, GDumb is also implemented with reservoir sampling (Chaudhry et al., 2019) instead of class-balanced sampling. We

Table 1: Comparison of ONE-MODEL and TWO-MODEL (DIANA)

Method	Split-CIFAR *AUC* (↑)	Split-CUB *AUC* (↑)	Split-AWA2 *AUC* (↑)	ISIC2019 *AUC* (↑)	EuroSat *AUC* (↑)
ONE-MODEL	58.7 ± 1.7	$\mathbf{73.5 \pm 2.3}$	79.1 ± 2.7	71.3 ± 9.2	72.4 ± 6.5
TWO-MODEL	$\mathbf{68.4 \pm 2.0}$	70.4 ± 0.7	$\mathbf{98.2 \pm 1.1}$	$\mathbf{73.6 \pm 4.8}$	$\mathbf{86.7 \pm 1.5}$

Table 2: Comparison of capacity on Split-CIFAR100

Method	architecture	AUC(↑) %	memory	GFLOPS	params
ONE-MODEL	ResNet34	56.2	1280	0.11	**2.50M**
TWO-MODEL	ResNet18(x2)	**68.4**	1280	**0.12**	2.24M

also compare our method with Rainbow-memory (RM) (Bang et al., 2021) and CBRS (Chrysakis & Moens, 2020), which use class-balanced sampling. The results are presented in Appendix A. We show that our method and class-balanced sampling can be combined to further improve performance.

5.2 RESULTS

In Figure 2, we present the evolution of average AUC during the lifelong learning process. We can observe that DIANA dominates the baselines in most cases. An interesting observation is that DIANA begins to gain advantages over baselines as the training proceeds. One plausible reason is that it is harder to learn useful features for maximizing the AUC objective in the initial training process. However, as shown in Figure 2, by optimizing the right objective as in the proposed DIANA, we can achieve a high average AUC score in the long run. This observation indicates that DIANA has the potential to handle large amounts of tasks.

We report results of AUC_T, ACC_T, and FGT_T for all algorithms on the imbalanced benchmarks. Details can be found in Appendix: Figure 4, Table 6 and Table 7. In terms of AUC score, our DIANA outperforms other baselines with a large margin. In particular, compared with the best baseline, DIANA improves 2.9% on Split-CIFAR, 9.0% on Split-CUB, 13.0% on Split-AWA2, 1.4% on ISIC2019, and 6.2% on EuroSat. This shows that it is more effective to directly optimize AUC as in the proposed DIANA. It can also be observed that DIANA achieves the highest Accuracy ACC_T on all the datasets except ISIC2019, this further shows the discriminativeness of the learned features.

One interesting observation is that GDumb generally achieves a high AUC than other baselines on the imbalanced benchmarks. However, it is worth noting that GDumb is trained *offline* on the replay buffer with *multiple* passes over the data. GDumb avoids the gradient interference problem by only leveraging the gradients on the replay buffer. However, in real-world lifelong learning applications, the model needs to quickly adapt to the data with a *single* pass. This impedes the practical applicability of GDumb. In contrast, DIANA is trained *online* with a *single* over the samples which can be applied in real-time applications.

One model vs. two models. To verify the gradient interference problem discussed in Section 4.2, we conduct ablation on ONE-MODEL and TWO-MODEL. ONE-MODEL denotes the method of solving Equation (6) which is a naive combination of AUC maximization and memory-based lifelong learning. It would suffer from the gradient interference problem due to conflict gradients computed on the current task and the replay buffer. TWO-MODEL denotes the method of solving Equation (8) (a.k.a., DIANA). We denote the model $\mathbf{w}$ and $\mathbf{v}$ in Equation (8) as the current model and reference model respectively. Then the gradients of TWO-MODEL approach are the following. The gradient w.r.t. current (reference) model is the summation over two parts: the gradient calculated based on current (reference) task data, and the gradient of the distance function between two models.

As shown in Table 1, TWO-MODEL outperforms ONE-MODEL in all benchmarks except for Split-CUB. In particular, TWO-MODEL is better than ONE-MODEL by $+10.3\%$, $+19.9\%$, $+2.3\%$, and $+14.3\%$ on Split-CIFAR, Split-AWA2, ISIC2019, and EuroSat respectively.

We conduct a detailed analysis on the reason why TWO-MODEL is better than ONE-MODEL on Split-CIFAR but a bit worse on Split-CUB. We probe this problem by observing the angle between the current gradient and reference gradient in ONE-MODEL. Concretely, we follow GEM and A-GEM by firstly storing the gradients computed on the current mini-batch and the episodic memory, then calculating the angle between them. We repeat this the process in each iteration and show the distribution the angles on two standard datasets (balanced and imbalanced) with a histograms in Figure 8 in Appendix. Especially, we experimentally find that 9.68% of the angles are obtuse in

Table 3: The multi-class results of average AUC, average ACC, and average Forgetting (FGT) of different methods on Split CIFAR100, Split CUB200, and Split AWA2.

Method	Split-CIFAR			Split-CUB			Split-AWA2		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
SINGLE	75.8 ± 2.0	42.8 ± 3.4	12.7 ± 2.1	91.9 ± 2.4	48.1 ± 6.3	3.2 ± 2.3	51.0 ± 2.1	58.0 ± 0.7	28.4 ± 2.9
EWC	76.4 ± 1.8	43.9 ± 2.4	11.9 ± 1.4	91.5 ± 2.6	47.2 ± 6.3	3.8 ± 2.5	51.7 ± 1.3	57.9 ± 1.6	28.7 ± 3.8
MAS	78.2 ± 1.7	44.5 ± 3.3	8.1 ± 1.9	92.7 ± 2.1	49.9 ± 6.2	2.6 ± 2.1	50.1 ± 0.1	53.6 ± 6.2	28.5 ± 3.9
A-GEM	81.6 ± 0.9	54.5 ± 1.7	7.2 ± 0.9	92.8 ± 0.8	54.4 ± 7.5	1.1 ± 1.1	51.1 ± 3.2	59.6 ± 0.8	27.3 ± 1.6
Gdumb	75.9 ± 0.9	49.4 ± 1.4	2.6 ± 0.4	92.8 ± 1.8	$\mathbf{65.5 \pm 2.3}$	$0.3 \pm$ s0.2	54.4 ± 3.4	60.8 ± 2.7	14.1 ± 6.1
DER	80.3 ± 1.2	40.7 ± 2.5	8.9 ± 1.2	92.6 ± 2.8	59.5 ± 9.0	1.5 ± 0.7	71.3 ± 16.0	63.1 ± 3.4	15.3 ± 9.2
MEGA	62.2 ± 2.7	32.3 ± 2.4	10.8 ± 2.2	89.6 ± 4.0	50.6 ± 7.3	4.3 ± 2.5	51.1 ± 1.0	59.6 ± 5.6	12.8 ± 2.9
DIANA	$\mathbf{89.5 \pm 0.3}$	$\mathbf{65.5 \pm 0.9}$	$\mathbf{1.2 \pm 1.2}$	$\mathbf{93.4 \pm 0.6}$	64.3 ± 1.8	$\mathbf{0.2 \pm 0.1}$	$\mathbf{92.6 \pm 1.4}$	$\mathbf{82.5 \pm 2.3}$	$\mathbf{0.3 \pm 0.2}$

imbalanced Split-CIFAR; while only 3.16% are obtuse in imbalanced Split-CUB. Fewer obtuse angles indicate the gradient interference problem is less severe, so ONE-MODEL is better on Split-CUB. One possible reason for the Split-CUB dataset to have fewer obtuse angles is that the images in Split-CUB dataset are fine-grained and different tasks have high similarity, so the gradients between tasks and memory are more likely to be similar.

We have the following conjecture based on the results above: when the data on different tasks has very different distributions, TWO-MODEL is preferred over ONE-MODEL and vice versa.

Furthermore, to eliminate the concern of unfair comparison, we compare the capacity in terms of episodic memory size, computation (GFLOPS, i.e., one billion (10^9) floating-point operations per second), and the number of parameters. We enlarged the one-model approach's model architecture to match the two-model storage. Table 2 presents the comparison under same capacity. With nearly the same capacity, TWO-MODEL still outperforms ONE-MODEL. In conclusion, it's the decoupled and aligned mechanism itself that benefits imbalanced lifelong learning, not the extra model parameters.

Multi-class Balanced Classification. To further verify the effectiveness of our algorithm on general lifelong learning, we conduct experiments on standard multi-class classification benchmarks without making datasets imbalanced. Similar to most of the existing AUC maximization literature, our algorithm focuses on binary classification. In order to apply DIANA for general lifelong learning problems, we extend our method to accommodate multi-class AUC maximization following works (Liu et al., 2020; Yang et al., 2021). If there are c classes, we have c output scores from the network, one score for each class. If a sample belongs to i-th class, the corresponding score at i-th position is treated as positive and the rest scores negative, a binary AUC loss is calculated based on these scores. By iterating over c classes, multi-class AUC loss accumulates. To obtain AUC metric in multi-class form, we calculate AUC score for each class pair (i.e., one versus all) and then perform the average.

Table 3 presents results on standard Split-CIFAR, Split-CUB200 and Split-AWA2. Because most of the baselines have reported the accuracy and tuned hyper-parameters on Split-CIFAR, we just follow their settings. As to Split-CUB200 and Split-AWA2, the setups follow Section 5.1. DIANA outperforms other baselines in terms of AUC, Accuracy, and forgetting. Particularly, on Split-CUB200, DIANA obtains a slightly lower accuracy than Gdumb but the highest on the AUC metric. The results show that our method is not only suitable for imbalanced scenarios but also works well under general lifelong learning settings such as multi-class classification on balanced datasets.

More ablations in Appendix. Due to limited space, more ablations are studied in Appendix, including results with balanced sampling in Appendix A, imbalanced ratio in Appendix C, balanced vs. imbalanced in Appendix D, and space and time complexity in Appendix E.

6 Conclusion

In this paper, we study AUC optimization under imbalanced continual learning settings. We propose a novel algorithm DIANA based on the minimax reformulation of the AUC objective. We systematically study the gradient interference problem on imbalanced data, both theoretically and empirically. We demonstrate that this problem can be alleviated by employing two models with model decoupling and alignment: one for the current task and the other for past tasks, and they gradually align during the training. We extend our algorithm to multi-class AUC maximization in general balanced lifelong learning. Compared to existing memory-based and regularization-based approaches, the proposed algorithm achieves a higher AUC score as well as less forgetting. One limitation is that our approach uses two models and increases slightly computational cost.

REFERENCES

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)*, pp. 139–154, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pp. 11816–11825, 2019.

Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8218–8227, 2021.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in Neural Information Processing Systems*, 33:14879–14890, 2020.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018a.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018b.

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.

Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pp. 1952–1961. PMLR, 2020.

Soham Dan and Dushyant Sahoo. Variance reduced stochastic proximal algorithm for auc maximization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 184–199. Springer, 2021.

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.

Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *European Conference on Computer Vision*, pp. 386–402. Springer, 2020.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. *arXiv preprint arXiv:1910.07104*, 2019.

Wei Gao, Rong Jin, Shenghuo Zhu, and Zhi-Hua Zhou. One-pass auc optimization. In *International conference on machine learning*, pp. 906–914. PMLR, 2013.

Yunhui Guo, Mingrui Liu, Tianbao Yang, and Tajana Rosing. Improved schemes for episodic memory-based lifelong learning. *Advances in Neural Information Processing Systems*, 33, 2020.

David A. Gutman, Noel C. F. Codella, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Nabin K. Mishra, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). *CoRR*, abs/1605.01397, 2016.

James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

James A Hanley and Barbara J McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

Tyler L Hayes, Giri P Krishnan, Maxim Bazhenov, Hava T Siegelmann, Terrence J Sejnowski, and Christopher Kanan. Replay in deep learning: Current approaches and missing biological elements. *Neural Computation*, 33(11):2908–2950, 2021.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ke Hu, Yi Qi, Jianqiang Huang, Jia Cheng, and Jun Lei. Continual learning for ctr prediction: A hybrid approach. *arXiv preprint arXiv:2201.06886*, 2022.

Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 590–597, 2019.

Xisen Jin, Arka Sadhu, Junyi Du, and Xiang Ren. Gradient-based editing of memory examples for online task-free continual learning. *Advances in Neural Information Processing Systems*, 34: 29193–29205, 2021.

Michael Kamp, Linara Adilova, Joachim Sicking, Fabian Hüger, Peter Schlicht, Tim Wirtz, and Stefan Wrobel. Efficient decentralized deep learning by dynamic model averaging. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 393–409. Springer, 2018.

Chris Dongjoo Kim, Jinseo Jeong, and Gunhee Kim. Imbalanced continual learning with partitioning reservoir sampling. In *European Conference on Computer Vision*, pp. 411–428. Springer, 2020.

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.

Yunwen Lei and Yiming Ying. Stochastic proximal auc maximization. *Journal of Machine Learning Research*, 22(61):1–45, 2021.

Mingrui Liu, Xiaoxuan Zhang, Zaiyi Chen, Xiaoyu Wang, and Tianbao Yang. Fast stochastic auc maximization with o (1/n)-convergence rate. In *International Conference on Machine Learning*, pp. 3195–3203, 2018.

Mingrui Liu, Zhuoning Yuan, Yiming Ying, and Tianbao Yang. Stochastic auc maximization with deep neural networks. In *International Conference on Learning Representations*, 2020.

David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pp. 6467–6476, 2017.

Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3589–3599, 2021.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.

Quang Pham, Chenghao Liu, Doyen Sahoo, and HOI Steven. Contextual transformation networks for online continual learning. In *International Conference on Learning Representations*, 2020.

Ameya Prabhu, Philip Torr, and Puneet Dokania. Gdumb: A simple approach that questions our progress in continual learning. August 2020.

D Ramyachitra and P Manikandan. Imbalanced dataset classification and solutions: a review. *International Journal of Computing and Business Research (IJCBR)*, 5(4):1–29, 2014.

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.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

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.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9630–9638, 2021.

Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. Incremental learning for semantic segmentation of large-scale remote sensing data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3524–3537, 2019.

Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9385–9394, 2021.

Zhiyong Yang, Qianqian Xu, Shilong Bao, Xiaochun Cao, and Qingming Huang. Learning with multiclass auc: Theory and algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Yiming Ying, Longyin Wen, and Siwei Lyu. Stochastic online auc maximization. *Advances in neural information processing systems*, 29:451–459, 2016.

Zhuoning Yuan, Yan Yan, Milan Sonka, and Tianbao Yang. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. 2021.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3987–3995. JMLR. org, 2017.

Peilin Zhao, Rong Jin, Tianbao Yang, and Steven C Hoi. Online auc maximization. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 233–240, 2011.

Table 4: The results of memory-based methods with Class Balanced Reservoir Sampling on Split CIFAR100, Split CUB200, and Split AWA2.

Method	Split-CIFAR			Split-CUB			Split-AWA2		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
A-GEM	58.3 ± 3.5	60.3 ± 0.3	19.5 ± 3.6	50.8 ± 4.0	50.2 ± 0.3	17.6 ± 2.5	55.4 ± 1.8	49.9 ± 4.0	30.6 ± 2.6
Gdumb	70.6 ± 1.4	63.1 ± 0.9	6.3 ± 0.5	66.8 ± 5.5	**55.3 ± 5.2**	4.3 ± 2.7	98.4 ± 0.8	92.5 ± 2.6	0.2 ± 0.6
DER	64.9 ± 3.2	58.3 ± 1.4	13.4 ± 3.3	65.0 ± 5.8	51.7 ± 2.0	10.3 ± 3.2	86.1 ± 14.1	71.6 ± 8.0	10.6 ± 11.4
MEGA	66.1 ± 5.7	59.4 ± 2.6	3.2 ± 1.5	50.8 ± 1.1	50.0 ± 0.4	13.0 ± 2.1	57.1 ± 6.3	50.5 ± 5.9	17.3 ± 1.4
RM	**77.4 ± 1.9**	**69.8 ± 1.7**	4.3 ± 1.9	58.8 ± 2.8	54.4 ± 1.1	11.3 ± 1.2	80.5 ± 6.0	71.1 ± 8.8	5.6 ± 2.0
DIANA	76.7 ± 1.0	69.3 ± 1.7	**3.1 ± 0.7**	**73.4 ± 2.1**	51.0 ± 0.5	**0.2 ± 0.3**	**99.5 ± 0.2**	**87.4 ± 2.9**	**0.1 ± 0.1**

Table 5: The results of memory-based methods with Class Balanced Reservoir Sampling on medical images ISIC2019 and satellite images EuroSat.

Method	ISIC2019			EuroSat		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
A-GEM	66.5 ± 10.3	**90.6 ± 0.1**	12.8 ± 11.9	79.4 ± 4.4	55.6 ± 4.0	14.4 ± 5.0
Gdumb	57.2 ± 9.0	71.1 ± 15.9	22.9 ± 9.1	81.6 ± 8.4	73.4 ± 8.9	2.4 ± 6.6
DER	61.8 ± 13.7	84.5 ± 12.6	17.7 ± 17.1	81.2 ± 4.5	65.3 ± 6.2	7.4 ± 4.2
MEGA	54.9 ± 9.6	62.3 ± 15.9	18.8 ± 10.7	77.8 ± 7.3	69.7 ± 10.3	2.0 ± 3.6
RM	72.8 ± 5.0	71.5 ± 3.7	10.3 ± 2.5	86.3 ± 2.6	79.3 ± 1.5	0.7 ± 4.2
DIANA	**77.7 ± 4.2**	78.2 ± 4.7	**1.9 ± 3.2**	**90.9 ± 1.7**	**83.8 ± 2.7**	**-1.0 ± 1.8**

A DIANA OUTPERFORMS OTHER METHODS WITH CLASS BALANCED SAMPLING

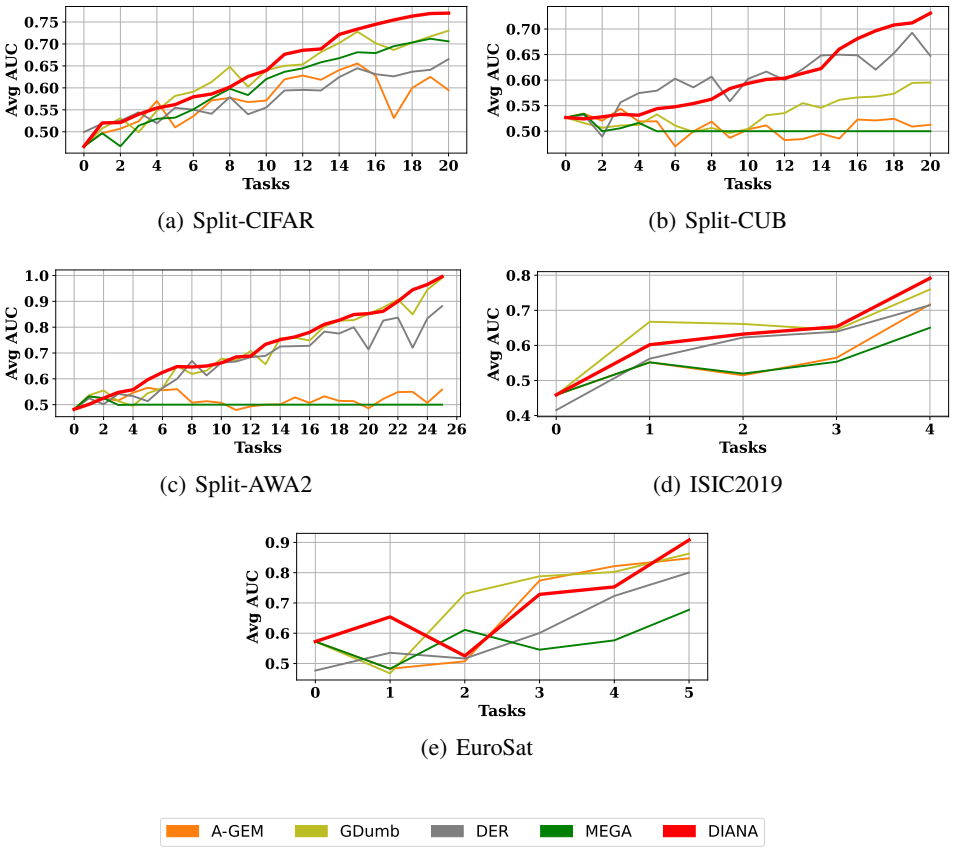
For imbalanced lifelong learning, class balanced sampling is an efficient way to alleviate the harm of imbalanced data stream in some cases. Class-Balancing Reservoir Sampling (CBRS) (Chrysakis & Moens, 2020) is an enhanced Reservoir Sampling strategy for memory-based methods under imbalanced lifelong learning settings. It maintains a class-balanced memory by replacing instances of the major class. Thus, in experience replay, training batches from episodic memory are collected in a balanced form. However, since memory-based methods require data from both the current task and episodic memory, they still need to handle an imbalanced data stream from the current task even though the memory is balanced. Moreover, the distributions of balanced and imbalanced data streams differ significantly, which may lead to server gradient interference between these two streams.

We examine memory-based approaches with CBRS instead of vanilla Reservoir Sampling, the learning rate and batch size are consistent with section 5.1. We implement CBRS according to the pseudo-code described in Chrysakis & Moens (2020). When the memory is not full, all samples are stored in memory. After the memory is full, all classes are categorized as the full class and the not-full class. Once a class has been the largest class in memory, it's marked as the full class and would be ignored by the population. On the contrary, the instances of the full class would be replaced by the not-full class.

Table 4 and 5 present the results with CBRS. The evolution of average AUC during the lifelong learning process is shown in Figure 3. GEM is not included because this algorithm is not compatible with CBRS, it uses Ring Buffer instead of Reservoir Sampling and each task has individual memory space. Compared with vanilla Reservoir sampling in Table 6 and 7, using CBRS significantly improves the performance of most algorithms, except A-GEM. On Split-CIFAR, A-GEM, GDumb, DER, MEGA, and DIANA increase 1.5%, 5.1%, 2.5%, 3.0%, and 8.3% in terms of AUC score respectively. We also compare DIANA with another class-balanced method Rainbow Memory (RM), it uses a different class-balanced sampling strategy. Our proposed DIANA beats RM except Split-CIFAR.

We give a detailed analysis of each method below.

- A-GEM obtains no benefits from CBRS. Since the current data stream is still imbalanced, we analyze that it's not suitable for A-GEM to rectify the current gradient based on the reference gradient which is computed on a balanced replay buffer.



(a) Split-CIFAR (b) Split-CUB

(c) Split-AWA2 (d) ISIC2019

(e) EuroSat

Figure 3: Evolution of average AUC of memory-based methods with Class Balanced Reservoir Sampling.

- GDumb has stable and significant increments on all 5 datasets, because it's only trained on balanced memory data, and doesn't suffer from the gradient interference problem.
- DER increases on Split-CIFAR, Split-AWA2, and EuroSat, while dropping performance on ISIC2019 and Split-CUB.
- MEGA gets boosted on Split-CIFAR and Split-CUB, but drops on the other three datasets.
- DIANA has much better AUC than other methods as shown in Table 4 and Table 5. There are two interesting observations. First, on ISIC2019 dataset, DIANA has worse accuracy than A-GEM (78.2% versus 90.6%) but better AUC value (77.7% versus 66.5%). The reason is that the dataset ISIC2019 is very imbalanced (See Section 5.1 Datasets). Second, on EuroSat dataset, DIANA has a negative forgetting measure, which means that DIANA with CBRS can help learn previous tasks when learning new tasks.

In summary, DIANA achieves higher AUC than other memory-based approaches when applying Class-Balancing Reservoir Sampling. DIANA has consistent improvements on all 5 datasets. We have a conjecture that the two-model framework alleviates the gradient interference between imbalanced data stream and balanced replay buffer, according to Section 4.2.

B Result Detail

We show the detailed results of all the methods on imbalanced benchmarks and practical data in Figure 4, Table 6 and Table 7.

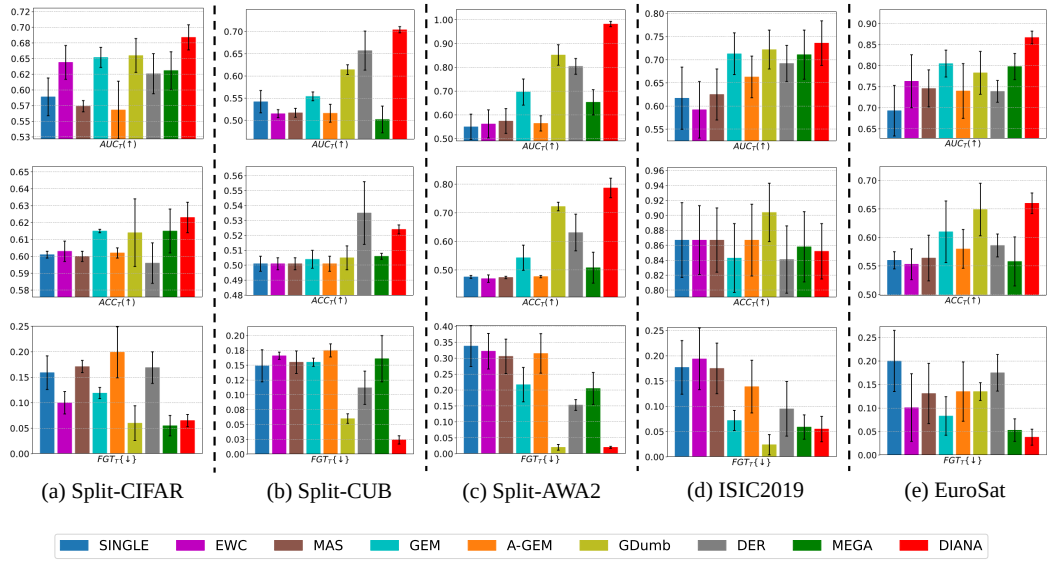


Figure 4: Performance of DIANA and other baselines on Split-CIFAR, Split-CUB, Split-AWA2, ISIC, and EuroSAT, each result is averaged after 5 runs with different random seeds. The Standard Deviation (std) is represented by the black line.

Table 6: The results of average AUC, average ACC, and average Forgetting (FGT) of different methods on Split CIFAR100, Split CUB200, and Split AWA2.

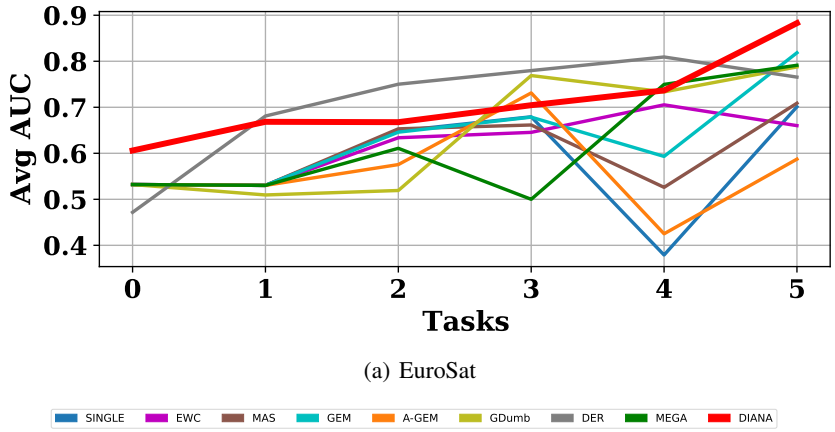
Method	**Split-CIFAR**			**Split-CUB**			**Split-AWA2**		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
SINGLE	58.9 ± 3.0	60.1 ± 0.2	15.9 ± 3.3	54.2 ± 2.5	50.1 ± 0.5	14.9 ± 2.7	55.0 ± 5.3	47.6 ± 0.5	33.8 ± 6.4
EWC	64.4 ± 2.7	60.3 ± 0.6	10.0 ± 2.2	51.5 ± 0.9	50.1 ± 0.4	16.6 ± 0.6	56.2 ± 5.9	47.0 ± 1.3	32.2 ± 5.6
MAS	57.4 ± 0.9	60.0 ± 0.3	17.1 ± 1.2	51.7 ± 1.0	50.1 ± 0.4	15.5 ± 1.9	57.4 ± 5.2	47.4 ± 1.3	30.6 ± 5.4
GEM	65.2 ± 1.6	61.5 ± 0.1	11.9 ± 1.1	55.4 ± 1.0	50.4 ± 0.6	15.5 ± 0.7	69.6 ± 5.5	54.3 ± 4.4	21.7 ± 5.4
A-GEM	56.8 ± 4.6	60.2 ± 0.3	19.9 ± 5.0	51.6 ± 2.0	50.1 ± 0.5	17.5 ± 1.1	56.4 ± 3.2	47.7 ± 0.4	31.5 ± 6.2
GDumb	65.5 ± 2.7	61.4 ± 2.0	6.0 ± 3.4	61.4 ± 1.1	50.5 ± 0.8	6.0 ± 0.8	85.2 ± 4.3	72.2 ± 1.5	**1.9 ± 0.9**
DER	62.6 ± 3.2	59.6 ± 1.2	16.9 ± 3.1	65.7 ± 4.4	**53.5 ± 2.1**	11.2 ± 3.8	80.4 ± 3.3	63.1 ± 6.4	15.3 ± 1.7
MEGA	63.1 ± 3.0	61.5 ± 1.3	**5.5 ± 2.0**	50.1 ± 3.0	50.6 ± 0.3	16.1 ± 3.9	65.3 ± 5.3	50.8 ± 5.4	20.5 ± 5.0
DIANA	**68.4 ± 2.0**	**62.3 ± 0.9**	6.6 ± 1.2	**70.4 ± 0.7**	52.3 ± 0.3	**2.4 ± 0.7**	**98.2 ± 1.1**	**78.7 ± 3.4**	2.0 ± 0.3

Table 7: The results of average AUC, average ACC, and average Forgetting (FGT) of different methods on medical images ISIC2019 and satellite images EuroSat.

Method	**ISIC2019**			**EuroSat**		
	AUC (↑)%	*ACC* (↑)%	*FGT* (↓)%	*AUC* (↑)%	*ACC* (↑)%	*FGT* (↓)%
SINGLE	61.7 ± 7.7	86.7 ± 5.0	17.7 ± 5.3	69.3 ± 6.0	56.0 ± 1.5	20.0 ± 6.5
EWC	59.2 ± 8.1	86.7 ± 5.0	19.4 ± 6.1	76.3 ± 6.3	55.3 ± 2.7	10.1 ± 7.2
MAS	62.5 ± 6.5	86.7 ± 5.0	17.5 ± 6.0	74.6 ± 4.4	56.4 ± 4.0	13.1 ± 6.4
GEM	71.3 ± 6.5	84.3 ± 4.6	7.2 ± 2.0	80.5 ± 3.2	61.0 ± 5.4	8.3 ± 4.1
A-GEM	66.3 ± 6.5	86.7 ± 4.8	13.9 ± 5.2	74.0 ± 6.5	58.0 ± 3.4	13.5 ± 6.3
GDumb	72.2 ± 4.2	**90.4 ± 0.5**	**4.4 ± 2.0**	78.3 ± 5.5	64.9 ± 4.6	4.8 ± 1.9
DER	69.2 ± 3.9	84.1 ± 4.5	9.5 ± 5.4	73.9 ± 2.6	58.6 ± 2.0	17.5 ± 3.9
MEGA	71.1 ± 5.3	85.8 ± 4.7	5.9 ± 2.4	79.8 ± 3.1	55.8 ± 4.3	5.3 ± 2.4
DIANA	**73.6 ± 4.8**	85.2 ± 3.7	5.5 ± 2.5	**86.7 ± 1.5**	**66.0 ± 1.8**	**3.8 ± 1.7**

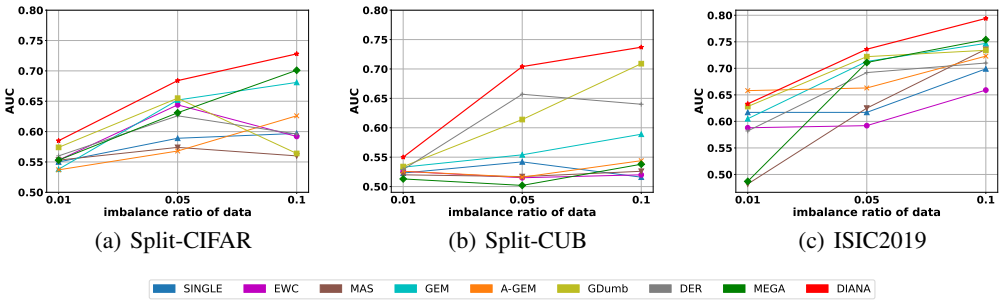
C IMBALANCED RATIO

In Figure 6, we vary the imbalanced ratio to evaluate the robustness of different methods. We consider three imratio settings: $\{0.01, 0.05, 0.1\}$. As expected, when the imratio increases, all the algorithms generally achieve better performance. Across all three settings, DIANA achieves the highest performance except when $imratio = 0.01$ on ISIC2019. It is worth mentioning that when imratio is 0.01, i.e., only 1% of samples are positive, all the methods drop drastically including



(a) EuroSat

Figure 5: Evolution of average AUC on EuroSat.



(a) Split-CIFAR (b) Split-CUB (c) ISIC2019

Figure 6: Average AUC on the data with different imbalanced ratio.

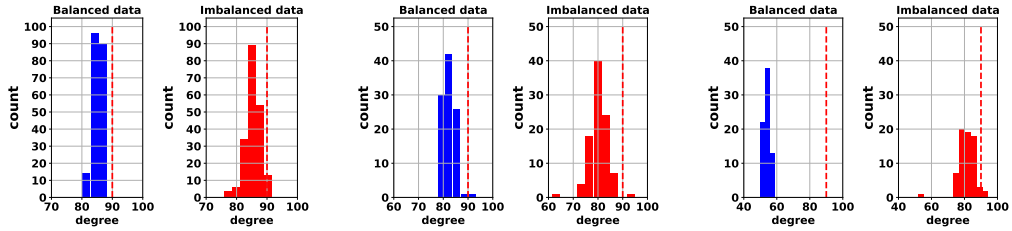
DIANA. The possible reason is that when imratio is 0.01, the positive samples are very rare so it is difficult for all the methods to learn efficiently.

D BALANCED VERSUS IMBALANCED

In this section, it's studied how imbalance affects lifelong learning. In hyperplane, if the direction of gradient descent changes a lot, it's probably unstable and hard to optimize. Thus, we can exploit the deviation of direction to indicate whether a task is getting harder or not. We first store all gradients in a task and get an averaged gradient as an anchor. Then calculate the angle between the anchor and each mini-batch gradient. Drawing the histograms of angle deviation, we can see the distribution of gradient direction before and after making it imbalanced. It's plotted in Figure 7. After making imbalanced, the histograms have a wider range and larger std, which means the directions of gradient vary more dramatically.

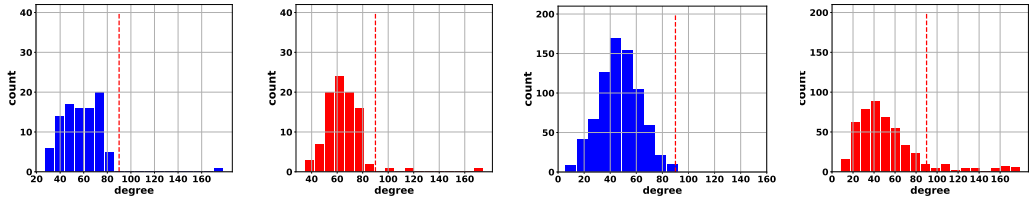
E TIME AND SPACE COMPLEXITY

We experimentally explore the time and space complexity of DIANA algorithm. All the experiments are performed on a single GTX-1080Ti GPU. The running time of the algorithm is reported in Table 8. Our TWO-MODEL-based approach increases slightly computational cost but improves significantly performance than A-GEM and MEGA. As for space complexity, DIANA uses the same memory size as those reply methods like A-GME and MEGA.



(a) Split-CIFAR (b) Split-CUB (c) ISIC2019

Figure 7: The distributions of the directions of the gradients on the current tasks. (a) balance std: 1.59, imbalance std: 3.50; (b) balance std: 2.58; imbalance std: 3.85; (c)balance std: 2.24, imbalance std: 5.52.



(a) Balanced Split-CUB (b) Imbalanced Split-CUB (c) Balanced Split-CIFAR (d) Imbalanced Split-CIFAR

Figure 8: The distributions of the angles between the current gradient and reference gradient, where the proportion of angles greater than 90 degrees is: (a) 1.05%, (b) 3.16%, (c) 0.13%, (d) 9.68%

Table 8: Comparison of time and space complexity on Split-CIFAR

Method	**Training time (s)**	**memory**
Single	8.8	0
EWC	78.6	0
MAS	77.3	0
GEM	109	1280
A-GEM	17.1	1280
Gdumb	8.8	1280
DER	20.0	1280
MEGA	15.7	1280
DIANA	34.5	1280