Continual Learning Through Retrieval and Imagination

Zhen Wang[1], Liu Liu[1], Yiqun Duan[2], Dacheng Tao[3,1]

[1]The University of Sydney, Australia, [2]University of Technology Sydney, Australia, [3]JD Explore Academy, China
{zwan4121,liu.liu1}@sydney.edu.au, yiqun.duan@student.uts.edu.au, dacheng.tao@gmail.com

Abstract

Continual learning is an intellectual ability of artificial agents to learn new concepts from sequential data. The main impediment to continual learning is catastrophic forgetting, a severe performance degradation on previously learned tasks. Although simply replaying all previous data or continuously adding the model parameters could alleviate the issue, it is impractical in real-world applications due to the limited available resources. Inspired by the mechanism of the human brain to deepen its past impression, we propose a novel framework, Deep Retrieval and Imagination (DRI), which consists of two components: 1) an embedding network that constructs a unified embedding space without adding model parameters on the arrival of new tasks; and 2) a generative model to produce additional (imaginary) data based on the limited memory. By retrieving the past experiences and corresponding imaginary data, DRI distills knowledge and rebalances the embedding space to further mitigate forgetting. Theoretical analysis demonstrates that DRI can reduce the loss approximation error and improve the robustness through retrieval and imagination, bringing better generalizability to the network. Extensive experiments show that DRI performs significantly better than the existing state-of-the-art continual learning methods and effectively alleviates catastrophic forgetting.

1 Introduction

Humans continuously learn new skills and accumulate knowledge over their lifetime (Alvarez et al. 1994; Smolen et al. 2019). By contrast, artificial neural networks suffer from catastrophic forgetting (McCloskey et al. 1989) which refers to the drastic drop in performance on the previous tasks while learning new tasks. The underlying reason is that training a network with new information severely interferes with the previously learned knowledge (McClelland et al. 1995). As trivial workarounds, devoting a whole new network to each task or storing all previous task data for re-training the model could alleviate the performance degradation on previous learning, but it is impractical for real-world applications in terms of required resources. Continual learning (also called lifelong or incremental learning) methods aim at training a neural network from a sequential stream of data, relieving catastrophic forgetting with limited resources (Rebuffi et al. 2017; Riemer et al. 2019; van de Ven et al. 2018).

Previous continual learning progresses majorly follow two directions: **(1)** trying to estimate the importance of each network parameter for previous tasks and penalizing the changes of important parameters during the learning of new tasks (Kirkpatrick et al. 2017; Schwarz et al. 2018; Serra et al. 2018; Zenke et al. 2017). However, it is difficult to find a reasonable metric to regularize all the parameters effectively, especially in long sequences of tasks and large networks. **(2)** trying to consolidate the knowledge in the original network by replaying a subset of past examples stored in a memory buffer (Buzzega et al. 2020; Hou et al. 2018; Rannen et al. 2017; Rebuffi et al. 2017; Buzzega et al. 2021). Whereas, the main additional challenge is the class imbalance between previous versus new tasks. The latest literature (Delange et al. 2021) proposed a series of guidelines for continual learning methods to be applicable in practice: i) good performance and less forgetting on previous tasks; ii) no oracle of task identifiers at inference time; and iii) bounded memory footprint throughout the entire training phase. Unfortunately, most of the exiting methods fail to satisfy all these guidelines mentioned above (Benjamin et al. 2019; Chaudhry et al. 2021; Wang et al. 2020; Lopez-Paz et al. 2017; Wei et al. 2021; Riemer et al. 2019; Schwarz et al. 2018; Zhu et al. 2020).

Contemporary biology suggests that humans can keep past knowledge with limited memory capacity, benefiting from the retrieval and imagination of a few past experiences (Schuster et al. 2011; Grilli et al. 2013). The Complementary Learning Systems (CLS) theory (Gelbard-Sagiv et al. 2008; Kumaran et al. 2016) holds that the hippocampus in human brain stores episodic-like memory that can be reactivated during sleep or in unconscious and conscious recall, thus consolidating knowledge in the neocortex via the retrieval and imagination of past experiences in terms of multiple internal replays (Schacter et al. 2012; Cheng et al. 2016). Inspired by the mechanism of the human brain to deepen past impressions, we propose to apply such imagination into machine vision systems for retrieving the limited memory, in order to consolidate knowledge and alleviate catastrophic forgetting.

In this work, we propose a novel framework, Deep Retrieval and Imagination (DRI), to solve the challenges in continual learning. DRI can imitate the ability of the brain to generate additional imaginary data based on the limited memory of past data and retrieve previous tasks in a balanced manner while learning the current data. Specifically,

DRI introduces an image-conditional Generative Adversarial Networks (GANs) (Goodfellow et al. 2014; Antoniou et al. 2017) to learn to strengthen the memory of a past data-item by generating other within-class imaginary data, and thus can mitigate forgetting. DRI maps the continuum of data into a unified embedding space by an embedding network without adding model parameters on the arrival of new tasks, as in other methods. We refer to the class embedding mean as the term *prototype* and conduct the classification by computing the distances to the prototype of each class, which do not require task identifiers at inference time. By retrieving the limited memory and corresponding imaginary data, DRI distills *dark knowledge* (Hinton et al. 2014) and re-optimizes the embedding space with a metric loss, which effectively consolidates knowledge and alleviates forgetting. Further, considering the adverse effects caused by the imbalance between past and new classes, we propose a *cosine herding* sampling strategy, which rebalances new data and past data during each retrieval batch and ensures that all classes are equally represented in memory.

To verify the effectiveness and the performance of DRI, we conduct both theoretical and experimental analyses. Theoretically, we demonstrate that DRI retrieves previous experiences to reduce loss approximation error and leverages the imaginary data to improve the robustness, bringing better generalizability for the network. Experimentally, extensive evaluations under three scenarios show that DRI performs significantly better than the existing state-of-the-art continual learning methods and effectively mitigates catastrophic forgetting. Ablation studies validate the impact of each component in DRI.

2 Related Work

Rehearsal-based methods prevent catastrophic forgetting by replaying a subset of the stored exemplars of the previous tasks (Benjamin et al. 2019; Hou et al. 2019). By interleaving the previous task exemplars with current task data, the network parameters can be jointly optimized. GSS (Aljundi et al. 2019) introduces a gradient-based sampling to store optimally selected exemplars in the memory buffer. HAL (Chaudhry et al. 2021) complements experience replay with an additional objective, keeping intact the predictions on some *anchor* points of past tasks. GEM (Lopez-Paz et al. 2017) and A-GEM (Chaudhry et al. 2019) leverage episodic memory to compute previous task gradients to constrain the current update step. Several methods exploit Knowledge Distillation (Hinton et al. 2014) to alleviate forgetting (Buzzega et al. 2020; Rebuffi et al. 2017). iCaRL (Rebuffi et al. 2017) trains a *nearest-class-mean* classifier while maintaining the representation in later tasks via a self-distillation loss term. DER (Buzzega et al. 2020) mixes rehearsal with distillation loss for retraining past experience and achieves state-of-the-art performance. Our proposal is a rehearsal-based method, which leverages a unified embedding network and imagination generator to achieve state-of-the-art performance.

Generative-based methods synthesize previous data produced by generative models and replay the synthesized data (Xiang et al. 2019; van de Ven et al. 2020). CL-GAN (Shin et al. 2017) employs a generative adversarial network (GAN) (Goodfellow et al. 2014) to generate past images and trains a classifier on both of synthesized and real images. DGM (Ostapenko et al. 2019) relies on conditional GANs with neural masking, which needs to expand network parameters for new tasks. BIR (van de Ven et al. 2020) replays hidden representations generated by the network's own feedback connections. However, generative-based approaches tend to produce blurry images or representations (not belonging to any class) on complex datasets, hurting the classification accuracy. Instead of synthesizing data, the generator in our proposal augments the real images, and can produce quality images as additional data to improve performance.

Parameter-based methods try to estimate the importance of each network parameter of prior tasks and penalize the changes of important parameters during the learning of new tasks (Aljundi et al. 2018). The difference between these works is the way to compute network parameter importance. For example, Elastic Weight Consolidation (EWC) (Kirkpatrick et al. 2017) and online EWC (oEWC) (Schwarz et al. 2018) compute synaptic importance with the diagonal Fisher information matrix as the approximation of Hessian. Synaptic Intelligence (SI) (Zenke et al. 2017) accumulates task-relevant information over time to measure the importance. However, it is difficult to find an effective metric to evaluate all the parameters, and thus leads to the failure of these methods for solving longer sequences of tasks and large networks.

Other Approaches. LwF (Li et al. 2017) computes the current responses for the new samples at the beginning of each task, minimizing their drift during training. SDC (Yu et al. 2020) utilizes metric learning and estimates the drift of prior tasks during the training of new tasks, however it requires pre-training the model on a large dataset to get a feature extractor.

3 Deep Retrieval and Imagination

3.1 Overall Framework

Formally, a continual learning problem is split in a sequence of $\mathcal{T}$ supervised learning tasks t, $t \in \{1, ..., \mathcal{T}\}$. For task t, input samples $x \in \mathcal{X}$ and the corresponding ground truth labels $y \in \mathcal{Y}$ are drawn from an i.i.d. distribution $\mathcal{D}_t$. A network f_θ with parameters θ observes one task t at a time in a sequential manner. Let $\Theta \subseteq \mathbb{R}^d$ be a network parameter space, and let $\ell(\theta; x, y) : \Theta \mapsto \mathbb{R}$ be the loss function of θ associated with data point (x, y), and $f_\theta(x)$ is the output of f_θ for x. The general objective is to minimize the population loss function of all observed tasks:

$$\mathcal{F}(\theta) = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \mathcal{L}_t(\theta), \text{ where } \mathcal{L}_t(\theta) \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}_t}[\![\ell(\theta; x, y)]\!].$$

Specifically, when learning on the t-th task, the model obtains access to N_t data points sampled from $\mathcal{D}_t$ and we define

$$\hat{\mathcal{L}}_t(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} \ell(\theta; x_i, y_i)$$

as the empirical loss function; then the network parameters update by these N_t training data. After the training procedure is finished, these N_t data are assumed to be unavailable, but a small amount of data can be stored in a limited memory. The

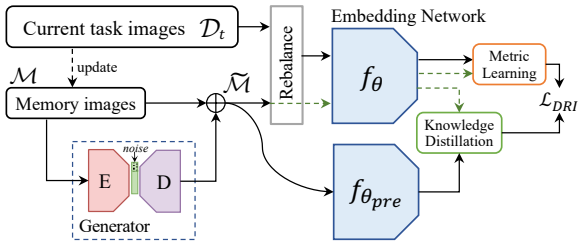


Figure 1: The overall DRI architecture.

goal is to avoid forgetting past tasks when trained on new tasks by utilizing the limited stored data.

We propose the brain-inspired framework Deep Retrieval and Imagination (DRI) to effectively alleviate catastrophic forgetting for continual learning. Our overall learning architecture is illustrated in Figure 1. DRI mainly consists of two components: 1) an embedding network f with parameters θ, mapping data onto a unified embedding space; and 2) a generative model, referred as Imagination GAN (IGAN), for generating imaginary data based on real images. As a rehearsal-based method, DRI is allocated a memory buffer $\mathcal{M}$ to store a tiny subset of past exemplars. Key to continual learning is how to utilize the limited past exemplars to mitigate catastrophic forgetting (Section 3.2). For current task t, we distill *dark knowledge* (Hinton et al. 2014) through retrieving past *experiences* and consolidate the representation of past tasks by metric learning, which can be achieved through optimizing the following objective:

$$\mathcal{L}_t(\theta) + \mathbb{E}_{(x,y)\sim\mathcal{M}}\left[\alpha \left\| f_\theta(x) - f_{\theta_{pre}}(x)\right\|_2^2 + \beta\, \ell(\theta; x, y)\right], \quad (1)$$

where $f_{\theta_{pre}}$ serves as the teacher model parameterized by last task θ_{pre} and remains frozen; f_θ as the student model mimics $f_{\theta_{pre}}$ by minimizing the Euclidean distance between their output embeddings; ℓ is a metric learning loss; α and β are coefficients balancing the terms.

Furthermore, we introduce a novel image-conditioned generative model (IGAN) to enrich the within-class images based on limited real images stored in $\mathcal{M}$ (Section 3.3). In that case, images generated through $IGAN_g(x)$ could help consolidate knowledge by replaying generated images as real sample-related imagination, thus alleviate the forgetting while training new tasks. Besides, we consider the adverse effects of an imbalance between new and past tasks, and introduce *cosine herding* (Welling 2009) sampling strategy, which rebalances new data and past data during each retrieval batch and ensures that all classes are equally represented in the memory. With these considerations in hands, DRI optimizes the objective:

$$\begin{aligned}\mathcal{L}_{DRI} = &\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_t\cup\widetilde{\mathcal{M}}}[\![\ell(\theta; x, y))]\!] \\ &+ \mathop{\mathbb{E}}_{(x,y)\sim\widetilde{\mathcal{M}}}\left[\alpha \left\| f_\theta(x) - f_{\theta_{pre}}(x)\right\|_2^2 + \beta\, \ell(\theta; x, y)\right],\end{aligned} \quad (2)$$

where $\widetilde{\mathcal{M}}$ represents a set including memory data and their corresponding generated data; $\cup$ denotes the Union operation.

Algorithm 1: - Deep Retrieval and Imagination (DRI)

Input: continuum dataset $\mathcal{D}$, memory capacity K
Require: parameters θ, *IGAN*, scalars α and β, learning rate η

```
M ← {}                                              ▷ Initialize memory with empty set
for t = 1, ..., T do
    θ_pre ← θ
    for (x, y) in D_t do
        (x', y') ← sample(M)
        (x'_a, y'_a) ← (IGAN_g(x'), y') ∪ (x', y')
        Δ ← α ||f_θ(x'_a) − f_{θ_pre}(x'_a)||_2^2 + β ℓ(θ; x'_a, y'_a)
        (x_b, y_b) ← rebalance((x, y), (x'_a, y'_a))
        θ ← θ − η∇_θ[ℓ(θ; x_b, y_b) + Δ]            ▷ Section 3.2
    end for
    IGAN ← updateIGAN(IGAN; D_t, M)                 ▷ Section 3.3
    M ← updateMemory(M; D_t, θ, K)                  ▷ Eq. (8)
end for
```

3.2 Deep Retrieval for Embeddings

DRI constructs a unified embedding space by an embedding network with a metric learning loss to continually learn from a sequential data stream. By retrieving the stored exemplars and corresponding imaginary data, DRI distills previous knowledge and rebalances the embedding space, which alleviates forgetting effectively and improves overall performance.

Embedding-Based Network Structure Most existing continual learning methods (Buzzega et al. 2020; Chaudhry et al. 2019; Li et al. 2017; Rebuffi et al. 2017; Riemer et al. 2019; Schwarz et al. 2018) employ one-hot classification networks, and have to add new weights (or multi-head classifiers) to accommodate newly arrived classes. Instead, we use embeddings as the network outputs, which naturally allow for modeling emerging new classes, and do not require direct changes to the network structure. Embedding networks map data into a low-dimensional output, where similar data are close together and dissimilar data are far apart (Chopra et al. 2005; Wang et al. 2021). In the learned space, general metrics, such as L2-distance, can be applied to determine the similarities between the original data. Metric learning losses (Hoffer et al. 2015; Wang et al. 2019) are used for training embedding networks, with the aim to reduce the distance between similar data and to increase the distance between dissimilar data. Next, by using embeddings as network outputs, we make the further regularized retrieval feasible.

Regularized Retrieval Retrieving limited memory data during learning new tasks is a crucial way to maintain previous knowledge. Ideally, we look for network parameters that are well adapted to the new task while approximating the behavior observed in the previous one. Effectively, we seek to encourage the network to mimic its previous representation for past exemplars by minimizing:

$$\mathbb{E}_{(x,y)\sim\widetilde{\mathcal{M}}}\left[\left\| f_\theta(x) - f_{\theta_{pre}}(x)\right\|_2^2\right], \quad (3)$$

where θ_{pre} is the network parameters after the completion of the previous task, and $\widetilde{\mathcal{M}}$ is the set containing memory data and their corresponding generated data from IGAN (described in Section 3.3). It is worth noting that we can only

save one previous model, at most, or we could get the network output on $\widetilde{\mathcal{M}}$ before starting a new task without saving any previous model.

During continual learning, a constant class representation in the embedding space is not necessarily optimal, and it needs to be adjusted to accommodate more classes and tasks. We expect the network could review the past and know the new, which requires the model not only to consolidate past knowledge, but also to jointly learn the interrelationship between past and new tasks. For this purpose, we seek to minimize:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_t\cup\widetilde{\mathcal{M}}}[\![\ell(\theta;x,y)]\!]+\mathbb{E}_{(x,y)\sim\widetilde{\mathcal{M}}}[\![\beta\,\ell(\theta;x,y)]\!], \quad (4)$$

where β is a coefficient balancing the past experience replay. Learning on the union $\mathcal{D}_t \cup \widetilde{\mathcal{M}}$ could form a unified embedding space covering past and new tasks. Eq. (3) with a coefficient α and Eq. (4) constitute the objective function of DRI (Eq. (2)). Here, we employ a pair-based metric learning loss, Multi-Similarity (MS) loss (Wang et al. 2019), as our loss function ℓ:

$$\ell=\frac{1}{N_b}\sum_{i=1}^{N_b}\Bigg\{\frac{1}{\mu}\log\Big[1+\sum_{k\in\mathcal{P}_i}e^{-\mu(S_{i,k}-\lambda)}\Big] +\frac{1}{\upsilon}\log\Big[1+\sum_{k\in\mathcal{N}_i}e^{\upsilon(S_{i,k}-\lambda)}\Big]\Bigg\}, \quad (5)$$

where $S_{i,k}$ denotes the similarity of a pair $\{x_i, x_k\}$; $\mathcal{P}_i$ and $\mathcal{N}_i$ are the positive and negative sets for the anchor x_i; N_b is the number of training samples in a batch; and μ, υ and λ are hyper-parameters.

Having trained an embedding network with MS loss, we can leverage the embedding space for classification. DRI uses a *nearest-class-mean* classification strategy, in which an image x is classified to a class c^* determined by:

$$c^*=\arg\min_{c\in C}\text{dist}(f_\theta(x),\rho_c), \quad (6)$$

$$\rho_c=\frac{1}{N_c}\sum_i\delta_{y_i=c}f_\theta(x_i), \quad (7)$$

where N_c is the number of examples for class c; ρ_c is referred to as the *prototype* of class c, the average presentation vector of all exemplars for a class c; and δ is the indicator function. In the evaluation, we compute ρ_c based on $\mathcal{M}$.

Rebalance in Training and Memory Different from other rehearsal-based methods simply interleaving the past exemplars with current task data (Buzzega et al. 2020; Rebuffi et al. 2017; Riemer et al. 2019), we consider the adverse effects of imbalance and sample past data and new data in a balanced manner, propelling all classes observed so far to be equally represented in a training batch. Given a batch size N_b in metric learning, we sample $N_b/|C|$ images per class from $\mathcal{D}_t \cup \widetilde{\mathcal{M}}$ for forming a batch, where C is the set of classes observed so far. However, even if classes are balanced in a batch, the magnitudes of each class embeddings are significantly uneven. To this end, we propose to leverage *cosine normalization* for embeddings and prototypes, as:

$$g(v_i,v_j)=\frac{<v_i,v_j>}{\|v_i\|_2\,\|v_j\|_2},$$

where v_i and v_j are vectors, and $\|\cdot\|_2$ denotes the l_2-norm. We deploy the cosine normalization for metric learning (Eq. (5)) and nearest-class-mean classification (Eq. (6)). Although cosine normalization has been adopted in other visual problem (Gidaris et al. 2018; Wei et al. 2019; Qi et al. 2018; Hou et al. 2019), we demonstrate that it can effectively eliminate the imbalance in the magnitude of class embeddings and facilitate continual learning (Section 5.3).

Whenever new classes or tasks arrive, the memory $\mathcal{M}$ is adjusted for the data stream. We expect that all classes are treated equally and a subset with the most representative exemplars of each class can be stored. Inspired by herding (Welling 2009) strategy, we design a *cosine herding* sampling to balance memory and retrieval, described as follows. Given the memory capacity K, we assign $M=\lfloor K/|C|\rfloor$ exemplars for each class, where C is the set of classes observed so far. For current task t with a set of classes C_t, we compute prototype ρ_c of each class on $\mathcal{D}_t$, $c \in C_t$ (Eq. (7)). We select the exemplars in a ranked order. The m-th exemplar e_m of class c is obtained by:

$$e_m \leftarrow \arg\min_{x\in\{\mathcal{D}_t|y=c\}} g\left(\rho_c, \frac{1}{m}[f(x)+\sum_{i=1}^{m-1}f(e_i)]\right). \quad (8)$$

The first M exemplars, $e_1, ..., e_M$, are added to the memory. Due to the limited memory capacity, we reduce the number of exemplars of each previous class to M in reverse order, i.e., removing $e_{M+1}, e_{M+2}, ...$ for each of the previous classes. The superiority of the cosine herding sampling is that the average normalized embedding over all exemplars can best approximate the average embedding over all training examples. Besides, we rebalance past and new data within a training batch to maintain a stable embedding space in the training.

3.3 Imagination GAN

As mentioned above, a novel Generative Adversarial Network referred as Imagination GAN (IGAN) is introduced to create imaginary images from given real images stored in the limited memory. During learning a new task, imaginary images are involved in the retrieval as additional data to consolidate previously learned knowledge. Figure 2 illustrates the brief architecture of IGAN, which we describe in detail below.

IGAN is composed of a generator network and a discriminator network. Consider the current task data $\mathcal{D}_t$ and the memory $\mathcal{M}$ as the training data $\{x_i^c | i = 1, 2, ..., N_c$ and $c \in C\}$, with each image labelled by its class c and indexed by i, taken from the set of classes C observed so far. The generator network consists of an encoder projecting an image from class c to a lower-dimensional vector and a decoder generating a within-class image from the bottleneck vector concatenated with a Gaussian noise $\boldsymbol{z}_i$. The discriminator network is trained to discriminate between *real* and *fake*, according to a critic. We expect the generator is able to produce diverse data that is related to, but different from the input image. Therefore, we improve WGAN (Arjovsky et al. 2017; Gulrajani et al. 2017) critic as the *3-tuple* fashion that takes:

- fake tuple $\{x_i^c, \hat{x}\}$, a real image x_i^c and the output $\hat{x}$ of generator that takes x_i^c as an input;
- stable tuple $\{x_i^c, x_i^c\}$, a real image x_i^c and itself;

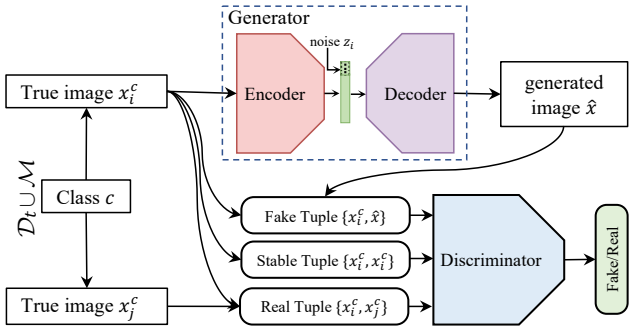


Figure 2: IGAN architecture. Adversarial training leads the network to generate within-class images with diversity according to the input image.

- real tuple $\{x_i^c, x_j^c\}$, a real image x_i^c and another images x_j^c from the same class c.

The critic tries to discriminate the fake tuple from the real tuple, where stable tuples are treated as real tuples in a certain proportion to maintain the stability of the network. IGAN can generate diverse data by making *fake tuples* look like *real tuples*. Moreover, IGAN also needs fractional *fake tuples* to look like *stable tuples* to prevent the model from collapsing.

Specifically, the generator is optimized by minimizing this discriminative ability measured by the Wasserstein distance (Arjovsky et al. 2017). By providing the real tuple to the discriminator, we propel the GAN to generate within-class images with diversity in a generalized way, rather than autoencoding the current images. However, for some complex datasets with large intra-class distances, such as CIFAR (Krizhevsky et al. 2009) and ImageNet (Deng et al. 2009), GAN will collapse on them because it has to learn the diversity of the huge differences (Antoniou et al. 2017). To this end, we utilize the stable tuple, treating a certain proportion of stable tuple as real tuple, to prevent the GAN from collapsing, balancing diversity with stability. At the end of each task t, IGAN is trained on $D_t \cup \mathcal{M}$ to update the generative ability of imagining data. The update of IGAN requires only slight re-training after the first task, therefore it necessitates only low computation.

4 Theoretical Analysis

In this section, we analyze the generalization of our learning framework based on robustness theory in machine learning (Bellet et al. 2015; Mohri et al. 2009, 2018; Yin et al. 2020). Our analysis demonstrates that DRI retrieves previous experiences to reduce loss approximation error and leverages the imaginary data to improve the algorithmic robustness, bringing better generalizability to the network.

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $z = (x, y) \in \mathcal{Z}$. A learning algorithm $\mathcal{A}$ takes as input a finite set of pairs from $(\mathcal{Z} \times \mathcal{Z})^n$ and outputs a function. We denote by $\mathcal{A}_{p_s}$ the function learned by an algorithm $\mathcal{A}$ from a sample p_s of pairs. We give the notion of $(E, \epsilon(\cdot))$ robust (Bellet et al. 2015) for an algorithm:

Definition 1 *An algorithm $\mathcal{A}$ is $(E, \epsilon(\cdot))$ robust for $E \in \mathbb{N}$ and $\epsilon(\cdot) : (\mathcal{Z} \times \mathcal{Z})^n \to \mathbb{R}$ if $\mathcal{Z}$ can be partitioned into E disjoints sets, denoted by $\{H_i\}_{i=1}^E$, such that for all samples $s \in \mathcal{Z}^n$ and the pair set $p(s)$ associated to this sample, the following holds: $\forall (s_1, s_2) \in p(s), \forall z_1, z_2 \in \mathcal{Z}, \forall i, j = 1, ..., E$: if $s_1, z_1 \in H_i$ and $s_2, z_2 \in H_j$, then*

$$|\ell(\mathcal{A}_{p_s}, s_1, s_2) - \ell(\mathcal{A}_{p_s}, z_1, z_2)| \leq \epsilon(p_s).$$

E and $\epsilon(\cdot)$ quantify the robustness of the algorithm and depend on the training sample. With an algorithmic perspective for DRI, our generalization analysis aims to bound the gap between average population loss and regularized training loss, as per the following theorem.

Theorem 1 *Assume the algorithm $\mathcal{A}$ is $(E, \epsilon(\cdot))$-robust, $\mathcal{L}_t(\theta)$ is ρ-Hessian Lipschitz, f_θ is L-Lipschitz and $|\ell(\theta; x, y)| \leq b$. With probability at least $1 - \delta$ over the random training examples $(x, y) \sim \mathcal{D}_t$, $t \in \{1, ..., \mathcal{T}\}$, we have*

$$\mathcal{F}(\theta) \leq \underbrace{\frac{1}{\mathcal{T}}(\hat{\mathcal{L}}_{\mathcal{T}}(\theta) + \mathcal{L}_{\mathcal{T}-1}^{prox}(\theta))}_{\text{regularized training loss}} + \underbrace{\frac{\rho}{2\mathcal{T}} \sum_{t=1}^{\mathcal{T}-1} ||\theta - \hat{\theta}_t||_2^3}_{\text{loss approximation error}}$$
$$+ \underbrace{\frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} \epsilon(p_{s_t}) + \mathcal{O}(b\sqrt{\frac{2E \ln 2 + 2 \ln 1/\delta}{N}} + R)}_{\text{finite-sample effect}},$$

where $\hat{\mathcal{L}}_{\mathcal{T}}(\theta)$ is the empirical loss of task $\mathcal{T}$; $\mathcal{L}_{\mathcal{T}-1}^{prox}(\theta)$ is the sum of second-order Taylor approximation of the first $\mathcal{T} - 1$ empirical loss functions; p_{s_t} is the sample of pairs for task t and $N = \min_t N_t$; R is a constant.

As we can see, the gap between $\mathcal{F}(\theta)$ and the regularized training loss includes two parts. The first part is the loss function approximation error, which could decay as our regularized retrieval. The second part is related to the robustness terms, which would be improved by leveraging the additional imaginary data.

5 Experiments

5.1 Experimental Setup

We consider a strict evaluation setting (Hsu et al. 2018), which models the sequence of tasks following three scenarios: Task Incremental Learning (**Task-IL**) splits the training samples into partitions of tasks, which requires task identities to select corresponding classifiers at inference time; Class Incremental Learning (**Class-IL**) sequentially increases the number of classes to be classified without requiring the task identities, as the hardest scenario (van de Ven et al. 2018); Domain Incremental Learning (**Domain-IL**) observes the same classes during each task, but the input-distribution is continuously changing; task identities remains unknown.

Datasets. We experiment with the following datasets:

- Split MNIST: the MNIST benchmark (LeCun et al. 1998) is split into 5 tasks by grouping together 2 classes.
- Split CIFAR-10: splitting CIFAR-10 (Krizhevsky et al. 2009) in 5 tasks, each of which introduces 2 classes.
- Split Tiny-ImageNet: Tiny-ImageNet (Stanford 2015) has 100,000 images across 200 classes. Each task consists of 20 disjoint subset of classes from these 200 classes.

Memory	Method	S-CIFAR-10		S-Tiny-ImageNet		R-MNIST
		Class-IL	*Task-IL*	*Class-IL*	*Task-IL*	*Domain-IL*
–	JOINT	92.20±0.15	98.31±0.12	59.99±0.19	82.04±0.10	95.76±0.04
	SGD	19.60±0.04	61.02±3.33	7.92±0.26	18.31±0.68	67.66±8.53
–	oEWC (Schwarz et al. 2018)	19.49±0.12	68.29±3.92	7.58±0.10	19.20±0.31	**77.35±5.77**
	SI (Zenke et al. 2017)	19.48±0.17	68.05±5.91	6.58±0.31	**36.32±0.13**	71.91±5.83
	LwF (Li et al. 2017)	19.61±0.05	63.29±2.35	**8.46±0.22**	15.85±0.58	-
	CL-GAN (Shin et al. 2017)	24.73±1.89	78.46±2.18	8.02±0.39	26.33±1.04	-
	DGM (Ostapenko et al. 2019)	38.62±1.28	83.51±0.67	8.27±0.46	27.51±1.78	-
	BIR (van de Ven et al. 2020)	**46.14±1.83**	**87.52±0.91**	-	-	73.26±0.89
200	ER (Riemer et al. 2019)	44.79±1.86	91.19±0.94	8.49±0.16	38.17±2.00	85.01±1.90
	GEM (Lopez-Paz et al. 2017)	25.54±0.76	90.44±0.94	-	-	80.80±1.15
	A-GEM (Chaudhry et al. 2019)	20.04±0.34	83.88±1.49	8.07±0.08	22.77±0.03	81.91±0.76
	iCaRL (Rebuffi et al. 2017)	49.02±3.20	88.99±2.13	7.53±0.79	28.19±1.47	-
	FDR (Benjamin et al. 2019)	30.91±2.74	91.01±0.68	8.70±0.19	40.36±0.68	85.22±3.35
	GSS (Aljundi et al. 2019)	39.07±5.59	88.80±2.89	-	-	79.50±0.41
	HAL (Chaudhry et al. 2021)	32.36±2.70	82.51±3.20	-	-	82.91±1.21
	DER (Buzzega et al. 2020)	61.93±1.79	91.40±0.92	11.87±0.78	40.22±0.67	90.04±2.61
	DRI (ours)	**65.16±1.13**	**92.87±0.71**	**17.58±1.24**	**44.28±1.37**	**91.17±1.53**
500	ER (Riemer et al. 2019)	57.74±0.27	93.61±0.27	9.99±0.29	48.64±0.46	88.91±1.44
	GEM (Lopez-Paz et al. 2017)	26.20±1.26	92.16±0.69	-	-	81.15±1.98
	A-GEM (Chaudhry et al. 2019)	22.67±0.57	89.48±1.45	8.06±0.04	25.33±0.49	80.31±6.29
	iCaRL (Rebuffi et al. 2017)	47.55±3.95	88.22±2.62	9.38±1.53	31.55±3.27	-
	FDR (Benjamin et al. 2019)	28.71±3.23	93.29±0.59	10.54±0.21	49.88±0.71	89.67±1.63
	GSS (Aljundi et al. 2019)	49.73±4.78	91.02±1.57	-	-	81.58±0.58
	HAL (Chaudhry et al. 2021)	41.79±4.46	84.54±2.36	-	-	85.00±0.96
	DER (Buzzega et al. 2020)	70.51±1.67	93.40±0.39	17.75±1.14	51.78±0.88	92.24±1.12
	DRI (ours)	**72.78±1.44**	**93.85±0.46**	**22.63±0.81**	**52.89±0.60**	**93.02±0.85**

Table 1: Classification results (accuracy %) for standard continual learning benchmarks. '-' indicates experiments we were unable to run, because of compatibility issues (*e.g.* iCaRL and LwF in Domain-IL) or intractable training time (*e.g.* HAL, GEM and GSS on Tiny ImageNet).

Dataset \ Method	ER	GEM	A-GEM	iCaRL	FDR	GSS	HAL	DER	**DRI (ours)**
S-Tiny-ImageNet (Class-IL)	53.51±1.90	-	62.14±2.29	28.78±0.84	53.72±1.56	-	-	32.12±0.34	**22.32±0.49**
S-CIFAR-10 (Task-IL)	1.34±0.13	6.91±2.33	11.36±1.68	1.59±0.57	1.93±0.48	7.71±2.31	5.21±0.50	2.59±0.08	**0.49±0.24**
R-MNIST (Domain-IL)	3.10±0.42	2.49±0.17	18.10±1.44	-	3.31±0.56	92.66±0.02	17.62±2.33	2.17±0.11	**0.78±0.47**

Table 2: Forgetting results (%) for continual learning benchmarks with 5120 memory capacity (lower is better).

- Rotated MNIST (Ioffe et al. 2015): containing 20 subsequent tasks (domains), each of which is generated by rotating all MNIST images with a random angle in the interval.

Implementation. To fairly compare each method, we trained all networks using the stochastic gradient descent (SGD) optimizer. For variants of MNIST dataset, we follow (Riemer et al. 2019; Lopez-Paz et al. 2017) and rely on Multi-Layer Perceptron (MLP) with two hidden layers, each one comprised of 100 ReLU units. For Tiny ImageNet and CIFAR-10, we follow (Buzzega et al. 2020; Rebuffi et al. 2017) by employing ResNet18 (He et al. 2016) (not pre-trained). The training images and generated examples are randomly cropped and flipped following (Buzzega et al. 2020; Yu et al. 2020). We select the hyper-parameters by performing a grid search on the validation set which is obtained by sampling 10% of the training set. Part of the experiment results of baselines are from (Buzzega et al. 2020).

5.2 Performance Comparison

In this section, we compare DRI against six rehearsal-based methods (ER (Riemer et al. 2019), GEM (Lopez-Paz et al. 2017), A-GEM (Chaudhry et al. 2019), GSS (Aljundi et al. 2019), FDR (Benjamin et al. 2019), and HAL (Chaudhry et al. 2021)); three generative-based methods (CL-GAN (Shin et al. 2017), DGM (Ostapenko et al. 2019), and BIR (van de Ven et al. 2020)); three methods leveraging Knowledge Distillation (LwF (Li et al. 2017), iCaRL (Rebuffi et al. 2017), and DER (Buzzega et al. 2020)); and two parameter-based methods (oEWC (Schwarz et al. 2018) and SI (Zenke et al. 2017)). We further provide an upper bound (JOINT) obtained by training all tasks jointly and a lower bound simply performing *SGD* without any countermeasure to forgetting.

Accuracy. First, we compare the performance in terms of overall accuracy at the end of all tasks, shown in Table 1. From the results, it is observed that DRI achieves state-of-

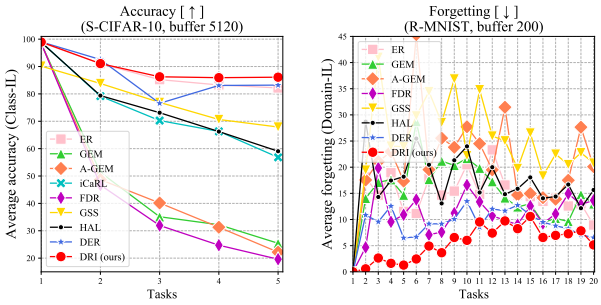
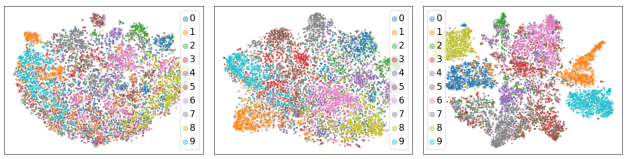


Figure 3: Incremental performance evaluated on all tasks observed so far during continual learning. [↑] higher is better, [↓] lower is better *(best seen in color)*.

	S-CIFAR-10 Class-IL	S-Tiny-ImageNet Task-IL	R-MNIST Domain-IL
DRI\Δ	54.04±1.81	38.97±1.76	87.93±1.72
DRI\IGAN$_g$	59.38±1.37	40.48±1.21	88.35±1.29
DRI\ML	62.07±2.21	40.79±0.85	89.82±1.33
DRI\KD	60.11±0.78	39.74±1.59	89.01±1.90
DRI\Re	61.34±1.92	41.05±2.14	90.16±2.31
DRI	**65.16±1.13**	**44.28±1.37**	**91.17±1.53**

Table 3: Ablation study for main components.



(a) DRI\Δ (b) DRI\IGAN$_g$ (c) DRI

Figure 4: The t-SNE visualization of embedding space.

the-art performance in almost all settings. Especially in the case of small memory, the advantage of DRI is more obvious, e.g., DRI can reduce the classification error by more than 5% on Tiny ImageNet with 200 memory capacity. This is supported by IGAN providing additional imaginary data to alleviate the memory-limitation problem. The gap is unbridgeable when comparing with oEWC and SI, which indicates that the regularization towards old parameters is not effective in preventing forgetting. The generative-based methods' performance on complex datasets, e.g., ImageNet, degrades significantly due to its difficulty in generating complex and clear images.

Forgetting. Second, to compare the preventing forgetting capability, we assess the *average forgetting* (Chaudhry et al. 2018) that measures the performance degradation in subsequent tasks. Table 2 shows forgetting results in different learning scenarios. Our approach suffers from less forgetting than all the other methods, as DRI constructs a unified embedding space throughout the learning and consolidates knowledge through deep retrieval and imagination.

Incremental Performance. Finally, we demonstrate the *average incremental performance* (Rebuffi et al. 2017) which is the result of evaluating on all the tasks observed so far after completing each task. As shown in Figure 3, the results are curves of accuracy and forgetting after each task. The performance of most methods degrades rapidly as new tasks arrive, while DRI consistently outperforms the state-of-the-art methods throughout the learning.

real	stable	fake	DA	S-MNIST	S-CIFAR-10
√	√			75.35±0.92	42.61±4.77
√		√		85.86±1.02	54.82±3.61
	√	√		82.14±1.81	59.11±2.06
			√	82.91±2.33	62.24±1.89
√	√	√		87.35±1.27	65.16±1.13

Table 4: Ablation experiments for IGAN.

5.3 Ablation Study

To assess the effects of the proposed components in DRI, we perform comprehensive ablation studies. Table 3 shows the comparison with the removal of each major component of DRI, including regularized retrieval (Δ), IGAN generator (IGAN$_g$), embedding network with metric learning (ML), knowledge distillation (KD) and rebalance (Re), where \ indicates the removal operation. The results of Table 3 demonstrate the effectiveness of each component of DRI. Specially, Figure 4 shows the t-SNE (Van der Maaten et al. 2008) visualization for the impact of retrieval (DRI\Δ) and imagination (DRI\IGAN$_g$) in embedding space. By leveraging the proposed Deep Retrieval and IGAN, the embeddings of DRI are better clustered and separated after the continual learning.

Furthermore, we investigate the impact of three types of tuples (*real*, *stable* and *fake*) in IGAN and compare IGAN with data augment (DA) techniques as shown in Table 4. IGAN can generate diverse data by making fake tuples look like real tuples. If we remove fake tuples, the generator cannot get the supervision and produces noises. If removing real tuples, IGAN will simply autoencode the current image. Moreover, IGAN also needs fractional stable tuples to prevent the model from collapsing, especially in complex datasets. Table 4 also reveals that naive data augmentation does not significantly improve the accuracy of the nearest-class-mean classifier (Eq. (6)).

6 Conclusion

In this paper, we propose a brain-inspired framework, Deep Retrieval and Imagination (DRI), to effectively mitigate the catastrophic forgetting for continual learning. DRI designs a generative model to produce imaginary data and leverages knowledge distillation for retrieving past experiences in a balanced manner, thereby can consolidate knowledge. Theoretical analysis demonstrates that DRI improves the generalizability of the network by leveraging the imaginary data and retrieving previous experiences. Extensive experimental results show that DRI significantly outperforms current state-of-the-art methods, and ablation studies validate the effectiveness of the proposed components.

7 Acknowledgements

Mr Zhen Wang and Dr Liu Liu are supported by Australian Research Council Project DP-180103424.

References

Abati, D.; Tomczak, J.; Blankevoort, T.; Calderara, S.; Cucchiara, R.; and Bejnordi, B. E. 2020. Conditional channel gated networks for task-aware continual learning. In *CVPR*, 3931–3940.

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 139–154.

Aljundi, R.; Chakravarty, P.; and Tuytelaars, T. 2017. Expert gate: Lifelong learning with a network of experts. In *CVPR*, 3366–3375.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. In *NeurIPS*.

Alvarez, P.; and Squire, L. R. 1994. Memory consolidation and the medial temporal lobe: a simple network model. *Proceedings of the National Academy of Sciences*, 91(15): 7041–7045.

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*, 214–223. PMLR.

Bellet, A.; and Habrard, A. 2015. Robustness and generalization for metric learning. *Neurocomputing*, 151: 259–267.

Benjamin, A. S.; Rolnick, D.; and Kording, K. P. 2019. Measuring and regularizing networks in function space. *ICLR*.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *NeurIPS*.

Buzzega, P.; Boschini, M.; Porrello, A.; and Calderara, S. 2021. Rethinking Experience Replay: a Bag of Tricks for Continual Learning. In *International Conference on Pattern Recognition (ICPR)*, 2180–2187. IEEE.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 532–547.

Chaudhry, A.; Gordo, A.; Dokania, P.; Torr, P.; and Lopez-Paz, D. 2021. Using Hindsight to Anchor Past Knowledge in Continual Learning. *AAAI*, 35(8): 6993–7001.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019. Efficient Lifelong Learning with A-GEM. In *ICLR*.

Cheng, J.; Yang, W.; Huang, M.; Huang, W.; Jiang, J.; Zhou, Y.; Yang, R.; Zhao, J.; Feng, Y.; Feng, Q.; et al. 2016. Retrieval of brain tumors by adaptive spatial pooling and fisher vector representation. *PloS one*, 11(6): e0157112.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, 539–546. IEEE.

Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE TPAMI*, 1–1.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *CVPR*, 5138–5146.

Foster, D. J.; Sekhari, A.; and Sridharan, K. 2018. Uniform convergence of gradients for non-convex learning and optimization. *arXiv preprint arXiv:1810.11059*.

Gelbard-Sagiv, H.; Mukamel, R.; Harel, M.; Malach, R.; and Fried, I. 2008. Internally generated reactivation of single neurons in human hippocampus during free recall. *Science*, 322(5898): 96–101.

Gidaris, S.; and Komodakis, N. 2018. Dynamic Few-Shot Visual Learning Without Forgetting. In *CVPR*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *NeurIPS*, volume 27.

Grilli, M. D.; and Glisky, E. L. 2013. Imagining a Better Memory: Self-Imagination in Memory-Impaired Patients. *Clinical Psychological Science*, 1(1): 93–99.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. 2017. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

He, J.; Erfani, S. M.; Ma, X.; Bailey, J.; Chi, Y.; and Hua, X.-S. 2021. α-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. In *NeurIPS*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2014. Distilling the Knowledge in a Neural Network. In *NeurIPS workshop*.

Hoffer, E.; and Ailon, N. 2015. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, 84–92. Springer.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2018. Lifelong learning via progressive distillation and retrospection. In *ECCV*, 437–452.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *CVPR*, 831–839.

Hsu, Y.-C.; Liu, Y.-C.; Ramasamy, A.; and Kira, Z. 2018. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. In *NeurIPS Continual learning Workshop*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kumaran, D.; Hassabis, D.; and McClelland, J. L. 2016. What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in cognitive sciences*, 20(7): 512–534.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE TPAMI*, 40(12).

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

McClelland, J. L.; McNaughton, B. L.; and O'Reilly, R. C. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3): 419.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, 109–165. Academic Press.

Mohri, M.; and Rostamizadeh, A. 2009. Rademacher complexity bounds for non-iid processes. In *NeurIPS*, 1097–1104.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of Machine Learning*. MIT press.

Ostapenko, O.; Puscas, M.; Klein, T.; Jahnichen, P.; and Nabi, M. 2019. Learning to remember: A synaptic plasticity driven framework for continual learning. In *CVPR*, 11321–11329.

Qi, H.; Brown, M.; and Lowe, D. G. 2018. Low-shot learning with imprinted weights. In *CVPR*, 5822–5830.

Rannen, A.; Aljundi, R.; Blaschko, M. B.; and Tuytelaars, T. 2017. Encoder based lifelong learning. In *CVPR*, 1320–1328.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *CVPR*.

Riemer, M.; Cases, I.; Ajemian, R.; Liu, M.; Rish, I.; Tu, Y.; and Tesauro, G. 2019. Learning to Learn without Forgetting by Maximizing Transfer and Minimizing Interference. In *ICLR*.

Schacter, D. L.; Addis, D. R.; Hassabis, D.; Martin, V. C.; Spreng, R. N.; and Szpunar, K. K. 2012. The future of memory: remembering, imagining, and the brain. *Neuron*, 76(4): 677–694.

Schuster, C.; Hilfiker, R.; Amft, O.; Scheidhauer, A.; Andrews, B.; Butler, J.; Kischka, U.; and Ettlin, T. 2011. Best practice for motor imagery: a systematic literature review on motor imagery training elements in five different disciplines. *BMC Medicine*, 9(1): 1–35.

Schwarz, J.; Czarnecki, W.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & Compress: A scalable framework for continual learning. In *ICML*.

Serra, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming Catastrophic Forgetting with Hard Attention to the Task. In *ICML*, volume 80, 4548–4557.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual Learning with Deep Generative Replay. In *NeurIPS*, volume 30.

Smolen, P.; Baxter, D. A.; and Byrne, J. H. 2019. How can memories last for days, years, or a lifetime? Proposed mechanisms for maintaining synaptic potentiation and memory. *Learning & Memory*, 26(5): 133–150.

Stanford. 2015. Tiny ImageNet Challenge (CS231n). http://tiny-imagenet.herokuapp.com/.

van de Ven, G. M.; Siegelmann, H. T.; and Tolias, A. S. 2020. Brain-inspired replay for continual learning with artificial neural networks. *Nature Communications*, 11(1): 1–14.

van de Ven, G. M.; and Tolias, A. S. 2018. Three continual learning scenarios. *NeurIPS Continual Learning Workshop*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *CVPR*, 5022–5030.

Wang, Z.; Duan, Y.; Liu, L.; and Tao, D. 2021. Continual Learning with Embeddings: Algorithm and Analysis. In *ICML 2021 Workshop on Theory and Foundation of Continual Learning*.

Wang, Z.; Liu, L.; and Tao, D. 2020. Deep Streaming Label Learning. In *ICML*, volume 119, 9963–9972.

Wei, T.; Shi, J.; and Li, Y. 2021. Probabilistic Label Tree for Streaming Multi-Label Learning. In *SIGKDD*, 1801–1811.

Wei, T.; Tu, W.-W.; and Li, Y.-F. 2019. Learning for Tail Label Data: A Label-Specific Feature Approach. In *IJCAI*, 3842–3848.

Welling, M. 2009. Herding Dynamical Weights to Learn. In *ICML*, 1121–1128.

Xi, H.; He, L.; Zhang, Y.; and Wang, Z. 2020. Bounding the efficiency gain of differentiable road pricing for EVs and GVs to manage congestion and emissions. *PloS one*, 15(7): e0234204.

Xiang, Y.; Fu, Y.; Ji, P.; and Huang, H. 2019. Incremental learning using conditional adversarial networks. In *ICCV*, 6619–6628.

Yin, D.; Farajtabar, M.; Li, A.; Levine, N.; and Mott, A. 2020. Optimization and Generalization of Regularization-Based Continual Learning: a Loss Approximation Viewpoint. *arXiv preprint arXiv:2006.10974*.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic drift compensation for class-incremental learning. In *CVPR*, 6982–6991.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *ICML*.

Zhu, J.; Luo, B.; Zhao, S.; Ying, S.; Zhao, X.; and Gao, Y. 2020. IExpressNet: Facial Expression Recognition with Incremental Classes. In *ACM International Conference on Multimedia*, 2899–2908.