On Learning the Geodesic Path for Incremental Learning

Christian Simon[†,§],   Piotr Koniusz[§,†],   Mehrtash Harandi[♣,§]

[†]The Australian National University   [♣]Monash University   [§]Data61-CSIRO

firstname.lastname@{`anu.edu.au`,`monash.edu`,`data61.csiro.au`}

Abstract

Neural networks notoriously suffer from the problem of catastrophic forgetting, the phenomenon of forgetting the past knowledge when acquiring new knowledge. Overcoming catastrophic forgetting is of significant importance to emulate the process of "incremental learning", where the model is capable of learning from sequential experience in an efficient and robust way. State-of-the-art techniques for incremental learning make use of knowledge distillation towards preventing catastrophic forgetting. Therein, one updates the network while ensuring that the network's responses to previously seen concepts remain stable throughout updates. This in practice is done by minimizing the dissimilarity between current and previous responses of the network one way or another. Our work contributes a novel method to the arsenal of distillation techniques. In contrast to the previous state of the art, we propose to firstly construct low-dimensional manifolds for previous and current responses and minimize the dissimilarity between the responses along the geodesic connecting the manifolds. This induces a more formidable knowledge distillation with smooth properties which preserves the past knowledge more efficiently as observed by our comprehensive empirical study. [1]

1. Introduction

Humans are able to gather and grow their knowledge gradually and effortlessly while preserving their *past knowledge*. In pursuing a similar capability at an algorithmic level and in the so-called Incremental Learning (IL), an artificial machine must learn and remember various tasks or concepts [22] to accomplish a more broadened set of objectives. Formally, a learning algorithm in IL will receive its objectives (*e.g.*, recognizing new classes) gradually. This is in contrast to conventional learning paradigm where the objectives and data are available from the beginning to the learning algorithm.

---

[1]Our code is available at https://github.com/chrysts/geodesic_continual_learning
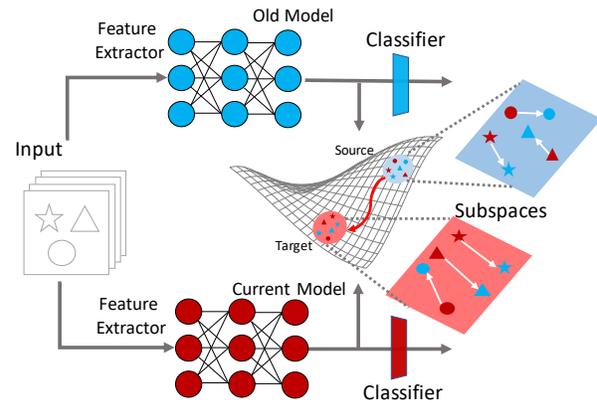
Figure 1: The visualization of our proposed method. To regularize the network for incoming tasks, knowledge from previous tasks is preserved by distilling the features following the geodesic path between two subspaces of different models. The distillation is based on the projection of two sets of features from two different networks.

The gradual nature of learning in IL poses serious difficulties as now the model has to sequentially learn and adapt to new tasks, while being vigilant to its needs and requirements (*e.g.*, limited memory to preserve the knowledge). As shown in the seminal works of Kirkpatrick *et al*. [20] and Rebuffi [34], the gradual nature of learning plus the presence of constraints and limitations in IL can degrade the performance of the model drastically, which is formally known as *"Catastrophic forgetting"* [7,28,29]. The phenomenon refers to a neural network experiencing performance degradation at previously learned concepts when trained sequentially on learning new concepts. This forgetting problem appears even worse when the model is sequentially updated for new tasks without considering the previous tasks as shown in [20,23] *i.e.*, learning new tasks overrides the knowledge from previous tasks. Finding the balance among tasks, also known as the stability-plasticity dilemma in [33], is crucial to achieve the IL ultimate goal.

In classical classification on visual data, a Convolutional Neural Network (CNN) is used to encode the input images

into features and a final layer consists of classifiers to map the features to the fixed number of classes. In class IL [38], a CNN keeps learning and grows its capacity to accommodate new classes or tasks. In order to achieve the equilibrium performance between learning new tasks and maintaining existing base performance, some IL methods store observed tasks in the memory and replay them [10, 25] to prevent *catastrophic forgetting*. However, learning through a large number of tasks limits the exemplars that can be reserved in the memory. Hou *et al*. [17] suggest to replay the exemplars in the memory on both old and current models and prevent the forgetting phenomenon with knowledge distillation.

The design of distillation loss remains an open research problem. A knowledge distillation on the output space (after the classifier) is firstly proposed in [23] so-called Learning without Forgetting (LwF). However, the study in [17, 34] shows that LwF lets the parameters of the new classes to become more dominant than the old parameters. As a result, the model tends to classify all test data to the new classes. Because of this reason, the classifier is created based on the *nearest mean of exemplars* and the distillation loss with the cosine distance in the feature space is suggested in [17]. All these prior methods do not consider the gradual change between tasks. In fact, human minds learn from one task to another task by *gradual walk* as described in [3]. Inspired by the idea of gradual change in human minds, we propose a distillation loss in IL by adopting the concept of geodesic flow between two tasks (see Fig. 1) called as **GeoDL**. To summarize, the contributions of our work are:

1. We propose a novel distillation loss that considers the information geometry aspect in incremental learning and leads to improved performance.

2. We show that using geodesic path for knowledge distillation yields less forgetful networks compared to the distillation losses using the Euclidean metric.

2. Related Work

In this section, we describe several approaches that prevent *catastrophic forgetting* for incremental learning.

Representative memories. In IL, the selection of exemplars in the memory is very crucial as we cannot access the whole data from past training phases. The selected exemplars are replayed as representatives from the past categories, as a result, the model does not overfit to the new categories. A method, so-called *herding* [34, 40], is used to pick potential exemplars for future use in training models. Rebuffi *et al*. [34] use prototypes (mean of each class) to find the closest samples to be stored in the memory. However, some selected samples may not be optimal for the future training phases, Liu *et al*. [24] propose a meta-learning approach to update the memory by the gradient descent. A memory

replay can also be achieved with a generative mechanism as proposed in [19, 35, 41], but this approach needs additional networks which are not easy to train and optimize. To efficiently keep exemplars in the memory, Iscen *et al*. [18] proposes a feature adaptation strategy.

Gradient trajectories. A family of these approaches regularize the update direction to be 'less forgetting'. Inspired by the synaptic consolidation from neurobiological models [8], Kirkpatrick *et al*. [20] propose Elastic Weight Consolidation (EWC) to regulate the update of DNN using the Fisher Information Matrix and accommodate both the solution from previous task and the new task, thus the training trajectories lead to low errors on both tasks. By allowing *positive* backward transfer, Gradient Episodic Memory (GEM) [25] adds a constraint considering the similarity between the gradients of the previous tasks and the gradients of the current task. Riemannian walk [4] provides a generalization framework, so-called EWC++, as an extension of EWC and an evaluation method for forgetting and intransigence.

Parameter updates. In this line of work, there strategy is to only update some parameters in which it is common using dropout [37] and attention [16] for standard classification. Mallya *et al*. [26, 27] encourage to use masking or pruning to update selected network parameters for every different task. Rajasegaran *et al*. [32] propose a simple controller with random search to choose the optimal paths for a new task.

Regularization with distillation losses. Training neural networks in the incremental fashion affects the model's capability to maintain the performance of existing categories. An effective strategy is to introduce knowledge distillation [15] between an old model and a new model. A knowledge distillation loss is performed on the output of neural networks as proposed in [23] such that it preserves old task performance. To avoid forgetting, Castro *et al*. [2] recommends balanced fine-tuning with temporary distillation loss to the classification layers. A distillation loss for incremental learning is extended to the feature space with a measurement using the Euclidean distance proposed in [6, 17]. However, all of these knowledge distillation methods do not consider to learn the underlying manifold structure between old and current tasks. Belouadah and Popescu [1] argue that knowledge distillation hurts the IL performance when there are at least a few examples. However, our setting in this work follows [17] in which the classifier weight is built based on the notion of cosine distance and not a fully connected layer as in [1].

Geodesic flows. The gradual changes between two different tasks can be modeled with the geodesic flow using projection to intermediate subspaces [9]. The projection to subspaces has been used previously in domain adaptation [12, 13] and few-shot learning [36]. In contrast, our proposed method employs the geodesic flow for knowledge distillation such that the learned features preserve similarity in the intermediate subspaces along the geodesic path.

3. Preliminaries

3.1. Class incremental learning setting

Below, we define the multi-class IL problem and closely follow the IL setup in [17]. Suppose a sequence of T tasks with associated data $\mathcal{D}_0, \cdots, \mathcal{D}_T$ is given. We denote the data in task t by $\mathcal{D}_t = \{(\boldsymbol{x}_i, y_i)_{i=1}^{N_t}\}$, with $\boldsymbol{x}_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. In contrast to classical training that has access to every training example, a certain budget is determined for the memory such that only limited amount of examples can be stored $\mathcal{M} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^{L}\}$. At time $t = 0$, the training data merely consists of $\mathcal{D}_0$. Afterwards, the training data is formed as the union of the data at given task and the memory (*i.e.*, $\mathcal{D}_t \cup \mathcal{M}$). Note that, each $\mathcal{D}_t$ for $t > 0$ often has less samples and novel categories (*e.g.* 2, 5, or 10) as compared to $\mathcal{D}_0$.

In IL, a DNN parameterized with $\boldsymbol{\Phi} = \{\boldsymbol{\theta}, \boldsymbol{\varphi}\}$ is used to realize a mapping in the form $\hat{y} = f(\boldsymbol{x}; \boldsymbol{\Phi})$. Here, $\boldsymbol{\theta}$ represents a part of network that encodes the input into a descriptor as $\boldsymbol{z} = f_{\mathrm{enc}}(\boldsymbol{x}; \boldsymbol{\theta})$. The descriptor is then passed through a classifier parameterized by $\boldsymbol{\varphi}$ to produce the prediction $\hat{y} = f_{\mathrm{cls}}(\boldsymbol{z}, \boldsymbol{\varphi})$. At time t, our goal is to update and improve our model at $\boldsymbol{\Phi}_t$ from the old model $\boldsymbol{\Phi}_{t-1}$ using $\mathcal{D}_t \cup \mathcal{M}$. This, in practice means that the classifier $\boldsymbol{\varphi}_t$ keeps expanding as more novel classes are observed.

Evaluating an IL Model. There are two important objectives to assess the performance of a model in IL problems. **1)**. The average accuracy ($\mathcal{A}$) to evaluate the capability of the model in learning new tasks. **2)**. Forgetting rate ($\mathcal{F}$) to evaluate how much the model suffers from *catastrophic forgetting*. For each task, we compute an accuracy score $\mathcal{A}_t$ for a test set that is representative for all classes seen so far (*i.e.*, classes seen from $\mathcal{D}_0$ up to $\mathcal{D}_t$). We quantify the forgetting rate by considering the accuracy for the test set at $t = 0$ using the model $\boldsymbol{\Phi}_0$ and the final model $\boldsymbol{\Phi}_T$. The average accuracy and forgetting rate are defined as:

$$\mathcal{A} = \frac{1}{T} \sum_{t=1}^{T} \mathcal{A}_t, \qquad \mathcal{F} = \mathcal{A}_0|_{\boldsymbol{\Phi}_0} - \mathcal{A}_0|_{\boldsymbol{\Phi}_T} . \qquad (1)$$

Here, we overloaded our notations slightly and denote the accuracy of a model $\boldsymbol{\Phi}_t$ using the test set of the task at time $t = 0$ by $\mathcal{A}_0|_{\boldsymbol{\Phi}_t}$. A successful model is one that has a high average accuracy and low forgetting rate.

3.2. Regularization with knowledge distillation

We start this part by reviewing how prior works mitigate the forgetting phenomenon with knowledge distillation. We then discuss how our proposed method extends the idea of distillation loss by considering the gradual change between tasks (see Fig. 2 for a conceptual diagram that highlights the differences between our approach and prior art). To prevent

catastrophic forgetting, distillation loss aims to minimize alteration on shared network parameters during the adaptation process (*i.e.*, learning a novel task). The distillation loss is employed along with the cross-entropy loss [17,24,34] or the triplet loss [43]. In the Learning without Forget (LwF) [23], the knowledge distillation is applied to minimize the dissimilarity between predictions of the old and new model. This is to ensure that predictions on previously seen classes do not vary drastically (*i.e.*, $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_t) \approx p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_{t-1})$), leading to maintaining prior knowledge. Formally and for a problem with K classes, the loss of LwF is defined as:

$$\mathcal{L}_{\mathrm{LwF}}(\boldsymbol{\Phi}_t, \boldsymbol{\Phi}_{t-1}) = \sum_{k=1}^{K} \left[p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_{t-1})\right]_k \log \left[p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_t)\right]_k$$

$$\left[p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\Phi}_t)\right]_k = \frac{\exp(f(\boldsymbol{x}, \boldsymbol{\Phi}_t)_k/\tau)}{\sum_{k=1}^{K} \exp(f(\boldsymbol{x}, \boldsymbol{\Phi}_t)_k/\tau)} . \qquad (2)$$

Here, $[\cdot]_k$ denotes the element at index k of a vector and τ is a temperature variable. Note that $\boldsymbol{y}$ belongs to the old class in $\mathcal{D}_{t-1}$.

Another form of distillation, again aiming to minimize the dissimilarity between predictions, is to constraint the latent features of the network (*i.e.*, $\boldsymbol{z} = f(\boldsymbol{x}; \boldsymbol{\theta})$). Hou *et al*. [17] propose the so-called *less-forget* constraint which employs a cosine embedding loss for knowledge distillation. Formally, we have:

$$\mathcal{L}_{\mathrm{Cos}}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}) = \sum_{\boldsymbol{x} \sim \{\mathcal{D}_t \cup \mathcal{M}\}} \left[1 - \mathrm{sim}\big(f(\boldsymbol{x}; \boldsymbol{\theta}_t), f(\boldsymbol{x}; \boldsymbol{\theta}_{t-1})\big)\right],$$

where:

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \frac{\boldsymbol{u}^\top \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}. \qquad (3)$$

The cosine distance used in Eq. 3 is based on the Euclidean metric between features from two models directly. The aforementioned distillation losses do not benefit from the manifold structure of each task.

4. Proposed Method

As described in §3.2, the existing methods for knowledge distillation do not take into account the gradual change between consecutive tasks. Furthermore, the Euclidean norm often used for distillation cannot capture the underlying geometry of two different feature spaces. Furthermore, the previous works did not explicitly benefit from the geometry and manifold structure of the tasks during performing distillation. To benefit from the manifold geometry during distillation, we propose to enforce consistency along the geodesic connecting the models $\boldsymbol{\Phi}_{t-1}$ and $\boldsymbol{\Phi}_t$ (see Fig. 3 for a conceptual diagram).
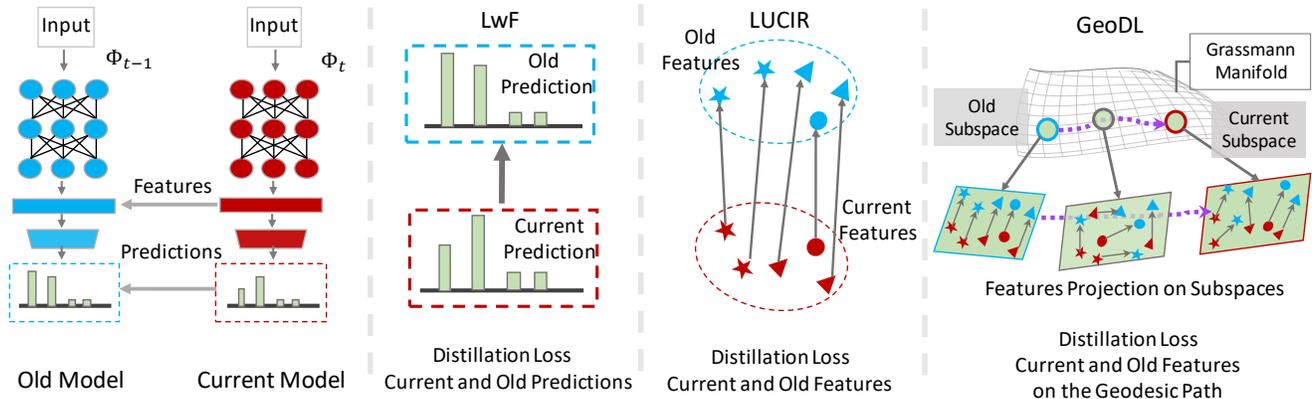
Figure 2: Comparison with prior knowledge distillation approaches for IL. Knowledge distillation is applied to the features or predictions from the old model Φ_{t-1} and the current model Φ_t. LwF [23] distillation loss is based on the predictions and LUCIR [17] preserves the knowledge from previous tasks using a cosine embedding loss. Our proposal, called GeoDL, makes use of *gradual walk* between two subspaces from old and current features.

4.1. Gradual walk with intermediate subspaces

We use the concept of subspaces to embed a set of features (*e.g.*, a mini-batch) from both old and current models. To this end, we model the geometry of latent samples in a model by a low-dimensional subspace with a basis[2] $P \in \mathbb{R}^{\mathrm{d}\times\mathrm{n}}$. Subspaces form a Riemannian manifold are studied using the geometry of Grassmann manifold.

Definition 1 (Grassmann Manifold) *The set of n-dimensional linear subspaces of $\mathbb{R}^d, 0 < n < d$ ("linear" will be omitted in the sequel) is termed the Grassmann manifold, and is denoted here by $\mathcal{G}(\mathrm{n},\mathrm{d})$. An element P of $\mathcal{G}(\mathrm{n},\mathrm{d})$ can be specified by a basis, i.e., a $d \times n$ matrix with orthonormal columns (i.e., $\mathcal{G}(\mathrm{n},\mathrm{d}) \ni P \Rightarrow P^\top P = \mathbf{I}_n$).*

The *gradual walk* is essentially the geodesic flow between two points on the Grassmann manifold. In contrast to the distillation losses using the Euclidean metric, our approach allows the features to be projected along the geodesic flow and can capture the smooth changes between tasks.

Given a batch of B data samples, let $\mathbf{Z}_{t-1}, \mathbf{Z}_t \in \mathbb{R}^{\mathrm{d}\times B}$ be the corresponding encodings by the model at time $t-1$ and t, respectively. We propose to model $\mathbf{Z}_{t-1}$ and $\mathbf{Z}_t$ by two low-dimensional manifolds. This, in our work is indeed the two subspaces spanning $\mathbf{Z}_{t-1}$ and $\mathbf{Z}_t$ which we denote by P_{t-1} and P_t. The geodesic flow between P_{t-1} and P_t denoted by $\Pi : \nu \in [0,1] \to \Pi(\nu) \in \mathcal{G}(\mathrm{n},\mathrm{d})$ is:

$$\Pi(\nu) = \begin{bmatrix} P_{t-1} & R \end{bmatrix} \begin{bmatrix} U_1 \Gamma(\nu) \\ -U_2 \Sigma(\nu) \end{bmatrix}, \qquad (4)$$

where $R \in \mathbb{R}^{\mathrm{d}\times(\mathrm{d-n})}$ is the orthogonal complement of P_{t-1} and the diagonal elements for $\Gamma(\nu)$ and $\Sigma(\nu)$ are

[2]Principal Component Analysis (PCA) can be used to obtain the basis of the subspace.

$\gamma_i = \cos(\nu\omega_i)$ and $\sigma_i = \sin(\nu\omega_i)$, respectively for $i = 1, 2, \cdots, \mathrm{n}$ and $0 \leq \omega_1 \leq \cdots \leq \omega_\mathrm{n} \leq \pi/2$. In the supplementary material of our paper, we provide more details about the geodesic flow on Grassmannian and how $R, \Gamma(\nu)$ and $\Sigma(\nu)$ can be obtained from P_{t-1} and P_t.

Note that both $P_{t-1}^\top P_t$ and $R^\top P_t$ share the same right singular vectors V, thus generalized Singular Value Decomposition (SVD) [39] can be employed to decompose the matrices.

4.2. Projection to the intermediate subspaces

We propose to distill the feature space of the previous model into the current one by considering the cosine similarity between projections along $\Pi(\nu)$. Let $z_{t-1} = f(x, \Phi_{t-1})$ and $z_t = f(x, \Phi_t)$ be the encodings of x according to the previous and current models, respectively. The inner product between projections of z_{t-1} and z_t on a subspace P is:

$$g_P(z_{t-1}, z_t) = \left\langle P^\top z_{t-1}, P^\top z_t \right\rangle = \left(P^\top z_{t-1}\right)^\top P^\top z_t$$
$$= \boxed{z_{t-1}^\top P P^\top z_t}.$$

From this, we can define the inner product along the geodesic flow as:

$$g_\Pi(z_{t-1}, z_t) := \int_0^1 g_{\Pi(\nu)}(z_{t-1}, z_t) \, d\nu$$
$$= \int_0^1 z_{t-1}^\top \Pi(\nu)\Pi(\nu)^\top z_t \, d\nu$$
$$= z_{t-1}^\top \underbrace{\left(\int_0^1 \Pi(\nu)\Pi(\nu)^\top \, d\nu\right)}_{Q} z_t.$$
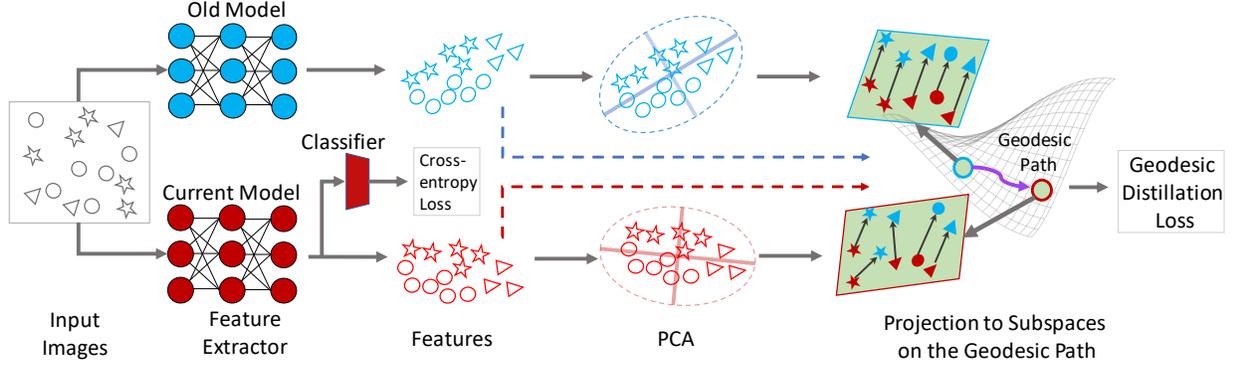
Figure 3: Pipeline of our proposed approach. We model the feature space of the previous $\boldsymbol{\Theta}_{t-1}$ and the current task $\boldsymbol{\Theta}_t$ by two low-dimensional subspaces and enforce distillation along the geodesic connecting them. This is achieved by aligning samples along projection on the subspaces on the geodesic flow.

Let $\boldsymbol{Q} = \boldsymbol{\Delta}\boldsymbol{\Lambda}\boldsymbol{\Delta}^{\top}$. It can be shown that:

$$
\begin{aligned}
\boldsymbol{\Delta} &= \begin{bmatrix} \boldsymbol{P}_{t-1}\boldsymbol{U}_1 & \boldsymbol{R}\,\boldsymbol{U}_2 \end{bmatrix}, \\
\boldsymbol{\Lambda} &= \int_0^1 \begin{bmatrix} \boldsymbol{\Gamma}(\nu)\boldsymbol{\Gamma}(\nu) & -\boldsymbol{\Gamma}(\nu)\boldsymbol{\Sigma}(\nu) \\ -\boldsymbol{\Sigma}(\nu)\boldsymbol{\Gamma}(\nu) & \boldsymbol{\Sigma}(\nu)\boldsymbol{\Sigma}(\nu) \end{bmatrix} d\nu.
\end{aligned}
\tag{5}
$$

Recall $\gamma_i = \cos(\omega_i)$ and $\sigma_i = \sin(\omega_i)$, we can calculate the diagonal elements:

$$
\begin{aligned}
\lambda_{1i} &= \int_0^1 \cos^2(\nu\omega)\, d\nu = 1 + \frac{\sin(2\omega_i)}{2\omega_i}, \\
\lambda_{2i} &= -\int_0^1 \cos(\nu\omega)\sin(\nu\omega)\, d\nu = \frac{\cos(2\omega_i)-1}{2\omega_i}, \\
\lambda_{3i} &= \int_0^1 \sin^2(\nu\omega)\, d\nu = 1 - \frac{\sin(2\omega_i)}{2\omega_i}.
\end{aligned}
\tag{6}
$$

This will enable us to compute $\boldsymbol{Q}$ in closed-form as (see our supplementary material for the detailed derivations):

$$
\boldsymbol{Q} = \boldsymbol{\Delta} \begin{bmatrix} \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 \\ \boldsymbol{\lambda}_2 & \boldsymbol{\lambda}_3 \end{bmatrix} \boldsymbol{\Delta}^{\top},
\tag{7}
$$

where $\boldsymbol{Q}$ is a $\mathrm{d} \times \mathrm{d}$ positive semi-definite matrix. In the final form, the inner product between features projected onto intermediate subspaces is:

$$
\mathrm{g}_\Pi \left(\boldsymbol{z}_{t-1}, \boldsymbol{z}_t \right) = \boldsymbol{z}_t^{\top} \boldsymbol{Q} \boldsymbol{z}_{t-1}.
\tag{8}
$$

Intuitively, $\boldsymbol{Q}$ is the matrix that defines the manifold structure between features of two different tasks. We argue that this facilitates distilling the knowledge by taking into account the smooth changes between the low-dimensional models (*i.e.*, manifolds) of the tasks. To use g_Π for training, we propose to minimize the following distillation loss, which can be understood as a general form of cosine similarity when the geodesic flow is considered:

$$
\mathcal{L}_{\text{GeoDL}} = 1 - \frac{\boldsymbol{z}_t^{\top}\boldsymbol{Q}\boldsymbol{z}_{t-1}}{\|\boldsymbol{Q}^{1/2}\boldsymbol{z}_t\|\|\boldsymbol{Q}^{1/2}\boldsymbol{z}_{t-1}\|}.
\tag{9}
$$

In practice, in order to better prevent *catastrophic forgetting*, we use an adaptive weighting scheme for GeoDL as $\beta_{ad} = \beta\sqrt{|N_{new}|/|N_{old}|}$. Note that this weighting scheme is a common practice for recent IL algorithms [17, 24]. Interestingly, one can recover the conventional distillation loss in Eq. 3 by choosing $\boldsymbol{P}_{t-1} = \boldsymbol{P}_t$, resulting in $\boldsymbol{Q} = \mathbf{I}$. This reveals that the conventional distillation loss is indeed a special case of our solution.

Algorithm 1 Train IL with GeoDL

Input: $\mathcal{D}_0, \cdots, \mathcal{D}_T$

1: $\boldsymbol{\Phi}_0 \leftarrow$ random initialization
2: Train $\boldsymbol{\Phi}_0$ on $\mathcal{D}_0$ minimizing $\mathcal{L}_{\text{CE}}$
3: Select and store exemplars to $\mathcal{M}$
4: **for** t in $\{1, ..., T\}$ **do**
5:     **while** not done **do**
6:         Sample a mini-batch $\{\boldsymbol{X}, \boldsymbol{Y}\}$ from $\{\mathcal{D}_t \cup \mathcal{M}\}$
7:         Get $\boldsymbol{Z}_{t-1} = f(\boldsymbol{X}, \boldsymbol{\Phi}_{t-1}), \boldsymbol{Z}_t = f(\boldsymbol{X}, \boldsymbol{\Phi}_t)$
8:         Compute $\boldsymbol{P}_{t-1} = \text{PCA}(\boldsymbol{Z}_{t-1})$
9:         Compute $\boldsymbol{P}_t = \text{PCA}(\boldsymbol{Z}_t)$
10:        Generate the geodesic flow using Eq. 7
11:        Project $\boldsymbol{Z}_t$ and $\boldsymbol{Z}_{t-1}$ to the geodesic path
12:        Minimize $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{GeoDL}}$
13:        Update $\boldsymbol{\Phi}_t$
14:     **end while**
15:     Evaluate on the test set
16:     Update exemplars from $\mathcal{D}_t$ and $\mathcal{M}$
17: **end for**

4.3. Classifiers and exemplars selection

Below we discuss the classifier design and sample selection which are also important for IL. Both of these

methods are widely known for tasks with visual data (*e.g.*, see [5, 11, 17, 30, 34]). Suppose the classifier encodes the template of class i by a normalized vector $\varphi_i/\|\varphi_i\|$. The likelihood of class i can be obtained as:

$$p(y_i|\boldsymbol{x}) = \frac{\exp\big(\text{sim}(\boldsymbol{\varphi}_i, \boldsymbol{z}_t)\big)}{\sum_j \exp\big(\text{sim}(\boldsymbol{\varphi}_j, \boldsymbol{z}_t)\big)}. \quad (10)$$

Note that φ is updated by SGD along other network parameters. The classification uses the cross-entropy loss $\mathcal{L}_{\text{CE}}$.

In order to select exemplars to be stored in the memory, we use the *herding* method by selecting top-k samples with the nearest cosine distance to the mean-embedding classifier. Algorithm 1 details out the steps of our approach. Note that our approach is an end-to-end training algorithm, thus the algorithm learns embeddings which transition smoothly between tasks thanks to the geodesic flow.

5. Experiments

In this section, we examine our proposed method and compare with the state of the art on several datasets: CIFAR-100, ImageNet-subset, and ImageNet-full.

5.1. Dataset and implementation

The CIFAR-100 dataset [21] contains 100 classes and each class has 500 train and 100 test images with the size of 32×32. The ImageNet dataset is evaluated for two different splits: ImageNet-subset with 100 categories and ImageNet-full with 1000 categories. Both ImageNet and CIFAR-100 datasets are used to examine our method. Basic data augmentation is employed when training our algorithm *e.g.* image flipping, color jitter, and random crop for all datasets. For a fair comparison, arrangement of all classes in our experiment closely follows a random seed (1993) in the same way as implemented in [17,24] and our performance is evaluated on three different runs. The first sequence ($t = 0$) is trained with 50 classes for CIFAR-100 and ImageNet-subset and 500 classes for ImageNet-full, afterwards the IL tasks *e.g.* 5, 10, and 25 are performed to evaluate and compare our proposed approach with the state-of-the-art techniques.

The CNN backbone used for comparison on CIFAR-100 is ResNet-32 [14] following the network architecture in [17]. On ImageNet-subset and ImageNet-full, we evaluate the proposed approach using ResNet-18 [14]. The Stochastic Gradient Descent (SGD) optimizer is applied to train the CNN for all experiments with an initial learning rate set to 0.1 and reduced by a factor 0.1 every half and three quarters of the total epochs. The model is trained on each sequence with 160, 90, and 90 epochs on CIFAR-100, ImageNet-subset, and ImageNet-full, respectively. We set the mini-batch size 128 and $\beta = 6.0$ for all datasets. We fix the memory size and set to 20 exemplars per class for all experiments unless

otherwise specified. We use the PyTorch package [31] with automatic differentiation in our implementation.

5.2. Evaluations

To evaluate our technique, we compare with the state of the art in IL, namely LwF [23], iCARL [34], BiC [42], LUCIR [17], and Mnemonics [24]. Note that, we cite the results of Mnemonics [24] from the corrected version on arXiv that is different from the published version (CVPR2020). Among all, LwF [23] and LUCIR [17] are the methods that propose knowledge distillation to prevent *catastrophic forgetting*. To provide reference, we also report the upper-bound (joint-CNN) in which all data from previous tasks is available and not limited to only samples from the memory. We adopt several baselines to compare with the-state-of-the-art methods. For selecting exemplars in the memory (*herding*), we employ the *nearest-mean-selection* technique and the *nearest-mean-of-exemplars* classifier in iCARL [34] with GeoDL. Furthermore, we also incorporate our method with LUCIR [17] by replacing the feature distillation loss $\mathcal{L}_{\text{cos}}$ by our distillation loss $\mathcal{L}_{\text{GeoDL}}$ while the cross-entropy loss and the margin ranking loss remain in the objective function. We show and compare our method on CIFAR-100, ImageNet-subset, and ImageNet-full in Table 1. In addition, we also plot the accuracy of GeoDL + LUCIR in comparison with prior methods in Fig. 4. Please refer to our supplementary material for additional plots and experiments.

Results on CIFAR-100. Table 1 provides comparisons with the state-of-the-art methods on CIFAR-100. GeoDL improves the basic iCARL [34] method (without knowledge distillation) by 8%, 13%, and 15% for 5, 10, and 25 tasks, respectively. Furthermore, our GeoDL also improves the LUCIR algorithm [17] by 2%, 5%, and 5% for 5, 10, and 25 tasks, respectively. Our method also has lower forgetting rates in contrast to recent IL algorithms across different numbers of tasks. Fig. 4 (left) also shows that GeoDL consistently achieves higher accuracy than knowledge distillation based methods *e.g.*, LUCIR [17] and LwF [23] for each task. Our method tops the balance to learn new tasks and avoid *catastrophic forgetting*.

Results on ImageNet. Table 1 also reports the average accuracy of our model. Fig. 4 (middle) shows the accuracy for each task in comparison to prior methods for IL on ImageNet-subset. Our results suggest that GeoDL improves the baseline methods by notable margin. The basic iCARL [34] method without GeoDL achieves the classification accuracy 70.10%, 70.86%, and 70.72% for 5, 10, and 25 tasks, respectively. Furthermore, GeoDL also improves the LUCIR algorithm [17] with the highest accuracy 73.87%, 73.55%, and 71.72% for 5, 10, and 25 tasks, respectively. Our method also has low forgetting rates across different numbers of tasks. The evaluation on ImageNet-full shows that our method outperforms the prior methods for IL where
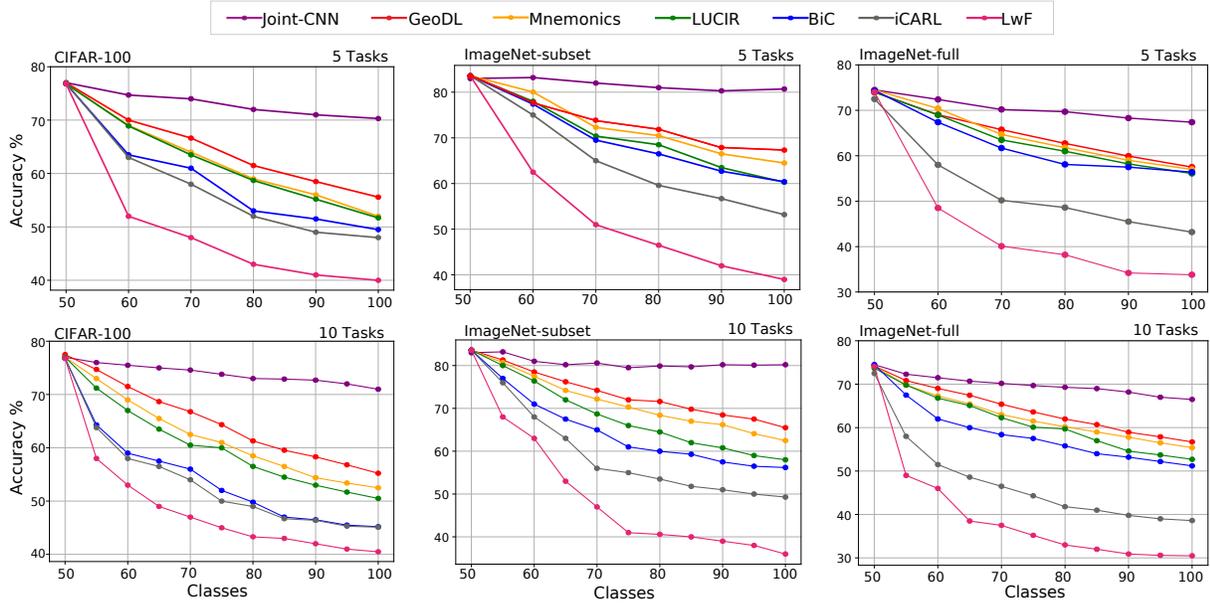
Figure 4: The accuracy for each task (5 and 10 tasks) on CIFAR-100 (left), ImageNet-subset (middle), and ImageNet-full (right). The x-axis shows the number of classes learned at a specific time and y-axis is the corresponding accuracy.

Method	CIFAR-100			ImageNet-subset			ImageNet-full		
Average accuracy (%)	5	10	25	5	10	25	5	10	25
LwF [23]	49.59	46.98	45.51	53.62	47.64	44.32	44.35	38.90	36.87
iCARL [34]	57.12	52.66	48.22	65.44	59.88	52.97	51.50	46.89	43.14
BiC [42]	59.36	54.20	50.00	70.07	64.96	57.73	62.65	58.72	53.47
LUCIR [17]	63.17	60.14	57.54	70.84	68.32	61.44	64.45	61.57	56.56
Mnemonics [24]	63.34	62.28	60.96	72.58	71.37	69.74	64.63	63.01	61.00
iCARL + GeoDL	62.54	61.40	61.84	70.10	70.86	70.72	60.02	57.98	56.70
LUCIR + GeoDL	**65.14**	**65.03**	**63.12**	**73.87**	**73.55**	**71.72**	**65.23**	**64.46**	**62.20**
Forgetting rate (%)	5	10	25	5	10	25	5	10	25
LwF [23]	43.36	43.58	41.66	55.32	57.00	55.12	48.70	47.94	49.84
iCARL [34]	31.88	34.10	36.48	43.40	45.84	47.60	26.03	33.76	38.80
BiC [42]	31.42	32.50	34.60	27.04	31.04	37.88	25.06	28.34	33.17
LUCIR [17]	18.70	21.34	26.46	31.88	33.48	35.40	24.08	27.29	30.30
Mnemonics [24]	10.91	13.38	19.80	17.40	17.08	20.83	13.85	15.82	19.17
iCARL + GeoDL	12.20	21.10	26.84	26.84	22.44	24.88	21.84	22.87	28.22
LUCIR + GeoDL	**9.49**	**9.10**	**12.01**	**13.78**	**12.68**	**15.21**	**11.03**	**12.81**	**15.11**

Table 1: The average accuracy and the forgetting rate on ImageNet-subset. The numbers of tasks T are set to 5, 10, and 25. Ideally, each method must find the balance to achieve the high average accuracy and the low forgetting rate.

it achieves high average accuracy and prevents *catastrophic forgetting*.

5.3. Ablation studies

Below, we investigate how our method improves the basic approach for IL without considering additional loss functions and exemplars selection. We only use the cross-entropy loss and the *herding* selection method [34] for exemplars in our ablation studies. We compare GeoDL with the knowl-

edge distillation approaches proposed in LwF [23] and LU-CIR [17]. In this study, we examine these distillation losses and ours w.r.t. various classifiers and number of exemplars in the memory on CIFAR-100.

Impact of different classifiers. We employ three different classifiers to examine the behavior of our method. The first classifier is based on learnable class prototypes φ, one per class. Following work [17], we refer this classifier as CNN. The second classifier is based on selecting the k-
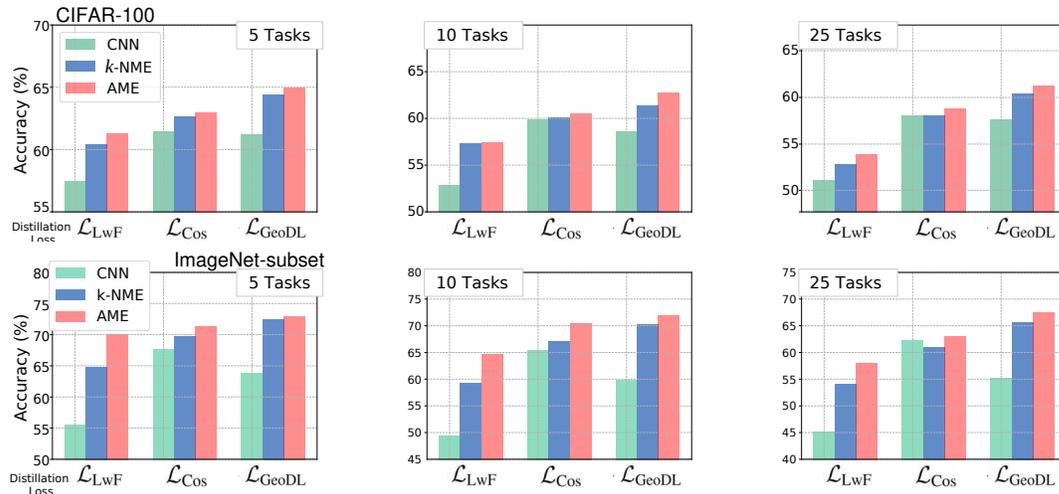
Figure 5: The average accuracy on CIFAR-100 (top) and ImageNet-subset (bottom) by varying the number of tasks (5, 10, 25) and classifiers (CNN, k-NME, AME). The objective function contains only cross-entropy and distillation losses. The memory size is 20 exemplars per class.

Method	Classifier	Average accuracy (%) Memory size per class				
		20	40	60	80	100
$\mathcal{L}_{CE}$	CNN	50.67	56.63	61.71	62.61	65.91
	k-NME	51.42	56.12	61.11	61.88	64.81
	AME	53.01	56.77	61.64	62.26	64.35
$\mathcal{L}_{CE} + \mathcal{L}_{LwF}$ [23]	CNN	52.84	58.84	63.14	64.94	67.24
	k-NME	57.31	59.07	63.14	64.35	66.53
	AME	57.41	59.21	63.47	64.57	66.73
$\mathcal{L}_{CE} + \mathcal{L}_{Cos}$ [17]	CNN	59.82	61.55	65.39	65.11	67.68
	k-NME	60.09	61.32	64.86	64.34	66.84
	AME	60.46	61.21	64.85	64.32	66.78
$\mathcal{L}_{CE} + \mathcal{L}_{GeoDL}$	CNN	58.54	58.72	**66.00**	**66.55**	**68.13**
	k-NME	61.40	61.86	65.82	65.54	67.31
	AME	**62.77**	**62.78**	65.92	65.58	67.38

Table 2: The average accuracy for 10 tasks on CIFAR-100 by varying the number of exemplars in the memory.

nearest neighbors from the samples within the same class (k-NME) [17]. The third classifier is based on the class embedding of all samples (AME) [17]. Fig. 5 shows that all distillation losses using AME achieve better accuracy in comparison to other classifiers. Furthermore, our proposed method achieves the best performance given AME. GeoDL improves the performance for IL without any additional losses. We note that our method does not improve the performance of CNN classifier by a large margin because the CNN classifiers trained on past tasks do not get updated when the model learns a new task.

Impact of increasing the number of exemplars in the memory. We investigate the accuracy for 20, 40, 60, 80, 100 exemplars in the memory. In order to investigate the trend of increasing the number of exemplars, only the cross-entropy loss is appended in addition to the baseline ablation

setting. Table 2 shows that GeoDL outperforms the accuracy of training across various numbers of exemplars in the memory given the LwF [23] with the feature distillation loss [17]. The CNN classifier attains higher performance given more exemplars compared to k-NME and AME due to training with a well-balanced number of exemplars per class. We also note that a high number of exemplars (*e.g.*, 100 on CIFAR-100) in the memory helps close the performance gap between training given only the standard cross-entropy loss *vs.* using the additional distillation loss.

Impact of subspace dimension on the result. In our method (see PCA in Algorithm 1), the subspace dimension n is a hyperparameter to be tuned. Table 3 shows the average accuracy for 10 tasks on CIFAR-100. Our method is robust to the choice of n and we set $n = 127$ in general.

Subspace Dimension n	16	32	64	127
GeoDL	61.87	61.04	61.66	62.77

Table 3: Results w.r.t. the subspace dimension (CIFAR-100).

6. Conclusions

We have presented a novel distillation loss for IL called GeoDL. In contrast to the prior methods, GeoDL considers the gradual change between consecutive tasks of IL to prevent *catastrophic forgetting*. To this end, our objective function uses the geodesic path between the representations of current task and old task, which results in a smooth transition of the learning process. Our approach achieves competitive results compared to the state of the art for IL on various datasets. Furthermore, GeoDL consistently improves existing baselines and outperforms prior knowledge distillation techniques. The ablation studies also highlight that GeoDL performs better than previous distillation losses for IL.

References

[1] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 583–592, 2019. 2

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2

[3] Stephan K Chalup. Incremental learning in biological and machine learning systems. *International Journal of Neural Systems*, 12(06):447–465, 2002. 2

[4] 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)*, pages 532–547, 2018. 2

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019. 6

[6] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[7] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1

[8] Stefano Fusi, Patrick J Drew, and Larry F Abbott. Cascade models of synaptically stored memories. *Neuron*, 45(4):599–611, 2005. 2

[9] Kyle A Gallivan, Anuj Srivastava, Xiuwen Liu, and Paul Van Dooren. Efficient algorithms for inferences on grassmann manifolds. In *IEEE Workshop on Statistical Signal Processing, 2003*, pages 315–318. IEEE, 2003. 2

[10] Alexander Gepperth and Cem Karaoguz. A bio-inspired incremental learning architecture for applied perceptual problems. *Cognitive Computation*, 8(5):924–934, 2016. 2

[11] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4367–4375, 2018. 6

[12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012. 2

[13] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006, 2011. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[15] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 2

[16] Jie Hong, Pengfei Fang, Weihao Li, Zhang Tong, Christian Simon, Lars Petersson, and Mehrtash Harandi. Reinforced attention for few-shot learning and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 2

[17] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 4, 5, 6, 7, 8

[18] Ahmet Iscen, Jeffrey Zhang, Svetlana Lazebnik, and Cordelia Schmid. Memory-efficient incremental learning through feature adaptation. *arXiv preprint arXiv:2004.00713*, 2020. 2

[19] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *International Conference on Learning Representations*, 2018. 2

[20] 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. 1, 2

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 6

[22] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17(4), 2007. 1

[23] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2, 3, 4, 6, 7, 8

[24] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2, 3, 5, 6, 7

[25] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pages 6467–6476, 2017. 2

[26] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision*, 2018. 2

[27] 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*, pages 7765–7773, 2018. 2

[28] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 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, 1995. 1

[29] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning prob-

lem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[30] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *European Conference on Computer Vision*, pages 488–501, 2012. 6

[31] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Autodiff Workshop*, 2017. 6

[32] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12669–12679, 2019. 2

[33] Björn Rasch and Jan Born. About sleep's role in memory. *Physiological reviews*, 2013. 1

[34] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2, 3, 6, 7

[35] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017. 2

[36] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4136–4145, 2020. 2

[37] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15:1929–1958, 2014. 2

[38] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 2

[39] Charles F Van Loan. Generalizing the singular value decomposition. *SIAM Journal on numerical Analysis*, 13(1):76–83, 1976. 4

[40] Max Welling. Herding dynamic weights for partially observed random field models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 599–606, 2009. 2

[41] Chenshen Wu, Luis Herranz, Xialei Liu, Joost van de Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. In *Advances in Neural Information Processing Systems*, pages 5962–5972, 2018. 2

[42] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019. 6, 7

[43] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991, 2020. 3