Towards Continual Learning Desiderata via HSIC-Bottleneck Orthogonalization and Equiangular Embedding

Depeng Li[1]*, Tianqi Wang[1]*, Junwei Chen[1], Qining Ren[1], Kenji Kawaguchi[2], Zhigang Zeng[1]†

[1]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]School of Computing, National University of Singapore
{dpli, tianqiwang, junwei_chen, qiningren, zgzeng}@hust.edu.cn, kawaguch@csail.mit.edu

Abstract

Deep neural networks are susceptible to catastrophic forgetting when trained on sequential tasks. Various continual learning (CL) methods often rely on exemplar buffers or/and network expansion for balancing model stability and plasticity, which, however, compromises their practical value due to privacy and memory concerns. Instead, this paper considers a strict yet realistic setting, where the training data from previous tasks is unavailable and the model size remains relatively constant during sequential training. To achieve such desiderata, we propose a conceptually simple yet effective method that attributes forgetting to layer-wise parameter overwriting and the resulting decision boundary distortion. This is achieved by the synergy between two key components: HSIC-Bottleneck Orthogonalization (HBO) implements non-overwritten parameter updates mediated by Hilbert-Schmidt independence criterion in an orthogonal space and EquiAngular Embedding (EAE) enhances decision boundary adaptation between old and new tasks with predefined basis vectors. Extensive experiments demonstrate that our method achieves competitive accuracy performance, even with absolute superiority of zero exemplar buffer and 1.02× the base model.

Introduction

Current deep learning models have shown promising performance in various fields, but they lack the ability of continual learning (CL) that humans possess (Kang et al. 2022; Smith et al. 2023). CL entails progressively acquiring knowledge from sequentially presented tasks, with access to only current task data and no past data (Li and Zeng 2023b). As a result, directly retraining a well-trained model on new task data using stochastic gradient descent (SGD) leads to the well-known phenomenon of *catastrophic forgetting* (McCloskey and Cohen 1989), which refers to abrupt and significant performance degradation on previously learned tasks.

Recent works have experienced a remarkable surge in addressing catastrophic forgetting (Wang et al. 2021a; Tong et al. 2023; Zhou et al. 2023). However, it is noteworthy that the merits of CL come with costs. *Rehearsal-based approaches*, as the mainstay of CL, explicitly buffer a small subset of past samples and retrain them with those from a new task jointly (Liu et al. 2020; Hayes et al. 2020; Boncicelli et al. 2022; Guo, Liu, and Zhao 2022; Luo et al. 2023). Critically, these methods pose a threat to data privacy and often decline performance as buffer size decreases, as depicted in Figure 1(a). *Architecture-based approaches* dynamically modify the network architecture to accommodate knowledge needed for new tasks (Serrà et al. 2018; Ke et al. 2021; Yang et al. 2023a; Hu et al. 2023). In particular, network expansion involves adding a sub-network for each task and utilizing aggregated feature representation for final prediction (Yan, Xie, and He 2021; Wang et al. 2022a). As shown in Figure 1(b), their model size expands rapidly as the number of tasks grows, which should be counted into the memory budget for a fair comparison (Zhou et al. 2023). *Regularization-based approaches* penalize parameter variations over an over-parameterized network, where each network parameter is associated with weight importance (Kirkpatrick et al. 2017; Zeng et al. 2019; Wołczyk et al. 2022). However, the performance of these methods that do not store any past data is yet unsatisfactory, especially in the class-incremental learning (class-IL) scenario, which addresses the most common problem of incrementally learning new classes without the provision of test-time task identities (Zhuang et al. 2022; Wang et al. 2022b) (see Figure 1(c)).

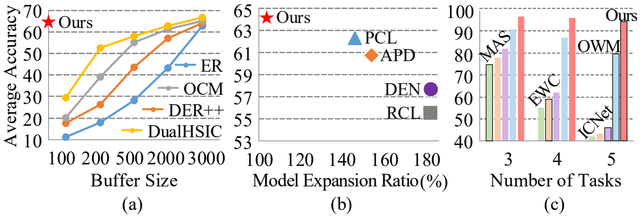

Figure 1: Comparison between our method and representative CL approaches. (a) Rehearsal-based ones are often sensitive to buffer sizes. (b) Some architecture-based ones scale rapidly during sequential training. (c) Most regularization-based ones struggle with the stability-plasticity dilemma whose performance is not satisfactory in the class-IL (hybrid with (a) or/and (b) excluded). By contrast, our method reaches multiple CL desiderata simultaneously.

*These authors contributed equally.

†Corresponding author.

In summary, for alleviating catastrophic forgetting, many CL methods prioritize accuracy performance to the detriment of other fronts. This motivates us to find new methods against forgetting while satisfying multiple CL desiderata: (i) *It should no longer access the training data of previous tasks*. While keeping prior observations demonstrates superior ability in combating forgetting, reliance on rehearsal buffers may not be memory-efficient (Zhang et al. 2020; Li et al. 2023b; Luo et al. 2023). Importantly, this involves violating practical constraints such as privacy and security issues (Shokri and Shmatikov 2015), which are common in domains like federated learning (Qi, Zhao, and Li 2023). (ii) *It should remain the model size relatively unchanged during sequential training.* Instead of buffering data, storing backbones from the history (e.g., network expansion) pushes the performance towards the upper bound progressively (Yan, Xie, and He 2021; Zhou et al. 2023; Hu et al. 2023). The drawback lies in the growth can be computationally expensive and it is intractable to customize the growth quota compactly matching the difficulty of a newly arriving task (Dai et al. 2019). (iii) *It should strike a balance between the stability and plasticity* (Rajasegaran et al. 2019; Kim and Han 2023), ensuring not only the persistent knowledge retention of past tasks but also the sufficient capacity to accommodate new ones. Intuitively, it is difficult, if not impossible, for pure parameter regularization to achieve such balance via the current learning paradigm.

With these considerations, a straightforward attempt is to seek alternative solutions to parameter regularization, with extra care to not infringe its inherent merits. Inspired by this insight, in this paper, we develop a drastically different training objective that *recasts* a representative regularizer in reaching multiple CL desiderata. Termed *CLDNet*, we avoid the conventional cross-entropy loss and instead incorporate statistical dependency and distance metric, achieving a better stability-plasticity trade-off in a rehearsal-free and minimal-expansion fashion. To this end, we decompose the CL process into dual problem-solving: (1) How to address the layer-wise parameter overwriting due to the scarcity of prior task data? (2) How to mitigate the inter-/intra-task confusion caused by distortions in decision boundaries?

To approach the first question, we leverage the interplay of Hilbert-Schmidt independence criterion (HSIC) and orthogonal projection, hence the name HSIC-Bottleneck Orthogonalization (HBO). Taking a close look at both: HSIC is a non-parametric kernel-based technique utilized to assess the statistical (in)dependence of different layers, which has been widely adopted for various learning tasks (Wang, Dai, and Liu 2021) but is under-investigated in CL community (Wang et al. 2023); And a basic idea behind the orthogonal projection is to regularize gradient update directions that do not disturb the weights of previous tasks (Zeng et al. 2019). Based on them, the introduced HBO implements non-overwritten parameter updates facilitated by the HSIC-bottleneck training in an orthogonal space, where one can exploit readily available gradient updates by measuring nonlinear dependencies between the inputs and outputs. It requires no access to or storing of previous data, no architecture growth, and no awareness of test-time task identities.

To address the aforementioned second question, we draw inspiration from the recently proposed equiangular basis vectors (EBVs) (Shen, Sun, and Wei 2023). Unlike the trainable fully-connected layer with softmax, the EBVs is parameter-free since its learning objective is to minimize the spherical distance of learned representations with predefined basis vectors. This ensures that the trainable parameters of deep neural networks are constant even with the growth of tasks. Though attractive, it remains unclear how to extend EBVs to CL. To bridge this gap, we design an EquiAngular Embedding (EAE) component that sits atop HBO. During each CL session, we vectorize the embedding of HBO output and optimize it towards its class-specific equiangular basis vector in a scalable manner. Compared to the standard classification, EAE exhibits a stronger discriminative ability through the tailored distance metric, thereby enhancing decision boundary adaptation between old and new tasks.

Benefiting from the synergy between HBO and EAE, our CLDNet reaches multiple CL desiderata. Empirical evaluation across a range of widely used benchmark datasets demonstrates the superiority of our approach in terms of exemplar buffers, network expansion, and competitive performance. On CIFAR-100, for instance, CLDNet outperforms the state-of-the-art rehearsal-based baseline by 7.54% with a minimal expansion ratio of 1.02× and buffer size of 0.

Related Work

Prior works to address catastrophic forgetting in CL can be broadly divided into three categories, from which we discuss a selection of representatives and focus on ones mostly related to our work. Rehearsal-based approaches preserve model stability by keeping a memory buffer of past samples at either input layer (Liu et al. 2020; Bonicelli et al. 2022) or hidden layer (Yang et al. 2023b) for joint training. GDumb (Prabhu, Torr, and Dokania 2020) employs a limited memory to buffer data in the order of arrival and dynamically replaces previously stored data. Rather than pixel-level exemplars, i-CTRL (Tong et al. 2023) is founded on compact and structured representations, while REMIND (Hayes et al. 2020) stores hidden representations and reconstructs synthesized features for rehearsal.

Architecture-based approaches either isolate model parameters (Serrà et al. 2018; Ke et al. 2021) or expand additional network branches (Yang et al. 2023a; Hu et al. 2023). PNN (Rusu et al. 2016) gradually adds new branches for all layers horizontally. RPS-Net (Rajasegaran et al. 2019) uses parallel modules at each layer where a possible searching space is formed to contain previous task-specific knowledge. Methods such as PCL (Hu et al. 2021), DER (Yan, Xie, and He 2021), and FOSTER (Wang et al. 2022a) acquire sufficient model plasticity by allocating a sub-network per task.

Regularization-based approaches mainly employ penalty terms to impose constraints on weights deemed important for old tasks (Zhang et al. 2020; Li and Zeng 2023a). The pioneering work was conducted by EWC (Kirkpatrick et al. 2017), followed by SI (Zenke, Poole, and Ganguli 2017), and MAS (Aljundi et al. 2018). On the other hand, orthogonal projection-driven methods address forgetting by designing network parameter updating rules (Farajtabar et al. 2020;

Li et al. 2023a). OWM (Zeng et al. 2019) constructs an orthogonal projector such that its gradient updates only occur in directions orthogonal to the input of previous tasks. In the multi-head setting (e.g., a separate classifier per task), GPM (Saha, Garg, and Roy 2021) stores the bases of core gradient space while FS-DGPM (Deng et al. 2021) further predicts the importance of such bases aided with a rehearsal buffer. Our work is also built on the basis of orthogonal projection but is very different from existing approaches as its training objective consists of the statistical dependency and distance metric, achieving a better stability-plasticity tradeoff in a rehearsal-free and minimal-expansion fashion.

Among the latest CL approaches, our work is closely related to AOP (Guo et al. 2022), OCM (Guo, Liu, and Zhao 2022), and DualHSIC (Wang et al. 2023). (1) AOP aims to improve OWM itself by introducing a rule of expectation serving to strengthen orthogonal projectors. By contrast, inspired by HSIC (Ma, Lewis, and Kleijn 2020), we reformulate the OWM process as dependence minimization or maximization problems in a unified way. To the best of our knowledge, it is the first orthogonal projector that utilizes HSIC for CL. (2) OCM turns to a complicated contrastive learning proxy over *two models* to maximize the mutual information (MI), while CLDNet's HBO detects nonlinear dependencies with the advantage of easy empirical estimation over MI. (3) DualHSIC realizes CL by considering the intertask relationship into task-specific and task-invariant knowledge, while CLDNet directly addresses forgetting via nonparameter overwriting and decision boundary adaptation. Note that both OCM and DualHSIC rely on rehearsal buffers that we do not. Interestingly, we observe that both OCM and DualHSIC require the additional trainable projection head; Instead, we bring a parameter-free classifier to CL, which enhances model decision ability as evidenced by the recent study (Shen, Sun, and Wei 2023). Therefore, our work differs significantly in terms of motivation and methodology.

Preliminaries

Hilbert-Schmidt Independence Criterion HSIC is a kernel-based measure of dependence between random variables (Gretton et al. 2005). With it, one can transform many existing learning tasks into statistical independence minimization (or maximization) problems, much akin to MI. Unlike the indirect variational bounds on MI (Poole et al. 2019), it can be directly estimated given a finite number of observations. Therefore, it has been used in a variety of applications for machine learning (Ma, Lewis, and Kleijn 2020; Wang et al. 2021b; Li et al. 2021; Kawaguchi et al. 2023).

Formally, given two random variables X and Y jointly drawn from probability distribution P_{XY}, HSIC identifies their dependency by first taking a nonlinear feature transformation of each, say $\phi : X \to \mathcal{H}$ and $\psi : Y \to \mathcal{G}$, with h and g being kernel functions in the $\mathcal{H}$ and $\mathcal{G}$ Hilbert spaces respectively. Let $(X^{'}, Y^{'})$ be independent copies of (X, Y):

$$\begin{aligned}\mathrm{HSIC}(P_{XY},\mathcal{H},\mathcal{G}) &= \mathbb{E}_{XYX'Y'}[h(X,X^{'})g(Y,Y^{'})] \\ &+ \mathbb{E}_{XX'}[h(X,X^{'})]\mathbb{E}_{YY'}[g(Y,Y^{'})] \\ &- 2\mathbb{E}_{XY}[\mathbb{E}_{X'}[h(X,X^{'})]\mathbb{E}_{Y'}[g(Y,Y^{'})]]\end{aligned} \tag{1}$$

The formulation suggests that HSIC captures nonlinear dependencies between X and Y, with the magnitude of the index indicating the strength of associations.

To render HSIC a practical measure for learning tasks, it has proven to be easily evaluated from mini-batches of data (Gretton et al. 2005; Song et al. 2012). Given n i.i.d. samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ drawn from P_{XY}, the empirical estimation of HSIC is:

$$\widehat{\mathrm{HSIC}}(\mathcal{D},\mathcal{H},\mathcal{G}) = (n-1)^{-2}\mathrm{tr}(HKGK) \tag{2}$$

where $\mathrm{tr}(\cdot)$ is the trace operator, $H_{ij} = h(x_i, x_j)$ and $G_{ij} = g(y_i, y_j)$ are kernel matrices, and $K = I_n - \frac{1}{n}1_n 1_n^{\mathrm{T}}$ is the centering matrix. In our CLDNet, we evaluate the HSIC term in this empirical expression and denote it as $\mathrm{HSIC}(X, Y)$.

Equiangular Basis Vectors Although a trainable classifier with softmax cross-entropy remains the predominant approach for classification tasks, the potential of using a preassigned classifier has been explored (Mettes, Van der Pol, and Snoek 2019; Zhou et al. 2022). The general idea is to replace the classifier weight with the class prototype, e.g., fixing the classifier as a Hadamard matrix (Hoffer, Hubara, and Soudry 2018), regular polytope (Pernici et al. 2021), and simplex ETF (Yang et al. 2022). In this paper, we are encouraged by the recently proposed equiangular basis vectors (EBVs) (Shen, Sun, and Wei 2023), characterized by simplicity and effortless implementation over prior studies.

Prior to learning, the EBVs assigns C d-dimensional basis vectors on the surface of a unit hypersphere $S^d \in \mathbb{R}^d$, denoted by the set $\mathcal{W} = \{w_c\}_{c=1}^{C}$. In the predefined process, these basis vectors are pairwise separated by the common angle $\gamma \in [0, 1)$, which satisfies:

$$-\gamma \leq \frac{w_i^{\mathrm{T}} \cdot w_j}{\|w_i\| \|w_j\|} \leq \gamma, \forall w_i, w_j \in \mathcal{W}, i \neq j \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm. Let φ denote the spherical distance function. The EBVs produces a distribution over classes based on softmax:

$$p(y = y_c|z) = \frac{\exp(\varphi(z, w_c))}{\sum_{c'=1}^{C}\exp(\varphi(z, w_{c'}))} \tag{4}$$

where $z \in \mathbb{R}^d$ denotes the learned feature representation by a backbone network given the input x and y is the class label. During the learning process, its objective is to minimize the spherical distance of feature representations with predefined basis vectors within such a set $\mathcal{W}$.

Methodology

We propose a conceptually simple yet effective method that reaches multiple Continual Learning Desiderata within a single Network (CLDNet). This is accomplished by the synergy between two key components: HBO implements non-overwritten parameter updates mediated by the Hilbert-Schmidt independence criterion in an orthogonal space, and EAE enhances decision boundary adaptation between old and new tasks with predefined basis vectors. As depicted in Figure 2, our model is formulated by two nested parts: the backbone network f_θ and the parameter-free classifier σ; For

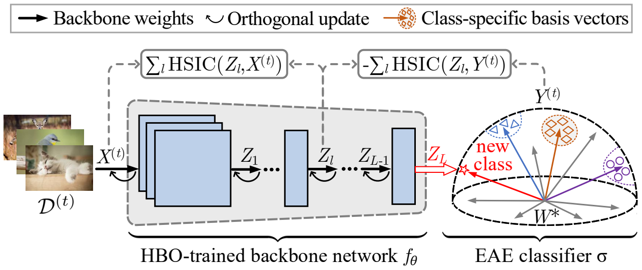
Backbone weights
Orthogonal update
Class-specific basis vectors
$\sum_l \mathrm{HSIC}(Z_l, X^{(t)})$
$-\sum_l \mathrm{HSIC}(Z_l, Y^{(t)})$
$Y^{(t)}$
$X^{(t)}$
Z_1
Z_l
$Z_{L\text{-}1}$
Z_L
new class
W^*
$\mathcal{D}^{(t)}$
HBO-trained backbone network f_θ
EAE classifier σ

Figure 2: Overview of CLDNet. HBO transforms learning task t into a constrained statistical dependency mini-max problem and EAE predicts by matching class-specific basis vectors. Systematically, the last-layer hidden representation Z_L is bound to any one of the available basis vectors in grey for recognizing a new class. We mark this process in red.

a single input x, we have its output $o(x) = \sigma(f_\theta(x))$. The following explains how CLDNet learns continually via the statistical dependency and distance metric.

Before elaboration, some definitions related to CL setting are introduced as follows. CL trains a model incrementally on a sequence of task datasets $\{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(T)}\}$, where $\mathcal{D}^{(t)} = \{(x_i^{(t)}, y_i^{(t)})_{i=1}^{|\mathcal{D}^{(t)}|}\}$ (or denoted by $X^{(t)}$ and $Y^{(t)}$) and $|\mathcal{D}^{(t)}|$ is the number of samples from task t. The output class $\mathcal{C}^{(t)}$ has no overlap, i.e., $\mathcal{C}^{(t)} \cap \mathcal{C}^{(t')} = \varnothing (t \neq t')$. Once a task is learned, its training data is often no longer accessible. At inference time, we mainly focus on one of the most challenging class-IL scenarios where task identities of test instances from classes seen so far are unknown.

HSIC-Bottleneck Orthogonalization (HBO)

We first present the learning objective of the HBO process. We then iteratively calculate the orthogonal projector facilitated by HSIC terms in batch learning form.

Suppose we have the backbone network f_θ with L hidden layers activated by the function $S_l(\cdot) : \mathbb{R}^{d_{l-1}} \to \mathbb{R}^{d_l}$, yielding hidden representations $Z_l \in \mathbb{R}^{m \times d_l}$ $(l = 1, 2, \ldots, L)$, where m is the batch size. Denote by θ_l the parameter to be updated, $Z_l = S_l(Z_{l-1}\theta_l)$. Following OWM (Zeng et al. 2019), we adopt the orthogonal projector[1] $P_l = \alpha(A_l^{\mathrm{T}} A_l + \alpha I)^{-1}$ that regularizes gradient update directions orthogonal to the input of previous tasks. The difference lies in that we recast it via the HSIC-bottleneck training. Then, HBO transforms CL into a constrained statistical dependency problem:

$$\begin{aligned} \min_{Z_l} &: \sum_{l=1}^{L} \mathrm{HSIC}(Z_l, X^{(t)}) - \beta \mathrm{HSIC}(Z_l, Y^{(t)}) \\ \text{s.t.} &: \theta_l^t = \theta_l^{(t-1)} - \lambda P_l^{\mathrm{HSIC}} \Delta\theta_l^{(t-1)}, P_l^{\mathrm{HSIC}} A_l = O \end{aligned} \tag{5}$$

where β is the balancing factor of HSIC terms, $\Delta\theta_l^{(t-1)}$ is the parameter update via HSIC-bottleneck training, λ is the learning rate, and P_l^{HSIC} is the orthogonal projector for modulating gradients. Unlike OWM, we construct it by capturing nonlinear dependencies of different layers. As formulated by Equation (5), for each layer of backbone network f_θ, the dependence between the input $X^{(t)}$ and hidden representations Z_l is minimized while that between the output $Y^{(t)}$ and hidden representations Z_l is maximized.

To gain insights into HBO, we want to emphasize that the optimal hidden representation Z_l is more amenable to CL, which inherently contributes to non-overwritten parameter updates in the perspective of reducing *feature bias*:

The conventional cross-entropy loss learns more on discriminative features (e.g., the dominant parts) that can recognize the classes of the task (Hu et al. 2021). During sequential training, some of those not previously learned features (e.g., the non-dominant parts) may become dominant for recognizing the new task classes, resulting in feature bias in the backbone network, as revealed by the recent study (Guo, Liu, and Zhao 2022). By contrast, HBO encourages learning all possible features from a sequence of tasks, implying that some of the features that may not be sufficiently discriminative for the current task are also holistically considered. This is achieved by optimizing hidden representation Z_l layer by layer to seek a balance between independence from unnecessary details of the input $X^{(t)}$ and dependence on the output $Y^{(t)}$. In this sense, the information needed to predict the label is well acquired and permeated in $Z_l (l = 1, 2, \ldots, L)$ when Eq. (5) converges. This not only mitigates the feature bias but also constructs a more accurate projector, thereby facilitating non-overwritten parameter updates. With this pitfall addressed, we achieve significant accuracy gains over OWM in empirical evaluation.

Now let us compute the orthogonal projector using the mini-batches of data, denoted by $Z_0(k) = X^{(t)}(k)$, $Z_l(k) = [z_l^1, z_l^2, \ldots, z_l^k]$. We embed the recursive least square algorithm (Golub and Van Loan 2013) into HSIC-bottleneck training. The derivation progress is based on Woodbury matrix identity, from which we get the following iterative expression:

$$\begin{aligned} P_l^{\mathrm{HSIC}}(k+1) = & P_l^{\mathrm{HSIC}}(k) - \\ & \frac{P_l^{\mathrm{HSIC}}(k) z_l^{k+1} (z_l^{k+1})^{\mathrm{T}} P_l^{\mathrm{HSIC}}(k)}{\alpha + (z_l^{k+1})^{\mathrm{T}} P_l^{\mathrm{HSIC}}(k) z_l^{k+1}} \end{aligned} \tag{6}$$

We make two remarks about the above derivation. (1) Equation (6) circumvents the matrix-inverse operation in the original matrix form. Meanwhile, both inputs and learned hidden representations from previous tasks are no longer revisited; Instead only the most recently updated $P_l^{\mathrm{HSIC}}(k)$ is required. (2) As mentioned in related work, AOP improves OWM itself by converting the small constant α into an adaptive one associated with each training batch. However, our work differs significantly in terms of motivation and methodology, and optionally, incorporates it for hyperparameter selection, as is demonstrated in the experiments.

EquiAngular Embedding (EAE)

This part first describes the indispensability of the EAE process for decision boundary adaptation and then elaborates on how it works together with HBO.

[1] This form is equivalent to $P_l = I - A_l(A_l^{\mathrm{T}} A_l + \alpha I)^{-1} A_l$ used in OWM, where A_l consists of all learned hidden representations Z_{l-1} and α is a small constant.

In our CLDNet, the HBO-trained backbone network outputs its last-layer hidden representation, which contains the information necessary for decision, but not necessarily in the available form like the logit of each class. One simple way is to append a single output layer (or projection head) trained with softmax cross-entropy (Ma, Lewis, and Kleijn 2020). However, this fully-connected layer would be added in a fully parametric manner, requiring additional consideration of parameter overwriting issues; Importantly, the decision boundary/output space of old tasks would be squeezed by the new task. To tackle these, inspired by the recently proposed EBVs (Shen, Sun, and Wei 2023), we design a novel EquiAngular Embedding (EAE) which replaces the trainable classifier parameters with predefined basis vectors.

EAE starts with a data-independent predefined process. Recall that, in the preliminary part, we denote by $\mathcal{W} = \{w_c\}_{c=1}^{C}$ the set composed of C d-dimensional basis vectors on the surface of a unit hypersphere $S^d \in \mathbb{R}^d$, and $\gamma \in [0, 1)$ the common angle pairwise separating these basis vectors. The question arises of how to construct such a set $\mathcal{W}$ which satisfies Equation (3) when given fixed γ, d, and C. Specifically, we randomly initialize a matrix $W \in \mathbb{R}^{d \times C}$ with normalized rows such that the angle between any two vectors $\arccos(w_i, w_j)$ equals $w_i^{\mathrm{T}} \cdot w_j$, in which $w_i, w_j \in \mathbb{R}^d$ ($w_i \leftarrow \frac{w_i}{|w_i|}$, $i, j = 1, 2, \ldots, C$, $i \neq j$). Then, we further tweak unit basis vectors in W by the following optimization function (Shen, Sun, and Wei 2023):

$$W^* = \arg\min_{W} \sum_{i=1}^{N-1} \sum_{j>i}^{N} \max(|w_i^{\mathrm{T}} \cdot w_j| - \gamma, 0) \quad (7)$$

This formulation cuts out the gradient of those unit vector pairs that hold $-\gamma \leq w_i^{\mathrm{T}} \cdot w_j \leq \gamma$ and optimizes the remaining ones.

Now let us consider EAE in the context of CL. When it comes to decision-making, classes from different tasks would be sequentially bound to some $w_c^* \in \mathbb{R}^d$ in $W^* \in \mathbb{R}^{d \times C}$, allowing models to scale to a large number of possible outputs, without a linear cost in the number of parameters. For a single input $x_i^{(t)}$ of task t, we have the backbone network output $z_i^{(t)}$ of the Lth-layer (we omit the subscript L for brevity); According to Equation (4), the probability of $z_i^{(t)}$ recognized as the class $y_i^{(t)}$ can be rewritten as:

$$P(y = y_i^{(t)} | z_i^{(t)}) = \frac{\exp(z_i^{(t)} \tilde{w}_c^*)}{\sum_{c'=1}^{\mathcal{C}^{(t)}} \exp(z_i^{(t)} \tilde{w}_{c'}^*)} \quad (8)$$

where $\tilde{w}_c^*$ denotes the ℓ_2-normalized w_c^*. Intuitively, this is equivalent to optimizing the cosine similarity between each $z_i^{(t)}$ and $\tilde{w}_c^*$. Therefore, based on Equations (5) and (8), the overall training objective of our CLDNet can be converted into minimizing the negative log-likelihood over task t:

$$\min_{\theta} : -\frac{1}{|\mathcal{D}^{(t)}|} \sum_{(x_i^{(t)}, y_i^{(t)})} \log P(y = y_i^{(t)} | f_\theta(x_i^{(t)})) \quad (9)$$

where $|\mathcal{D}^{(t)}|$ is the number of training samples from task t. Therefore, as an alternative to logits, the prediction is made by keeping the input $x_i^{(t)}$ with representation $z_i^{(t)}$ as close as possible to its class-specific basis vectors. Owe to our unique training objective, we only use a simple loss without storing any old task exemplars or backbones, suggesting that our CLDNet differs from most existing CL methods significantly. Please refer to Algorithm 1 for more.

Algorithm 1: CLDNet Training and Test algorithm

```
Input: A sequence of task datasets $\{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \ldots, \mathcal{D}^{(T)}\}$,
backbone network $f_\theta$ with $L$ hidden layers, followed by the
parameter-free classifier $\sigma$, learning rate $\lambda$, batch size $m$, etc.
Output: $\sigma(f_\theta(\cdot))$
# predefine basis-vector matrix prior to training
Initialize $W \in \mathbb{R}^{d \times C}$ with normalized rows;
Obtain $W^*$ by tweaking $W$ with Equation (7);
# during sequential training in batch learning
for $t = 1, 2, \ldots, T$ do
    for $j = 1, 2, \ldots, |\mathcal{D}^{(t)}|/m$ do
        Calculate hidden representations $\{Z_l\}_{l=1}^{L}$;
        Assign last-layer hidden representation $Z_L$ class-specific basis vectors from $W^*$;
        Solve the constrained statistical dependency problem with Equation (5);
        Update the backbone network parameter $\theta$ with Equation (9);
    end for
end for
# at test time
Draw test instances from any of tasks 1 to $T$;
return predicted labels by retrieving their closest class-specific basis vectors.
```

Reaching Continual Learning Desiderata

Here we discuss how CLDNet reaches multiple CL desiderata. The above formulations in our method clearly require no *rehearsal buffer* and substantial *network expansion*, which are non-trivial. For example, although a limited rehearsal buffer is allowed in the CL community, some prior works opt for sufficiently large buffer sizes that even suffice to train a supervised counterpart, as revealed by GDumb (Prabhu, Torr, and Dokania 2020). The same goes for arbitrarily expanding task-specific backbone networks, which would result in misleading high-accuracy performance. We think understanding this question is very important for future research, e.g., one should count these *non-desiderata* into the memory budget for a fair comparison (Zhou et al. 2023).

Focusing on such a strict yet realistic setting, it is natural to think of whether our CLDNet achieves the stability and plasticity trade-off. We would still like to emphasize that this balance is realized by the synergy of HBO and EAE. On the one hand, HBO addresses the layer-wise parameter overwriting in the backbone network, followed by the parameter-free EAE classifier. *This ensures the persistent knowledge retention of past tasks.* On the other hand, the rank of orthogonal projectors theoretically matters in the backbone network capacity available for incoming tasks and we construct such a nonzero matrix $P_l^{\mathrm{HSIC}}(k)$, allowing for some degree of freedom to learn new tasks; Meanwhile, since predefined basis vectors are exactly equivalent, the output space is no longer constrained to the number of classes. *This maintains the required network capacity to accommodate new tasks.*

Empirical Evaluation

We perform extensive experiments to evaluate the proposed CLDNet in the challenging class-IL setting. First, we introduce the experimental setup. We then provide the experimental results and discussion, following which we conduct ablation studies on the core components in our algorithm.

Experiment Setting

Dataset and Split. We experiment on multiple evaluation benchmarks for class-IL. **Small Scale:** MNIST (LeCun et al. 1998) contains 60,000 handwritten digit images in the training set and 10,000 samples in the test set, which is split into 5 disjoint tasks with 2 classes per task; FashionMNIST (Xiao, Rasul, and Vollgraf 2017) is an MNIST-like fashion product benchmark where the ten objects are split into five two-class classification tasks; CIFAR-10 (Krizhevsky, Hinton et al. 2009) has 10 classes with 50,000 samples for training and 10,000 for testing, which is divided into 5 tasks with 2 classes per task. **Medium Scale:** CIFAR-100 (Krizhevsky, Hinton et al. 2009) comprises 60,000 images belonging to 100 distinct classes, which are further divided into 10 tasks with each task containing 10 disjoint classes. **Large Scale:** ImageNet-R (Hendrycks et al. 2021) has 200 classes with 24,000 samples for training and 6,000 for testing. It is split into 10 tasks with 20 classes in each task. ImageNet-R incorporates newly curated data encompassing diverse styles, such as cartoons, graffiti, and origami, alongside challenging examples from ImageNet that conventional models (e.g., ResNet) fail to recognize. The substantial intra-class variability renders it more akin to intricate real-world problems.

Training Details. Architectures: In our experiments, all methods use similar-sized neural network architectures. For MNIST and FashionMNIST, following the setting in (Wołczyk et al. 2022), we use a standard MLP with 2 hidden layers of size 400; For CIFAR-10, following the setting in (Zeng et al. 2019; Guo et al. 2022), we use a CNN with 3 convolutional layers; For CIFAR-100, following the similar setting in (Bonicelli et al. 2022; Wang et al. 2023), we use a wide-adopted ResNet18; For ImageNet-R, following the setting in (Wang et al. 2022b), we use the ViT-B/16 pre-trained on ImageNet and allow all methods to start from the same pre-training for a fair comparison. **Hyper-parameters:** We either reproduce results using suggested hyper-parameters in their source code repositories or directly take existing results reported in state-of-the-art (SOTA) baselines. In our CLDNet, for HBO we set the coefficient $\beta = 500$ and adopt the Gaussian kernel as suggested by (Ma, Lewis, and Kleijn 2020), as well as the adaptive α with an initial value 0.01 for the orthogonal projector, like (Guo et al. 2022); For EAE we set $\gamma = 0.04$, $d = 1000$, and $C = 1000$ following the recommendations by EBVs (Shen, Sun, and Wei 2023). **Computing Infrastructure:** All experiments are run in PyTorch using NVIDIA RTX 3080-Ti GPUs with 12GB memory.

Method	MNIST	FMNIST	CIFAR-10
EWC	36.52 ± 2.54	35.16 ± 5.33	18.92 ± 4.88
MAS	38.74 ± 2.67	33.78 ± 6.42	17.79 ± 6.04
OEWC	40.52 ± 6.84	38.17 ± 4.02	16.98 ± 5.21
SI	45.28 ± 0.57	40.23 ± 3.34	17.38 ± 4.13
ICNet	40.73 ± 3.26	35.11 ± 0.02	19.07 ± 0.15
DMC	90.76 ± 0.25	72.54 ± 1.25	51.28 ± 0.95
OWM	91.60 ± 0.13	80.32 ± 0.73	52.83 ± 0.87
AOP	94.43 ± 0.21	82.97 ± 0.95	53.56 ± 0.29
CRNet	94.45 ± 0.36	90.98 ± 0.83	50.01 ± 0.58
Ours	**96.61** ± **0.15**	**95.37** ± **0.68**	**56.12** ± **0.33**

Table 1: Average accuracy (%) across all five tasks of the Split MNIST, FashionMNIST (FMNIST), and CIFAR-10, evaluated after learning the whole sequence. All methods are run 5 times, with the mean and standard deviation reported.

Results and Comparison

This paper considers a strict yet realistic setting for defying forgetting, covering three aspects of CL desiderata: ideally (i) accessing no training data of previous tasks, (ii) maintaining the model size relatively unchanged during sequential training, and (iii) striking a balance between stability and plasticity. To demonstrate the superiority of our work, we extensively compare it with the representative and SOTA competitors. Since different methods have very different requirements in data, networks, and computation, it is intractable to compare all in the completely same experimental conditions. Therefore, we compare our CLDNet with regularization-, rehearsal-, and architecture-based approaches respectively.

Comparison with Regularization-based Approaches. Table 1 compares our CLDNet with regularization-based approaches, which typically penalize parameter variations over an over-parameterized network by regularizers or orthogonal projectors. Without rehearsal buffers or network expansion, class-IL is particularly difficult for these methods. The competitors include EWC (Kirkpatrick et al. 2017), SI (Zenke, Poole, and Ganguli 2017), MAS (Aljundi et al. 2018), OEWC (Schwarz et al. 2018), OWM (Zeng et al. 2019), DMC (Zhang et al. 2020), ICNet (Wołczyk et al. 2022), and more recent SOTA methods, AOP (Guo et al. 2022), and CRNet (Li and Zeng 2023b). During class-IL, methods like EWC, MAS, OEWC, and SI struggle with the stability and plasticity dilemma as it is difficult to correctly assign credit to weights with the number of tasks increasing. Similar findings can be observed in (Wołczyk et al. 2022). Recent AOP and CRNet are strong baselines, but our method consistently outperforms them on all three task sequences by clear margins. Compared with the second-best results, CLDNet gets an improvement of 2.16% 4.39%, and 2.56% on the Split MNIST, FMNIST, and CIFAR-10, respectively.

Comparison with Rehearsal-based Approaches. As reported in Table 2, we evaluate our CLDNet with rehearsal-based approaches on CIFAR-100, which assume access to partial old data. The competitors include ER (Chaudhry et al. 2019), DER++ (Buzzega et al. 2020), X-DER-RPC (Boschini et al. 2022), ER-ACE (Caccia et al. 2021), more re-

	Buffer size for CIFAR-100		
Method	200	500	2000
ER	18.09±1.33	28.25±0.69	43.18±2.00
X-DER-RPC	51.40±2.17	57.45	62.46
ER-ACE	41.85±0.83	48.19	57.34
DER++	26.25±0.96	43.65	58.05
OCM	52.08±1.13	56.93±0.86	61.79±0.42
LiDER	51.23±1.93	57.76±0.75	62.78±0.51
DualHSIC	52.67±1.81	57.88±1.04	62.70±0.57
Ours	**65.42**±**0.36** (Buffer size = 0)		

Table 2: Average accuracy (%) across all ten tasks of the Split CIFAR-100. All results except ours, OCM, and LiDER are from (Wang et al. 2023). LiDER and DualHSIC are built upon X-DER-RPC to report their best performance.

	Buffer size for ImageNet-R		
Method	0	1000	5000
ER	-	55.13±1.29	65.18±0.40
BiC	-	52.14±1.08	64.63±1.27
GDumb	-	38.32±0.55	65.90±0.28
DER++	-	55.47±1.31	66.73±0.81
Co^{2}L	-	53.45±1.55	65.90±0.14
L2P	61.57±0.66	-	-
DualPrompt	68.13±0.49	-	-
Ours(1)	65.37±0.39	-	-
Ours(2)	**71.43**±**0.22**	-	-
Upper bound		79.13±0.18	

Table 3: Average accuracy (%) across all ten tasks of the Split ImageNet-R. When buffer size = 0, "-" denotes most rehearsal-based methods are not applicable anymore; When buffer size = 1000 or 5000, "-" denotes the omitted results.

Method	Buffer size	Capacity	Avg. Acc
PNN	0	1.71×	54.90±0.92
DEN		1.81×	57.38±0.56
RCL		1.80×	55.26±0.13
APD		1.53×	61.18±0.20
PCL		1.46×	62.58±0.32
Ours		**1.02×**	**64.99**±**0.24**

Table 4: Performance comparison on the Split CIFAR-100. The metric Capacity (lower is better) measures what extent a model scales after learning the whole sequence using the convolutional architecture in (Yoon et al. 2020).

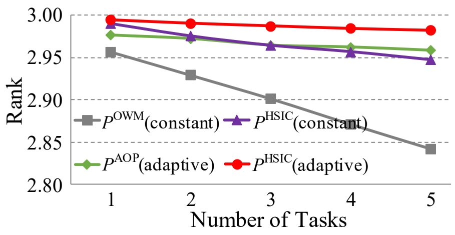

Figure 3: Changes of the rank of orthogonal projectors.

cent SOTA methods, OCM (Guo, Liu, and Zhao 2022), LiDER (Bonicelli et al. 2022), and DualHSIC (Wang et al. 2023). Both LiDER and DualHSIC serve to improve the performance of rehearsal-based counterparts. We observe that these baselines deteriorate when buffer size decreases, which may paralyze when buffer size is zero. By contrast, the performance gains of CLDNet are 12.75%, 7.54%, and 2.64% from buffer size 200 to 2000, respectively, thanks to the synergy between HBO and EAE in our CLDNet.

Table 3 further compares with rehearsal-based approaches on challenging ImageNet-R, including BiC (Wu et al. 2019), GDumb (Prabhu, Torr, and Dokania 2020), Co^{2}L (Cha, Lee, and Shin 2021). Since we following the pre-training used in L2P (Wang et al. 2022c) and DualPrompt (Wang et al. 2022b), these two prompt-based methods are also considered. This corresponds to two versions of the proposed method: Ours(1) refers to only one pre-trained ViT being used while Ours(2) involves two ViT like L2P and DualPrompt. We observe that the performance of rehearsal-based methods exhibits an obvious decline as the buffer size decreases—the significant intra-class diversity of ImageNet-R poses a great challenge for rehearsal-based methods to work effectively with the buffer size of 1000. This suggests again the necessity of CLDNet as a rehearsal-free method. Compared with prompt-based methods, Ours(1) beats L2P by 3.80% and falls short of DualPrompt by 2.76%; Ours(2) surpasses DualPrompt by 3.30%. This additionally indicates that our method can accommodate the real-world scenario where pre-training is usually involved as a base session.

Comparison with Architecture-based Approaches. To make the comparison more complete, we also compare our CLDNet with architecture-based (i.e., network expansion) approaches, which assign new branches for each task. The competitors include PNN (Rusu et al. 2016), DEN (Yoon et al. 2018), RCL (Xu and Zhu 2018), APD (Yoon et al. 2020), PCL (Hu et al. 2021). Table 4 reports the results on CIFAR-100. CLDNet achieves the best Avg. Acc 64.99% with minimal Capacity 1.02×. For a fair comparison, we do not compare with hybrid methods that deviate significantly from the CL desiderata that our method is designed for: this excludes methods that heavily rely on both network expansion and rehearsal buffers. On the one hand, hybrid methods like RPS-Net and EDR do push the performance towards the upper bound achieved by offline training. On the other hand, e.g., for CIFAR-100, they explicitly use a buffer size of 2000 and require about 5× (RPS-Net) or 10× (DER w/o P) more parameters than the base network.

Ablation Study. Figure 3 plots the rank of different orthogonal projectors under a *constant* or *adaptive* α, where the same MLP with 3 hidden layers is trained for the split MNIST. Interestingly, AOP improves OWM by shrinking the change in rank as tasks increase; The HSIC used in our method contributes to a higher rank for maintaining model plasticity. In addition, we provide an empirical analysis of the effectiveness of HBO for learning (see Figure 4) and EAE for decision (see Table 5). In summary, both compo-

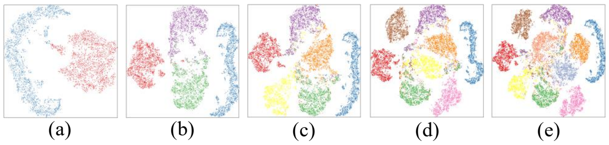

Figure 4: t-SNE visualization based on split FashionMNIST. Each color represents a class. We visualize two classes in each task as a session. (a)-(e) represents the corresponding representation visualization of classes trained so far.

Component	MNIST	FMNIST	CIFAR-10
HBO + $\sigma_{\theta'}$	19.96	19.74	15.56
HBO + $\sigma_{\theta''}$	95.12	92.77	53.89
HBO + σ (i.e., EAE)	**96.62**	**95.35**	**56.10**

Table 5: Effectiveness of the core designs in our CLDNet. $\sigma_{\theta'}$ ($\sigma_{\theta''}$) represents a single fully-connected output layer, parameterized by θ' (θ''), and is trained by cross-entropy loss without (with) an orthogonal projector for prediction.

nents contribute to the final performance improvement, e.g., parameter-free EAE classifier σ facilitates decision boundary adaptation.

Conclusion

This present study considers a stringent yet practical setting to reach multiple CL desiderata. Taking the statistical dependency and distance metric as training objectives, we propose CLDNet with two pivotal components, i.e., HBO for non-overwritten parameter updates and EAE for decision boundary adaptation. We perform extensive experiments to show that CLDNet achieves a better stability-plasticity trade-off in a rehearsal-free and minimal-expansion way. Moreover, we hope that our study inspires further research in reaching CL desiderata, e.g., this makes sense for real-world applications under privacy-sensitive and resource-limited CL scenarios.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2021ZD0201300, the National Natural Science Foundation of China under Grants U1913602 and 61936004, and the 111 Project on Computational Intelligence and Intelligent Control under Grant B18024.

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, 139–154.

Bonicelli, L.; Boschini, M.; Porrello, A.; Concetto, S.; Calderara, S.; et al. 2022. On the Effectiveness of Lipschitz-Driven Rehearsal in Continual Learning. In *Advances in Neural Information Processing Systems*, volume 35, 31886–31901.

Boschini, M.; Bonicelli, L.; Buzzega, P.; Porrello, A.; and Calderara, S. 2022. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5497–5512.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark experience for general continual learning: a strong, simple baseline. In *Advances in Neural Information Processing Systems*, volume 33, 15920–15930.

Caccia, L.; Aljundi, R.; Asadi, N.; Tuytelaars, T.; Pineau, J.; and Belilovsky, E. 2021. New insights on reducing abrupt representation change in online continual learning. *arXiv preprint arXiv:2104.05025*.

Cha, H.; Lee, J.; and Shin, J. 2021. Co^{2}L: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9516–9525.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H.; and Ranzato, M. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.

Dai, W.; Li, D.; Zhou, P.; and Chai, T. 2019. Stochastic configuration networks with block increments for data modeling in process industries. *Information Sciences*, 484: 367–386.

Deng, D.; Chen, G.; Hao, J.; Wang, Q.; and Heng, P.-A. 2021. Flattening Sharpness for Dynamic Gradient Projection Memory Benefits Continual Learning. In *Advances in Neural Information Processing Systems*, volume 34, 18710–18721.

Farajtabar, M.; Azizan, N.; Mott, A.; and Li, A. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, 3762–3773. PMLR.

Golub, G. H.; and Van Loan, C. F. 2013. *Matrix Computations*. JHU Press.

Gretton, A.; Bousquet, O.; Smola, A.; and Schölkopf, B. 2005. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory*, 63–77. Springer.

Guo, Y.; Hu, W.; Zhao, D.; and Liu, B. 2022. Adaptive orthogonal projection for batch and online continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6783–6791.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online continual learning through mutual information maximization. In *International Conference on Machine Learning*, 8109–8126. PMLR.

Hayes, T. L.; Kafle, K.; Shrestha, R.; Acharya, M.; and Kanan, C. 2020. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, 466–483. Springer.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8349.

Hoffer, E.; Hubara, I.; and Soudry, D. 2018. Fix your classifier: the marginal value of training the last weight layer. In *International Conference on Learning Representations*.

Hu, W.; Qin, Q.; Wang, M.; Ma, J.; and Liu, B. 2021. Continual learning by using information of each class holistically. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7797–7805.

Hu, Z.; Li, Y.; Lyu, J.; Gao, D.; and Vasconcelos, N. 2023. Dense network expansion for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11858–11867.

Kang, H.; Mina, R. J. L.; Madjid, S. R. H.; Yoon, J.; Hasegawa-Johnson, M.; Hwang, S. J.; and Yoo, C. D. 2022. Forget-free continual learning with winning subnetworks. In *International Conference on Machine Learning*, 10734–10750. PMLR.

Kawaguchi, K.; Deng, Z.; Ji, X.; and Huang, J. 2023. How Does Information Bottleneck Help Deep Learning? In *International Conference on Machine Learning*, 16049–16096. PMLR.

Ke, Z.; Liu, B.; Ma, N.; Xu, H.; and Shu, L. 2021. Achieving Forgetting Prevention and Knowledge Transfer in Continual Learning. In *Advances in Neural Information Processing Systems*, volume 34, 22443–22456.

Kim, D.; and Han, B. 2023. On the Stability-Plasticity Dilemma of Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20196–20204.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 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. *Handbook of Systemic Autoimmune Diseases*.

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, D.; Wang, T.; Chen, J.; Kawaguchi, K.; Lian, C.; and Zeng, Z. 2023a. Multi-view class incremental learning. *arXiv preprint arXiv:2306.09675*.

Li, D.; Wang, T.; Xu, B.; Kawaguchi, K.; Zeng, Z.; and Suganthan, P. N. 2023b. IF2Net: Innately forgetting-free networks for continual learning. *arXiv preprint arXiv:2306.10480*.

Li, D.; and Zeng, Z. 2023a. Complementary learning subnetworks for parameter-efficient class-incremental learning. *arXiv preprint arXiv:2306.11967*.

Li, D.; and Zeng, Z. 2023b. CRNet: A fast continual learning framework with random theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–14.

Li, Y.; Pogodin, R.; Sutherland, D. J.; and Gretton, A. 2021. Self-supervised learning with kernel dependence maximization. *Advances in Neural Information Processing Systems*, 34: 15543–15556.

Liu, Y.; Su, Y.; Liu, A.-A.; Schiele, B.; and Sun, Q. 2020. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12245–12254.

Luo, Z.; Liu, Y.; Schiele, B.; and Sun, Q. 2023. Class-incremental exemplar compression for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11371–11380.

Ma, W.-D. K.; Lewis, J.; and Kleijn, W. B. 2020. The HSIC bottleneck: Deep learning without back-propagation. In *Proceedings of the AAAI conference on artificial intelligence*, 5085–5092.

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. Elsevier.

Mettes, P.; Van der Pol, E.; and Snoek, C. 2019. Hyperspherical prototype networks. In *Advances in neural information processing systems*, volume 32.

Pernici, F.; Bruni, M.; Baecchi, C.; Turchini, F.; and Del Bimbo, A. 2021. Class-incremental learning with pre-allocated fixed classifiers. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 6259–6266. IEEE.

Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; and Tucker, G. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, 5171–5180. PMLR.

Prabhu, A.; Torr, P. H.; and Dokania, P. K. 2020. GDumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, 524–540. Springer.

Qi, D.; Zhao, H.; and Li, S. 2023. Better generative replay for continual federated learning. *arXiv preprint arXiv:2302.13001*.

Rajasegaran, J.; Hayat, M.; Khan, S.; Khan, F. S.; and Shao, L. 2019. Random path selection for incremental learning. In *Advances in Neural Information Processing Systems*, volume 32, 12669–12679.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Saha, G.; Garg, I.; and Roy, K. 2021. Gradient projection memory for continual learning. In *International Conference on Learning Representations*.

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 *International Conference on Machine Learning*, 4528–4537.

Serrà, J.; Suris, D.; Miron, M.; and Karatzoglou, A. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, 4548–4557. PMLR.

Shen, Y.; Sun, X.; and Wei, X.-S. 2023. Equiangular Basis Vectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11755–11765.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC*

Conference on Computer and Communications Security, 1310–1321.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. CODA-Prompt: COntinual Decomposed Attention-based Prompting for Rehearsal-Free Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11909–11919.

Song, L.; Smola, A.; Gretton, A.; Bedo, J.; and Borgwardt, K. 2012. Feature Selection via Dependence Maximization. *Journal of Machine Learning Research*, 13(5).

Tong, S.; Dai, X.; Wu, Z.; Li, M.; Yi, B.; and Ma, Y. 2023. Incremental learning of structured memory via closed-loop transcription. In *International Conference on Learning Representations*.

Wang, F.-Y.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2022a. FOSTER: Feature boosting and compression for class-incremental learning. In *European Conference on Computer Vision*, 398–414. Springer.

Wang, S.; Li, X.; Sun, J.; and Xu, Z. 2021a. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 184–193.

Wang, T.; Dai, X.; and Liu, Y. 2021. Learning with Hilbert–Schmidt independence criterion: A review and new perspectives. *Knowledge-based systems*, 234: 107567.

Wang, Z.; Jian, T.; Masoomi, A.; Ioannidis, S.; and Dy, J. 2021b. Revisiting Hilbert-Schmidt information bottleneck for adversarial robustness. In *Advances in Neural Information Processing Systems*, volume 34, 586–597.

Wang, Z.; Zhan, Z.; Gong, Y.; Shao, Y.; Ioannidis, S.; Wang, Y.; and Dy, J. 2023. DualHSIC: HSIC-Bottleneck and Alignment for Continual Learning. *arXiv preprint arXiv:2305.00380*.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022b. Dual-Prompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, 631–648. Springer.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022c. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 139–149.

Wołczyk, M.; Piczak, K.; Wójcik, B.; Pustelnik, L.; Morawiecki, P.; Tabor, J.; Trzcinski, T.; and Spurek, P. 2022. Continual learning with guarantees via weight interval constraints. In *International Conference on Machine Learning*, 23897–23911. PMLR.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 374–382.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xu, J.; and Zhu, Z. 2018. Reinforced continual learning. In *Advances in Neural Information Processing Systems*, volume 31.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3014–3023.

Yang, B.; Lin, M.; Zhang, Y.; Liu, B.; Liang, X.; Ji, R.; and Ye, Q. 2023a. Dynamic Support Network for Few-shot Class Incremental Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 2945–2951.

Yang, Y.; Chen, S.; Li, X.; Xie, L.; Lin, Z.; and Tao, D. 2022. Inducing Neural Collapse in Imbalanced Learning: Do We Really Need a Learnable Classifier at the End of Deep Neural Network? In *Advances in Neural Information Processing Systems*, volume 35, 37991–38002.

Yang, Y.; Yuan, H.; Li, X.; Lin, Z.; Torr, P.; and Tao, D. 2023b. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. In *International Conference on Machine Learning*. PMLR.

Yoon, J.; Kim, S.; Yang, E.; and Hwang, S. J. 2020. Scalable and order-robust continual learning with additive parameter decomposition. In *International Conference on Learning Representations*.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong learning with dynamically expandable networks. In *International Conference on Machine Learning*. PMLR.

Zeng, G.; Chen, Y.; Cui, B.; and Yu, S. 2019. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8): 364–372.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 3987–3995.

Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; and Kuo, C.-C. J. 2020. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1131–1140.

Zhou, D.-W.; Wang, F.-Y.; Ye, H.-J.; Ma, L.; Pu, S.; and Zhan, D.-C. 2022. Forward compatible few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9046–9056.

Zhou, D.-W.; Wang, Q.-W.; Ye, H.-J.; and Zhan, D.-C. 2023. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *International Conference on Learning Representations*.

Zhuang, H.; Weng, Z.; Wei, H.; Xie, R.; Toh, K.-A.; and Lin, Z. 2022. ACIL: Analytic class-incremental learning with absolute memorization and privacy protection. In *Advances in Neural Information Processing Systems*, volume 35, 11602–11614.