Non-exemplar Online Class-Incremental Continual Learning via Dual-Prototype Self-Augment and Refinement

Fushuo Huo[1], Wenchao Xu[1], Jingcai Guo[1, 2], Haozhao Wang[3]*, Yunfeng Fan[1]

[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
[2]The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China
[3]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China
{fushuo.huo, yunfeng.fan}@connect.polyu.hk, {jc-jingcai.guo, wenchao.xu}@polyu.edu.hk, hz_wang@hust.edu.cn

Abstract

This paper investigates a new, practical, but challenging problem named Non-exemplar Online Class-incremental continual Learning (NO-CL), which aims to preserve the discernibility of base classes *without* buffering data examples and efficiently learn novel classes continuously in a *single-pass* (i.e., online) data stream. The challenges of this task are mainly two-fold: (1) Both base and novel classes suffer from severe catastrophic forgetting as no previous samples are available for replay. (2) As the online data can only be observed once, there is no way to fully re-train the whole model, e.g., re-calibrate the decision boundaries via prototype alignment or feature distillation. In this paper, we propose a novel Dual-prototype Self-augment and Refinement method (DSR) for NO-CL problem, which consists of two strategies: 1) Dual class prototypes: vanilla and high-dimensional prototypes are exploited to utilize the pre-trained information and obtain robust quasi-orthogonal representations rather than example buffers for both privacy preservation and memory reduction. 2) Self-augment and refinement: Instead of updating the whole network, we optimize high-dimensional prototypes alternatively with the extra projection module based on self-augment vanilla prototypes, through a bi-level optimization problem. Extensive experiments demonstrate the effectiveness and superiority of the proposed DSR in NO-CL.

Introduction

With the ubiquitously prevalent personal smart devices, a massive amount of data are being continually generated, which requires adaptive machine learning models to learn new tasks without forgetting the old knowledge (De Lange et al. 2022; Zhu et al. 2021a, 2022). In privacy-sensitive online scenarios, a practical Online Class-incremental continual Learning (OCL) system is expected to learn novel classes incrementally while keeping the prior knowledge without restoring any streaming data due to privacy and computation concerns. However, (1) existing OCL solutions heavily rely on the example buffer for replay to re-calibrate the decision boundaries, between data batches and tasks (Fini et al. 2020). (2) OCL mainly concerns the setting of totally online continual learning, which is a relatively uncommon scenario for dynamic environments, and the state-of-the-art method (Gu et al. 2022) achieves impractical low accuracy (<**20**% for CIFAR100 with 1000 buffer size), which significantly hinders the deployment of OCL methods.

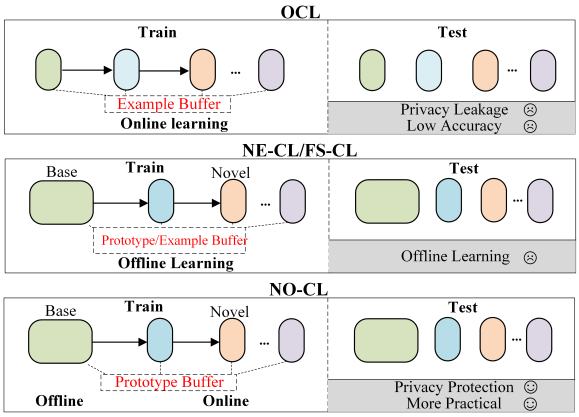

Figure 1: The overall concept of proposed NO-CL (*bottom*), compared with OCL (*top*) and NE-CL/FS-CL (*middle*).

Considering dynamic continual learning scenarios, we investigate a new, practical, yet challenging protocol named Non-exemplar Online Class-incremental continual Learning (NO-CL), as demonstrated in Figure 1(*bottom*). Concretely, in practice, an intelligent system conducts online class-incremental learning without restoring stream examples, while utilizing and preserving the knowledge from previous training. Such under-explored practical settings are in line with Non-Exemplar Class-incremental continual Learning (NE-CL) (Yu et al. 2020; Zhu et al. 2021a, 2022) and Few-Shot Class-incremental Learning (FS-CL) (Zhu et al. 2021d; Kalla and Biswas 2022; Peng et al. 2022), as shown in Figure 1(*middle*), where base classes are well trained and retained, but novel classes need to be explored. However, NE-CL conducts class-incremental learning without example buffers in an offline fashion, which enables the network to align the prior information (i.e., prototypes and/or features) gradually like semantic drift compensation (Yu et al. 2020), dual augmentation (Zhu et al. 2021a), and prototype selection (Zhu et al. 2022). FS-CL aims to continually learn with few shot samples also in an offline way, like gradually refining the prototypes (Zhu et al. 2021d), finetuning

*Corresponding author

the classifier heads (Kalla and Biswas 2022), or focusing on training robust embedding network (Peng et al. 2022). Therefore, NE-CL and FS-CL methods can hardly solve the NO-CL problem. Figure 2 demonstrates the brief quantitative comparisons of OCL, FS-CL, NE-CL, and the proposed method in *the same NO-CL training protocols*.

To solve the proposed NO-CL problem, we devise a simple but effective method called Dual-prototype Self-augment and Refinement (DSR). As the single-pass data can not be revisited, unlike previous example-base continual learning methods, directly finetuning the feature extractor will cause severe catastrophic forgetting. Therefore, we freeze the pre-trained feature extractor and translate the NO-CL problem to bi-level optimizing (Sinha, Malo, and Deb 2018) privacy-preserved prototypes with the extra projection module. Specifically, vanilla and high-dimensional prototypes (V-P and H-P) of base classes are restored to preserve learned information. For incremental sessions, direct calculation and reasoning V-P tends to accumulate errors and fails to fully explore online data. The project module is introduced to translate V-P to the high-dimensional embedding (Gayler 2004; Kanerva 2009; Karunaratne et al. 2020), which has been proven robust to noise. In detail, a random vector from high-dimensional embedding is quasi-orthogonal to other vectors with high probability (the "curse" of dimensionality (Kanerva 2009)), and fine-tuning the prototype in the high-dimensional embedding provides a sufficiently large capacity to accommodate novel classes over time, with minimal interference with learned knowledge. Therefore, we formulate a bi-level optimization strategy to optimize the extra high-dimensional prototypes alternatively with the projection module, to refine the decision boundaries and recalibrate projection module based on optimized prototypes. In summary, our contributions are as follows:

1) We propose a novel yet practical problem called Non-exemplar Online Class-incremental continual Learning (NO-CL), where an intelligent system with pre-trained base classes information can efficiently learn novel classes continually from the single-pass (i.e., online) data stream, without example buffers. Meanwhile, previous knowledge should also be preserved.
2) We develop a novel Dual-prototype Self-augment and Refinement (DSR) method, which transfers training the whole network to bi-level optimizing prototypes and the extra projection module.
3) Extensive quantitative results demonstrate DSR performs significantly better than existing OCL, NE-CL, and FS-CL methods under the *same training protocols* of proposed NO-CL, both in accuracy and efficiency.

Related Work

Class-incremental Learning (CL) Existing methods can be generally divided into three categories: regularization-based (Aljundi et al. 2018; Lee et al. 2017; Lopez-Paz and Ranzato 2017; Chaudhry et al. 2018), structure-based (Lee et al. 2020; Mallya and Lazebnik 2018; Kang et al. 2022), and replay-based methods (Douillard et al. 2020; Hu et al. 2021). The detailed review can refer to (De Lange et al. 2022; Mai et al. 2022). Recently, some practical yet challenging settings of class-incremental learning, including Online Class-Incremental continual Learning (OCL), Non-Exemplar Class-incremental continual Learning (NE-CL), and Few-Shot Class-incremental Learning (FS-CL) are also proposed. Here we give a brief introduction.

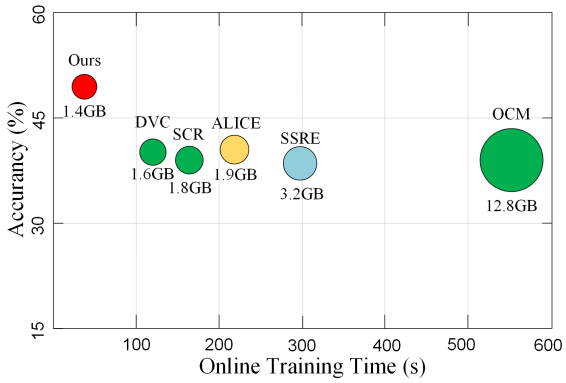

Figure 2: Class-wise accuracy, online training time, and memory overhead comparisons on the CIFAR100 with the same protocols of NO-CL. Stat-of-the-art OCL (SCR(Mai et al. 2021), DVC(Gu et al. 2022), and OCM (Guo, Liu, and Zhao 2022), all with 1000 example buffer), FS-CL (ALICE (Peng et al. 2022)), and NE-CL (SSRE (Zhu et al. 2022)) methods are illustrated. Online training batchsize is 10.

Online Class-incremental continual Learning (OCL) OCL aims to learn new classes continually from online data streams (each sample is seen only once). As the model needs to learn novel classes from the data stream while not forgetting previous classes, OCL methods (Mai et al. 2022; Caccia et al. 2022; Aljundi et al. 2019b; Mai et al. 2021; Guo, Liu, and Zhao 2022; Gu et al. 2022; Zhang et al. 2022; Lin et al. 2023) follow replay-based protocols, where example buffers are stored and retrieved between data batches and tasks. Concretely, (Guo, Liu, and Zhao 2022; Gu et al. 2022) dig the critical information by maximizing their mutual information. (Zhang et al. 2022) design augmentation strategies to address the underfitting-overfitting dilemma of online rehearsal. However, as pointed out by (Fini et al. 2020), example buffers in OCL violate privacy and computation restrictions, especially in online learning scenarios. However, the example-free task-incremental online continual learning method (Fini et al. 2020) needs the prior task information. Also, even equipped with the example buffers, the state-of-the-art method (Gu et al. 2022) only achieves relatively low accuracy ($<20\%$ for CIFAR100 (Krizhevsky and Hinton 2009) with 1000 buffer size). Considering these problems, we proposed a novel yet practical setting called NO-CL, which aims to better preserve privacy and utilize pre-trained offline knowledge in practical online applications.

Non-Exemplar Class-incremental Learning (NE-CL) Due to computation burden or privacy security, some works (Yu et al. 2020; Zhu et al. 2022; Yin et al. 2020; Zhu et al. 2021c) develop non-exemplar class-incremental learn-

ing methods where no past data can be stored. (Yu et al. 2020) compensates unknown prototype drifts of old classes via the drifts of current data. (Zhu et al. 2021c) employs self-supervised learning to obtain more transferable features. Also, prototypes are also augmented to preserve the decision boundaries of previous classes. Recently, (Zhu et al. 2022) considers to adjust the joint representation learning and distillation process. However, NE-CL needs to *offline* train novel classes to adjust prototypes and *gradually* distill features. Therefore, NE-CL methods fail to solve the proposed NO-CL problem as analysis and experiments below.

Few-Shot Class-incremental Learning (FS-CL) Compared to NE-CL, FS-CL assumes that novel classes come with few reference images. State-of-the-art FS-CL methods are mainly divided into two types. Some methods (Cheraghian et al. 2021; Dong et al. 2021; Tao et al. 2020; Kang et al. 2023) update the backbone to accommodate new classes while preserving base class via gradual knowledge distillation (Zhao et al. 2023), meanwhile, heavily relying on complex example buffers to retain the learned information of the previous network. Some methods (Peng et al. 2022; Hersche et al. 2022; Kalla and Biswas 2022) freeze the backbone and re-adopt features from the base classes to recognize new classes. Therefore, contrastive learning (Song et al. 2023), meta-learning (Hersche et al. 2022), self-supervised learning (Kalla and Biswas 2022), and data augmentation (Peng et al. 2022; Zhou et al. 2022) strategies have been employed to obtain the backbone with high transferable representations. However, such FS-CL methods finetune the network *offline* and/or do not *fully* explore novel classes, leading to relatively poor performance on NO-CL.

Bi-level Optimization (BO) Problem Bi-level optimization aims to solve a nested optimization problem, where the outer-level optimization is subjected to the result of the inner-level optimization (Sinha, Malo, and Deb 2018; Liu et al. 2022). It has been widely employed in machine learning areas like meta-learning and hyperparameter selection. For CL problems, (Liu et al. 2020) uses BO to alternatively optimize the CL and the exemplar models. (Liu, Schiele, and Sun 2021) applies BO to learn the aggregation weights of the plastic and elastic branches of CL models. (Luo et al. 2023) solves the bi-level optimization of the CL model and example compression model. For the proposed NO-CL problem, we formulate dual prototypes and bi-level optimize prototypes and the projection module. The optimization process quickly converges, as shown in Figure 2 and Appendix E[1].

Problem Formulation

As shown in Figure 1, the NO-CL problem comprises base classes from pre-training data and novel classes from online training data. During online learning, only the raw data of the current classes is available, and the network aims to incrementally learn online new classes whilst retaining learned information before the current session. Concretely, assuming an m-step NO-CL problem, let $\{\mathcal{D}^0_{train}, \mathcal{D}^1_{train}, ..., \mathcal{D}^m_{train}\}$ and $\{\mathcal{D}^0_{test}, \mathcal{D}^1_{test}, ..., \mathcal{D}^m_{test}\}$ denote the training and testing data from sessions $\{0, 1, ..., m\}$, respectively. Each training and testing sessions i have the corresponding label sets denoted by $\mathcal{C}^i_{train}$ and $\mathcal{C}^i_{test}$. $\mathcal{C}^i_{train}$ are mutually exclusive across different training sets, i.e., $\forall i \neq j, \mathcal{C}^i_{train} \cap \mathcal{C}^j_{train} = \phi$. While during evaluation, the model will be tested on all seen classes so far, i.e., for session i, the corresponding label space is $\mathcal{C}^0_{test} \cup \mathcal{C}^1_{test} ... \cup \mathcal{C}^i_{test}$. Besides, the base session ($i = 0$) provides a large number of classes and also allows offline pre-training. For the incremental sessions ($i > 0$), the data comes in the online stream state without rehearsal. During incremental sessions, like NE-CL (Yu et al. 2020; Zhu et al. 2022), considering privacy and computation constraints, buffers with raw data are not permitted.

Dataset Partition. Similar to (Yu et al. 2020; Zhu et al. 2022, 2021d; Kalla and Biswas 2022; Peng et al. 2022), the benchmark datasets are divided into $(60\%+4\%\times10)$, where the base session contains 60% classes for pre-training, and the rest classes are online incrementally learned within 10 sessions. Also, the results of $(40\%+6\%\times10)$, $(80\%+2\%\times10)$, and $(60\%+2\%\times20)$ are also provided in ***Appendix B***.

Methodology

As for the proposed NO-CL problem, we aim to fully explore single-pass data stream novel classes while preserving previous information without example buffers. The stability-plasticity dilemma is intractable as the single-pass data stream results in the overfitting of novel classes while severely interfering with previously learned information. As NO-CL has no example buffers to rehearse between data batches and tasks to eliminate forgetting. we transfer training the whole network to alternatively update the extra projection module and dual prototypes in the bi-level optimization problem. Figure 3 shows the framework of the proposed DSR method. In the following section, we introduce the base session training protocol, Dual-prototype Self-augment and Refinement (DSR) for online continual learning, including vanilla-prototype self-augment, bi-level optimization procedure for dual prototypes and projection module.

Base Session Training

For the base session pre-training, we aim to obtain vanilla- and high-dimensional prototypes for sequentially online sessions. Therefore, we employ loss regularizations on the outputs of the feature extractor and projection module:

$$L^{base} = L^{base}_{vp} + L^{base}_{hp} \tag{1}$$

where $L^{base}_{vp} = Loss(Proj_{vp}(\theta_1(x)), y)$ and $L^{base}_{vp} = Loss(Proj_{hp}(\theta_2(\theta_1(x))), y)$. x, y, θ_1, and θ_2 denote input samples, labels, feature extractor, and projection module. $Proj_{vp/hp}$ are linear layers to align vanilla- and high-dimensional prototypes for loss calculations. For loss functions ($Loss$), we adopt two variations: cross-entropy (CE) loss and supervised contrastive (SC) loss (Khosla et al. 2020) (Details please refer to *Appendix A*). To train the robust embedding, recent NE-CL, FS-CL, and OCL methods focus on training diverse features that are transferable across sessions, like data augmentation (Peng et al. 2022; Zhu et al. 2021a,c), self-supervised learning (Zhu et al. 2021c; Kalla

[1]Appendix is in https://arxiv.org/abs/2303.10891

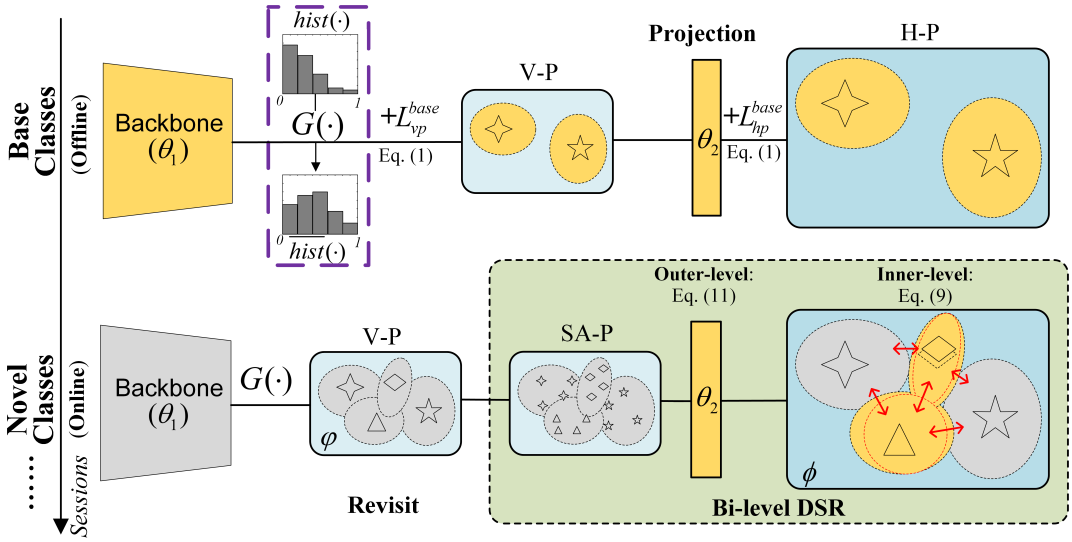

Figure 3: Overview of the proposed DSR method. The base and novel sessions train in offline and online ways, respectively. V-P, SA-P, and H-P mean vanilla, self-augment, and high-dimensional prototypes. Backbone, projection module, V-P, and H-P are represented by θ_1, θ_2, φ, and ϕ. $hist(\cdot)$ and $G(\cdot)$ denote histogram and feature transformation (Eq. (2), i.e., purple dotted line). Yellow and Gray mean the learnable and frozen components, and red dotted lines represent the refined decision boundaries.

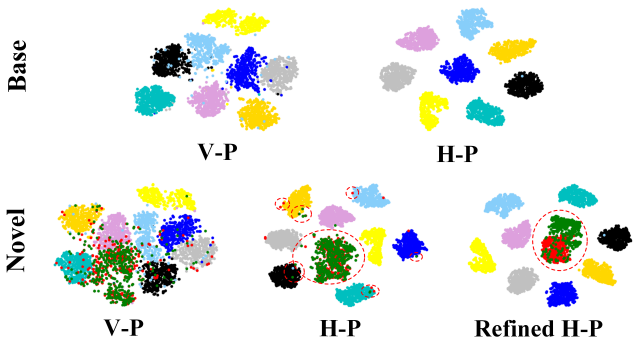

Figure 4: t-SNE (Laurens and Hinton 2008) visualization of the feature embeddings. For better visualization, we train eight classes on the base session and incremental learn two classes (marked in green and red dots) sequentially. Red circles in H-P mean the confusion in novel classes and among base and novel classes. Best viewed in color.

and Biswas 2022), mutual information regularization (Gu et al. 2022; Guo, Liu, and Zhao 2022), supervised contrastive regularization (Mai et al. 2021; Lin et al. 2023). Similarly, for the NO-CL problem, sophisticated pre-training strategies also improves the generalization and transfer ability of our method to accommodate online new classes (refer to Ablation studies with data augmentation (Peng et al. 2022; Zhu et al. 2021b) (w/ DA) in Table 3 and Appendix C). As shown in Table 1, even the vanilla cross-entropy variation of our method surpasses other dedicated training methods.

Dual-prototype Self-augment and Refinement

Overview. After the base session training, the backbone maps the data from the input domain $\mathcal{X}$ to a feature space: $\theta_1 : \mathcal{X} \to \mathbb{R}^{d_f}$. θ_1 are parameters of backbone. The prototypes in $\mathbb{R}^{d_f}$ are computed and restored to retain previous knowledge. Previous example-free prototype-based methods (Yu et al. 2020; Zhu et al. 2021a,c) offline refine prototypes and/or features together with samples from novel classes to achieve the plasticity and stability trade-off. However, as for online learning, the single-pass data stream fails to gradually update the prototypes and network parameters. Besides, directly classifying novel classes based on frozen backbone fails to fully explore data samples of novel classes. Therefore, we devise dual prototypes strategy and optimize prototypes alternatively with the projection module through bi-level optimization. Specially, we introduce the vanilla-prototype self-augment, high-dimensional prototypes and projection module bi-level optimization procedure.

Vanilla-prototype Self-augment. We restore the vanilla prototype of the base and novel class to rehearse for retaining and calibrating learned and online information. However, vanilla retrieving previous prototypes will confuse the decision boundary. As a solution, (Zhu et al. 2021c) tries to augment prototypes via Gaussian noise when learning new classes. However, the distribution tends to skew to *0* and loses the discriminative representation, due to the relu (Nair and Hinton 2010) activation function in the final layer of the backbone (i.e., ResNet (He et al. 2016)). Therefore, to make feature distribution more Gaussian-like, we transform features similar to Tukey's Ladder of Powers Transformation (Tukey 2010), which is a kind of power transformation that can reduce the skewness of distributions. The distribution is

rectified and normalized as follows:

$$G(x) = \begin{cases} \frac{x^{\lambda}}{\max(\|x^{\lambda}\|_2, \epsilon)} & if\ \lambda \neq 0 \\ \frac{\log(x)}{\max(\|\log(x)\|_2, \epsilon)} & if\ \lambda = 0 \end{cases}. \tag{2}$$

where $\epsilon = e^{-6}$ and λ is the hyper-parameter to control the distribution, i.e., decreasing λ makes the distribution less positively skewed and vice versa. The Gaussian-like rectification (denoted as $G(\cdot)$) are applied after the backbone (θ_1) in both base and online sessions as $f = G(\theta_1(x))$. For class i in the session m, the vanilla prototype (vp) and its relative variance (v) are computed as:

$$vp_i^m = \frac{1}{|k_i|}\sum_{j=1}^{k_i} f_j^m,\ v_i^m = \frac{1}{|k_i|}\sum_{j=1}^{k_i}\left(f_j^m - vp_i^m\right)^2 \tag{3}$$

where k_i represents the number of samples (x) of class i in $\mathcal{D}_{train}^m$. Previous knowledge is retained by self-augmenting prototypes from the class-specific Gaussian distribution: $\mathbb{D}(\varphi_i^m) = \{(\varphi_i^m, i) | \sim \mathcal{N}(vp_i^m, v_i^m)\}$. Without example buffers, we freeze the backbone θ_1 and translate online class-incremental learning into the bi-level optimization, where the high-dimensional embedding and projection module are proposed to facilitate prototype refinement and calibration.

High-dimensional Prototypes Refinement. Recently, Hyperdimensional Computing has been used in computer vision tasks like few shot learning (Karunaratne et al. 2021), out-of-distribution detection (Wilson et al. 2023), and image translation (Theiss et al. 2022), which leverage quasi-orthogonal high-dimensional representations without inducing much training and inference overhead. As for NO-CL, we project vanilla prototypes into high-dimensional prototypes (H-P) to accommodate online new classes with minimal interference with learned knowledge. The initial H-P (ϕ^m) in the m-th online session is obtained based on the single-pass raw data:

$$\phi_i^m = \frac{1}{|k_i|}\sum_{j=1}^{k_i}\left(Proj_{\theta_2}\left(f_j^m\right)\right) \tag{4}$$

where $Proj_{\theta_2}$ represents the projection module with parameters (θ_2) and k means the number of raw samples of class i. As we can see in Figure 4 (Novel: H-P), though the prototypes have been clustered and separated to some extent in high-dimensional embedding, the overlaps among novel classes and between base and novel classes also exist. Therefore, we refine the high-dimensional prototypes of novel classes (ϕ_n) and re-calibrate the projection module (θ_2) based on refined online H-P (ϕ_n), pre-computed base H-P (ϕ_b), and V-P (φ). The bi-level optimization object is formulated as follows:

$$\min_{\theta_2}[L_1(\theta_2; \varphi; \phi_n^* \cup \phi_b)] \tag{5a}$$

$$\text{s.t. } \phi_n^* = \arg\min_{\phi_n} L_2(\phi_n; \phi_b) \tag{5b}$$

In the following, we elaborate on the implementation details for the bi-level optimization.

In the **inner-level optimization**, i.e., Eq. (5b), as analyzed above, to eliminate the overlaps in H-P (ϕ), we refine ϕ_n by decreasing the cosine similarity of H-P among inter-novel classes (L^{in}), and between base and novel classes (L^{bn}), respectively. The formulas are as follows:

Algorithm 1: Training procedure of DSR.

Input: Training data $\{\mathcal{D}_{train}^0, \mathcal{D}_{train}^1, ..., \mathcal{D}_{train}^m\}$, base epoch n_1, online iteration T
Output: Optimal θ_1, θ_2, and ϕ_n

Initialize: θ_1, θ_2;
Base Session: *// train θ_1^0 and θ_2^0*
while *epoch* $< n_1$ **do**
  train θ_1^0 and θ_2^0 with Eq. (1).
end
Obtain φ^0 and ϕ^0 with Eq. (3)&(4);
Online Session: *// bi-level optimize θ_2 and ϕ_n*
for *incremental sessions* $M \in \{1, 2...m\}$ **do**
  Input: θ_1^{M-1}, θ_2^{M-1}, φ^{M-1} ϕ^{M-1}, $\mathcal{D}_{train}^M$.
  Output: θ_1^M, θ_2^M, φ^M, ϕ^M.
  $\theta_1^M \leftarrow \theta_1^{M-1}$;
  Obtain φ^M and ϕ^M with Eq. (3)&(4);
  while $t < T$ *or not converged* **do**
    –inner-level optimization–
    Update ϕ^M with Eq. (9);
    –outer-level optimization–
    Update θ_2^M with Eq. (11);
  end
end

$$L_2(\phi_n; \phi_b) = L^{in}(\phi_n) + L^{bn}(\phi_n, \phi_b) \tag{6}$$

$$L^{in}(\phi_n) = \sum_{i,j=1,s.t.\ i\neq j}^{|\phi_n|} < \sigma(\phi_n^i), \sigma(\phi_n^j) > \tag{7}$$

$$L^{bn}(\phi_n) = \sum_{i=1}^{|\phi_n|}\sum_{j=1}^{|\phi_b|} < \sigma\left(\phi_n^i\right), \sigma\left(\phi_b^j\right) > \tag{8}$$

where $<\cdot,\cdot>$ and σ mean cosine similarity and tanh activation function. Moreover, to avoid significant deviations from the original representations, ϕ_n updates in the exponential moving average strategy (EMA):

$$\phi_n^* = \alpha\phi_n + (1-\alpha)\left(\phi_n - \gamma \bigtriangledown_{\phi_n} L_2(\phi_n; \phi_b)\right) \tag{9}$$

where α is the momentum hyper-parameter(α = 0.9 in this paper), and γ is the learning rate.

The aim of the **outer-level optimization**, i.e., Eq. (5a), is to re-calibrate projection module (θ_2) based on refined ϕ_n^* and ϕ_b by self-augmenting vanilla prototypes from pre-computed $\mathbb{D}(\varphi)$. The optimization of θ_2 is as follows:

$$L_1 = -\sum_{i=1}^{\mathcal{C}}\sum_{k=1}^{K} < \theta_2\left(\mathbb{D}(\varphi)\right), \phi_n^* \cup \phi_b > \tag{10}$$

$$\theta_2 = \theta_2 - \beta \bigtriangledown_{\theta_2} L_1(\theta_2; \varphi; \phi_n^* \cup \phi_b) \tag{11}$$

where β is the learning rate, $\mathcal{C}$ and K are the number of learned classes and sampled prototypes, respectively.

For inference, cosine similarities are computed between sample embeddings and H-P (ϕ) of learned classes ($\mathcal{C}$) for classification: $\text{pred} = \arg\max_{i=1,..,\mathcal{C}} < \theta_2(G(\theta_1(x))), \phi_i >$.

Methods	**CORE-50**		**CIFAR100**		**Mini-ImageNet**	
Metrics	**Acc**(base/novel)|**HM**		**Acc**(base/novel)|**HM**		**Acc**(base/novel)|**HM**	
FACT	43.0(53 /29)|37.1		44.8(56/28)|37.2		45.5(58/26)|36.3	
ALICE	41.5(51/28)|25.9		43.5(56/24)|33.8		45.8(59/26)|35.8	
PASS	37.9(63/1)|1.0		38.6(64/1)|1.8		40.4(65/2)|3.4	
SSRE	—		39.8(66/1)|1.0		—	
MS	*1000*	*2000*	*1000*	*2000*	*1000*	*2000*
MIR	22.6(25/20)|21.9	24.5(27/21)|23.5	24.2(26/21)|23.3	25.1(27/22)|24.3	22.9(25/20)|22.2	23.8(26/21)|23.1
GD	25.8(27/24)|25.8	27.5(29/25)|26.9	25.8(27/24)|25.4	27.1(29/25)|26.6	23.2(25/22)|23.1	24.4(25/23)|24.4
ASER	29.4(31/28)|29.0	31.4(34/28)|30.5	30.7(31/30)|30.5	33.6(35/32)|33.2	25.4(28/22)|24.5	29.7(30/29)|29.5
SCR	39.4(38/42)|39.7	40.7(41/40)|40.5	37.1(41/36)|38.3	41.9(45/38)|41.1	36.2(36/36)|36.1	38.8(44/31)|36.4
SCR$_{ft}$	39.6(46/30)|36.2	43.6(52/32)|39.2	39.6(50/24)|32.3	42.1(54/25)|33.8	38.8(44/31)|36.2	42.8(48/36)|40.8
OCM	41.0(42/40)|40.8	42.5(43/42)|42.3	37.3(38/37)|37.2	41.6(43/40)|41.3	37.2(36/39)|36.1	40.9(46/34)|38.8
OCM $_{ft}$	41.1(47/33)|38.7	43.7(48/37)|41.9	40.8(46/33)|38.3	42.3(47/35)|40.3	39.0(41/37)|38.2	41.2(43/39)|40.8
DVC	39.9(40/41)|40.0	41.8(43/41)|41.6	38.6(38/39)|38.9	41.8(43/39)|41.2	35.6(34/37)|35.9	38.4(37/41)|38.9
DVC$_{ft}$	41.9(48/33)|38.9	43.7(49/36)|41.5	39.0(43/33)|37.2	40.5(44/36)|39.4	36.2(40/31)|34.8	39.3(41/37)|38.8
Ours(+CE)	46.8+3.1(48/46)|46.6		45.8+1.0(50/40)|44.2		47.7+1.9(53/40)|45.6	
Ours(+SC)	**49.1**+5.4(50/48)|**50.0**		**48.6**+3.8(52/43)|**47.2**		**50.7**+4.9(56/43)|**48.4**	

Table 1: Class-wise accuracy (Acc) by the end of the training of all classes, base classes, and novel classes. Harmonic accuracy (HM) is also illustrated. MS and $_{ft}$ mean the example memory size and finetuning versions. The best results are in bold.

Experiments

Datasets and Evaluation Protocols. As mentioned in section 3, the benchmark datasets are divided into ($60\%+4\%\times 10$), where the base session contains 60% classes for base session training, and the rest of the classes are online incrementally learned within 10 sessions. Other splits are also provided in ***Appendix B***. We conduct experiments on three widely used datasets, including CORE-50 (Lomonaco and Maltoni 2017), CIFAR 100 (Krizhevsky and Hinton 2009), and Mini-ImageNet (Vinyals et al. 2016), which have 50, 100, and 100 classes, respectively. Following recent class-incremental learning methods (De Lange et al. 2022; Mai et al. 2022), class-wise average accuracy (Acc) and average forgetting (A_f) are applied to evaluate the performance. Meanwhile, for the NO-CL problem, the number of base and novel classes is unbalanced. To evaluate the overall performance of the stability-plasticity dilemma, we also employ harmonic metric (HM, i.e., $\mathrm{HM}=\frac{2\times \mathrm{Acc_b}\times \mathrm{Acc_n}}{\mathrm{Acc_b}+\mathrm{Acc_n}}$, Acc_b and Acc_n denote the accuracy of base and novel classes) like (Kalla and Biswas 2022; Peng et al. 2022).

Comparison Methods. We compare DSR with three categories baselines: (1) OCL: GD(Prabhu, Torr, and Dokania 2020), MIR (Aljundi et al. 2019a), ASER (Shim et al. 2021), SCR (Mai et al. 2021), OCM (Guo, Liu, and Zhao 2022), DVC (Gu et al. 2022). (2) NE-CL: PASS (Zhu et al. 2021c), SSRE (Zhu et al. 2022). (3) FS-CL: FACT (Zhou et al. 2022), ALICE (Peng et al. 2022). All comparisons are trained and inferred in the **same protocols** of NO-CL.

Implementation details. Following (Gu et al. 2022; Guo, Liu, and Zhao 2022), we employ a reduced ResNet-18 as the backbone without pre-training. We use stochastic gradient descent with a learning rate of 0.1 with a batch size of 100 during the base session. The dimension of high-dimensional embedding is 2048, and $Proj_{\theta_2}$ is implemented as a two-layer MLP with a hidden layer of 512 dimensions with relu as the activation function. For other hyper-parameters, we set the base session training epoch n_1, online iteration T to 100, 20, set online learning rate γ, β all to 0.01, set feature transform coefficient λ and the number of sampled prototypes K to 0.5, 20. Analysis of hyper-parameters is performed in ***Appendix D***. As for compared methods, we adopt the **same training protocols** of NO-CL as ours and adopt the defaulted hyper-parameters of their methods (please refer to ***Appendix A***). We report the mean result of all methods over ten different runs.

Results and Ablation Studies

Comparing with the State-of-the-art. We compare our method (+CE and +SC loss versions) with other SOTA methods in the setting of the proposed NO-CL problem. The results are illustrated in Tables 1 and 2 and Figure 5, which give the following observations. **1).** Overall, in terms of class-wise accuracy, our method with CE and SC loss achieves pleasant results, especially with SC loss, which outperforms others by 5.4%, 3.8%, and 4.9% in CORE-50, CIFAR100, and Mini-ImageNet, respectively. Note that SOTA methods like FACT, ALICE, PASS SCR, OCM, and DVC adopt sophisticated pre-training strategies like self-supervised learning, supervised contrastive learning, data augmentation etc. For harmonic accuracy (HM), which measures the performance of stability-plasticity trade-offs, ours also exceed other methods by a large margin. Our method also outperforms OCL methods, which employ large example buffers, in most cases for average forgetting metrics. **2).** Concretely. for OCL methods, though equipped with large example buffers and pre-trained information, the over-fitting and catastrophic forgetting problems are also severe compared to ours. To avoid these issues, similar to ours, we freeze the backbone after base session training

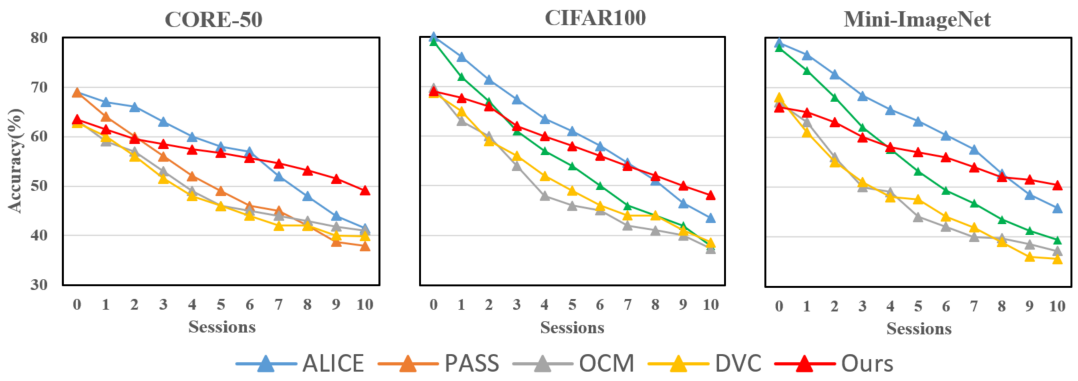

Figure 5: The line chart represents class-wise average accuracy of SOTA methods (DVC(Gu et al. 2022), OCM(Guo, Liu, and Zhao 2022), PASS(Zhu et al. 2021c), and ALICE (Peng et al. 2022)) along the incremental sessions.

Method	CORE-50	CIAFR 100	Mini-ImageNet
MS	*1000/2000*	*1000/2000*	*1000/2000*
SCR	10.8/9.2	13.2/9.6	15.3/12.3
SCR$_{ft}$	8.3/7.6	7.2/**6.7**	9.3/8.4
OCM	15.2/12.6	16.1/13.4	16.9/12.3
OCM$_{ft}$	9.3/8.1	10.6/7.6	11.7/9.1
DVC	11.3/10.8	14.3/12.0	16.9/13.4
DVC$_{ft}$	8.7/7.9	9.2/8.6	11.8/9.7
Ours(+CE)	7.6	7.8	9.0
Ours(+SC)	**7.3**	6.9	**8.2**

Table 2: Average forgetting (A$_f$, lower is better) results.

and only jointly finetune the classifier head with online data and buffer samples (denoted as $_{ft}$). As we can see from SCR$_{ft}$, OCM$_{ft}$, and DVC$_{ft}$, simply freezing the backbone can not achieve the stability-plasticity trade-off that though base knowledge can be better preserved while damaging the ability to learn online novel classes. Although large example buffers and frequent rehearsal can eliminate these issues, which somewhat violate the online learning protocols. **3).** As for NE-CL methods, PASS and SSRE fail to learn the novel classes in online sessions. Due to the lack of old class samples, PASS and SSRE promote knowledge transfer in the progressive knowledge distillation process, while the distillation constraints hamper the online learning of novel classes. **4).** For the FS-CL methods, FACT and ALICE focus on training generalization feature representations during the base session. They directly infer novel classes based on robust embedding. However, without adjusting the representation, ALICE fails in a large number of sessions, as shown in the last few sessions in Figure 5 and $60\% + 2\% \times 20$ configuration in ***Appendix B***. Note that NE-CL methods, i.e., PASS and SSRE, employ the more sophisticated pre-training strategy, which perform slightly better than ours in the base session. **3).** Moreover, as for computation overhead during online learning, which is usually considered in OCL scenarios (Fini et al. 2020), as shown in Figure 2, our method only consumes $\sim$ 35 seconds and minimal memory overhead in CIFAR100 dataset. More quantitative results of computation overhead are in ***Appendix E***.

Ablation Studies. (1) The necessity of DSR strategy: To overall validate the necessity of DSR strategy, we directly infer with the prototypes from the high-dimensional embedding (denoted as baseline). For example-free online sessions, Though quasi-orthogonal high-dimensional representations preserve the pre-trained information and accommodate online new classes to some extent, the baseline does not fully leverage the online data stream and fails in all class-wise accuracy and HM metrics. Moreover, based on the baseline, we directly optimize the H-P with the L_2 (Eq. (6)) function (baseline+L_2). Note that for the proposed NO-CL, we have no example buffer to re-calibrate the backbone (θ_1). Direct optimization prototypes results in degrading performance. We ablate inner-level updating (**w/o** L_2). Novel classes degrade to some extent. **(2) The effectiveness of each component:** We ablate Gaussian-like rectification (w/o $G(\cdot)$) and directly revisit the prototype without sampling repeatedly from Gaussian distributions (w/o SAP). We can see that the Gaussian-like rectification brings $\sim$1.5% gains via reducing the skewness of distributions. Augmented prototypes significantly preserve the decision boundaries of previous classes while also being beneficial to the overall performance through joint optimization. To prove the effectiveness of amending prototypes in the high-dimensional embedding (w/o HD), we project the vanilla prototypes to low-dimensional embedding (256) instead of high dimension (2048). The performance of both base and novel classes degrades, particularly in novel classes. The reason is that prototypes in low-dimensional embedding require dedicated alignments, otherwise resulting in confusion both in base and novel classes, which is not suitable for NO-CL. EMA updating HD better realizes the trade-off between learned and refined knowledge. More ablations can refer to ***Appendix D***. **(3) The importance of the base session training:** We adopt the data augmentation strategy (w/ DA) proposed by (Peng et al. 2022; Zhu et al. 2021b) in the base training session to obtain diverse and transferable representations. More base session training strategies please refer to ***Appendix C***. We learn that the robust embedding improves our method by a large margin, both in preserving old classes and online accommodating novel classes, which provides a

Ablations	CIFAR100	Mini-ImageNet
Metrics	**Acc**(base/novel)|**HM**	**Acc**(base/novel)|**HM**
baseline	43.4(54.2/27.4)|36.3	43.7(57.1/23.6)|33.4
baseline+L_2	38.9(52.3/18.9)|27.8	40.2(54.9/18.1)|27.2
w/o L_2	46.8(51.6/39.7)|44.9	48.9(55.1/39.8)|46.2
w/o $G(\cdot)$	47.1(51.0/41.3)|45.6	49.1(54.3/41.4)|47.0
w/o SA-P	46.2(50.5/39.7)|44.4	47.5(53.9/37.9)|44.5
w/o HD	46.7(51.6/39.4)|44.7	48.5(54.7/39.3)|45.7
w/o EMA	48.0(51.4/43.0)|46.8	50.1(55.3/42.3)|47.9
Ours	48.6(52.4/42.9)|47.2	50.7(56.1/42.6)|48.4
Ours w/ DA	51.2(55.7/44.6)|49.5	53.2(58.2/45.8)|51.3

Table 3: Ablation studies on CIFAR100 and Mini-ImageNet. Experiments are conducted with the SC loss.

direction to solve the proposed NO-CL problem.

Conclusion

In this paper, we formulate a novel, practical, but challenging problem named NO-CL, which aims to preserve pre-trained base classes information, while efficiently learning novel classes continually from the single-pass (i.e., online) data stream, without example buffers. To solve this problem, we have proposed a novel Dual-prototype Self-augment and Refinement (DSR) method, which presents two solutions: 1) Dual class prototypes: vanilla and high-dimensional prototypes (V-P and H-P) are maintained to utilize the pre-trained information and obtain robust quasi-orthogonal representations. 2) Self-augment and refinement: Without buffers and offline training, we bi-level optimize the extra high-dimensional prototypes alternatively with the projection module, to refine the decision boundaries and recalibrate projection module based on optimized H-P and self-augment V-P. Extensive experiments demonstrate the effectiveness of DSR in handling the NO-CL problem.

Acknowledgements

This research was partially supported by Project PolyU15222621 and PolyU15225023, the National Natural Science Foundation of China under grants 62302184, Hong Kong RGC General Research Fund (No. 152211/23E), the National Natural Science Foundation of China (No. 62102327), and PolyU Internal Fund (No. P0043932).

References

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory Aware Synapses: Learning what (not) to forget. In *ECCV*.

Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; and Page-Caccia, L. 2019a. Online Continual Learning with Maximal Interfered Retrieval. In *NeurIPS*.

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019b. Gradient based sample selection for online continual learning. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *NeurIPS*.

Caccia, L.; Aljundi, R.; Asadi, N.; Tuytelaars, T.; Pineau, J.; and Belilovsky, E. 2022. New Insights on Reducing Abrupt Representation Change in Online Continual Learning.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2018. Efficient Lifelong Learning with A-GEM. *ICLR*.

Cheraghian, A.; Rahman, S.; Fang, P.; Roy, S. K.; Petersson, L.; and Harandi, M. 2021. Semantic-Aware Knowledge Distillation for Few-Shot Class-Incremental Learning. In *CVPR*, 2534–2543.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE TPAMI*, 44(7): 3366–3385.

Dong, S.; Hong, X.; Tao, X.; Chang, X.; Wei, X.; and Gong, Y. 2021. Few-Shot Class-Incremental Learning via Relation Knowledge Distillation. *AAAI*, 1255–1263.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *ECCV*, 86–102.

Fini, E.; Lathuilière, S.; Sangineto, E.; Nabi, M.; and Ricci, E. 2020. Online Continual Learning under Extreme Memory Constraints. In *ECCV*, 720–735.

Gayler, R. W. 2004. Vector Symbolic Architectures answer Jackendoff's challenges for cognitive neuroscience. *Joint International Conference on Cognitive Science*.

Gu, Y.; Yang, X.; Wei, K.; and Deng, C. 2022. Not Just Selection, but Exploration: Online Class-Incremental Continual Learning via Dual View Consistency. In *CVPR*, 7442–7451.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online Continual Learning through Mutual Information Maximization. In *ICML*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

Hersche, M.; Karunaratne, G.; Cherubini, G.; Benini, L.; Sebastian, A.; and Rahimi, A. 2022. Constrained Few-Shot Class-Incremental Learning. In *CVPR*, 9057–9067.

Hu, X.; Tang, K.; Miao, C.; Hua, X.-S.; and Zhang, H. 2021. Distilling Causal Effect of Data in Class-Incremental Learning. In *CVPR*, 3957–3966.

Kalla, J.; and Biswas, S. 2022. S3C: Self-Supervised Stochastic Classifiers for Few-Shot Class-Incremental Learning. In *ECCV*, 432–448. Cham.

Kanerva, P. 2009. Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computation*.

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 *ICML*, 10734–10750.

Kang, H.; Yoon, J.; Madjid, S. R. H.; Hwang, S. J.; and Yoo, C. D. 2023. On the Soft-Subnetwork for Few-Shot Class Incremental Learning. In *ICLR*.

Karunaratne, G.; Le Gallo, M.; Cherubini, G.; Benini, L.; Rahimi, A.; and Sebastian, A. 2020. In-memory hyperdimensional computing. In *Nature Electronics*.

Karunaratne, G.; Schmuck, M.; Le Gallo, M.; Cherubini, G.; Benini, L.; Sebastian, A.; and Rahimi, A. 2021. Robust high-dimensional memory-augmented neural networks. In *Nature Communications*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised Contrastive Learning. In *NeurIPS*, 18661–18673.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. In *Technical Report*.

Laurens, V. D. M.; and Hinton, G. 2008. Visualizing Data using t-SNE. *JMLR*, 9(2605): 2579–2605.

Lee, S.; Ha, J.; Zhang, D.; and Kim, G. 2020. A Neural Dirichlet Process Mixture Model for Task-Free Continual Learning. *ICLR*.

Lee, S.-W.; Kim, J.-H.; Jun, J.; Ha, J.-W.; and Zhang, B.-T. 2017. Overcoming Catastrophic Forgetting by Incremental Moment Matching. In *NeurIPS*.

Lin, H.; Zhang, B.; Feng, S.; Li, X.; and Ye, Y. 2023. PCR: Proxy-Based Contrastive Replay for Online Class-Incremental Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 24246–24255.

Liu, R.; Gao, J.; Zhang, J.; Meng, D.; and Lin, Z. 2022. Investigating Bi-Level Optimization for Learning and Vision From a Unified Perspective: A Survey and Beyond. *IEEE TPAMI*, 44(12): 10045–10067.

Liu, Y.; Schiele, B.; and Sun, Q. 2021. Adaptive Aggregation Networks for Class-Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2544–2553.

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 (CVPR)*.

Lomonaco, V.; and Maltoni, D. 2017. CORe50: a New Dataset and Benchmark for Continuous Object Recognition. In *CoRL*, 17–26.

Lopez-Paz, D.; and Ranzato, M. A. 2017. Gradient Episodic Memory for Continual Learning. In *NeurIPS*. Curran Associates, Inc.

Luo, Z.; Liu, Y.; Schiele, B.; and Sun, Q. 2023. Class-Incremental Exemplar Compression for Class-Incremental Learning. In *CVPR*, 11371–11380.

Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; and Sanner, S. 2022. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469: 28–51.

Mai, Z.; Li, R.; Kim, H.; and Sanner, S. 2021. Supervised Contrastive Replay: Revisiting the Nearest Class Mean Classifier in Online Class-Incremental Continual Learning. In *CVPR Workshops*, 3589–3599.

Mallya, A.; and Lazebnik, S. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *CVPR*.

Nair, V.; and Hinton, G. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*.

Peng, C.; Zhao, K.; Wang, T.; Li, M.; and Lovell, B. C. 2022. Few-Shot Class-Incremental Learning from an Open-Set Perspective. In *ECCV*, 382–397.

Prabhu, A.; Torr, P. H. S.; and Dokania, P. K. 2020. GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In *ECCV*, 524–540.

Shim, D.; Mai, Z.; Jeong, J.; Sanner, S.; Kim, H.; and Jang, J. 2021. Online Class-Incremental Continual Learning with Adversarial Shapley Value. *AAAI*.

Sinha, A.; Malo, P.; and Deb, K. 2018. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Transactions on Evolutionary Computation*, 22(2): 276–295.

Song, Z.; Zhao, Y.; Shi, Y.; Peng, P.; Yuan, L.; and Tian, Y. 2023. Learning with Fantasy: Semantic-Aware Virtual Contrastive Constraint for Few-Shot Class-Incremental Learning. In *CVPR*, 24183–24192.

Tao, X.; Hong, X.; Chang, X.; Dong, S.; Wei, X.; and Gong, Y. 2020. Few-Shot Class-Incremental Learning. In *CVPR*.

Theiss, J.; Leverett, J.; Kim, D.; and Prakash, A. 2022. Unpaired Image Translation via Vector Symbolic Architectures. In *ECCV*, 17–32.

Tukey, J. W. 2010. Exploratory data analysis. In *Addison-Wesley Series in Behavioral Science*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; kavukcuoglu, k.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *NeurIPS*.

Wilson, S.; Fischer, T.; Sünderhauf, N.; and Dayoub, F. 2023. Hyperdimensional Feature Fusion for Out-of-Distribution Detection. In *WACV*, 2644–2654.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to Distill: Data-Free Knowledge Transfer via DeepInversion. In *CVPR*.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and Weijer, J. v. d. 2020. Semantic Drift Compensation for Class-Incremental Learning. In *CVPR*.

Zhang, Y.; Pfahringer, B.; Frank, E.; Bifet, A.; Lim, N. J. S.; and Jia, Y. 2022. A simple but strong baseline for online continual learning: Repeated Augmented Rehearsal. *NeurIPS*.

Zhao, L.; Lu, J.; Xu, Y.; Cheng, Z.; Guo, D.; Niu, Y.; and Fang, X. 2023. Few-Shot Class-Incremental Learning via Class-Aware Bilateral Distillation. In *CVPR*, 11838–11847.

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 *CVPR*, 9046–9056.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021a. Class-Incremental Learning via Dual Augmentation. In *NeurIPS*, 14306–14318.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021b. Class-Incremental Learning via Dual Augmentation. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 14306–14318. Curran Associates, Inc.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021c. Prototype Augmentation and Self-Supervision for Incremental Learning. In *CVPR*, 5871–5880.

Zhu, K.; Cao, Y.; Zhai, W.; Cheng, J.; and Zha, Z.-J. 2021d. Self-Promoted Prototype Refinement for Few-Shot Class-Incremental Learning. In *CVPR*, 6801–6810.

Zhu, K.; Zhai, W.; Cao, Y.; Luo, J.; and Zha, Z.-J. 2022. Self-Sustaining Representation Expansion for Non-Exemplar Class-Incremental Learning. In *CVPR*, 9296–9305.