Learning to Predict Gradients for Semi-Supervised Continual Learning

Yan Luo, *Member, IEEE*, Yongkang Wong, *Member, IEEE*, Mohan Kankanhalli, *Fellow, IEEE*, and Qi Zhao, *Senior Member, IEEE*

***Abstract*— A key challenge for machine intelligence is to learn new visual concepts without forgetting the previously acquired knowledge. Continual learning (CL) is aimed toward addressing this challenge. However, there still exists a gap between CL and human learning. In particular, humans are able to continually learn from the samples associated with known or unknown labels in their daily lives, whereas existing CL and semi-supervised CL (SSCL) methods assume that the training samples are associated with known labels. Specifically, we are interested in two questions: 1) how to utilize unrelated unlabeled data for the SSCL task and 2) how unlabeled data affect learning and catastrophic forgetting in the CL task. To explore these issues, we formulate a new SSCL method, which can be generically applied to existing CL models. Furthermore, we propose a novel gradient learner to learn from labeled data to predict gradients on unlabeled data. In this way, the unlabeled data can fit into the supervised CL framework. We extensively evaluate the proposed method on mainstream CL methods, adversarial CL (ACL), and semi-supervised learning (SSL) tasks. The proposed method achieves state-of-the-art performance on classification accuracy and backward transfer (BWT) in the CL setting while achieving the desired performance on classification accuracy in the SSL setting. This implies that the unlabeled images can enhance the generalizability of CL models on the predictive ability of unseen data and significantly alleviate catastrophic forgetting. The code is available at https://github.com/luoyan407/grad_prediction.git.**

***Index Terms*— Continual learning (CL), gradient prediction, semi-supervised learning (SSL).**

NOMENCLATURE

Symbol	Definition
x_i	ith training sample.
y_i	Label of the ith training sample.
t_i	The i-th task training sample.
$\tilde{x}_i$	ith unlabeled sample.
ℓ	Loss function.
z_i	Logit of the ith training sample.
θ	Parameters of the classification model f.
ω	Parameters of the gradient learner h.
η	Learning rate for updating θ.
$\hat{\eta}$	Learning rate for updating ω.
α	Hyperparameter controlling the proportion of the magnitude.
λ	Coefficient with respect to the fitness loss.

Manuscript received 21 September 2022; revised 17 May 2023, 24 August 2023, and 6 November 2023; accepted 23 January 2024. This work was supported in part by the NSF under Grant 1908711 and Grant 2143197; and in part by the National Research Foundation, Singapore, under its Strategic Capability Research Centers Funding Initiative. *(Corresponding author: Qi Zhao.)*

Yan Luo was with the Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA. He is now with the Harvard Ophthalmology AI Laboratory, Harvard University, Boston, MA 02114 USA (e-mail: yluo16@meei.harvard.edu).

Yongkang Wong and Mohan Kankanhalli are with the School of Computing, National University of Singapore, Singapore 117417 (e-mail: yongkang.wong@nus.edu.sg; mohan@comp.nus.edu.sg).

Qi Zhao is with the Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA (e-mail: qzhao@cs.umn.edu).

Digital Object Identifier 10.1109/TNNLS.2024.3361375

I. INTRODUCTION

CONTINUAL learning (CL) models observe sets of labeled data through a sequence of tasks [1], [2]. The tasks may vary over time, e.g., images with novel visual concepts (i.e., classes) or addressing different problems from the previous tasks [3]. CL is analogous to human learning. Humans are able to continually acquire, adjust, and transfer knowledge and experiences throughout their lifespan. The key challenges are twofold. First, the learning models can abruptly forget previously absorbed knowledge while learning new information in novel tasks, i.e., suffer from catastrophic forgetting [4]. Second, how to employ the knowledge learned from previous tasks to quickly adapt to novel tasks.

Previous CL methods presume that the labels associated with the data are known [5], [6], [7], [8], [9], [10], [11]. This assumption may be divergent from human learning, where a considerable amount of labels associated with the unlabeled data could be novel and unrelated to the known labels. Furthermore, large-scale labeled data may not always be available due to the limits of labor-intensive and expensive human annotations. Moreover, the classes of a task are distinct from the ones in the other tasks, or the task's labels may be of a different form, e.g., category versus bounding box. Therefore, we do not presume any constraint that restricts the correlation between the labels associated with unlabeled data and the ones associated with labeled data. Instead, the unlabeled data could have either the same or different class labels as the labeled data, which is shown in Fig. 1. As a result, the fundamental challenge lies in the generalizability of learning in this SSCL setting. The CL models not only generalize the knowledge learned from preceding tasks to the current task but also should leverage unlabeled data that are associated with unknown labels to boost the learning process.

The labels that are known to the learning process play an important role in an end-to-end learning paradigm, even in the SSL setting. Through the labels and the predefined loss functions, the gradients are computed to back-propagate to neurons in each layer. This gradient-based learning process is key to updating the models to make more precise predictions [5], [6], [7], [8], [9], [10], [12], [13], [14], [15]. However, when the underlying labels of unlabeled data are unknown,

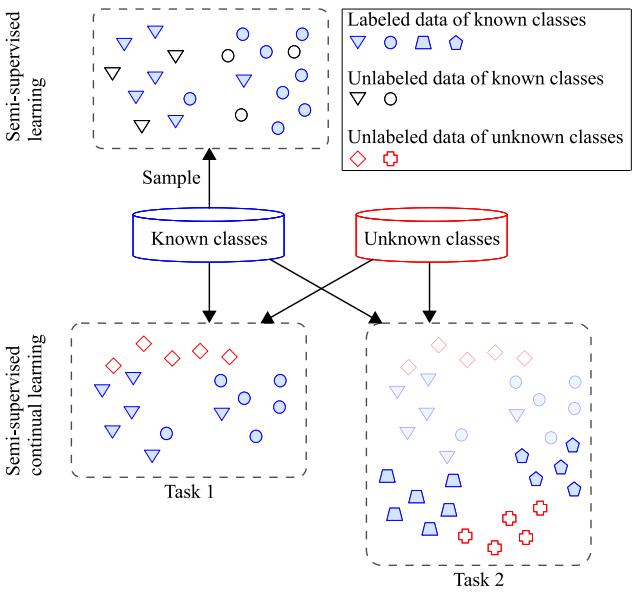


Fig. 1. Conceptual comparison between the challenge in the novel semi-supervised CL (SSCL) problem and the one in the semi-supervised learning (SSL) problem. The key difference is that the underlying classes with respect to the unlabeled data could be unknown in the SSCL problem, while the ones in SSL are assumed to be from the known classes. To suitably adapt to the CL paradigm, we do not impose such a constraint on the novel SSCL problem. Instead, the underlying labels of unlabeled data can be from either known classes or unknown classes. The faded-out samples in task 2 indicate that the samples in task 1 are not available in task 2 according to the protocol.

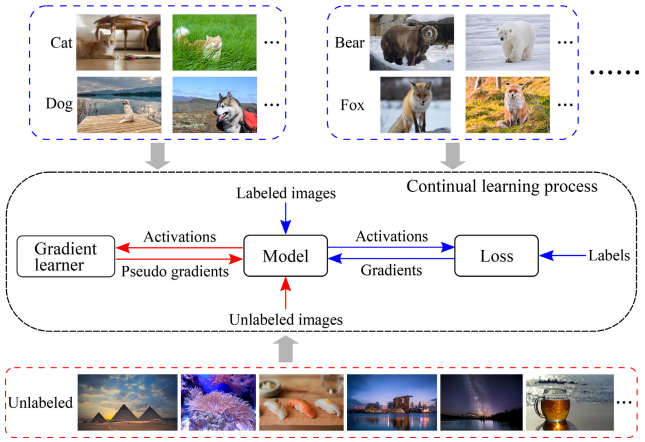


Fig. 2. Problem of semi-supervised CL. Conventional supervised CL requires labels to compute gradients for model update (see blue flows). In contrast, this work proposes to *predict gradients* so that the unlabeled images can be incorporated in the CL paradigm for better generalizability (see red flows).

it is a challenge to generate the gradients that improve the generalizability of models in the SSCL setting.

Conventionally, pseudo labeling, i.e., predicting labels by a teacher network for the unlabeled data and taking them as the ground-truth labels for training a student network, is widely used for SSL [16], [17], [18], [19]. However, it may not work in the SSCL setting as the classes of a task are distinct from the ones in the other tasks or the task's labels may be of a different form. In contrast to the pseudo-labeling methods, learning to predict pseudo gradients on unlabeled samples is straightforward and effective as predicting labels is skipped. Moreover, the pseudo gradients are aligned with the knowledge learned from samples in various categories, while the gradients generated by pseudo-labeling methods are aligned with a specific category as an unlabeled sample is conventionally labeled as a category in the CL set.

To utilize unlabeled data in the supervised CL framework, we propose a novel gradient-based learning method that learns from the labeled data to predict *pseudo gradients* for the unlabeled data, as shown in Fig. 2. Specifically, a novel gradient learner learns the mapping between features and the corresponding gradients generated with labels. We follow [5], [14] to conduct extensive experiments on CL benchmarks, i.e., MNIST-R, MNIST-P, iCIFAR-100, CIFAR-100, and miniImageNet. To verify the generalization ability of the proposed method, we follow [20] to evaluate the proposed method on SVHN, CIFAR-10, and CIFAR-100. The main contributions of this work are summarized as follows.

1) We propose a novel SSCL method that leverages the rich information from unlabeled data to improve the generalizability of CL models.
2) We propose a learning method that learns to predict gradients for unlabeled data. To the best of our knowledge, this is the first work that generates pseudo gradients without ground-truth labels.
3) Extensive experiments and ablation studies show that the proposed method improves the generalization performance on all metrics. This implies that learning with unlabeled data is helpful for improving the predictive ability and alleviating catastrophic forgetting of CL models.
4) We provide empirical evidence to show that the proposed method can generalize to the SSL task.

II. Related Work

A. Continual Learning

CL is a learning paradigm where a model learns through a sequence of tasks [1], [2], [21]. CL is a branch of online learning [22], [23], where the challenge is to balance the retention of knowledge from preceding tasks with the acquisition of new information for future tasks. However, catastrophic forgetting is a common issue in this paradigm [4]. There are three types of CL, namely, task-incremental learning, domain-incremental learning, and class-incremental learning [24]. Task-incremental learning solves a sequence of distinct tasks when the task ID is provided along the process. Domain-incremental learning adapts to changing input data distributions while maintaining performance on the original task. Class-incremental learning solves a sequence of distinct tasks and infers the task ID. Compared to domain-incremental learning, task-incremental learning, and class-incremental learning emphasize recognizing and classifying new classes without forgetting previous knowledge, which are closely related to the proposed method.

There are a number of works that can be cast into the category of task-incremental learning [5], [7], [8], [12], [14], [15], [25], [26], [27]. Specifically, Lopez-Paz and Ranzato [5] proposed a memory-based method, namely, GEM, to impose a constraint on the gradients with respect to the training samples and the memory. Along the same line, Luo et al. [15] introduced a gradient alignment method DCL that enhances the correlation between the gradient and the accumulated gradient. Recently, Ebrahimi et al. [14] proposed an adversarial CL (ACL) approach that aims to factorize task-specific and task-invariant features simultaneously. Unlike GEM, where

the training samples are observed one by one, the training process of ACL would repeat multiple times on every task. All the aforementioned works follow a supervised CL paradigm that requires ground-truth labels. However, how unlabeled data may influence the CL problem remains unclear. In this work, we propose the SSCL paradigm, where the model occasionally observes unlabeled data. The class-incremental learning problem aims to learn visual concepts in new tasks while retaining the visual concepts learned in the previous tasks [11], [26]. Correspondingly, the samples in the coresets would be repeatedly observed in this problem, whereas CL only observes each sample once. In this setting, Carvalho et al. introduce a catastrophic forgetting solution based on knowledge amalgamation (CFA) that learns a student network from multiple heterogeneous teacher models. Lee et al. [11] leverage unlabeled data with a knowledge distillation method to boost class-incremental learning. Notably, the experimental protocol in [11] is different from that of SSCL. The class-incremental learning with unlabeled data maintains three sets of samples through the learning process, i.e., the samples that have been seen in the previous tasks, the samples that are related to the current task and have not been seen before, and the unlabeled samples are selected by a confidence-based strategy from a data pool. In contrast, SSCL only observes the samples that are related to the current task and the unlabeled samples randomly selected from the data pool. For a fair comparison, we utilize the knowledge distillation method in [11] to generate pseudo labels for unlabeled samples as the baselines. This work follows the experimental protocols used in GEM [5] and ACL [14], which are widely adopted task-incremental learning schemes.

Except for the aforementioned methods, the task incremental learning problem and class incremental learning problem can be solved by dark experience replay (DER) [28] and eXtended-DER (X-DER) [29] simultaneously. DER exploits a buffer (i.e., dark experience) storing data from previous tasks to train a student model. X-DER leverages memory updates and future preparation to improve DER.

From the perspective of the strategies, the common strategies tackling the continual problem can be divided into three categories: rehearsal-based, regularization-based, and knowledge distillation-based methods. Rehearsal-based methods address catastrophic forgetting by replaying training samples stored in a memory buffer [28], [30]. In contrast, regularization-based methods prevent catastrophic forgetting by regularizing the model's parameters so that they do not change much when new data is presented [28]. Moreover, knowledge distillation can be used to prevent catastrophic forgetting by transferring knowledge from a previous model (teacher) to a new model (student) [28], [30].

B. Semi-Supervised Learning

SSL, a machine learning technique, involves training a model using both labeled and unlabeled data [31], [32], [33], [34]. This task aims to utilize a small set of labeled data along with a larger set of unlabeled data, enabling the model to establish connections and make predictions on unseen data. For example, Fierimonte et al. [31] proposed a fully decentralized approach to SSL using privacy-preserving matrix completion, specifically addressing the challenge of distributed learning. Duan et al. [32] introduced a novel method that incorporates low-confidence samples into SSL through mutex-based consistency regularization. Another approach by Yang et al. [35] leverages a contrastive learning-based loss function and augmented samples generated via an interpolation-based approach to guide training.

Existing methods are mainly based on pseudo-labeling or self-training, i.e., leveraging the labeled data to predict artificial labels for the unlabeled data [36], [37]. Most modern deep learning-based models follow this line of research [17], [20], [38], [39], [40]. Particularly, the noisy student model [19] employs the teacher–student method to train on ImageNet [41] with unlabeled images from JFT [42], which is an in-house dataset at Google and has 100 million labeled images with 15 000 labels, to achieve state-of-the-art performance. In addition, Zhang et al. [20] proposed a meta-objective to alternatingly optimize the weights and the pseudo labels such that the learning process can leverage unlabeled data. To utilize the abundant unlabeled data, these SSL models assign predicted labels to unlabeled data to generate gradients for back-propagation. In contrast, the proposed method instead predicts pseudo gradients for back-propagation, bypassing the need for a loss function with pseudo-labeled data. Different from conventional (semi-)supervised learning, where visual concepts are unchanging during the learning process, the visual concepts of a task in CL are different from those of the other tasks through the whole learning process. As a result, unlabeled data that are labeled as known classes would break the protocol of the split of classes in various tasks of CL [5]. Instead, CL is in favor of a more generic hypothesis of unlabeled data, that is, the underlying labels of unlabeled data could be unknown. A natural choice is to sample unlabeled images from external datasets, rather than treating training images as unlabeled images. In conventional SSCL [20], the visual concepts that are related to the unlabeled samples are presumed to be known for computing gradients. Different from SSCL, the proposed SSCL in this work does not require this hypothesis. As a result, without known labels, it is unable to compute the gradients for back-propagating the errors. Instead of computing the gradients with the labels, we study how to predict the pseudo gradients by measuring the suitability between unlabeled samples and predicted pseudo gradients in learning a certain visual concept.

C. Gradient-Based Methods

Gradient-based methods refer to a family of optimization algorithms used to find the parameters of a machine learning model that minimize a certain objective function [43]. These methods rely on computing the gradients of the objective function with respect to the model parameters and using those gradients to iteratively update the parameters until convergence. The gradient is a measure of how the loss function changes as a function of the model's parameters [5], [15], [39], [44].

The learning process is composed of forward propagation and back-propagation. Jaderberg et al. [45] proposed a learning framework with synthetic gradients to allow layers to be updated in an asynchronous fashion. The proposed pseudo gradients can be used as ground-truth gradients in such a learning framework when the labels of training images are missing. In particular, [5] and [15] use the information of gradients to form a constraint to improve the performance of CL. Reference [44] have a similar flavor, that is, they aim to learn an optimizer to adaptively compute the step length for the

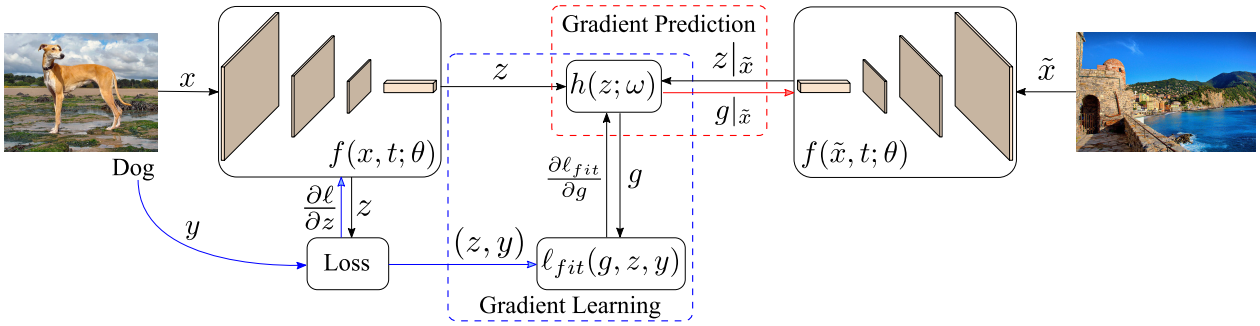


Fig. 3. Overview of the proposed gradient learning and gradient prediction process with the gradient learner $h(\cdot;\omega)$. The backbone network is shared between the two processes.

vanilla gradients. In contrast to these gradient-based methods that rely on labeled data to compute gradients, the proposed method predicts pseudo gradients by maximizing their fitness within the loss function applied to labeled data.

Stochastic optimization methods often use gradients to update a model's parameters. In the literature, stochastic gradient descent (SGD) [46] takes the anti-gradient as the parameters' update for the descent, using the first-order approximation [47]. In a similar manner, several first- and second-order methods are devised to guarantee convergence to local minima under certain conditions [48], [49], [50]. Nevertheless, these methods are computationally expensive and may be not feasible for learning settings with large-scale high-dimensional data. In contrast, adaptive methods, such as Adam [51], RMSProp [52], and Adabound [53], show remarkable efficacy in a broad range of machine learning tasks [52], [53]. Moreover, Zhang et al. [54] proposed an optimization method that wraps an arbitrary optimization method as a component to improve learning stability. These methods are contingent on vanilla gradients to update a model. In this work, we study how the predicted gradients influence the learning process.

III. Problem Set-Up

The training process of supervised learning methods generally requires a training dataset $D_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{i}$ that consists of samples $s_i = (x_i, y_i)$, where $x_i \in \mathcal{X}$ represents a sample and $y_i \in \mathcal{Y}$ represents a target vector, where $\mathcal{Y}$ is the target label space. The samples presumably are identically and independently distributed variables that follow a fixed underlying distribution $\mathcal{D}$ [5]. With all samples, supervised learning methods attempt to find a model $f : \mathcal{X} \overset{\theta}{\rightarrow} \mathcal{Y}$ to map feature vectors to the target vectors, where θ are the parameters of f. In contrast to supervised learning, SCL is more human-like and will observe the continuum of data

$$D_{\text{tr}} = \left\{(x_i, t_i, y_i) | (x_i, y_i) \sim \mathcal{D}_{t_i}, t_i \in \mathcal{T}\right\}$$

where t_i indicates the ith task and $\mathcal{T}$ is a set of tasks. A task is a specific learning problem. Different from supervised learning, which has a fixed distribution, each task is associated with an underlying distribution in the SCL setting. The SCL models are defined as $f : \mathcal{X} \times \mathcal{T} \overset{\theta}{\rightarrow} \mathcal{Y}$. Correspondingly, the loss of SCL is defined as

$$\mathcal{L}(f_\theta, D_{\text{tr}}) = \frac{1}{|D_{\text{tr}}|} \sum_{(x_i,t_i,y_i)\in D_{\text{tr}}} \ell(f_\theta(x_i, t_i), y_i) \tag{1}$$

where $f(\cdot;\theta)$ is simplified as $f_\theta(\cdot)$. With the loss function ℓ and a training sample (x_i, t_i, y_i), the gradient can be computed, i.e., $(\partial \ell / \partial z_i)(\partial z_i / \partial \theta)$, where $z_i = f_\theta(x_i, t_i)$. Typically, ℓ is the cross-entropy loss in the classification task. Note that we follow the convention of classification literature [55], [56], [57] to define the input of ℓ as logits z and ground-truth labels y, instead of predicted labels and ground-truth labels. Finally, the model is updated with the computed gradient, that is,

$$\theta \leftarrow \theta - \eta \frac{\partial \ell}{\partial z_i} \frac{\partial z_i}{\partial \theta} \tag{2}$$

where η is the learning rate for updating θ. Let $\Omega(X)$ be the set of classes associated with all labeled data X, and $\Omega(\tilde{X})$ be the set of classes associated with all unlabeled data $\tilde{X}$. We assume that $\Omega(X) \subset \Omega(\tilde{X})$. In other words, the underlying labels associated with unlabeled data are likely to be unknown to the learning process.

In this study, we introduce the concept of SSCL, which involves the use of both labeled and unlabeled data to train CL models. If the input is unlabeled data, the model update is shown in (2) cannot be performed. This is because the underlying labels associated with the unlabeled data are unknown to the learning process. We assume that the set of classes associated with all labeled data X is a subset of the set of classes associated with all unlabeled data $\tilde{X}$, denoted as $\Omega(X) \subset \Omega(\tilde{X})$. When training with unlabeled data, there is no label available to feed into the loss function, which makes it impossible to compute the update shown in (2). Thus, it is crucial to use unlabeled samples to predict pseudo gradients, represented as $(\partial g|_{\tilde{x}_i} / \partial \theta)$, which can then be used to update the model through back-propagation.

IV. Methodology

In this section, we introduce how to train a gradient learner in a CL framework, and how to use the resulting gradient learner to predict gradients of unlabeled data. We also discuss the sampling policy for unlabeled data and the geometric interpretation of the proposed gradient prediction. Fig. 3 shows an overview of the proposed SSCL method, which includes *gradient learning* and *gradient prediction* process.

A. Gradient Learning

In a CL framework, a model is designed to learn the mapping from raw data to the logits that minimize the predefined continual loss. During the training process, at the ith training step or episode, the generated logits z_i with respect to the input x_i is passed to the continual loss ℓ. With the corresponding y_i, $\ell(z_i, y_i)$ is computed to yield the gradient $(\partial \ell / \partial z_i)$. We aim

to compute the pseudo gradient $\bar{g}$ and use it to back-propagate the error and update the parameters θ by the chain rule. They can be mathematically summarized as

$$\text{Forward:} \quad z_i = f_\theta(x_i, t_i) \tag{3}$$

$$\text{Backward:} \quad \frac{\partial \ell}{\partial \theta} = \frac{\partial \ell}{\partial z_i}\frac{\partial z_i}{\partial \theta}, \ \theta \leftarrow \theta - \eta \frac{\partial \ell_{fit}}{\partial \bar{g}|_{\tilde{x}_i}}\frac{\partial \bar{g}_i}{\partial \omega}. \tag{4}$$

When the learning process is fed with unlabeled data $\tilde{x}_i$, it is desirable to have the logits and corresponding gradient so that $\tilde{x}_i$ can straightforwardly fit into the SCL framework. Therefore, we propose a gradient learner h that aims to learn the mapping from the logits z_i to the gradients $(\partial \ell / \partial z_i)$, that is,

$$g_i = h(z_i; \omega) \tag{5}$$

where ω is the parameters of h and g_i is the predicted gradient that is expected to work as $(\partial \ell / \partial z_i)$ for back-propagation.

To guarantee that the predicted gradients can mimic the gradients' efficacy in the learning process, we formulate the fitness of the predicted gradients with respect to the logits as a learning problem. We define the fitness loss function to quantify the effect of the predicted gradients by fitting them back in the loss, i.e.,

$$\ell_{\text{fit}}(z_i, g_i, y_i) = \ell(z_i - \eta g_i, y_i). \tag{6}$$

By observing triplet (z_i, g_i, y_i) at each training step, the minimization of ℓ_{fit} will iteratively update the proposed gradient learner $h(\cdot; \omega)$ through back-propagation. As depicted in (6), the predicted gradients aim to minimize the fitness loss, rather than mimicking the vanilla gradients $(\partial \ell / \partial z)$ in terms of direction and magnitude.

However, the gradients are sensitive in the learning process and a small change in gradients could lead to a divergence of training. To obtain robustly predicted gradients, instead of directly using the output of $h(\cdot; \omega)$ in the fitness loss (6), we reference the magnitude τ_i of the vanilla gradient. With τ_i, the predicted gradient can be accordingly normalized, i.e.,

$$\bar{g}_i = \alpha \tau_i g_i / \|g_i\|, \quad \tau_i = \|\frac{\partial \ell}{\partial z_i}\| \tag{7}$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the proportion of the magnitude of the predicted gradient to τ_i and z_i is generated by (x_i, t_i, y_i). On the other hand, the output of the proposed gradient learner is a gradient that is subtle and crucial to the learning process. To properly update the proposed gradient learner, we apply a simple yet practically useful version of the loss scale technique [58], [59], [60] to the fitness function. Specifically, the left-hand side in (6) is multiplied with a predefined coefficient λ. Finally, the fitness loss is computed with more robust $\bar{g}_i$, that is,

$$\ell_{\text{fit}}\big(z_i, \bar{g}_i, y_i\big) = \lambda \ell\big(z_i - \eta \bar{g}, y_i\big). \tag{8}$$

Once the fitness loss is set, triplet $(z, \bar{g}, y)$ at each training step suffices to fit into the model learning. Specifically, the proposed gradient learner would be updated with $(\partial \ell_{\text{fit}} / \partial \bar{g}_i)$, i.e.,

$$\omega \leftarrow \omega - \hat{\eta} \frac{\partial \ell_{\text{fit}}}{\partial \bar{g}_i}\frac{\partial \bar{g}_i}{\partial \omega} \tag{9}$$

where $\hat{\eta}$ is the learning rate for updating ω. The model learning formed by the fitness loss (6) and the update function (9) is generic and any gradient-based methods, e.g., multilayer perceptron (MLP) [61], deep networks [55], [56], [57], or transformer [62], can be used. Without loss of generality, we use the baseline gradient-based method, i.e., MLP, in this work.

Algorithm 1 Gradient Learning & Prediction

1: **Input**: $(x_i, t_i, y_i) \in D_{tr}$, $\tilde{x}_i$, θ, ω, α, λ, η, $\hat{\eta}$
2: $z_i = f(x_i, t_i; \theta)$
3: $\ell_i = \ell(z_i, y_i)$
4: Compute the gradient w.r.t. z_i, *i.e.* $\frac{\partial \ell_i}{\partial z_i}$
5: Update the model $\theta \leftarrow \theta - \eta \frac{\partial \ell_i}{\partial z_i}\frac{\partial z_i}{\partial \theta_i}$
6: $g_i = h(z_i; \omega)$
7: $\bar{g}_i = \alpha \tau_i g_i / \|g_i\|, \quad \tau_i = \|\frac{\partial \ell}{\partial z_i}, \|$
8: $\ell_{fit} = \lambda \ell(z_i - \eta \bar{g}, y_i)$
9: Compute the gradient w.r.t $\bar{g}_i$, *i.e.* $\frac{\partial \ell_{fit}}{\partial \bar{g}_i}$
10: Update the gradient learner $\omega \leftarrow \omega - \hat{\eta} \frac{\partial \ell_{fit}}{\partial \bar{g}_i}\frac{\partial \bar{g}_i}{\partial \omega}$
11: **if** $\tilde{x}_i$ is not equal to $\varnothing$ **then**
12: $z|_{\tilde{x}_i} = f(\tilde{x}_i, t_i; \theta), \ \ g|_{\tilde{x}_i} = h(z|_{\tilde{x}_i}; \omega)$
13: $\bar{g}|_{\tilde{x}_i} = \alpha \tau_i g|_{\tilde{x}_i} / \|g|_{\tilde{x}_i}\|, \quad \tau_i = \|\frac{\partial \ell}{\partial z_i}\|$
14: $\theta \leftarrow \theta - \eta \bar{g}|_{\tilde{x}_i}\frac{\partial \bar{g}|_{\tilde{x}_i}}{\partial \theta}$
15: **end if**

The process of learning to predict pseudo gradients is described in lines 6–10 in Algorithm 1.

B. Gradient Prediction

To avail the additional unlabeled data in the learning process for better generalizability, the proposed gradient learner h will predict gradients when the learning process is fed with unlabeled data $\tilde{x}$. Given $\tilde{x}_i$, the predicted gradient is computed in a similar way as (5) and (7) describe, but we use $\tau_{i-1} = \|(\partial \ell / \partial z_{i-1})\|$ (i.e., the last labeled sample prior to the nth step), rather than τ_i, as the label of $\tilde{x}_i$ is absent to produce $\|(\partial \ell / \partial z_i)\|$. Once the predicted gradient $\bar{g}|_{\tilde{x}_i}$ is computed, the model can be updated as

$$\theta \leftarrow \theta - \eta \bar{g}|_{\tilde{x}_i}\frac{\partial \bar{g}|_{\tilde{x}_i}}{\partial \theta}. \tag{10}$$

To maintain flexibility in leveraging external unlabeled data, we follow the basic idea of probability theory to presume that unlabeled data are sampled from a distribution. In contrast to the use of labeled data, where we assume all labeled data will be used during the training process, it is possible that no unlabeled data are sampled at some learning steps. In other words, the training process will revert to supervised learning if no unlabeled data is used. Mathematically, it can be formulated as

$$\tilde{x} = \begin{cases} \tilde{x}_i \sim \mathcal{D}_{\tilde{x}}, & \text{if } q < p \\ \varnothing, & \text{otherwise} \end{cases} \tag{11}$$

where q is a random variable following a distribution and p is a predefined threshold. Without loss of generality, we assume the distribution is a standard uniform distribution $\mathcal{U}(0, 1)$. When p is set to 1, it indicates that the learning process will always draw several unlabeled data from a set $\tilde{X}$ of unlabeled data. When p is set to 0, it indicates that the learning process will not draw any unlabeled data. In other words, p manages the transition from SCL to SSCL.

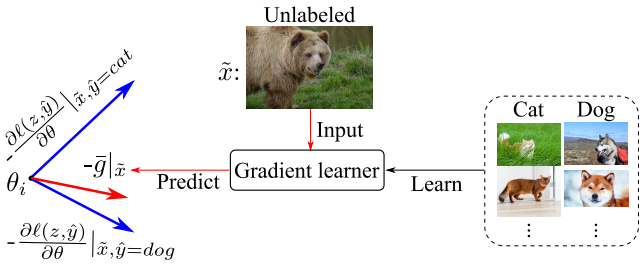


Fig. 4. Comparison between the predicted gradient $\bar{g}$ and the gradients $(\partial \ell(z, \hat{y})/\partial \theta)$ generated with pseudo labels $\hat{y}$. Assume the proposed gradient learner is trained with the samples in categories cat and dog, given an unlabeled image $\tilde{x}$, the proposed gradient learner would take all learned class-specific knowledge (i.e., with respect to categories cat and dog) into account, instead of taking one category (i.e., cat or dog) into account in pseudo labeling methods.

The process of predicting gradients is described in lines 11–14 in Algorithm 1, and the symbols used in the algorithm are depicted in Nomenclature.

C. Connection to Pseudo Labeling

Note that we do not assume that the underlying classes that are associated with the unlabeled data are the same as or similar to the known classes. As a result, the distributions of the unlabeled samples could be very different from the labeled samples. Hence, directly predicting pseudo-label for back-propagation may not be suitable in the SSCL setting.

When labels are unavailable, a common practice to utilize unlabeled samples is by training a teacher model with labeled samples and then predicting pseudo labels on unlabeled samples [11], [19], [63], [64]. Pseudo labeling [11], [19], [64] is viewed as a teacher–student learning framework, i.e.,

$$\underset{\theta'}{\text{minimize}} \quad \ell\left(f^{tch}_{\theta'}(x_i, t_i), y_i\right) \tag{12}$$

$$\hat{y} = \arg\max_j \left[f^{tch}_{\theta'}(\tilde{x}, t_i)\right]_j \tag{13}$$

$$\underset{\theta}{\text{minimize}} \quad \ell\left(f^{stn}_{\theta}(x_i, t_i), \hat{y}\right) \tag{14}$$

where *tch* (respective *stn*) stands for teacher (respective student), θ' (respective θ) are the weights of the teacher (respective student), $((x_i, t_i), y_i)$ is a labeled sample, and $\tilde{x}$ is an unlabeled sample. In short, the teacher would be trained with labeled samples by (12). When it comes across unlabeled sample $\tilde{x}$, the teacher first predicts an one-hot pseudo label $\hat{y}$ by (13) and then $\hat{y}$ is viewed as the label for training the student by (14). A common alternative to one-hot pseudo labels in (13) is the probabilities with respect to each class, which is used in leveraging unlabeled data in the class-incremental learning [11]. To generate probabilistic labels, the softmax function is usually applied. We denote the one-hot pseudo labeling method and the probabilistic pseudo labeling method as 1-PL and P-PL for simplicity.

The difference between gradient prediction and pseudo labeling is shown in Fig. 4. As teacher models have a chance of generating incorrect labels, the resulting gradients would vary with different pseudo-labeling. Instead, the proposed gradient learner is trained with labeled samples so the predicted gradients are generated with implicit knowledge that maps visual appearance to various visual concepts, rather than one. For example, given training samples of a cat and dog, when the proposed gradient learner observes a fox image to predict the pseudo gradient, the pseudo gradient would be aligned with the learned knowledge of both cat and dog, instead of only cat or only dog. Therefore, the pseudo gradients generated by the proposed gradient learner have better generalizability than the ones generated by pseudo-labeling methods. Last but not least, as indicated in (6), the predicted gradients are generated to minimize the fitness loss, while the gradients generated by pseudo labeling methods aim to reproduce the gradients generated with ground-truth labels. Ideally, if the pseudo labels are identical to the ground-truth labels, the gradients generated with pseudo labels would be identical to the gradients generated with ground-truth labels. However, this rarely happens in practice as unlabeled data have no labels or the underlying labels are unknown.

On the other hand, each task in SSCL has a limited number of labeled samples, and the visual concepts of any two tasks are different. With limited labeled samples, it is difficult to predict correct pseudo labels. Thus, predicting pseudo gradients is more straightforward and effective in this case.

Furthermore, pseudo-labeling methods have many more parameters than the proposed gradient learner. Although the outputs of the teacher model and the proposed gradient learner are supposed to be of the same dimension, the inputs are different. The former takes images as input whereas the latter takes CL models' output as input. Thus, the teacher models usually have the same as or more parameters ($> 1M$) than the student models [11], [17], [19], whereas the proposed gradient learner is a small MLP with fewer parameters ($< 10K$).

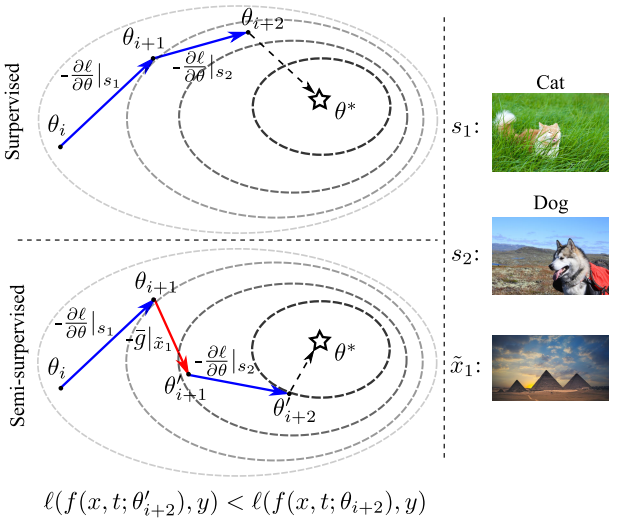


Fig. 5. Geometric interpretation of supervised learning (top) and SSL (bottom). Through leveraging the semantics of unlabeled images, the generalizability of models is expected to be improved. Experimental results in Tables II–VI validate this finding.

D. Geometric Interpretation

Fig. 5 shows the geometric interpretation of gradient prediction by comparing SSL (bottom) with supervised learning (top). In this illustrative example, given two labeled images, s_1 and s_2, and one unlabeled image, $\tilde{x}_1$, the predicted gradient $-\bar{g}|_{\tilde{x}_1}$ helps boost the convergence, i.e., θ'_{i+2} is closer to the underlying local minimum θ^* than θ_{i+2}. This also impacts the generalizability. Given an unseen labeled triplet (x, t, y), we have inequality $\ell(f(x;\theta'_{i+2}), t, y) < \ell(f(x;\theta_{i+2}), t, y)$. This implies that the CL model with pseudo gradients is likely to be closer to a local minimum than the one that is not using it.

E. Tradeoff: Overwhelming Versus Generalizing

It is desirable to use as much unlabeled data as possible, as long as the data improves the generalizability of the CL models. Unfortunately, this goal is difficult to achieve. The reasons are twofold. First, as shown in Fig. 1, since the underlying classes of the unlabeled images are unknown, the distributions of unlabeled data could be considerably different from the ones of the labeled data. Second, gradient learning and prediction are challenging as it is a regression task in a high-dimensional space and the values of gradients are usually small but influential. Last but not least, in contrast to the classification task, where the labels are one-hot vectors that are in [0, 1], the ranges of vanilla gradients are determined by the labeled data and lie in $(-\infty, +\infty)$. Therefore, the gradient learning task is by nature very challenging.

As a result, when more unlabeled data are used in the learning process, it is more prone to accumulate prediction errors that harm the training quality. Specifically, predictive errors in the back-propagation could overwhelm the knowledge learned from the given labeled data. Therefore, achieving a good tradeoff between overwhelming and generalizing is important in the SSCL problem. In this work, we use a probabilistic threshold p to implement this tradeoff.

V. Experiment

A. Experimental Set-Up

We follow the experimental protocols used in GEM [5] and ACL [14], which are cast into the category of task-incremental learning. In the training scheme of [5], the models will observe training samples and no training samples will be observed for a second time. The training and test samples are randomly assigned to n tasks according to classes and each task has training and test samples with different classes from the other tasks. Similarly, [14] randomly assigns the samples into n tasks, but in each task, there are multiple epochs that repeat the stochastic process over the training samples as the conventional supervised learning. After the training on every task is done, the trained models would be evaluated with all the samples of all tasks, including the tasks that the CL process has gone through and the tasks that have not been executed yet. Furthermore, to understand the generalization ability of the proposed method in the SSL task, we follow the experimental protocols used in [20] to evaluate the proposed method.

B. Datasets

In the GEM training scheme, we use the following datasets. **MNIST permutation (MNIST-P)** [25] is a variant of MNIST [65], which consists of 70k images of size 28×28. Each image is transformed by a fixed permutation of pixels. **MNIST rotation (MNIST-R)** [5] is similar to MNIST-P, but each image is rotated by a fixed angle between 0° and 180°. **Incremental CIFAR-100 (iCIFAR-100)** [26] is a variant of the CIFAR-100 [66], which consists of 60k images of size 32×32 that are split into multiple subsets by the classes.

In the ACL training scheme, we use **CIFAR-100** [14] and **miniImageNet** [67]. CIFAR-100 is also split into multiple subsets like iCIFAR-100. Instead of being used once, the images in each task are repeatedly used in every epoch. miniImageNet is a variant of ImageNet [41], which consists of 60k images of size 84×84 with 100 classes.

Following GEM and ACL, all training samples are split into 20 tasks. Briefly, each task on iCIFAR-100, CIFAR-100, and miniImageNet has five classes. For MNIST-P and MNIST-R, each task has ten classes and is performed with different permutations or rotations from the other tasks.

For the experiments on MNIST-R, MNIST-P, iCIFAR-100, and CIFAR-100, we use Tiny ImageNet as the unlabeled dataset. For the experiments on miniImageNet, we use the unlabeled images from MS COCO [68]. Both unlabeled datasets are widely used large-scale real-world datasets. Hence, the unlabeled pool is representative and general for various CL tasks.

In the semi-supervised training scheme, we follow the same experimental protocols used in [20] to evaluate the proposed method on SVHN [69], CIFAR-10, and CIFAR-100 [66]. The numbers of labeled data are 1k, 4k, and 10k for SVHN, CIFAR-10, and CIFAR-100, respectively.

C. Metrics and Methods

To comprehensively validate the performance of the proposed method, we conduct experiments based on the training schemes of GEM and ACL. DCL [15] achieves state-of-the-art performance on MNIST-P, MNIST-R, and iCIFAR-100, and is considered as another baseline in the GEM training scheme.

CL has three key metrics, namely, average accuracy (ACC), backward transfer (BWT), and forward transfer (FWT) [5], i.e.,

$$\text{ACC} = \frac{1}{T}\sum_{i=1}^{T} R_{T,i} \tag{15}$$

$$\text{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1} R_{T,i} - R_{i,i} \tag{16}$$

$$\text{FWT} = \frac{1}{T-1}\sum_{i=2}^{T} R_{i-1,i} - \bar{b}_i \tag{17}$$

where $R_{i,j}$ is the test classification accuracy that is evaluated on the test set of the jth task when training on the ith task, T is the number of tasks, and $\bar{b}_i$ is the test classification accuracy at random initialization at the ith task. Average accuracy indicates the predictive ability of the trained models on all tasks. BWT measures the effect of how learning a task t influences the performance on previous tasks $k < t$. A large negative score is referred to as catastrophic forgetting while a positive score implies that learning new tasks generalizes to previous tasks. Correspondingly, FWT measures the effect of how learning a task t influences the performance of future tasks $k > t$. A positive score implies that learning a task generalizes to future tasks, which is similar to zero-shot learning. In the ACL training scheme, we use the same metrics, i.e., average accuracy and BWT, as [14]. We denote a baseline as *backbone* (if any) *continual algorithm*, e.g., ResNet GEM. Similarly, we denote the proposed method as *backbone* (if any) *continual algorithm* + proposed, which indicates the proposed method is used to leverage the information from unlabeled images. Also, following [11] and [20], we report the performance of 1-PL, P-PL, and MG for the purpose of comparison in the CL setting. Specifically, the teacher model takes images as input to predict pseudo labels when the learning process encounters unlabeled images. The teacher model is composed of the same backbone as the CL model

TABLE I

HYPERPARAMETERS WITH RESPECT TO THE PROPOSED METHOD. BS DENOTES BATCH SIZE OF UNLABELED IMAGES

Setting	Dataset	Method	Backbone	BS	p	α	λ	h_ω
SSCL	MNIST-R	GEM	MLP	4	0.15	0.001	0.30	(64,16)
	MNIST-R	DCL	MLP	4	0.15	0.001	0.30	(64,16)
	MNIST-P	GEM	MLP	4	0.15	0.001	0.50	(64,16)
	MNIST-P	DCL	MLP	4	0.15	0.001	0.49	(64,16)
	iCIFAR-100	GEM	ResNet-18	4	0.30	0.005	2.00	(128,32)
	iCIFAR-100	DCL	ResNet-18	4	0.30	0.005	2.50	(128,32)
	iCIFAR-100	GEM	EffNet-B1	4	0.20	0.005	2.00	(128,32)
	iCIFAR-100	DCL	EffNet-B1	4	0.35	0.005	2.00	(128,32)
	CIFAR-100	ACL	AlexNet	64	0.30	0.001	0.20	(128,32)
	miniImageNet	ACL	AlexNet	64	0.35	0.001	0.15	(128,32)
SSL	SVHN	-	Conv-Large [70]	-	-	0.001	2.00	(128,32)
	CIFAR-10	-	Conv-Large	-	-	0.001	1.00	(128,32)
	CIFAR-100	-	Conv-Large	-	-	0.001	1.00	(128,32)

and a linear transformation layer that generates pseudo labels in each task. In other words, 1-PL and P-PL have many more parameters than the baseline and the proposed method.

D. Hyperparameters and Implementation Details

We use the same training hyperparameters in GEM [5], DCL [15], and ACL [14]. More details can be found in these works or in our code repository. Here, we focus on the hyperparameters that are related to the proposed method. There are five hyperparameters, namely, threshold p, magnitude ratio α, loss scale λ, network architecture h_ω, and batch size of unlabeled images. The hyperparameters with respect to the proposed method used in the SSCL and SSL setting are reported in Table I. In particular, we follow [20] to use the unlabeled data with $p = 1.0$ in SSL.

Specifically, this work follows the same experimental protocol used in [5] and [15] to evaluate the proposed method on MNIST-R, MNIST-P, and iCIFAR-100, while it follows the same experimental protocol used in [14] to evaluate the proposed method on CIFAR-100 and miniImageNet. All hyperparameters that are used with the baselines are used with the proposed method as well.

Without loss of generality, we use MLP as the gradient learner $h(\cdot;\omega)$ (h_ω for short). Assume the gradient is in $\mathbf{R}^m$, we denote (dimension of the first layer output, dimension of the second layer output, ..., dimension of the penultimate layer output) for simplicity. For instance, given $m = 5$, architecture $(64, 16)$ indicates the MLP consists of three layers, the first one is a linear operation with a coefficient matrix of size 5×64, the second one is with a coefficient matrix of 64×16, and the last one is with a coefficient matrix of size 16×5.

Similar to other supervised learning methods, a few learning steps may not be adequate to train a good gradient learner. Hence, the gradient learner is trained from the very beginning, but the predicted gradients are used after 50 learning steps in the GEM and DCL training scheme, and after five learning steps in the ACL training scheme.

Note that restricted to the shared and private module design in ACL [14], which requires a fixed dimension of the input features, the batch size of unlabeled images has to be the same as the batch size of training samples, that is, 64.

E. Generalization Performance

Tables II–IV report the performance of the proposed method with comparison to the compared baselines on MNIST-R, MNIST-P, and iCIFAR-100. EWC [25], iCARL [26], MAS [71], A-GEM [72], LUCIR [30], BiC [73], HAL [74], DER [28], X-DER [29], CFA [75], MutexMatch [32], interpolation-based contrastive learning (ICL) [35], and the glimpse network [76] use the same ResNet backbone. Compared to these baselines, the proposed method achieves higher average accuracy and BWT, e.g., ResNet GEM + proposed. This implies that the proposed method effectively utilizes the information of unlabeled images to improve the predictive ability, alleviate catastrophic forgetting, and enhance zero-shot learning ability. Moreover, the proposed method consistently improves the average accuracy, BWT, and FWT of the baselines with the same backbone, e.g., ResNet GEM reproduced versus ResNet GEM + proposed.

In the ACL setting (i.e., Tables V and VI), the average accuracy and BWT of the baseline are improved by the proposed method. Moreover, the standard deviation with respect to the proposed method over 5 runs is smaller than the corresponding baseline. This implies the proposed method is more stable than the baseline.

On the other hand, 1-PL and P-PL yield lower accuracies than the proposed method. This is because the pseudo labels are likely to be incorrect as the training samples are not

TABLE II

PERFORMANCE ON MNIST-R. ALL METHODS USE MLP AS THE BACKBONE NETWORK [5], [15]. THE PROPOSED GRADIENT LEARNER HAS 1824 PARAMETERS. ACCURACY IS IN (%). THE TOP PERFORMANCE IS HIGHLIGHTED IN BOLD. MG AND PG STAND FOR META GRADIENT [20] AND PREDICTED GRADIENT (PROPOSED), RESPECTIVELY. FT AND COS INDICATE THE FITNESS LOSS AND THE COSINE SIMILARITY LOSS, RESPECTIVELY

Methods	Accuracy	BWT	FWT
EWC [25]	54.61	-0.2087	0.5574
GEM [5]	83.35	-0.0047	0.6521
DCL [15]	84.08	0.0094	0.6423
GEM + 1-PL	74.58	-0.0782	0.6319
GEM + P-PL	79.39	-0.0380	0.6453
GEM reproduced	83.03	-0.0061	0.6482
GEM + MG	84.97	0.0051	0.6552
GEM + proposed	**86.54**	**0.0227**	0.6537
DCL + 1-PL	82.12	0.0022	0.6275
DCL + P-PL	83.34	0.0033	0.6359
DCL reproduced	84.88	0.0088	0.6526
DCL + MG	85.74	0.0168	0.6518
DCL + proposed	86.26	0.0106	**0.6620**

TABLE III

PERFORMANCE ON MNIST-P. ALL METHODS USE MLP AS THE BACKBONE NETWORK [5], [15]. THE PROPOSED GRADIENT LEARNER HAS 1824 PARAMETERS

Methods	Accuracy	BWT	FWT
EWC [25]	59.31	-0.1960	-0.0075
GEM [5]	82.44	0.0224	-0.0095
DCL [15]	82.58	0.0402	-0.0092
GEM + 1-PL	80.61	0.0327	-0.0014
GEM + P-PL	80.58	0.0224	-0.0039
GEM reproduced	82.35	0.0251	-0.0101
GEM + MG	82.30	0.0332	-0.0170
GEM + proposed	82.91	0.0316	-0.0072
DCL + 1-PL	81.57	**0.0479**	**0.0002**
DCL + P-PL	80.95	0.0219	-0.0083
DCL reproduced	82.83	0.0279	-0.0100
DCL + MG	82.48	0.0423	-0.0078
DCL + proposed	**82.97**	0.0402	-0.0038

TABLE IV

PERFORMANCE ON ICIFAR-100. *ResNet* INDICATES RESNET-18. *EffNet* STANDS FOR EFFICIENTNET (B1) [57]. THE PROPOSED GRADIENT LEARNER HAS 4896 PARAMETERS

Methods	Accuracy	BWT	FWT
EWC [25]	48.33	-0.1050	**0.0216**
iCARL [26]	51.56	-0.0848	0.0000
MAS [71]	49.45	-0.0674	0.0157
A-GEM [72]	67.14	0.0037	0.0087
LUCIR [30]	58.71	0.0177	-0.0067
BiC [73]	60.92	-0.0010	-0.0023
HAL [74]	63.85	0.0017	0.0088
DER [28]	65.72	0.0011	0.0053
X-DER [29]	68.32	0.0223	0.0017
CFA [75]	67.41	0.0124	-0.0026
MutexMatch [32]	68.09	0.0156	0.0021
ICL [35]	67.23	0.0084	-0.0012
Glimpse [76]	66.87	0.0169	-0.0031
ResNet GEM [5]	66.67	0.0001	0.0108
ResNet DCL [15]	67.92	0.0063	0.0102
EffNet GEM [15]	80.80	0.0318	-0.0050
EffNet DCL [15]	81.55	0.0383	-0.0048
ResNet GEM + 1-PL	65.44	0.0861	-0.0030
ResNet GEM + P-PL	65.55	0.0511	-0.0033
ResNet GEM reproduced	66.92	0.0132	-0.0048
ResNet GEM + MG	67.24	0.0614	-0.0001
ResNet GEM + proposed	68.74	0.0619	0.0055
ResNet DCL + 1-PL	66.43	0.0765	0.0051
ResNet DCL + P-PL	67.78	0.0704	0.0078
ResNet DCL reproduced	67.55	0.0048	-0.0117
ResNet DCL + MG	66.07	0.0524	0.0184
ResNet DCL + proposed	68.53	0.0574	-0.0038
EffNet GEM + 1-PL	78.33	0.0855	-0.0106
EffNet GEM + P-PL	77.46	0.0535	0.0077
EffNet GEM reproduced	81.44	0.0128	0.0105
EffNet GEM + MG	83.95	0.0294	-0.0256
EffNet GEM + proposed	85.51	0.0219	0.0148
EffNet DCL + 1-PL	77.12	**0.0862**	-0.0160
EffNet DCL + P-PL	76.82	0.0821	0.0097
EffNet DCL reproduced	83.47	0.0266	-0.0185
EffNet DCL + MG	85.06	0.0488	-0.0043
EffNet DCL + proposed	**85.70**	0.0378	0.0017

TABLE V

PERFORMANCE ON CIFAR-100 IN ACL SETTING. THE TRAINING PROCESS IS REPEATED FIVE TIMES, AND THE AVERAGE ACCURACY AND STANDARD DEVIATION ARE REPORTED [14]. ACL USES ALEXNET [55] AS BACKBONE. THE PROPOSED GRADIENT LEARNER HAS 1427 PARAMETERS

Methods	Accuracy	BWT
A-GEM [12]	54.38±3.84	-0.2199±0.0405
ER-RES [13]	66.78±0.48	-0.1501±0.0111
PNN [77]	75.25±0.04	0
HAT [9]	76.96±1.23	0.0001±0.0002
ACL [14]	78.08±1.25	0±0.0001
ACL reproduced	78.17±1.32	**0.01**±0.0168
ACL + proposed	**78.46**±1.05	**0.01**±0.0123

adequate and the visual concepts vary from task to task (the analysis of pseudo labeling is provided in Section VI). Incorrect pseudo-labels lead to the gradients that guide the learning process in unpredictable directions. Note that the BWTs of 1-PL and P-PL are higher than the others. This results from lower accuracy. As indicated in the definition of BWT (16), when the test classification accuracy $R_{i,i}$ on the ith task with the model trained in the ith task is low, it will lead to high BWT. In other words, when overall ACC is high, BWT tends to be relatively low. Similarly, FWT tends to be high (i.e., 0.0216) when the corresponding accuracies over tasks are low (i.e., 48.33%).

TABLE VI

PERFORMANCE ON MINIIMAGENET IN ACL SETTING. THE TRAINING PROCESS IS REPEATED FIVE TIMES, AND THE AVERAGE ACCURACY AND STANDARD DEVIATION ARE REPORTED [14]. ACL USES ALEXNET [55] AS BACKBONE. THE PROPOSED GRADIENT LEARNER HAS 1427 PARAMETERS

Methods	Accuracy	BWT
A-GEM [12]	52.43±3.10	-0.1523±0.0145
ER-RES [13]	57.32±2.56	-0.1134±0.0232
PNN [77]	58.96±3.50	0
HAT [9]	59.45±0.05	-0.0004±0.0003
ACL [14]	62.07±0.51	0±0
ACL reproduced	62.69±1.01	**0**±0.0042
ACL + proposed	**63.88**±0.39	**0**±0.0000

TABLE VII

SEMI-SUPERVISED CLASSIFICATION ERROR RATES (%) OF THE CONV-LARGE [70] ARCHITECTURE ON THE SVHN, CIFAR-10, AND CIFAR-100 DATASETS. THE NUMBERS OF LABELED DATA ARE 1K, 4K, AND 10K FOR THESE THREE DATASETS, RESPECTIVELY. WE FOLLOW THE EXACT EXPERIMENTAL PROTOCOL USED IN [20] AND USE THE OFFICIAL IMPLEMENTATION CODE TO CONDUCT THIS EXPERIMENT. THE META-OBJECTIVE DEFINED IN [20] IS USED AS THE FITNESS LOSS TO LEARN TO PREDICT PSEUDO GRADIENTS FOR UNLABELED IMAGES

Method	SVHN	CIFAR-10	CIFAR-100
Co-training [78]	3.29	8.35	34.63
TNAR-VAE [79]	3.74	8.85	-
ADA-Net [80]	4.62	10.30	-
DualStudent [81]	-	8.89	32.77
MG [20]	3.15	7.78	30.74
MG reproduced	3.53	7.82	30.74
MG + proposed	3.45	7.46	30.02

Since the proposed method is generic, we also evaluate it in the SSL setting [20]. The meta-objective defined in [20] is used as the fitness loss to learn to predict pseudo gradients for unlabeled images. The experimental results on SVHN [69], CIFAR-10, and CIFAR-100 are reported in Table VII. The proposed method can improve the performance of the SSL task. This implies that the proposed method generally works with the unlabeled data with the pseudo labels that share the same or similar distributions as the labeled data.

VI. ANALYSIS

This section provides a series of experiments to analyze the proposed method. All analyses are based on iCIFAR-100.

A. Effects of Visual Diversity

Here, we study the influence of the variance between training images and the unlabeled images on the model performance. In addition to the Tiny ImageNet from Section V-B, we selected a variety of datasets, namely, MS COCO [68], CUB-200 [82], FGVC-aircraft [83], and Stanford-cars [84], as the source of unlabeled images. The classes in these datasets overlap with the ones in CIFAR-100 to various degrees. The

TABLE VIII

EFFECTS OF VISUAL DIVERSITY OF $\tilde{x}$ ON THE CLASSIFICATION PERFORMANCE (%) ON ICIFAR-100 WITH RESNET GEM

Source	Accuracy	BWT	FWT
$\tilde{x} = \varnothing$	66.92	0.0132	-0.0048
Tiny ImageNet [41]	68.74	0.0619	0.0055
MS COCO [68]	67.78	0.0562	0.0006
CUB-200 [82]	68.03	0.0460	0.0041
FGVC-aircraft [83]	67.05	0.0385	0.0159
Stanford-cars [84]	67.41	0.0465	-0.0028

TABLE IX

EFFECTS OF RANDOM NOISE ON THE PERFORMANCE (%) WITH RESNET GEM. THE NOISE FOLLOWS A UNIFORM DISTRIBUTION $\mathcal{U}(-1, 1)$ OR A UNIT NORMAL DISTRIBUTION $\mathcal{N}(0, 1)$, AND IS USED AS PREDICTED GRADIENTS. THE EXPERIMENTAL DETAILS ARE DESCRIBED IN SECTION VI-B

Setting	Accuracy	BWT	FWT
No noise	66.92	0.0132	-0.0048
No noise + proposed	68.74	0.0619	0.0055
$\mathcal{U}(-1, 1)$	54.10	0.1978	-0.0121
$\mathcal{U}(-1, 1)$ + proposed	67.71	0.0533	0.0004
$\mathcal{N}(0, 1)$	45.29	0.2121	0.0032
$\mathcal{N}(0, 1)$ + proposed	67.08	0.0502	0.0007

performance is reported in Table VIII. Overall, the proposed method shows improvement with all unlabeled image sources. The images in Tiny ImageNet are similar to the ones in MS COCO, where both are natural images but have different image resolutions. The resolution of images in Tiny ImageNet is closer to that in CIFAR-100 than MS COCO. Therefore, using Tiny ImageNet images leads to the most performance improvement. In contrast, the images in FGVC-aircraft are the most dissimilar to the ones in CIFAR-100 and the accuracy improvement is marginal. On the other hand, using CUB-200 leads to higher accuracy than using MS COCO. This is because CUB-200 shares similar visual concepts with CIFAR (i.e., bird), and both the two datasets are object-centered, whereas the images of MS COCO contain multiple objects and are nonobject-centered.

B. Using Random Noise as Pseudo Gradients

To evaluate the efficacy of the proposed method, we use random noise as the predicted gradients. The random noise is either generated by a uniform distribution $\mathcal{U}(-1, 1)$ or a normal distribution $\mathcal{N}(0, 1)$. The results with the same experimental set-up as Table IV are shown in Table IX. Specifically, $\mathcal{U}(-1, 1)$ or $\mathcal{N}(0, 1)$ indicates that the corresponding noise is used to replace $\bar{g}|_{\tilde{x}_i}$ (see line 13 in Algorithm 1), while proposed indicates that the corresponding noise is used to replace $g|_{\tilde{x}_i}$ (see line 12 in Algorithm 1) and they will be the input to the equations in line 13 in Algorithm 1. As shown, $\mathcal{U}(-1, 1)$ or $\mathcal{N}(0, 1)$ produces much lower accuracy than the other settings. Note that the random noise disturbs the training for all the tasks so that the accuracies of preceding tasks are low when computing the BWT scores for the current task. As discussed in Section V-E, this leads to high BWTs, according to the definition of BWT (16).

TABLE X

EFFECTS OF DIFFERENT NUMBERS OF LABELED IMAGES ON ICIFAR-100. *L-Ratio* INDICATES THE AMOUNT OF LABELED IMAGES IN ICIFAR-100 USED FOR TRAINING. ABOUT 20% OF UNLABELED IMAGES ARE SAMPLED FROM TINY IMAGENET. THE SETTING IS THE SAME AS THE ONE USED IN TABLE IV AND RESNET GEM IS USED IN THIS ANALYSIS

Method	L-Ratio	Accuracy	BWT	FWT
MG [20]	20%	48.02	0.0420	0.0033
	40%	60.09	0.0822	-0.0004
	60%	61.40	0.0699	-0.0047
	80%	63.21	0.0541	0.0006
	100%	67.24	0.0614	-0.0001
Proposed	20%	50.38	0.0693	-0.0011
	40%	61.10	0.1045	-0.0039
	60%	61.38	0.0582	0.0107
	80%	64.60	0.0552	0.0003
	100%	68.74	0.0619	0.0055

C. Effects of Number of Labeled/Unlabeled Images

To understand how the numbers of labeled and unlabeled images affect the performance of CL, we conduct an analysis to show the performance of using different amounts (range from 0% to 100%) of labeled and unlabeled images. Without using any unlabeled images, it implies the method is a regular SCL method. We compare our proposed method with Meta-gradient [20] and the results with different amounts of labeled (unlabeled) images are reported in Table X (see Table XI). For results in Table X, we use 20% of unlabeled images for training. An observation is that the performance increases as more labeled images are used for training. On the contrary, using more unlabeled images, which follow very different distributions in comparison to the labeled images, does not always lead to better performance. As discussed in Section IV-E and shown in Fig. 1, the distributions of unknown classes' samples could be very different from the ones of known classes' samples. Therefore, using more unlabeled images of the unknown classes would lead to a performance drop.

D. How Hyperparameters Range Across Datasets

In this section, we study how key hyperparameters p, α, and λ are robust to the training on different datasets when using the same unlabeled data. Fig. 6 shows the curves of the accuracy with respect to p, α, and λ. Overall, the curves with respect to iCIFAR-100 and MNIST-R are similar to each other. Specifically, as the values of p, α, and λ exceed a certain point, it would lead to a significant drop in accuracy. The hyperparameters used in this work (see Table I) are selected in the optimal range.

E. Ablation Study

As introduced in the experimental setup, the proposed method depends on five hyperparameters. This section shows the corresponding ablation studies and the results are shown in Fig. 7. As discussed in Section IV, p reflects the tradeoff between overwhelming and generalizing. As p increases, the accuracy drops significantly. This is as expected in the earlier discussion. Moreover, we can observe that the architecture of the proposed gradient learner is more critical to the proposed method in terms of accuracy, BWT, and FWT, compared to the other hyperparameters.

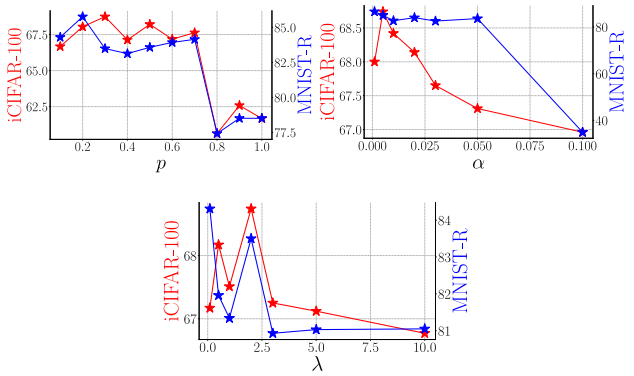


Fig. 6. Effects of p (top), α (middle), and λ (bottom) on accuracy across datasets (i.e., iCIFAR-100 and MNIST-R).

TABLE XI

EFFECTS OF DIFFERENT NUMBERS OF UNLABELED IMAGES ON ICIFAR-100. *U-Ratio* IS THE AMOUNT OF UNLABELED IMAGES IN TINY IMAGENET USED FOR TRAINING. THE SETTING IS THE SAME AS THE ONE USED IN TABLE IV AND RESNET GEM IS USED IN THIS ANALYSIS

Method	U-Ratio	Accuracy	BWT	FWT
MG [20]	0%	66.92	0.0132	-0.0048
	10%	66.43	0.0539	-0.0118
	20%	67.24	0.0614	-0.0001
	40%	67.03	0.0601	-0.0010
	60%	66.95	0.0579	-0.0037
	80%	66.71	0.0536	-0.0059
	100%	66.27	0.0456	0.0018
Proposed	0%	66.92	0.0132	-0.0048
	10%	67.80	0.0525	0.0084
	20%	68.74	0.0619	0.0055
	40%	67.53	0.0630	0.0140
	60%	67.99	0.0644	0.0099
	80%	67.91	0.0581	-0.0021
	100%	66.96	0.0573	0.0000

F. Training Loss, Validation Accuracy, and Fitness Loss

The losses and accuracy against tasks are shown in Fig. 8. As shown, the proposed method can improve the predictive ability of CL models, i.e., EfficientNet GEM and EfficientNet DCL, when unlabeled data and corresponding predicted gradients are used. The loss is decreased and the accuracy is increased. On the bottom row, the curves of the fitness loss versus task show that the fitness loss (6) across tasks is minimized by the proposed gradient learner.

G. Pseudo Labeling Versus Gradient Prediction

This section examines how pseudo labeling and gradient prediction work in the CL method. Moreover, we investigate the correlations between the gradients generated by various methods and its performance.

To understand the efficacy of pseudo-labeling methods, we first inspect the gradients generated with pseudo-labels and the accuracy of pseudo-label prediction on training samples. We take the gradients generated with ground-truth labels as a reference and compute the cosine similarity $\cos((\partial\ell/\partial z)|_{x,t,y}, (\partial\ell/\partial z)|_{x,t,\hat{y}})$ between the gradients generated with two types of labels, where x, t, y are training samples and $\hat{y}$ are pseudo labels. In this way, the discrepancy can be quantified as the cosine similarity. In other words, if pseudo labels are the same as the ground-truth labels, the cosine similarity between the gradients generated with pseudo labels and ground-truth labels should be 1, which indicates the resulting gradients are fully aligned. As shown in Fig. 9, the cosine similarities generated by 1-PL and P-PL are stably around 0.4. The drop from 1 to 0.4 results from the incorrect pseudo labels. The accuracies of 1-PL and P-PL are lower than 1%. The reasons for the low accuracy are twofold. First, in CL, all samples are only observed once and the number of training samples with respect to a class is relatively small, e.g., 500 on iCIFAR-100. Thus, there is not enough data to train a high-performance teacher model. Second, the classes of samples used for training at a task are distinct from those of the other tasks. This dynamic results in the difficulty of training a strong teacher model.

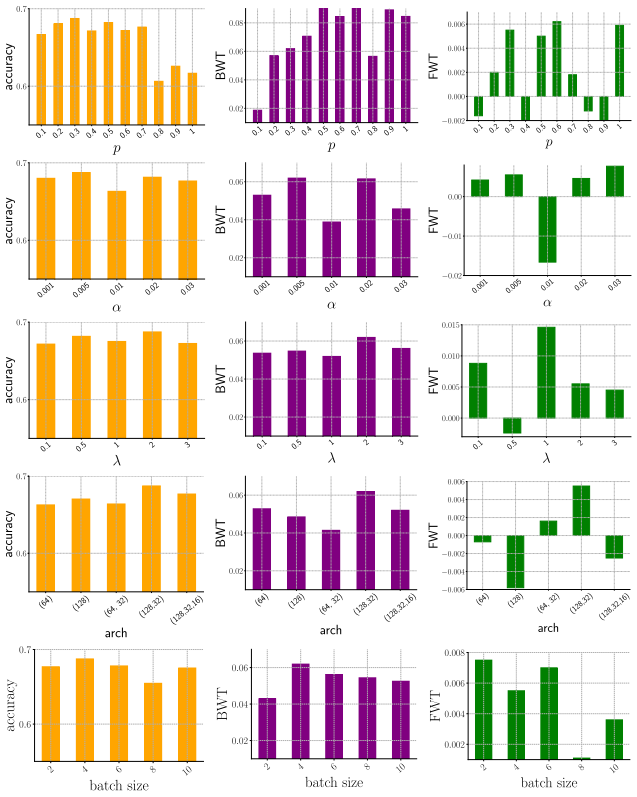


Fig. 7. Ablation study of the proposed method with various hyperparameters detailed in Section V. The metrics are classification accuracy (left), BWT (middle), and FWT (right). ResNet GEM is used for the analysis.

Next, we examine how the gradients generated by various methods correlate with the performance. Note that the predicted gradients aim to minimize the fitness loss (6), while the pseudo labeling methods aim to maximize the similarity between the gradients generated with pseudo labels and ground-truth gradients labels. Hence, the predicted gradients are expected to differ from the ground-truth-generated gradients. As shown in Fig. 10, ResNet GEM + P-PL yields a higher cosine similarity than ResNet DCL + P-PL, but achieves a lower accuracy. In contrast, the proposed method's (i.e., with gradient prediction) accuracy is clearly proportional to the cosine similarity. On the other hand, we observe that discriminative features produced by a strong backbone will lead to better-predicted gradients in terms of the geometric relationship.

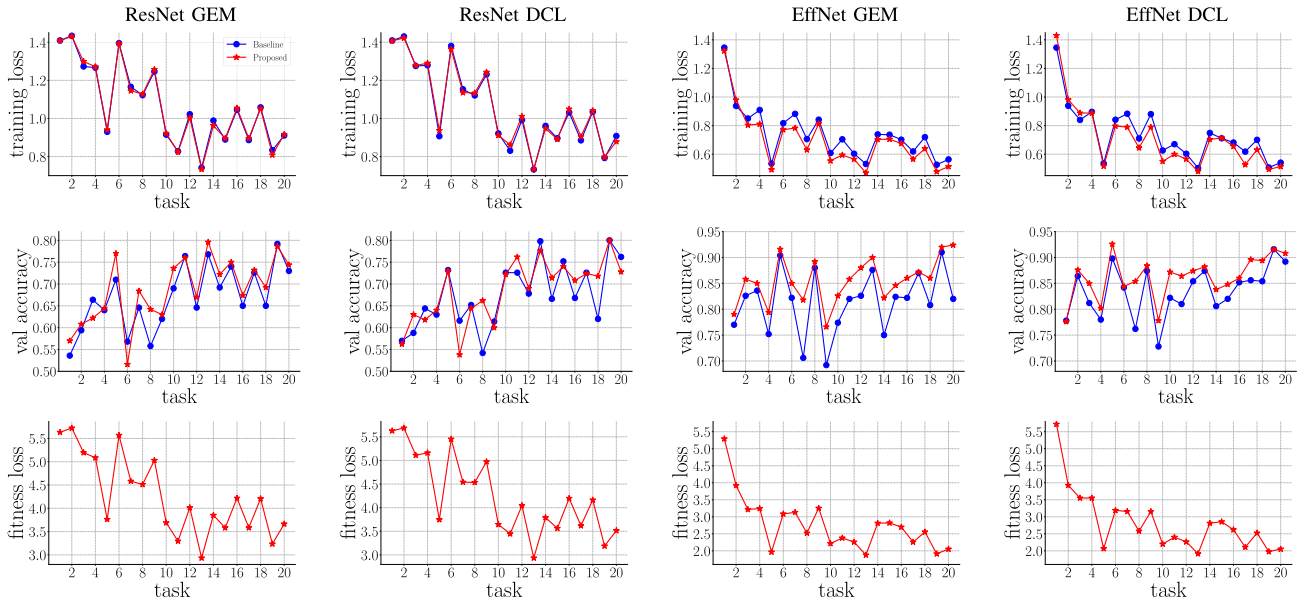


Fig. 8. Plots of the training loss curves (top row), the validation accuracy curves (middle row), and fitness loss curves (bottom row) on iCIFAR-100 with various pairs of methods and backbones.

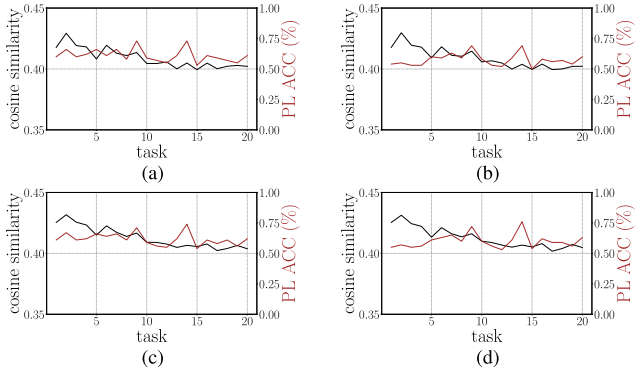


Fig. 9. Plots of cosine similarity between the gradients generated with ground-truth labels and pseudo labels (black curve), as well as the corresponding pseudo label prediction accuracy (brown curve). Due to the lack of training samples and the dynamical change of visual concepts at each task, the pseudo label prediction performs badly (lower than 1%). This is consistent with the drop in the cosine similarity, which should be 1 if the predicted pseudo labels are correct. (a) ResNet GEM 1-PL. (b) ResNet DCL 1-PL. (c) ResNet GEM P-PL. (d) ResNet DCL P-PL.

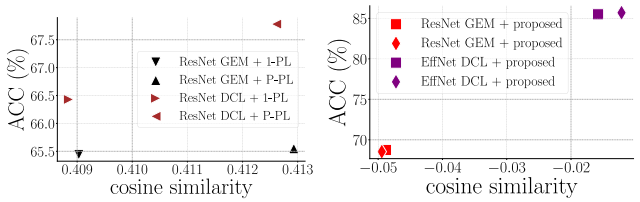


Fig. 10. Correlation between cosine similarity and averaged accuracy. Left: experiments with 1-PL and P-PL. Right: experiments with the proposed method.

TABLE XII
COMPUTATIONAL COMPLEXITY ON ICIFAR-100

Methods	Training Time (ms/Image)	GPU Mem (MB/Image)
ResNet GEM + 1-PL	25	194
ResNet GEM + P-PL	25	194
ResNet GEM	18	193
ResNet GEM + MG	14	194
ResNet GEM + proposed	19	193
ResNet DCL + 1-PL	24	194
ResNet DCL + P-PL	25	194
ResNet DCL	15	193
ResNet DCL + MG	13	193
ResNet DCL + proposed	16	193
EffNet GEM + 1-PL	108	753
EffNet GEM + P-PL	109	753
EffNet GEM	60	756
EffNet GEM + MG	61	750
EffNet GEM + proposed	62	756
EffNet DCL + 1-PL	108	763
EffNet DCL + P-PL	108	763
EffNet DCL	57	759
EffNet DCL + MG	56	757
EffNet DCL + proposed	61	759

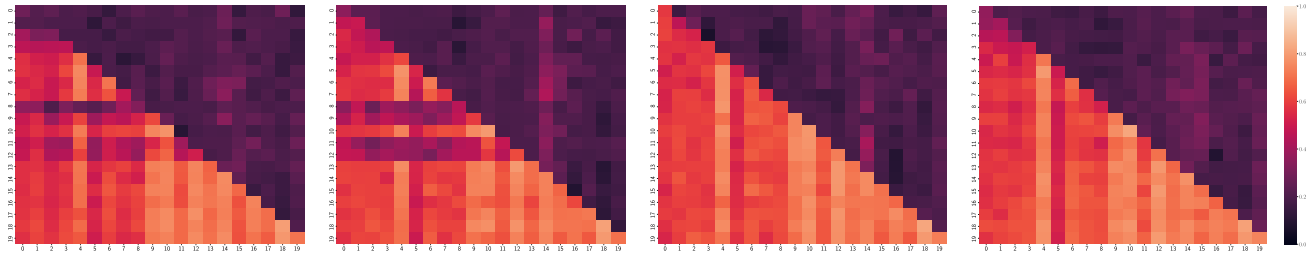
H. Computational Complexity for Training

Table XII reports the runtime and GPU memory for training models. For ResNet models, the training time per image ranges from 13 to 25 ms, with MG being the fastest and P-PL being the slowest. For EfficientNet models, training is generally slower, ranging from 56 to 109 ms per image. GPU memory usage per image is around 190–200 MB for ResNet models and 750–760 MB for EfficientNet. There is little difference between methods. In general, MG and the proposed method are faster than the other methods (i.e., baseline, 1-PL, and P-PL). In particular, the proposed gradient learner method has comparable speed to MG for both ResNet and EfficientNet models, while using slightly less memory.

VII. CONFUSION MATRIX

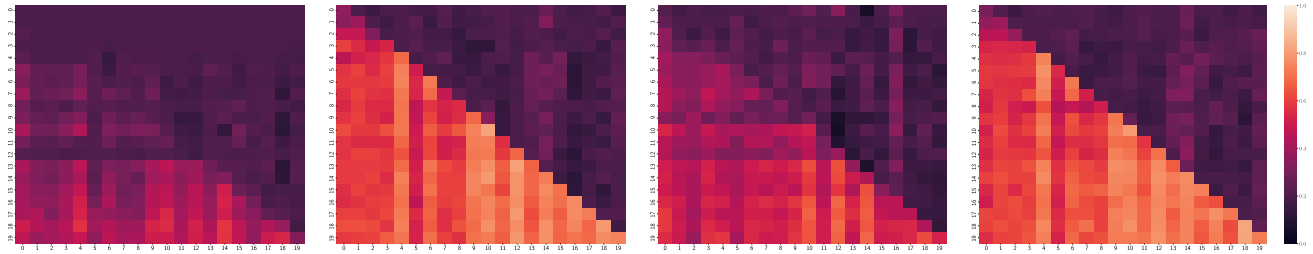
To comprehensively understand the efficacy of the proposed predicted gradients, we visualize the confusion matrices generated by various methods on iCIFAR-100 in Figs. 11 and 12. The ith row of the confusion matrix indicates the test classification accuracies over 20 tasks with the model trained on the ith task. Similarly, the jth column indicates the results are evaluated on the test set of the jth task.

As shown in Fig. 11, leveraging extra unlabeled images with the proposed method will have lower accuracies on early



(a) (b) (c) (d)

Fig. 11. Confusion matrix generated by different methods. The ith row indicates it is the training stage with the ith task's samples, while the ith row jth column indicates the model trained with the ith task's samples is evaluated on the jth task. ResNet GEM is used for the analysis. (a) 1-PL. (b) P-PL. (c) Baseline. (d) Proposed.



(a) (b) (c) (d)

Fig. 12. Confusion matrix generated with random noise. The experimental details are described in Section VI-B. (a) $\mathcal{N}(0, 1)$. (b) $\mathcal{N}(0, 1)$ + proposed. (c) $\mathcal{U}(-1, 1)$. (d) $\mathcal{U}(-1, 1)$ + proposed.

tasks than the baseline as the proposed gradient learner does not have sufficient training samples for learning. With more and more training samples being observed, better-predicted gradients are produced to improve the performance of late tasks. In addition, the accuracies of 1-PL and P-PL are overall lower due to the disturbance caused by the incorrect pseudo labels. As discussed in Section V-E, low $R_{i,i}$ leads to a high BWT score.

Table IX shows that using random noise as predicted gradients yields higher BWT than the other settings. Again, this is because the random noise disturbs the learning process, which leads to low accuracies (see Fig. 12). More importantly, Fig. 12 shows that random noise + proposed is more robust than the method with only random noise.

VIII. Conclusion

In this work, we study how to exploit the semantics of the unlabeled data to improve the generalizability of CL methods. Existing semi-supervised (continual) learning presumes that the labels associated with unlabeled data are known to the learning process. We relax the constraint, i.e., the labels associated with unlabeled data could be known or unknown to the learning process. Correspondingly, we propose a new SSCL method, where a novel gradient learner is trained with labeled data and utilized to generate pseudo gradients when the input label is absent. The proposed method is evaluated in the CL and ACL settings. The experimental results show that the average accuracy and BWT are both improved by the proposed method and achieve state-of-the-art performance. This implies that utilizing the semantics of the unlabeled data improves the generalizability of the model and alleviates catastrophic forgetting. Last but not least, we provide empirical evidence to show that the proposed method can generalize to the SSL task.

Acknowledgment

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

References

[1] M. B. Ring, "Continual learning in reinforcement environments," Ph.D. dissertation, Dept. Comput. Sci., Univ. Texas Austin, Austin, TX, USA, 1994.

[2] S. Thrun, "A lifelong learning perspective for mobile robot control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Jan. 1994, pp. 23–30.

[3] J. C. Schlimmer and D. Fisher, "A case study of incremental concept induction," in *Proc. AAAI*, vol. 86, 1986, pp. 496–501.

[4] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Jan. 1989.

[5] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6467–6476.

[6] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2990–2999.

[7] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[8] M. Riemer et al., "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[9] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 4555–4564.

[10] S. Sinha, S. Ebrahimi, and T. Darrell, "Variational adversarial active learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5972–5981.

[11] K. Lee, K. Lee, J. Shin, and H. Lee, "Overcoming catastrophic forgetting with unlabeled data in the wild," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 312–321.

[12] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with a-GEM," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[13] A. Chaudhry et al., "On tiny episodic memories in continual learning," in *Proc. Workshop Multi-Task Lifelong Reinforcement Learn. ICML*, 2019.

[14] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 12356, 2020, pp. 386–402.

[15] Y. Luo, Y. Wong, M. Kankanhalli, and Q. Zhao, "Direction concentration learning: Enhancing congruency in machine learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1928–1946, Jun. 2021.

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[17] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.

[18] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Workshop Challenges Represent. Learn. ICML*, 2013, vol. 3, no. 2.

[19] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves ImageNet classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10687–10698.

[20] X. Zhang, H. Jia, T. Xiao, M.-M. Cheng, and M.-H. Yang, "Semi-supervised learning with meta-gradient," in *Proc. AISTATS*, 2021, pp. 73–81.

[21] S. Thrun, "Lifelong learning algorithms," in *Learning To Learn*. Boston, MA, USA: Springer, 1998, pp. 181–209.

[22] L. Bottou and Y. Cun, "Large scale online learning," in *Advances in Neural Information Processing Systems*, vol. 16, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA, USA: MIT Press, 2004.

[23] S. C. Hoi, J. Wang, and P. Zhao, "LIBOL: A library for online learning algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 15, pp. 495–499, 2014.

[24] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," in *Proc. NeurIPS Continual Learn. Workshop*, 2018.

[25] K. James, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[26] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[27] A. Chaudhry, N. Khan, P. Dokania, and P. Torr, "Continual learning in low-rank orthogonal subspaces," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020.

[28] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: A strong, simple baseline," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020. Accessed: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin.

[29] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, and S. Calderara, "Class-incremental continual learning into the extended der-verse," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5497–5512, Apr. 2023.

[30] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.

[31] R. Fierimonte, S. Scardapane, A. Uncini, and M. Panella, "Fully decentralized semi-supervised learning via privacy-preserving matrix completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2699–2711, Nov. 2017.

[32] Y. Duan et al., "MutexMatch: Semi-supervised learning with mutex-based consistency regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, pp. 1–15, 2022.

[33] Y. Cui, W. Deng, H. Chen, and L. Liu, "Uncertainty-aware distillation for semi-supervised few-shot class-incremental learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, pp. 1–14, 2023.

[34] P. Zhu, J. Li, B. Cao, and Q. Hu, "Multi-task credible pseudo-label learning for semi-supervised crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 8, 2023, doi: 10.1109/TNNLS.2023.3241211.

[35] X. Yang, X. Hu, S. Zhou, X. Liu, and E. Zhu, "Interpolation-based contrastive learning for few-label semi-supervised learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, pp. 1–12, 2022.

[36] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Trans. Inf. Theory*, vol. IT-11, no. 3, pp. 363–371, Jul. 1965.

[37] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 5, pp. 1087–1095, Sep. 1994.

[38] V. S. Lokhande, S. Tasneeyapant, A. Venkatesh, S. N. Ravi, and V. Singh, "Generating accurate pseudo-labels in semi-supervised learning and avoiding overconfident predictions via Hermite polynomial activations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11432–11440.

[39] Z. Ren, R. A. Yeh, and A. G. Schwing, "Not all unlabeled data are equal: Learning to weight data in semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.

[40] K. Sohn et al., "FixMatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[42] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[43] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. 19th Int. Conf. Comput. Statist.* Paris, France: Springer, 2010, pp. 177–186.

[44] M. Andrychowicz et al., "Learning to learn by gradient descent by gradient descent," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2016, pp. 3981–3989.

[45] M. Jaderberg et al., "Decoupled neural interfaces using synthetic gradients," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1627–1635.

[46] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep. 1951.

[47] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[48] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, "How to escape saddle points efficiently," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1724–1732.

[49] Y. Carmon, J. C. Duchi, O. Hinder, and A. Sidford, "Accelerated methods for nonconvex optimization," *SIAM J. Optim.*, vol. 28, no. 2, pp. 1751–1772, 2018.

[50] S. Reddi et al., "A generic approach for escaping saddle points," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 1233–1242.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–11.

[52] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning. lecture 6a. overview of mini-batch gradient descent," Coursera, 2012.

[53] L. Luo, Y. Xiong, and Y. Liu, "Adaptive gradient methods with dynamic bound of learning rate," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[54] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: K steps forward, 1 step back," *Proc. Adv. Neural Inf. Process. Syst.*, pp. 9593–9604, 2019.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, vol. 16, 2016, pp. 770–778.

[57] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.

[58] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 794–803.

[59] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[60] R. Zhao, B. Vogel, and T. Ahmed, "Adaptive loss scaling for mixed precision training," 2019, *arXiv:1910.12385*.

[61] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, Germany: Springer, 2009.

[62] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst.*, 2017, pp. 5998–6008.

[63] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, "Learning from noisy labels with distillation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1910–1918.

[64] H. Pham, Z. Dai, Q. Xie, M.-T. Luong, and Q. V. Le, "Meta pseudo labels," 2020, *arXiv:2003.10580*.

[65] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

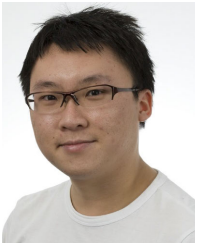
[66] A. Krizhevsky, "Learning multiple layers of features from tiny images," Master's thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.

[67] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3630–3638.

[68] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. ECCV*, in Lecture Notes in Computer Science, vol. 8693, 2014, pp. 740–755.
[69] Y. Netzer et al., "Reading digits in natural images with unsupervised feature learning," in *Proc. NeurIPS Workshop*, Granada, Spain, 2011.
[70] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017.
[71] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 11207. Munich, Germany: Springer, 2018, pp. 144–161.
[72] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Represent.*, 2019.
[73] Y. Wu et al., "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 374–382.
[74] A. Chaudhry, A. Gordo, P. Dokania, P. Torr, and D. Lopez-Paz, "Using hindsight to anchor past knowledge in continual learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 6993–7001.
[75] M. de Carvalho, M. Pratama, J. Zhang, and Y. Sun, "Class-incremental learning via knowledge amalgamation," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science), vol. 13715, M. Amini, S. Canu, A. Fischer, T. Guns, P. K. Novak, and G. Tsoumakas, Eds. Ghent, Belgium: Springer, 2022, pp. 36–50.
[76] J. Li, H. Shi, W. Chen, N. Liu, and K.-S. Hwang, "Semi-supervised detection model based on adaptive ensemble learning for medical images," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, pp. 1–12, 2023.
[77] A. A. Rusu et al., "Progressive neural networks," 2016, *arXiv:1606.04671*.
[78] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. L. Yuille, "Deep co-training for semi-supervised image recognition," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, vol. 11219, 2018, pp. 142–159.
[79] B. Yu, J. Wu, J. Ma, and Z. Zhu, "Tangent-normal adversarial regularization for semi-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10676–10684.
[80] Q. Wang, W. Li, and L. Van Gool, "Semi-supervised learning by augmented distribution alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1466–1475.
[81] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6727–6735.
[82] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-UCSD birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[83] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," 2013, *arXiv:1306.5151*.
[84] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2013, pp. 554–561.

Yan Luo (Member, IEEE) received the B.Sc. degree in computer science from the Xi'an University of Science and Technology, Xi'an, China, in 2008, and the Ph.D. degree from the University of Minnesota (UMN) Twin Cities, Minneapolis, MN, USA, in 2022.

Prior to UMN, he joined the Sensor-Enhanced Social Media (SeSaMe) Centre, Interactive and Digital Media Institute, National University of Singapore (NUS), Singapore, as a Research Assistant. Also, he joined the Visual Information Processing Laboratory, NUS, as a Ph.D. Student. He is currently a Post-Doctoral Research Fellow with the Harvard Ophthalmology AI Laboratory, Harvard University, Boston, MA, USA. He worked in the industry for several years on distributed systems. His research interests include responsible AI and equitable deep learning.



Yongkang Wong (Member, IEEE) received the B.Eng. degree from The University of Adelaide, Adelaide, SA, Australia, in 2006, and the Ph.D. degree from The University of Queensland, Brisbane, QLD, Australia, in 2012.

He worked as a Graduate Researcher at NICTA (Queensland laboratory), Brisbane, from 2008 to 2012. He is a Senior Research Fellow at the School of Computing, National University of Singapore, Singapore. He is also the Assistant Director of the NUS Centre for Research in Privacy Technologies (N-CRiPT). His current research interests are in the areas of image/video processing, machine learning, trusted multimodal analysis, and human-centric analysis.

Mohan Kankanhalli (Fellow, IEEE) received the B.Tech. degree from IIT Kharagpur, Kharagpur, India, in 1986, and the M.S. and Ph.D. degrees from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1988 and 1990, respectively.

He is currently a Provost's Chair Professor of computer science at the National University of Singapore (NUS), Singapore. He is also the Deputy Executive Chairman of AI Singapore Dean of the NUS School of Computing and he also directs the NUS Centre for Research in Privacy Technologies (N-CRiPT) which conducts research on privacy on structured as well as unstructured (multimedia, sensors, and IoT) data. His research interests are in multimedia computing, computer vision, information security and privacy, and image/video processing. He has made many contributions in the area of multimedia and vision image and video understanding, data fusion, and visual saliency as well as in multimedia security content authentication and privacy, and multicamera surveillance.

Qi Zhao (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the University of California, Santa Cruz, Santa Cruz, CA, USA, in 2009.

She was a Post-Doctoral Researcher with the Computation and Neural Systems, California Institute of Technology, Pasadena, CA, USA, from 2009 to 2011. She is currently an Associate Professor with the Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN, USA. Before joining the University of Minnesota Twin Cities, she was an Assistant Professor with the Department of Electrical and Computer Engineering and the Department of Ophthalmology, National University of Singapore, Singapore. She has published more than 100 journals and conference papers in computer vision, machine learning, and neuroscience venues, and edited a book with Springer, titled *Computational and Cognitive Neuroscience of Vision*, that provides a systematic and comprehensive overview of vision from various perspectives. Her main research interests include computer vision, machine learning, cognitive neuroscience, and healthcare.

Dr. Zhao serves as the Program Chair for WACV 22 and an Organizer and/or Area Chair for CVPR and other major venues in computer vision and AI regularly. She serves as an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS.