Parameter Merging with Gradient-Guided Supermasks in Online Continual Learning

Benliu Qiu, Heqian Qiu*, Lanxiao Wang*, Taijin Zhao, Yu Dai, Lili Pan, Hongliang Li*

University of Electronic Science and Technology of China
qbenliu@std.uestc.edu.cn, {hqqiu, lanxiaowang}@uestc.edu.cn, {zhtjww, ydai}@std.uestc.edu.cn,
lilipan@uestc.edu.cn, hlli@uestc.edu.cn

Abstract

Online continual learning (OCL) aims at learning a non-stationary data stream in a way of reading each data sample only once, and hence suffers from the trade-off of catastrophic forgetting and insufficient learning. In this work, we firstly analytically establish relationship between loss functions and model parameters from the Bayesian perspective. Based on our analysis, we subsequently propose a parameter merging method with gradient-guided supermasks. Our method leverages 1-order and 2-order gradient information to construct supermasks that determine the merging weights between the old and new models. Our method performs direct arithmetic operations on parameters to update models, beyond traditional gradient descent. We further discover that a widely-used premise that 1-order gradients can be negligible is invalid in OCL, due to slow convergence incurred by insufficient learning. Additionally, we utilize a dual-model dual-view distillation strategy that can align output distributions of the new and merged models for each sample, further enhancing model performance. Extensive experiments are conducted on four benchmarks in OCL settings, including CIFAR-10, CIFAR-100, Tiny-ImageNet, and ImageNet-100. Experimental results demonstrate that our method is effective, and achieves a substantial boost over previous methods.

1 Introduction

Continual learning (CL) empowers machines with human-like capability of learning and memorizing knowledge from a gradually observed data stream. Therefore, it has extensive prospects in real-world applications, e.g. autonomous driving (Ma et al. 2024), embodied AI (Gao et al. 2024), etc. According to availability of task identity and overlap of classes between different tasks, continual learning can be further divided into task-incremental, domain-incremental and class-incremental scenarios (van de Ven, Tuytelaars, and Tolias 2022). Currently, most CL methods learn a model on training data for multiple epochs, i.e. offline training, and hence usually suffer more from forgetting learned knowledge. However, online training is more common and practical in realistic applications, in which data arrive sequentially and each sample is accessed for training only once, posing great challenge for continual learning. For online continual learning (OCL), both retaining past knowledge and assimilating new knowledge are necessitated to achieve strong performance on all observed tasks. In this work, we focus on the most challenging online class-incremental learning scenario.

Existing literatures mostly address OCL from the perspective of rehearsing exemplars of past tasks (Chaudhry et al. 2019; Caccia et al. 2022). This rehearsal strategy selects a subset of training exemplars to be stored in memory, and adopts them during the learning process of a new task. Apart from exemplars, intermediate representations and prototypes can also be buffered for rehearsal (Buzzega et al. 2020; Wei et al. 2023). Some works leverage knowledge distillation (Gou et al. 2021) to alleviate imbalance problem (Guo, Liu, and Zhao 2023) or maximize mutual information (Guo, Liu, and Zhao 2022). Data augmentation combined with contrastive learning is also adopted to learn representative features, and new augmentation techniques (Zhu et al. 2021; Guo, Liu, and Zhao 2022) are proposed specially for sufficiently exploiting information contained in exemplars. To enhance the informativeness of each exemplar, Gu et al. (2024) made use of dataset condensation (Yu, Liu, and Wang 2024) that can reduce the number of stored exemplars. However, these prior methods focus on utilizing data or representations to indirectly affect parameter update while ignore the direct use of model parameters, although knowledge is intuitively stored in parameters. A natural question is: *Is there an intuitive way to incorporate knowledge via directly integrating parameters in OCL?* Furthermore, due to the reliance on data or representations, the prior methods implicitly require multi-step local gradient descent to recover from the transient forgetting (Lange, Ven, and Tuytelaars 2022), facing more challenge in restricted online scenarios.

In this paper, we explore direct parameter merging to incorporate old and new knowledge, which are respectively stored in parameters of the old and new models. Specifically, we firstly derive the posterior probability of model parameters from the perspective of Bayes' optimization, and further establish an optimization objective of OCL, which is minimizing a cumulative loss that is calculated based on data and parameters. Next, we make use of the Taylor expansion of parameters to approximate the cumulative loss, since the cumulative loss of OCL can be viewed as a function of model parameters. This expansion decomposes the cumulative loss into more fine-grained terms, and reveals

*Corresponding authors

the underlying relationship between model parameters and the cumulative loss, which will guide our design of parameter merging operations. Subsequently, based on the loss expansion, we propose a method of parameter merging that infuses 1-order and 2-order gradient information via supermasks, i.e. masks on top of parameters. Our parameter merging is implemented by twice linear interpolations according to the relationship between parameters and loss functions built by the cross-task linearity. In prior works on continual learning, the 1-order gradients are usually negligible because model parameters lie in a stable state after multi-epoch training. However, we discover that 1-order gradients cannot be ignored in OCL because of their larger norm values than 2-order gradients, and omitting them will make obvious adverse effect on model capacity of learning new knowledge. Furthermore, for the optimization objective, apart from a commonly-used cross-entropy loss, we leverage a dual-model dual-view distillation strategy to align output distributions of the new and merged models over original and augmented image pairs. Extensive experiments are conducted on CIFAR-10, CIFAR-100, Tiny-ImageNet, and ImageNet-100 with various memory sizes. Results indicate that the performance of our method surpasses compared methods by a substantial margin, and further analyses demonstrate the effectiveness of the proposed method.

The contributions of this work are summarized as follows:

- A theoretical analysis on OCL is provided from the perspective of Bayesian learning to establish the relationship between loss functions and model parameters, guiding the design of our method for OCL.
- We propose a parameter merging method for OCL, which incorporates old and new models on the parameter-wise level, with gradient-guided supermasks that inherit information contained on 1-order and 2-order gradients.
- We discover that a common practice in continual learning that omitting 1-order gradients of loss over parameters is not valid under online setting, because of slower convergence speed arising from insufficient one-pass training.
- Extensive experiments on four widely-used OCL benchmarks validate that the proposed method is effective and achieves significant improvement over the prior methods.

2 Related Works

Continual Learning

Continual learning (Delange et al. 2022) enables a model to learn gradually from a sequence of tasks. The mainstream CL methods can be categorised into three strategies: rehearsal, regularization and dynamic architecture. Rehearsal strategy stores a portion of exemplars from previous tasks in a memory buffer or a generative model, and reuses them in the training process of a current task (Rolnick et al. 2019). Regularization strategy constrains variation of important model parameters or intermediate outputs in order to retain previous knowledge (Kirkpatrick et al. 2017). Dynamic architecture strategy expands or isolates network parameters for each task (Wortsman et al. 2020). Among them, rehearsal strategy is the most popular one due to its simplicity and effectiveness. Recently, a series of works about mode connectivity (Garipov et al. 2018; Mirzadeh et al. 2021; Zhou et al. 2024) provides new possibility of fusing multiple models trained from different tasks. Matena and Raffel (2022) proposed to weight parameters of multiple models via fisher information and average them, to obtain a unified multi-task model. Li et al. (2025) explored a closed-form optimal model merging in continual learning, which adaptively balances stability and plasticity.

Online Continual Learning

Online continual learning is a more practical setting in realistic applications, where each sample is learned for just one pass. This online setting satisfies the demand of privacy protection and real-time processing. Most continual learning methods are devised for multi-epoch training, i.e. offline setting, and hence have inferior performance for addressing OCL. Existing OCL methods primarily adopt the rehearsal and regularization strategies, and achieve noticeable performance. ER (Rolnick et al. 2019) pointed out that storing a few samples of previous tasks and rehearsing them in the current task can achieve favorable performance. On the basis of ER, DER++ (Buzzega et al. 2020) saved extra logits for regularizing training loss when learning a new task. OCM (Guo, Liu, and Zhao 2022) learned more holistic features and preserved the previously learned knowledge via maximizing mutual information. DVC (Gu et al. 2022) replayed samples interfered most by new samples, and obtained consistent representations by maximizing mutual information between dual views of each image. ERACE (Caccia et al. 2022) modified the conventional cross-entropy loss by separating losses on the incoming stream and the stored data in memory. GSA (Guo, Liu, and Zhao 2023) recognized the gradient imbalance on logits, and proposed a gradient-based self-adaptive loss to mitigate it. PCR (Lin et al. 2023) replayed old samples and calculated contrastive loss using proxies instead of anchors. CCLDC (Wang et al. 2024) utilized collaborative learning and distillation chain to improve model plasticity. MOSE (Yan et al. 2024) expanded the network architecture with auxiliary branches, and leveraged multi-level supervision and reverse self-distillation to address the overfitting-underfitting dilemma. S6MOD (Liu et al. 2025) was built on selective state space model and class-conditional mixture of discretization, and developed a class-conditional routing strategy to balance model stability and plasticity. Recently, multi-objective optimization (Wu et al. 2024) and equiangular tight frame (He et al. 2024; Seo et al. 2024; Liu et al. 2025) were also explored in OCL.

3 Methodology

Preliminary

Our OCL protocol adopts the class-incremental scenario. The training data $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T}$ contain a sequence of T tasks, each of which owns data $\mathcal{D}_t = \{(\boldsymbol{x}_i^t, y_i^t)\}_{i=1}^{N_t}$ where $\boldsymbol{x}_i^t$ is an image, y_i^t is the corresponding label, and N_t is the number of samples. We use $\mathcal{C}_t$ to represent the task-specific classes, and $y_i^t \in \mathcal{C}_t$. Due to no class overlap among tasks in class-incremental scenario, we have $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$.

Our model is denoted as $F(x;\boldsymbol{\theta})$ with parameter set $\boldsymbol{\theta}$. A rehearsal-based method usually utilizes a memory buffer $\mathcal{M}$ to store a few samples of seen classes. During the training process of the current task, a small batch of data $\mathcal{B}_m$ is sampled from $\mathcal{M}$, and another small batch of new data $\mathcal{B}_s$ is fetched from the data stream. The model is trained jointly with data $\mathcal{B}_s \cup \mathcal{B}_m$ for each update step. The data stream of all learned tasks up to the task t is denoted as $\mathcal{D}_{1:t} = \bigcup_{i=1}^{t} \mathcal{D}_i$. Note that each sample is trained for one pass, i.e. one-epoch training, to satisfy the requirement of online setting. We hence have $\mathcal{D}_{1:t} = \{\mathcal{B}_s\}_{s=1}^{s_t} = \mathcal{B}_{1:s_t}$, where s_t is the total number of batches for the entire learning process with t tasks. The ultimate optimization objective of OCL is to minimize the cumulative loss on all learned tasks:

$$\begin{aligned}\boldsymbol{\theta}^*_{s_t} &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{1:t}(\boldsymbol{\theta};\mathcal{D}_{1:t}) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{1:s_t}(\boldsymbol{\theta};\mathcal{B}_{1:s_t}) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{s_t} \mathcal{L}_i(\boldsymbol{\theta};\mathcal{B}_i). \qquad (1)\end{aligned}$$

All data $\mathcal{B}_{1:s_t}$ are not accessible arbitrarily, but seen one by one, to meet the limitation of continual learning.

Parameter Merging via Supermasks

In this section, we will analytically derive the proposed method for OCL from the perspective of Bayesian learning.

Online continual learning methods aim to achieve model parameters which can maximize the posterior $p(\boldsymbol{\theta}|\mathcal{B}_{1:s})$ after learning s batches. According to the Bayes' rule, the posterior probability is expressed as

$$p(\boldsymbol{\theta}|\mathcal{B}_{1:s}) = \frac{p(\mathcal{B}_{1:s}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{B}_{1:s})}, \qquad (2)$$

where $p(\boldsymbol{\theta})$ and $p(\mathcal{B}_{1:s})$ are the prior probabilities over the parameters and all data learned so far, respectively, and $p(\mathcal{B}_{1:s}|\boldsymbol{\theta})$ is the likelihood probability. After several steps of derivation, we finally can obtain

$$p(\boldsymbol{\theta}|\mathcal{B}_{1:s}) = \frac{p(\mathcal{B}_s|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{B}_{1:s-1})}{p(\mathcal{B}_s|\mathcal{B}_{1:s-1})}. \qquad (3)$$

The detailed derivation is described in Sec. A of supplementary material. This equation establishes a recursive relation between $p(\boldsymbol{\theta}|\mathcal{B}_{1:s})$ and $p(\boldsymbol{\theta}|\mathcal{B}_{1:s-1})$. According to this equation, the posterior probability $p(\boldsymbol{\theta}|\mathcal{B}_{1:s})$ is also formulated as

$$\begin{aligned}p(\boldsymbol{\theta}|\mathcal{B}_{1:s}) &\propto p(\mathcal{B}_s|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{B}_{1:s-1}) \qquad (4)\\ &\propto \prod_{i=1}^{s} p(\mathcal{B}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}), \qquad (5)\end{aligned}$$

where the denominator $p(\mathcal{B}_s|\mathcal{B}_{1:s-1})$ in Eq. 3 is omitted since it is a constant and has no effect on optimization of $\boldsymbol{\theta}$. The optimal parameters $\boldsymbol{\theta}^*_s$ are expected to maximize the posterior probability $p(\boldsymbol{\theta}|\mathcal{B}_{1:s})$. Therefore, we can derive:

$$\begin{aligned}\boldsymbol{\theta}^*_s &= \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{B}_{1:s}) \qquad (6)\\ &= \arg\max_{\boldsymbol{\theta}} \log \left(\prod_{i=1}^{s} p(\mathcal{B}_i|\boldsymbol{\theta})p(\boldsymbol{\theta}) \right) \qquad (7)\\ &= \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{s} \log p(\mathcal{B}_i|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \qquad (8)\\ &= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{s} \mathcal{L}_i(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta}). \qquad (9)\end{aligned}$$

The second last equation is satisfied due to a loss function is commonly expressed as the negative likelihood $-\log p(\mathcal{B}_i|\boldsymbol{\theta})$, and $\log p(\boldsymbol{\theta})$ has negligible effect on the optimization process because of the randomized initialization. During the optimization process, model parameters are evolved along the advent of data stream as $\boldsymbol{\theta}_0 \xrightarrow{\mathcal{B}_1} \boldsymbol{\theta}_1 \cdots \xrightarrow{\mathcal{B}_s} \boldsymbol{\theta}_s$. Eq. 9 validates the rationality of Eq. 1 from the perspective of Bayesian learning.

In the following, we will revisit the cumulative loss $\mathcal{L}_{1:s}(\boldsymbol{\theta})$, and based on its decomposition and the mode connectivity theory (Garipov et al. 2018), we will derive the update approach of model parameters. According to the Taylor expansion of the loss function, we can express $\mathcal{L}_{1:s}(\boldsymbol{\theta}_s)$ as

$$\begin{aligned}\mathcal{L}_{1:s}(\boldsymbol{\theta}_s) &\approx \mathcal{L}_{1:s}(\boldsymbol{\theta}^*_{s-1}) + (\boldsymbol{\theta}_s - \boldsymbol{\theta}^*_{s-1})^\top \nabla_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta}^*_{s-1})\\ &+ \frac{1}{2}(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*_{s-1})^\top \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta}^*_{s-1})(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*_{s-1})\\ &+ \mathcal{O}\left((\boldsymbol{\theta}_s - \boldsymbol{\theta}^*_{s-1})^2\right), \qquad (10)\end{aligned}$$

where $\mathcal{O}\left((\boldsymbol{\theta}_s - \boldsymbol{\theta}^*_{s-1})^2\right)$ denotes the terms with higher order than the second, $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta}^*_{s-1})$ and $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta}^*_{s-1})$ are simplified expressions of the corresponding values of $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{1:s}(\boldsymbol{\theta})$ when $\boldsymbol{\theta} = \boldsymbol{\theta}^*_{s-1}$.

In recent years, researchers establish a theory of mode connectivity that describes relationships of model parameters in the loss landscape (Garipov et al. 2018; Mirzadeh et al. 2021; Zhou et al. 2024). Zhou et al. (2024) proposed the cross-task linearity among model parameters, and confirmed that it often occurs for the finetuned models, which is described as follows:

Definition (Cross-Task Linearity). Given a pair of finetuned models $(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) \in \Theta^2$ and downstream tasks $\mathcal{D}_i$ and $\mathcal{D}_j$, we say them satisfy Cross-Task Linearity (CTL) on $\mathcal{D}_i \cup \mathcal{D}_j$, if $\forall l \in [L], \forall \alpha \in [0,1]$,

$$f^{(l)}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \approx \alpha f^{(l)}(\boldsymbol{\theta}_i) + (1-\alpha) f^{(l)}(\boldsymbol{\theta}_j). \qquad (11)$$

Since the loss is deterministically calculated according to the output of last layer of neural networks, i.e. $f^{(L)}$, the CTL can further lead to the following relation of loss functions:

$$\mathcal{L}(\alpha\boldsymbol{\theta}_i + (1-\alpha)\boldsymbol{\theta}_j) \approx \alpha\mathcal{L}(\boldsymbol{\theta}_i) + (1-\alpha)\mathcal{L}(\boldsymbol{\theta}_j). \qquad (12)$$

The CTL tells us that the linearity of loss functions can be transmitted to the model parameters in the same way. Consequently, from the Taylor expansion of $\mathcal{L}_{1:s}(\boldsymbol{\theta}_s)$, we can establish a corresponding relationship of model parameters, which is similar to the one of loss functions. In other words, we can now decompose a complete neural network into several subnetworks. The decomposition is implemented with the help of supermasks that incorporate information of first-order gradients and second-order gradients. Specifically, we discover that the relationship of model parameters can be

constructed via applying twice the linear interpolation operation that is implied in the cross-task linearity, as follows:

$$\begin{aligned}\boldsymbol{\theta}_s^* &= \left((1-\alpha\mathbf{m}'')\odot\boldsymbol{\theta}_{s-1}^* + \alpha\mathbf{m}''\odot\boldsymbol{\theta}_s\right)\\ &\quad\odot(1-\alpha\mathbf{m}') + \alpha\mathbf{m}'\odot\boldsymbol{\theta}_s\\ &= (1-\alpha\mathbf{m}'')\odot\boldsymbol{\theta}_{s-1}^*\odot(1-\alpha\mathbf{m}')\\ &\quad+\alpha\mathbf{m}''\odot\boldsymbol{\theta}_s\odot(1-\alpha\mathbf{m}') + \alpha\mathbf{m}'\odot\boldsymbol{\theta}_s\\ &= \boldsymbol{\theta}_{s-1}^* + \alpha\mathbf{m}'\odot(\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s-1}^*)\\ &\quad+\alpha(1-\alpha\mathbf{m}')\odot\mathbf{m}''\odot(\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s-1}^*), \end{aligned} \tag{13}$$

where $\odot$ is the Hadamard product, and $\mathbf{m}'$ and $\mathbf{m}''$ are parameter-wise supermasks that incorporate information of 1-order and 2-order gradients, respectively. These two supermasks are calculated as below:

$$\mathbf{m}' = \sigma\left(\frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}_{1:s}(\boldsymbol{\theta}_{s-1}^*)}{\|\boldsymbol{\theta}_{s-1}^*\|_1}\right), \tag{14}$$

$$\mathbf{m}'' = \sigma\left(\frac{\nabla_{\boldsymbol{\theta}}^2\mathcal{L}_{1:s}(\boldsymbol{\theta}_{s-1}^*)}{\|\boldsymbol{\theta}_{s-1}^*\|_2}\right), \tag{15}$$

where $\sigma(\cdot)$ is the sigmoid function. Comparing Eq. 10 and Eq. 13, we can intuitively observe their connection that 1-order and 2-order gradients correspond to 1-order and 2-order supermasks. Prior works are usually built upon a basic premise that the 1-order gradients are near zero and hence can be omitted. In Figure 1, we plot the norm values of 1-order and 2-order gradients of layer-wise model parameters. As shown in Figure 1, the norm values of 1-order gradients are even larger than those of 2-order gradients, indicating that the 1-order gradients are non-trivial for OCL, presumably because of the slow convergence speed caused by one-pass training. Through elementary arithmetic derivation, we can further find that applying Eq. 13 is equivalent to firstly calculating $\hat{\alpha}$ as follows:

$$\hat{\alpha} \leftarrow \alpha\mathbf{m}' + \alpha\mathbf{m}'' - \alpha^2\mathbf{m}'\odot\mathbf{m}'', \tag{16}$$

and then performing the linear interpolation operation once, which is proved as the following:

$$\begin{aligned}\boldsymbol{\theta}_s^* &= (1-\hat{\alpha})\boldsymbol{\theta}_{s-1}^* + \hat{\alpha}\boldsymbol{\theta}_s = \boldsymbol{\theta}_{s-1}^* - \hat{\alpha}(\boldsymbol{\theta}_{s-1}^* - \boldsymbol{\theta}_s)\\ &= \boldsymbol{\theta}_{s-1}^* - \alpha(\mathbf{m}'+\mathbf{m}''-\alpha\mathbf{m}'\odot\mathbf{m}'')\odot(\boldsymbol{\theta}_{s-1}^*-\boldsymbol{\theta}_s)\\ &= \boldsymbol{\theta}_{s-1}^* + \alpha\mathbf{m}'\odot(\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s-1}^*)\\ &\quad+\alpha(1-\alpha\mathbf{m}')\odot\mathbf{m}''\odot(\boldsymbol{\theta}_s-\boldsymbol{\theta}_{s-1}^*).\end{aligned}$$

In regarding to Eq. 15, computing the Hessian matrix $\nabla_{\boldsymbol{\theta}}^2\mathcal{L}_{1:s}$ can be formidable due to the large scale of model parameters. To alleviate the computation cost, we approximate the Hessian matrix using the Fisher information matrix (Kirkpatrick et al. 2017), which is constructed by

$$F_{1:s}(\boldsymbol{\theta}) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}}\log p(\mathcal{B}_{1:s}|\boldsymbol{\theta})\left(\nabla_{\boldsymbol{\theta}}\log p(\mathcal{B}_{1:s}|\boldsymbol{\theta})\right)^\top\right]. \tag{17}$$

Furthermore, $F_{1:s}(\boldsymbol{\theta})$ can be empirically simplified as

$$\tilde{F}_{1:s}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(\nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{x}_i, y_i|\boldsymbol{\theta})\right)^2, \tag{18}$$

where each sample $(\boldsymbol{x}_i, y_i)$ belongs to $\mathcal{B}_{1:s}$ in the implementation of our method.

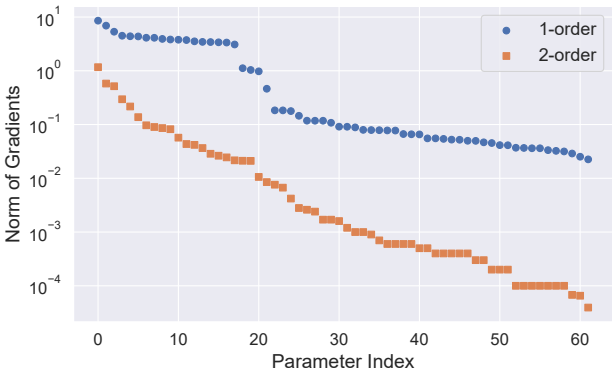

Figure 1: Norm of 1-order and 2-order gradients on parameters in OCL. l_2-norm is used as measure in this experiment.

Optimization Objective

The optimization loss of our method is composed of two parts. One is a common cross-entropy loss computed as:

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{e^{F_{y_i}(\boldsymbol{x}_i;\boldsymbol{\theta})}}{\sum_y e^{F_y(\boldsymbol{x}_i;\boldsymbol{\theta})}}\right), \tag{19}$$

where $F_{y_i}(\boldsymbol{x}_i;\boldsymbol{\theta})$ denotes the output logit of class y_i for the input image $\boldsymbol{x}_i$. Additionally, Wang et al. (2024) and Michel et al. (2024) validated the effectiveness of multi-model and multi-view distillation. In our method, there are two models $\boldsymbol{\theta}_s$ and $\boldsymbol{\theta}_s^*$ whose knowledge can be mutually exchanged, because the former captures new knowledge while the latter mainly memorizes past knowledge. Therefore, we utilize the dual-model dual-view distillation strategy to further boost model performance, which is formulated as the other loss term:

$$\begin{aligned}\mathcal{L}_{kd} = \frac{1}{2N}\sum_{i=1}^{N}\Bigg[&F(\boldsymbol{x}_i;\boldsymbol{\theta}_s^*)\log\frac{F(\boldsymbol{x}_i;\boldsymbol{\theta}_s^*)}{F(\hat{\boldsymbol{x}}_i;\boldsymbol{\theta}_s)}\\ &+F(\hat{\boldsymbol{x}}_i;\boldsymbol{\theta}_s^*)\log\frac{F(\hat{\boldsymbol{x}}_i;\boldsymbol{\theta}_s^*)}{F(\hat{\boldsymbol{x}}_i;\boldsymbol{\theta}_s)}\Bigg],\end{aligned} \tag{20}$$

where $\hat{\boldsymbol{x}}_i = \mathrm{DataAug}(\boldsymbol{x}_i)$. Our final loss is a weighted sum of these two losses with a balance coefficient λ:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{kd}. \tag{21}$$

Our complete method is summarized in Algorithm 1.

4 Experiments

Experimental Setup

Baselines. We primarily compare the proposed method with ER (Chaudhry et al. 2019), DER++ (Buzzega et al. 2020), ERACE (Caccia et al. 2022), GSA (Guo, Liu, and Zhao 2023), OnPro (Wei et al. 2023), OCM (Guo, Liu, and Zhao 2022), CCLDC (Wang et al. 2024), MOSE (Yan et al. 2024), and S6MOD (Liu et al. 2025). Among them, ER and DER++ can tackle both online and offline CL settings, while the rest are specifically devised for OCL. CCLDC and S6MOD are

Dataset	CIFAR-10		CIFAR-100			Tiny-ImageNet		
Memory	500	1000	1000	2000	5000	2000	5000	10000
ER	55.73±2.04	62.99±2.10	23.00±0.80	31.55±1.27	38.05±1.08	11.39±0.75	18.97±1.16	21.52±3.37
DER++	55.53±1.05	58.51±0.68	22.80±1.80	25.89±1.46	25.71±2.40	3.89±0.64	4.28±0.51	4.16±0.32
ERACE	52.65±1.37	61.45±1.47	27.40±0.60	32.88±0.63	39.61±0.53	14.79±0.95	22.25±1.69	26.64±0.91
GSA	61.45±1.95	67.63±1.24	29.68±1.54	36.96±0.79	45.86±1.89	15.77±0.72	22.48±0.4	28.46±1.85
PCR	60.61±2.23	61.66±1.39	30.68±0.81	38.63±1.01	45.27±0.78	12.47±3.56	20.41±2.84	23.85±4.21
OCM	**68.46**±0.79	72.79±2.36	29.30±1.55	36.70±0.58	41.87±1.52	<u>19.58</u>±0.63	27.85±1.03	32.56±1.37
CCLDC	66.43±2.48	<u>74.10</u>±1.71	33.43±1.06	44.45±1.04	<u>53.81</u>±1.16	16.56±1.63	<u>29.39</u>±1.23	37.73±0.85
MOSE	61.02±1.47	70.74±1.18	<u>35.05</u>±0.34	<u>45.06</u>±0.32	**54.53**±0.78	18.23±0.73	30.98±0.63	**38.71**±0.44
S6MOD	57.88±3.30	65.80±2.16	26.50±2.23	34.55±1.66	39.61±3.16	10.94±1.47	19.67±1.36	25.62±1.73
Ours	<u>67.71</u>±1.29	**74.12**±0.38	**38.35**±0.43	**45.56**±0.42	52.19±0.66	**23.59**±0.63	**31.82**±0.25	<u>37.86</u>±0.35

Table 1: Average end accuracy (%) on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets with varied memory sizes. The best results are bold while the second best results are underlined.

Algorithm 1: Parameter Merging via Supermasks

Input: stream data $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T} = \{\mathcal{B}_s\}_{s=1}^{s_T}$, initialized parameters $\boldsymbol{\theta}_0$, balance coefficient λ, learning rate η
Output: updated parameters $\boldsymbol{\theta}_s^*$

Initialize memory $\mathcal{M} \leftarrow \emptyset$, and parameters $\boldsymbol{\theta}_0^* \leftarrow \emptyset$
for $t \in \{1, \cdots, T\}$ **do**
  for $\mathcal{B}_s \sim \mathcal{D}_t$ **do**
    Sample a mini-batch $\mathcal{B} = \mathcal{B}_s \cup \mathcal{B}_m$, $\mathcal{B}_m \sim \mathcal{M}$
    Compute $\mathcal{L}_{ce}$ and $\mathcal{L}_{kd}$ based on Eq. 19 and 20
    Backpropagate and Update parameters:
      $\boldsymbol{\theta}_s \leftarrow \boldsymbol{\theta}_{s-1} - \eta \cdot \nabla_{\boldsymbol{\theta}} \left[\mathcal{L}_{ce} + \lambda \cdot \mathcal{L}_{kd}\right]$
    Compute supermasks $\mathbf{m}'$ and $\mathbf{m}''$ by Eq. 14 and 15
    Compute parameters $\boldsymbol{\theta}_s^*$ according to Eq. 13
    Update memory: $\mathcal{M} \leftarrow \mathrm{Update}(\mathcal{M}, \mathcal{B}_s)$
  end for
end for
return Updated parameters $\boldsymbol{\theta}_s^*$

implemented based on ER. Our method is also built upon ER and follows its memory update strategy.

Datasets. The performance of our method is evaluated on popular image classification datasets, including CIFAR-10 (Krizhevsky and Hinton 2009), CIFAR-100 (Krizhevsky and Hinton 2009), Tiny-ImageNet (Le and Yang 2015), and ImageNet-100 (Deng et al. 2009). We split CIFAR-10 into 5 disjoint tasks with 2 classes per task, CIFAR-100 into 10 disjoint tasks with 10 classes per task, Tiny-ImageNet into 100 disjoint tasks with 2 classes per task, and ImageNet-100 into 20 disjoint tasks with 5 classes per task.

Implementation details. For fair comparison, we follow (Guo, Liu, and Zhao 2022) and use full ResNet18 (He et al. 2016) as the feature extractor and a linear layer as the classifier on all four datasets. Our model is trained from scratch with the Adam optimizer and the learning rate is set to 0.0005. The batch size of $\mathcal{B}_s$ is set to 10 and that of $\mathcal{B}_m$ is fixed as 64, following the guidelines of (Guo, Liu, and Zhao 2022). Data augmentations of our method include random crop, random horizontal flip, color jitter and grayscale, and the baselines retain their original data augmentations. The coefficient λ in Eq. 21 is set to 5.5, and α in Eq. 13 is set to 0.01 by default.

Metrics. In our experiments, we use the average end accuracy (EndACC) to measure the overall performance, the learning accuracy (LACC) to evaluate model capability of fastly learning a new seen task, and the average forgetting measure (FM) to capture the maximal performance decline on previous tasks. Reported results of our method is averaged over 5 runs with different random seeds set to $0, 1, 2, 3, 4$. More specifications of experimental setup are provided in Sec. B of supplementary material.

Main Results

The proposed method is compared with a few typical and state-of-the-art methods in various settings of OCL. Experimental results on CIFAR-10, CIFAR-100, Tiny-ImageNet and ImageNet-100 are reported in Table 1, Table 2 and Table 7 (in Sec.C of supplementary material).

Table 1 shows the average end accuracy of multiple methods with varied memory sizes. As can be observed, our method achieves five best and two second best average end accuracies under the total eight settings. For example, on Tiny-ImageNet, our method surpasses the second best methods by 4.01% and 2.43% with memory size 2000 and 5000, respectively. From the results, we find that MOSE has strong performance due to its expanded structure, and achieves two best performance on CIFAR-100 (M=5000) and Tiny-ImageNet (M=10000). Nevertheless, our method outperforms it on the remaining six settings with gains ranging from 0.5% to 6.69%. Moreover, the reported results in Table 1 suggest that our method can achieve better performance when the memory size is small, indicating that our method does not rely on a sufficient large memory buffer compared to the remaining methods. For example, when the memory size decreases from 2000 to 1000 on CIFAR-100, the performance gap between our method and the second best method MOSE increases from 0.5% to 3.3%. The similar phenomena can also be found on CIFAR-10 and Tiny-ImageNet. This is because our method endows the vision

Dataset	CIFAR-10		CIFAR-100			Tiny-ImageNet		
Memory	500	1000	1000	2000	5000	2000	5000	10000
ER	33.16±3.50	20.94±6.79	32.65±1.78	22.20±2.26	13.29±1.98	58.38±1.69	46.87±1.60	40.77±2.45
DER++	26.79±4.21	18.21±6.32	33.21±4.77	30.86±3.42	30.28±3.90	29.31±1.97	26.98±1.31	26.48±1.79
ERACE	15.25±0.99	11.73±2.71	15.33±0.63	11.13±1.57	6.26±1.50	20.66±0.29	18.93±0.58	16.53±0.52
GSA	21.10±2.17	16.09±0.76	30.24±0.90	19.97±1.66	9.02±0.90	29.03±0.65	25.05±0.68	22.48±0.90
PCR	27.88±4.97	26.53±14.52	34.34±1.03	25.88±0.49	16.09±1.03	38.23±6.45	28.43±4.88	26.40±8.60
OCM	13.68±4.25	11.63±2.62	14.99±1.55	9.16±1.75	3.76±1.16	26.12±1.63	19.74±1.30	15.92±1.47
CCLDC	30.22±3.75	19.85±2.55	43.28±1.67	29.35±1.50	16.88±1.99	69.56±1.54	53.13±0.85	42.63±0.80
MOSE	30.36±1.69	20.27±1.27	37.54±0.43	25.89±0.45	13.60±0.59	47.16±1.41	24.96±0.62	15.51±0.33
S6MOD	31.25±3.44	16.71±2.86	30.96±2.07	19.02±2.13	12.97±2.61	58.03±2.19	45.83±1.49	37.62±1.53
Ours	22.33±1.67	11.88±1.52	16.83±1.15	11.02±0.52	5.65±1.24	23.78±0.41	17.83±0.35	12.27±0.49

Table 2: Average forgetting measure (%) on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets with varied memory sizes.

Method Components				CIFAR-10 (M=500)			CIFAR-100 (M=1000)		
Base	1-order	2-order	KD	EndACC	LACC	Avg.	EndACC	LACC	Avg.
✓				56.71±2.65	83.20±2.54	69.96	23.35±1.25	54.81±1.16	39.08
✓	✓	✓		60.98±0.64	**86.21**±0.88	73.60	28.43±0.32	**57.36**±0.75	42.90
✓	✓		✓	68.04±0.86	78.29±0.59	73.17	38.34±0.31	26.08±1.69	32.21
✓		✓	✓	**68.41**±1.15	79.69±1.25	74.05	**38.64**±0.39	26.06±1.60	32.35
✓	✓	✓	✓	67.71±1.29	85.34±0.65	**76.53**	38.35±0.43	49.57±1.14	**43.96**

Table 3: Ablation study on different components of our method. "Base" is the baseline method - ER; "1-order" and "2-order" represent using supermasks built on 1-order and 2-order gradients, respectively; "KD" denotes the dual-model dual-view distillation strategy. "Avg." denotes the average of EndAcc and LACC.

model with the capacity of memorizing knowledge partially via direct parameter operations, beyond repetitive use of old data stored in memory. In addition, we also conduct experiments on a large-scale dataset ImageNet-100, of which images have higher resolution, as reported in Table 7 of supplementary material. Our method consistently surpasses compared ones with varied memory sizes by 28.70% at most.

Table 2 reports the average forgetting of different methods. This metric reflects the model capacity of retaining past knowledge. Although OCM achieves the lowest average forgetting in most settings, its capacity of learning new knowledge is compromised as revealed in Table 1. Our method obtains the top three best average forgetting on CIFAR-10 and CIFAR-100, primarily behind ERACE and OCM. Nevertheless, our method still performs best on a larger scale dataset Tiny-ImageNet. From Table 2, we observe that the average forgetting of MOSE is particularly severe, nearly double of ours, reflecting that its high average end accuracies benefit more from the ability of learning new knowledge. The similar phenomenon occurs on CCLDC as well. These results validate that our method achieves better balance between retaining past knowledge and learning new knowledge.

In-Depth Studies

In this section, we delve deep into several studies to analyze the effectiveness and contributions of different components and choices made in our method. In addition, we visualize loss landscape and detailed accuracy curve to provide more intuitive comparisons between our method and others. We also study performance of our method in more restricted and realistic conditions, e.g. smaller memory size and blurry boundary between tasks. Some results are reported in Sec. C of supplementary material, due to excess of page limit.

Component analysis. To validate the effectiveness of each component, we conduct an ablation study on CIFAR-10 and CIFAR-100, and the related results are presented in Table 3. It is observed that only adding our supermasks of 1-order and 2-order gradients on the baseline will improve the average end accuracy and simultaneously bring large gain on the learning accuracy, revealing that they improve model capacity of learning new knowledge. On this basis, applying dual-model dual-view distillation can further improve the average end accuracy. This distillation strategy mainly focuses on anti-forgetting, and hence slightly weakens learning capacity. Despite this, applying these components together can achieve a relatively better balance between anti-forgetting and learning new knowledge. These results demonstrate the effectiveness of each component of our method.

Significance of two types of supermasks. To verify that the 1-order and 2-order supermasks are both indispensable, we compare the accuracy matrices on CIFAR-100 in Figure 2. We can observe that using only 1-order or 2-order supermasks will result in compromised accuracies on current new tasks, indicating the model's weak capability of learning new knowledge. However, applying both 1-order and 2-order supermasks will eliminate this negative effect.

λ	0.0	4.0	4.5	5.0	5.5	6.0	6.5	7.0
Ours	28.43±0.32	38.21±0.32	38.14±0.62	38.32±0.27	38.35±0.43	38.26±0.24	38.58±0.20	38.36±0.54

Table 4: Sensitivity analysis on the balance coefficient λ of our method. The experiments are conducted on 10-Split CIFAR-100 with memory size 1000. Average end accuracy is reported as mean±std, which is computed over 5 runs with different seeds.

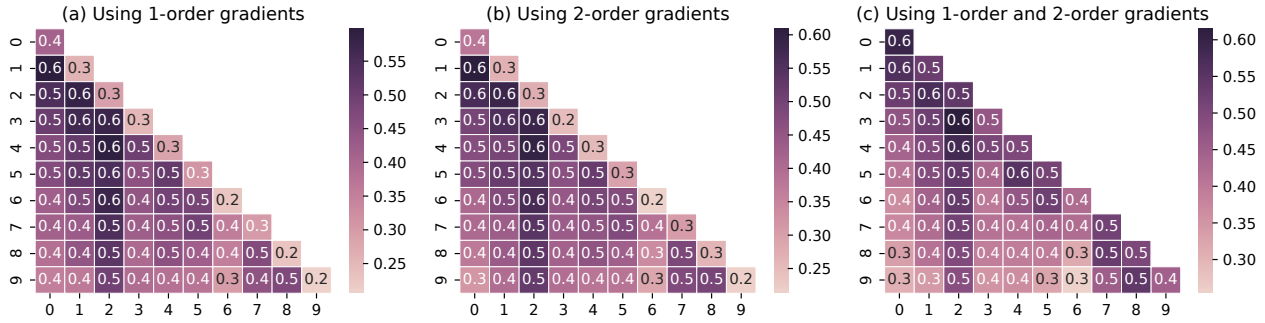

Figure 2: Effect of 1-order and 2-order gradients on accuracies of each task over 10-Split CIFAR-100. The abscissa represents the task step while the ordinate describes the task identifier.

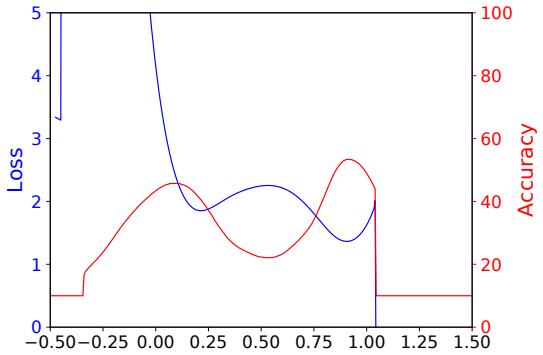

Figure 3: Loss curve and accuracy curve along a linear path between two models separately trained by ER and our method under the OCL setting. The two models are trained on CIFAR-10 with memory size 1000.

This is because both 1-order and 2-order gradients are non-negligible in OCL as revealed by Figure 1, and play an important role in correctly capturing influence of loss on updating model parameters. More results and analyses are reported in Sec. C of supplementary material.

Sensitivity to balance coefficient. In order to analyze the sensitivity to the balance coefficient λ in Eq. 21, we vary its value to record the variation of average end accuracy. This experiment is conducted on CIFAR-100 (M=1000), and the results are reported in Table 4. We can find that setting λ to 0, i.e. no use of dual-model dual-view distillation, will vastly decrease the average end accuracy. Nonetheless, the slight perturbation around our pre-defined value ($\lambda = 5.5$) has negligible effect on model performance, indicating that our method is insensitive to the value of balance coefficient.

Visualizing loss landscape. To study generalization and robustness (Foret et al. 2021), we plot a loss curve and an accuracy curve of a model with the parameters that are obtained by linear interpolation between the baseline model and ours. The curves are depicted in Figure 3, where our model lies at 1.00 whereas the baseline model lies at 0.00. We can observe that the path linearly connecting them will undergo an increase of loss and a prominent drop of accuracy, revealing that the two models are not located in a small local region. Moreover, we visualize loss landscapes of these two models with random perturbed directions in Figure 6 of supplementary material. The loss landscape of our model is flatter than that of the baseline model, implying that the model obtained by our method has greater generalization and robustness. These results indicate that our method substantially achieves a more generalizable and robust solution.

5 Conclusions

In this paper, we study the online continual learning (OCL). We firstly formulate the optimization objective of OCL based on the Bayesian perspective, and then derive the relationship between loss functions and parameters. Built upon our theoretical analysis, we subsequently propose a parameter merging approach that leverages supermasks guided by 1-order and 2-order gradient information. We further point out that a widely-used premise that 1-order gradients are negligible is invalid in the context of OCL. This is because the model does not converge to a stable state due to insufficient training. Moreover, we utilize an extra dual-model dual-view distillation to enable the entire learning process of our method. Extensive experiments validate that our method is effective and shows superior performance over prior advanced methods. Our method uncovers a theory-grounded approach to decompose neural networks. Benefitting from the constructed relationship between loss and parameters, the basic operations on loss can be transmitted to parameters in an analogous manner. We think it is a promising avenue to explore wider implications of mode connectivity theory.

Acknowledgements

This work was supported by the National Science and Technology Major Project under Grant 2021ZD0112001, the National Natural Science Foundation of China under Grant U23A20286 and Grant 62301121, and the Postdoctoral Fellowship Program (Grade B) of China Postdoctoral Science Foundation under Grant GZB20240120.

References

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and CALDERARA, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *Advances in Neural Information Processing Systems*, volume 33, 15920–15930.

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

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H. S.; and Ranzato, M. 2019. On Tiny Episodic Memories in Continual Learning. In *Proceedings of the International Conference on Machine Learning*.

Delange, 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 Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366–3385.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

Gao, C.; Liu, S.; Chen, J.; Wang, L.; Wu, Q.; Li, B.; and Tian, Q. 2024. Room-Object Entity Prompting and Reasoning for Embodied Referring Expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(2): 994–1010.

Garipov, T.; Izmailov, P.; Podoprikhin, D.; Vetrov, D. P.; and Wilson, A. G. 2018. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. In *Advances in Neural Information Processing Systems*, volume 31.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6): 1789–1819.

Gu, J.; Wang, K.; Jiang, W.; and You, Y. 2024. Summarizing Stream Data for Memory-Constrained Online Continual Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12217–12225.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7442–7451.

Guo, Y.; Liu, B.; and Zhao, D. 2022. Online Continual Learning through Mutual Information Maximization. In *Proceedings of the International Conference on Machine Learning*, 8109–8126. PMLR.

Guo, Y.; Liu, B.; and Zhao, D. 2023. Dealing With Cross-Task Class Discrimination in Online Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11878–11887.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

He, Y.; Chen, Y.; Jin, Y.; Dong, S.; Wei, X.; and Gong, Y. 2024. DYSON: Dynamic Feature Space Self-Organization for Online Task-Free Class Incremental Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23741–23751.

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

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report TR-2009, University of Toronto, Toronto.

Lange, M. D.; Ven, G. M. v. d.; and Tuytelaars, T. 2022. Continual evaluation for lifelong learning: Identifying the stability gap. In *International Conference on Learning Representations*.

Le, Y.; and Yang, X. 2015. Tiny ImageNet Visual Recognition Challenge. CS 231N.

Li, M.; Lu, Y.; Dai, Q.; Huang, S.; Ding, Y.; and Lu, H. 2025. BECAME: Bayesian Continual Learning with Adaptive Model Merging. In *Proceedings of the International Conference on Machine Learning*.

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*, 24246–24255.

Liu, S.; Yang, Y.; Li, X.; Clifton, D. A.; and Ghanem, B. 2025. Enhancing Online Continual Learning with Plug-and-Play State Space Model and Class-Conditional Mixture of Discretization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20502–20511.

Ma, X.; Ouyang, W.; Simonelli, A.; and Ricci, E. 2024. 3D Object Detection From Images for Autonomous Driving: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5): 3537–3556.

Matena, M. S.; and Raffel, C. 2022. Merging Models with Fisher-Weighted Averaging. In *Advances in Neural Information Processing Systems*.

Michel, N.; Wang, M.; Xiao, L.; and Yamasaki, T. 2024. Rethinking Momentum Knowledge Distillation in Online Continual Learning. In *Proceedings of the International Conference on Machine Learning*, 35607–35622.

Mirzadeh, S. I.; Farajtabar, M.; Gorur, D.; Pascanu, R.; and Ghasemzadeh, H. 2021. Linear Mode Connectivity in Multitask and Continual Learning. In *International Conference on Learning Representations*.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T.; and Wayne, G. 2019. Experience Replay for Continual Learning. In *Advances in Neural Information Processing Systems*.

Seo, M.; Koh, H.; Jeung, W.; Lee, M.; Kim, S.; Lee, H.; Cho, S.; Choi, S.; Kim, H.; and Choi, J. 2024. Learning Equiangular Representations for Online Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23933–23942.

van de Ven, G. M.; Tuytelaars, T.; and Tolias, A. S. 2022. Three types of incremental learning. *Nature Machine Intelligence*, 1–13.

Wang, M.; Michel, N.; Xiao, L.; and Yamasaki, T. 2024. Improving Plasticity in Online Continual Learning via Collaborative Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23460–23469.

Wei, Y.; Ye, J.; Huang, Z.; Zhang, J.; and Shan, H. 2023. Online Prototype Learning for Online Continual Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision*, 18764–18774.

Wortsman, M.; Ramanujan, V.; Liu, R.; Kembhavi, A.; Rastegari, M.; Yosinski, J.; and Farhadi, A. 2020. Supermasks in superposition. In *Advances in Neural Information Processing Systems*.

Wu, Y.; Wang, H.; Zhao, P.; Zheng, Y.; Wei, Y.; and Huang, L.-K. 2024. Mitigating Catastrophic Forgetting in Online Continual Learning by Modeling Previous Task Interrelations via Pareto Optimization. In *Proceedings of the International Conference on Machine Learning*, 53892–53908.

Yan, H.; Wang, L.; Ma, K.; and Zhong, Y. 2024. Orchestrate Latent Expertise: Advancing Online Continual Learning with Multi-Level Supervision and Reverse Self-Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yu, R.; Liu, S.; and Wang, X. 2024. Dataset Distillation: A Comprehensive Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1): 150–170.

Zhou, Z.; Chen, Z.; Chen, Y.; Zhang, B.; and Yan, J. 2024. On the Emergence of Cross-Task Linearity in Pretraining-Finetuning Paradigm. In *Proceedings of the International Conference on Machine Learning*.

Zhu, F.; Cheng, Z.; Zhang, X.-y.; and Liu, C.-l. 2021. Class-Incremental Learning via Dual Augmentation. In *Advances in Neural Information Processing Systems*.