Efficient Non-Exemplar Class-Incremental Learning with Retrospective Feature Synthesis

Liang Bai, Hong Song, Yucong Lin, Tianyu Fu, Deqiang Xiao, Danni Ai, Jingfan Fan, Jian Yang

arXiv:2411.01465v1 [cs.CV] 3 Nov 2024

Abstract—Despite the outstanding performance in many individual tasks, deep neural networks suffer from catastrophic forgetting when learning from continuous data streams in real-world scenarios. Current Non-Exemplar Class-Incremental Learning (NECIL) methods mitigate forgetting by storing a single prototype per class, which serves to inject previous information when sequentially learning new classes. However, these stored prototypes or their augmented variants often fail to simultaneously capture spatial distribution diversity and precision needed for representing old classes. Moreover, as the model acquires new knowledge, these prototypes gradually become outdated, making them less effective. To overcome these limitations, we propose a more efficient NECIL method that replaces prototypes with synthesized retrospective features for old classes. Specifically, we model each old class's feature space using a multivariate Gaussian distribution and generate deep representations by sampling from high-likelihood regions. Additionally, we introduce a similarity-based feature compensation mechanism that integrates generated old class features with similar new class features to synthesize robust retrospective representations. These retrospective features are then incorporated into our incremental learning framework to preserve the decision boundaries of previous classes while learning new ones. Extensive experiments on CIFAR-100, TinyImageNet, and ImageNet-Subset demonstrate that our method significantly improves the efficiency of non-exemplar class-incremental learning and achieves state-of-the-art performance.

Index Terms—Class-incremental learning, non-exemplar, multivariate Gaussian sampling, feature compensation.

I. INTRODUCTION

AUTOMATIC visual recognition based on deep neural networks is a fundamental task in computer vision that has undergone substantial advancements in recent years [1]–[3]. However, most existing methods assume a static scenario, where the entire training process occurs in a single session with all training data fully available. In contrast, real-world scenarios are dynamic, with new data continuously emerging, requiring continuous updates to the model. Therefore, it is essential to develop efficient visual recognition algorithms that can learn continuously, much like the human ability to acquire new concepts without forgetting previously learned knowledge.

Class incremental learning [6]–[8] is a real-world scenario that considers adding new classes to an already trained classi-

Corresponding authors: Hong Song and Jian Yang.

Liang Bai and Hong Song are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China.

Yucong Lin, Deqiang Xiao, Danni Ai, Jingfan Fan, and Jian Yang are with the School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China.

Tianyu Fu is with the School of Medical Technology, Beijing Institute of Technology, Beijing 100081, China.
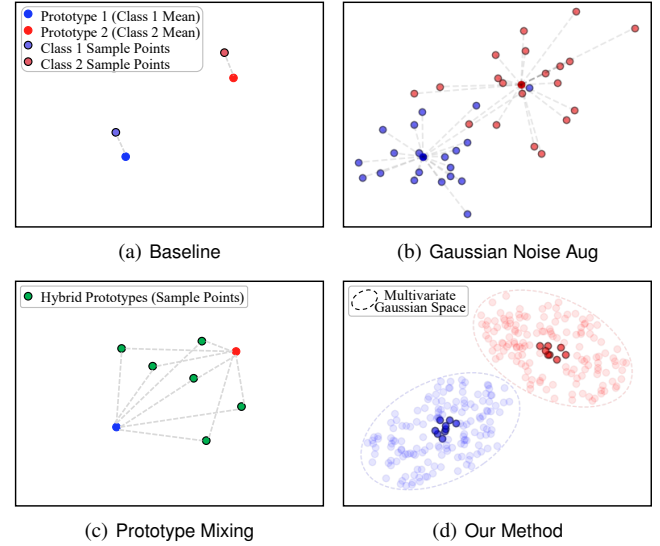

Fig. 1. Idea illustration. (a) Baseline: Use the stored prototypes as the classification features of the old classes to rebuild the classification boundaries. (b) Gaussian Noise Aug [4]: Use random Gaussian noise to enhance the prototypes, but it introduces inaccurate features and overlaps between class distributions due to noise interference to the representation space. (c) Prototype Mixing [5]: Combine the prototypes of the old classes randomly linearly to form the hybrid prototypes, but these obtained prototypes lack authenticity. (d) Our Method: Model the representation space of each old class using a multivariate Gaussian distribution, then generate classification features for old classes by sampling from the high-likelihood regions, thereby better capturing the diversity and accuracy of the generated representations.

fication model. This approach aims to mitigate the catastrophic forgetting that neural networks underfit previously learned classes while absorbing new class data [9]. Catastrophic forgetting typically arises due to the overlap or confusion between old and new class representations within the feature space [4]. As a large number of new class instances are introduced, the unified classifier is biased to these new classes, causing the decision boundaries of previous classes to shift significantly [4]. Some methods [10]–[14] mitigate this issue by replaying a limited set of exemplars from old classes while learning new classes. However, past data is often unavailable due to privacy concerns or restrictions on the long-term storage of raw samples in real-world applications. Therefore, in this paper, we focus on Non-Exemplar Class-Incremental Learning (NECIL), a more challenging scenario where no raw data from previous tasks is retained.

In NECIL research, one of the primary challenges is to reconstruct the decision boundaries of previous classes with-

out accessing their original samples. Some methods [15]–[18] leverage deep generative models to generate pseudo-samples of previous classes. However, these methods frequently encounter issues like model collapse and unreliable sample generation. To address these limitations, prototype-based methods have gained popularity for their ability to balance privacy security with robust representational capability [4], [5], [9], [19], [20]. As shown in Figure 1(a), PASS [4] first proposes storing a representative prototype for each old class, typically the class mean in the deep feature space. When learning new classes, these stored prototypes are used as old class representations to inject previous information. Furthermore, as illustrated in Figure 1(b), PASS [4] augments these prototypes by adding Gaussian noise to increase the diversity of the injected representations. However, this method struggles with maintaining representation quality and ensuring clear distinctions between classes, as the added noise can interfere with the feature space. RRFE [5], depicted in Figure 1(c), randomly combines old class prototypes linearly to create hybrid prototypes with mixed labels. While innovative, these hybrid prototypes lack authenticity, making it difficult to accurately reconstruct the original class boundaries. Other methods, such as SSRE [19] and FeTrIL [9], either directly use these augmented prototype variants or introduce enhancements based on them. Overall, these methods often overlook the distribution's shape and scope in the feature space of old classes, resulting in imprecise or limited diversity in class representations.

To overcome this challenge, we propose a Multivariate Gaussian Sampling (MGS) strategy, which offers a more robust generation of old class representations, replacing traditional prototypes. As depicted in Figure 1(d), our approach models the feature space of each old class using a multivariate Gaussian distribution. By sampling from the high-likelihood regions of these distributions, we generate diverse and high-quality representations of old classes. This strategy is inspired by recent work in out-of-distribution detection [21]. Unlike prototype-based methods, which focus primarily on the class mean, our approach accounts for both the shape and scope of the entire feature distribution space. As a result, the generated old class representations exhibit both good diversity and higher accuracy and can more effectively reconstruct the decision boundaries of previous classes.

In addition to generating old class representations, another often overlooked challenge in NECIL is the increasing deviation between the classifier and the generated old class representations. Specifically, as the model continuously learns new class data, both the feature extractor and the classifier must be updated to maintain the model's plasticity. This ongoing process gradually leads to the outdated of the old class representations, whether obtained through prototype-based methods or our proposed strategy [20]. In NECIL, this deviation is particularly problematic because it cannot be directly measured without access to raw samples from the old classes. To alleviate this issue, PRAKA [20] introduces random bidirectional interpolation between new class features and stored old class prototypes to enhance the quality of old class representations. While this random interpolation somewhat

mitigates representation deviation in scenarios with a limited number of incremental phases, it also introduces instability in the generated features due to its reliance on a random interpolation factor. As a result, representation deviation may increase over longer sequences of incremental tasks, leading to performance degradation, as demonstrated in the experimental results in Section IV-A.

To further address this bottleneck, we propose a Similarity-based Feature Compensation (SFC) mechanism to reduce the deviation between generated representations and the evolving classifier. Specifically, we use normalized cosine similarity to measure the similarity between the arriving new class features and generated old class features, selecting the most similar new class feature for each generated old class feature. We then apply a simple yet effective element-wise averaging between the generated and most similar feature sets to form the retrospective representations of old classes. These compensated features, along with their original labels, are used to retain the previously acquired knowledge. This compensatory mechanism ensures that synthesized old class representations not only remain closely aligned with real distributions over longer incremental tasks but also preserve a stable ability to reconstruct classification boundaries.

Finally, we build a robust baseline model that integrates self-supervised label augmentation [22] and knowledge distillation [23] to combine the two key components described above, forming the proposed efficient non-exemplar class-incremental learning method with **R**etrospective **F**eature **S**ynthesis, termed RFS. Our main contributions are as follows:

- We propose an effective old class representation generation strategy for NECIL, which models the feature space with multivariate Gaussian distribution and sample representations from high-likelihood regions to resist catastrophic forgetting.
- We contribute a similarity-based feature compensation mechanism, which selects the most similar new class features to compensate for the generated old class features through element-wise averaging, further improving the efficiency of incremental learning.
- Extensive experiments and analysis demonstrate that our method achieves state-of-the-art performance on the NECIL benchmarks of CIFAR-100, TinyImageNet, and ImageNet-Subset.

II. RELATED WORK

A. Class Incremental Learning

Class-Incremental Learning (CIL) addresses the challenge of continuous learning in real-world scenarios where new data classes are dynamically introduced. By providing flexible expansion capabilities to the existing end-to-end models, CIL has broad prospects and applications in object recognition [24]–[26], object detection [27]–[29], semantic segmentation [30]–[32], and other related fields. A key issue in CIL is mitigating the performance degradation on previously learned classes as models incrementally learn new ones. To tackle this, researchers have developed various CIL methods [24]–[26], typically categorized into three types: parameter isola-

tion methods, regularization-based methods, and replay-based methods.

Parameter isolation methods assign different sets of parameters to different tasks, preventing interference between tasks during continuous learning [24]. Some approaches [33]–[35] maintain a fixed model architecture, activating only task-specific parameters. Other works [36]–[38] employ dynamically growing network architectures, adding new parameters for new tasks while preserving those from previous tasks. While parameter isolation effectively avoids catastrophic forgetting, it limits knowledge sharing across tasks and can lead to rapid model growth or decreased accuracy, as sparse knowledge density hinders adaptability to more tasks.

Regularization-based methods introduce additional terms into the loss function to consolidate previously acquired knowledge while learning new tasks, avoiding the need to store raw samples. This approach prioritizes privacy security and reduces memory requirements [24]. For example, LwF [39] uses the previous model's output as a soft label to constrain the current model's output, ensuring the current model behaves similarly to the previous one on old classes, thereby alleviating forgetting. EWC [40] selectively reduces plasticity in specific weights to protect previously learned knowledge. DMC [41] trains a separate model for new classes, then uses double distillation to combine it with models trained on old classes, addressing performance limitations caused by asymmetric supervision between new and old classes.

Replay-based methods store raw samples or use generative models to create pseudo-samples of old classes, replaying these samples while learning new tasks to alleviate forgetting [24]. Methods like iCaRL [10], ER [42], and SER [43] store a limited subset of raw samples to preserve recognition ability on old classes. However, these methods face challenges when raw samples cannot be stored due to privacy concerns or long-term data storage limitations. Some methods [15]–[17] generate pixel-level samples by modeling the pixel space of previous classes, but this increases the complexity of training the generative model and often results in issues such as model collapse and unreliable sample generation [24]. In this paper, we bypass the complexity and instability of pixel-level sample generation by directly generating high-dimensional features for old classes, offering higher efficiency and more reliable quality.

B. Non-Exemplar Class-Incremental Learning

In real-world applications, data privacy concerns and limitations on storing original data have increased interest in studying non-exemplar class-incremental learning methods. PASS [4] stores a representative prototype for each old class and enhances these prototypes with Gaussian noise to maintain decision boundaries for previous tasks. SSRE [19] introduces a self-sustaining representation expansion scheme that preserves invariant knowledge while selectively incorporating new samples into the distillation process to improve discrimination between old and new classes. FeTrIL [9] combines a fixed feature extractor with a pseudo-feature generator, which creates old class representations by using the geometric translation

of new class features, balancing stability and plasticity in incremental learning. PRAKA [20] utilizes random bidirectional interpolation to blend previous class prototypes with new class features, dynamically reshaping old class feature distributions to preserve classification decision boundaries. Additionally, RRFE [5] removes the batch normalization layer during incremental training to enhance model stability and employs prototype mixing to expand the feature space of old classes. In contrast to these methods, we propose modeling the high-dimensional feature space using multivariate Gaussian distributions to generate higher-quality representations of old classes, ensuring both greater diversity and accuracy of old class representations.

III. METHODOLOGY

A. Problem Definition and Overall Framework

NECIL focuses on developing the classification model that can incrementally learn new classes while preserving its ability to recognize all previously learned classes, without the need to store raw samples from past tasks. Let C_t represent the set of classes learned during task t, and $D_t = \{X_t, Y_t\} = \{x_t^i, y_t^i\}_{i=1}^{N_t}$ denote the data for task t, where x_t^i is an input sample, $y_t^i \in C_t$ is its corresponding label, and N_t is the total number of samples in D_t. Classes learned across different tasks are distinct, i.e., $C_1 \cap C_2 \cap \cdots \cap C_T = \emptyset$. The classification model consists of two key components: the feature extractor $\mathcal{F}$ with parameters θ, and the unified classifier $\mathcal{G}$ with parameters φ. The goal for task t is to minimize a predefined loss function $\mathcal{L}_t$ on the new dataset D_t:

$$\underset{\theta_t, \varphi_t, \delta}{\operatorname{argmin}} \mathcal{L}_t \left(\mathcal{G} \left(\mathcal{F} \left(X_t; \theta_t \right); \varphi_t \right), Y_t \right) + \sum \delta_p$$
$$\text{s.t. } \mathcal{L}_t \left(X_p, Y_p \right) - \mathcal{L}_p \left(X_p, Y_p \right) \leqslant \delta_p, \delta_p \geqslant 0, \quad \forall p \in [1, t-1] \quad (1)$$

where $\mathcal{L}_t(X_p, Y_p) = \mathcal{L}(\mathcal{G}(\mathcal{F}(X_p; \theta_t); \varphi_t), Y_p)$ represents the loss of the current model at task t on the old dataset D_p, while $\mathcal{L}_p(X_p, Y_p) = \mathcal{L}(\mathcal{G}(\mathcal{F}(X_p; \theta_p); \varphi_p), Y_p)$ denotes the loss of the previous model trained during task p on the same dataset. The last term, $\delta = \{\delta_p\}$, is a slack variable that allows for a small permissible increase in the loss on the old dataset.

Figure 2 illustrates the overall framework of our approach. After each phase, we store statistical information for each learned class, specifically the mean and covariance matrix of deep features derived from the training data. This information is used in the subsequent incremental learning phase to synthesize retrospective features of previously learned classes. Specifically, the proposed multivariate Gaussian sampling strategy models the feature space using a multivariate Gaussian distribution, generating representations of old classes by sampling from high-likelihood regions. To enhance these generated features, we apply a similarity-based feature compensation mechanism that selects similar new class features to complement the generated old class representations to effectively synthetic retrospective features.

During incremental learning of new classes, the new data is augmented using rotation transformations before being passed into the feature extractor. The deep features extracted from these new instances are used to optimize both the classifier
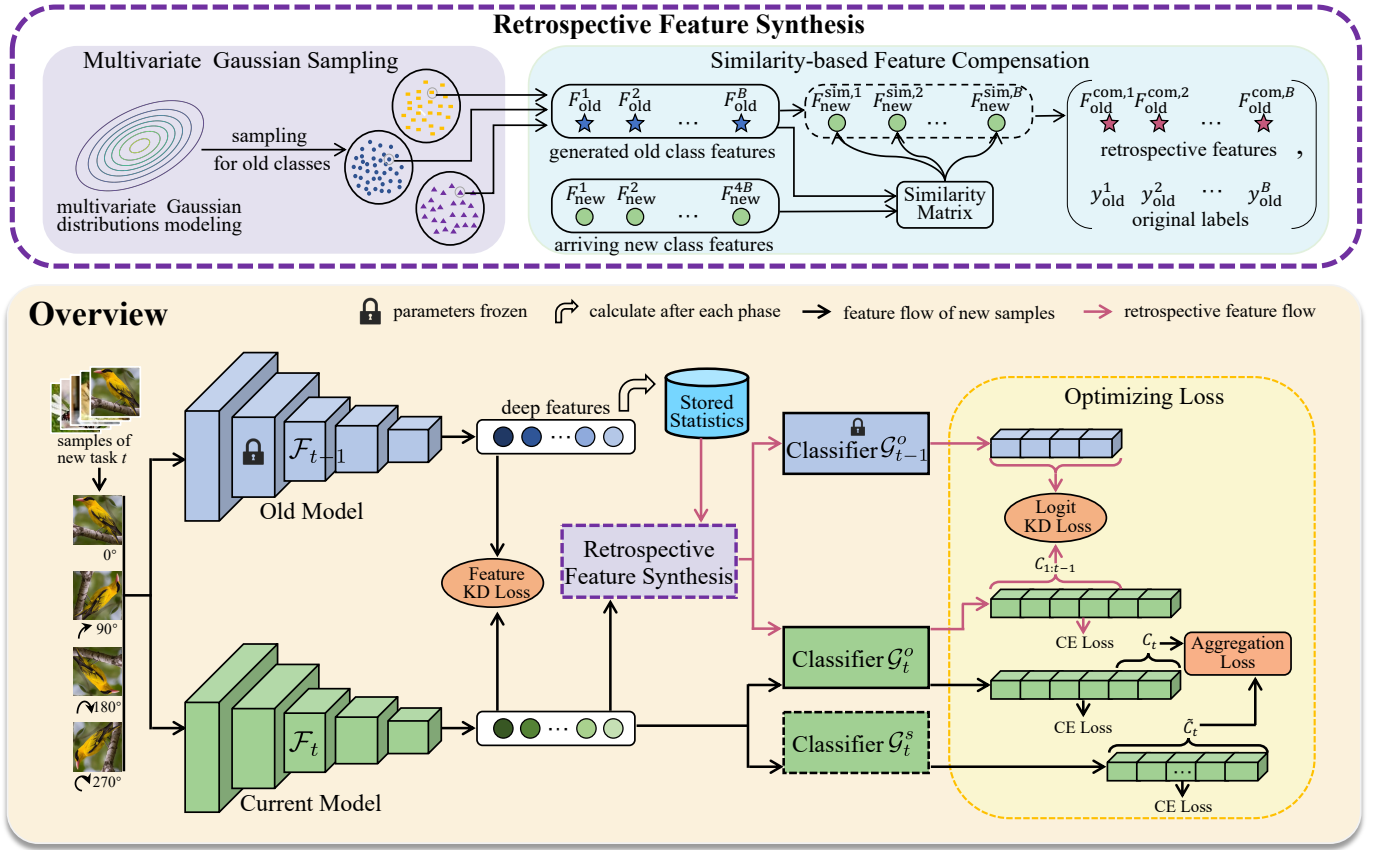
Fig. 2. Overall framework of our proposed method for NECIL. The new samples from the current task are augmented using rotation transformations. These augmented new class samples, along with synthesized retrospective features, are then utilized to optimize the incremental classification model to consolidate previously acquired knowledge while learning new classes. In the process of retrospective feature synthesis, we model the feature space of each old class using a multivariate Gaussian distribution and sample old class representations from high-likelihood regions. Additionally, we select the most similar new class features to complement the sampled old class features, thereby creating efficient retrospective representations of the old classes. "CE loss" represents the cross-entropy classification loss. "Stored Statistics" refers to the feature means and covariance matrices, which are calculated and stored after each task is learned. $\tilde{C}_t$ represents the classes within task t after rotation-based label augmentation.

$\mathcal{G}_t^s$ with self-supervised label augmentation [20], [22] and the standard non-augmented classifier $\mathcal{G}_t^o$. The augmented classifier $\mathcal{G}_t^s$ classifies only the classes learned in the current phase, treating each rotation transformation as a distinct class. In contrast, $\mathcal{G}_t^o$ is a unified classifier that recognizes the non-rotation-augmented classes, including both previously learned and newly learned classes. The synthesized retrospective features for the old classes are injected into $\mathcal{G}_t^o$ to reconstruct decision boundaries of these classes. To maintain model stability and prevent significant degradation on old classes when new data is introduced, we apply knowledge distillation [23] at both the feature level and the classification logits. During the evaluation (inference) phase, the current feature extractor $\mathcal{F}_t$ and classifier $\mathcal{G}_t^o$ are used for testing.

B. Retrospective Feature Synthesis

Multivariate Gaussian sampling. Inspired by VOS [21], we assume that the representations of old classes follows a class-conditional multivariate Gaussian distribution:

$$p(\mathcal{F}(x; \theta_t)|y = k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (2)$$

where $\boldsymbol{\mu}_k$ is the mean of class $k \in \{1, 2, \ldots, T\}$, $\boldsymbol{\Sigma}_k$ is the tied covariance matrix, and $\mathcal{F}(x) \in \mathbb{R}^m$ represents the latent feature extracted from image x by the feature extractor, with m being the dimension of the deep feature space. To estimate the parameters of this Gaussian distribution, we compute the empirical mean $(\widehat{\boldsymbol{\mu}}_k)$ and covariance $(\widehat{\boldsymbol{\Sigma}}_k)$ from the high-dimensional features of all training samples:

$$
\begin{aligned}
\widehat{\boldsymbol{\mu}}_k &= \frac{1}{N^k} \sum_{i: y_i = k} \mathcal{F}(x_i; \theta_p), \\
\widehat{\boldsymbol{\Sigma}}_k &= \frac{1}{N^k} \sum_{i: y_i = k} (\mathcal{F}(x_i; \theta_p) - \widehat{\boldsymbol{\mu}}_k)(\mathcal{F}(x_i; \theta_p) - \widehat{\boldsymbol{\mu}}_k)^\top
\end{aligned}
\quad (3)
$$

where N^k is the number of images in class k, and p represents the incremental learning phase which class k is learned in. After each phase of training is completed, the multivariate Gaussian parameters of the classes learned in this phase are evaluated.

We propose generating deep representations for old classes by sampling from the estimated multivariate Gaussian distribution. To ensure that the generated features have high

confidence and diversity, we sample old class features $\mathcal{O}_k$ from the ζ likelihood region of the estimated class-conditional distribution:

$$\mathcal{O}_k = \{\mathbf{o}_k \mid \frac{1}{\sqrt{(2\pi)^m \widehat{\mathbf{\Sigma}}_k}} \exp\left(\frac{(\mathbf{o}_k - \widehat{\boldsymbol{\mu}}_k)^\top (\mathbf{o}_k - \widehat{\boldsymbol{\mu}}_k)}{-2\widehat{\mathbf{\Sigma}}_k}\right) > \zeta\} \tag{4}$$

Here, $\mathcal{O}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k)$ represents the deep features sampled for class k, specifically from the sub-level set with likelihood greater than ζ. These sampled features should ideally aid the classifier in reconstructing the decision boundaries of old classes. For consistency, we generate the same number of old class features as the batch size of new data. In practice, we implement this feature generation process by selecting the sample with the highest likelihood from a set of 1000 ($K = 1000$) randomly sampled instances for each old class. We also conduct ablation experiments on different values of K to verify its robustness (as shown in Figure 9). Each iteration generates different samples to ensure feature diversity, while the high-likelihood constraint ensures feature quality. Together, these guarantees enhance the classifier's ability to accurately reconstruct the decision boundaries of previous classes.

Similarity-based feature compensation. While the features generated in the previous step help retain information about previous classes, these features gradually become outdated as the model is updated incrementally [20]. Specifically, maintaining the model's plasticity requires updating the classifier when learning new classes. This update can lead to a discrepancy between the synthesized old class features and the current classifier, potentially resulting in a bias towards new classes. To address this, we compensate for the generated old class representations by incorporating new class features, which are better aligned with the updated classifier.

As depicted in Figure 2, we first rotate each new class sample x_t^i by 0°, 90°, 180°, and 270° to generate augmented features $\tilde{x}_t^i$:

$$\tilde{x}_t^i = \{x_t^{i,j}\}_{j=0}^3 = \{\text{rotate}(x_t^i, j \times 90°)\}_{j=0}^3 \tag{5}$$

A batch of these rotation-augmented images is then passed through the feature extractor to obtain the new class features $\tilde{F}_{\text{new}}$.

Let C_o denote the set of all old classes at current phase, and let B represent the batch size. We then determine a batch of old classes, denoted as y_{old}, as follows:

$$y_{\text{old}} = \begin{cases} \text{rand}(C_o, B), & \text{if } B < \text{size}(C_o) \\ C_o, & \text{if } B = \text{size}(C_o) \\ C_o \cup \text{rand}(C_o, B - \text{size}(C_o)), & \text{if } B > \text{size}(C_o) \end{cases} \tag{6}$$

where $\text{rand}(C_o, B)$ represents a random selection of B classes from C_o. For each class in y_{old}, a corresponding feature is generated using the MGS strategy, forming the generated feature set F_{old}. The cosine similarity S between the old class features F_{old} and the new class features $\tilde{F}_{\text{new}}$ is calculated as follows:

$$S \in \mathbb{R}^{B \times 4B} = \cos(F_{\text{old}}, \tilde{F}_{\text{new}}) \tag{7}$$

For each old class feature in F_{old}, the most similar new class features in $\tilde{F}_{\text{new}}$ is selected to form the similar feature set $F_{\text{new}}^{\text{sim}}$:

$$F_{\text{new}}^{\text{sim}} = \left[\tilde{F}_{\text{new}, j_1^*}, \tilde{F}_{\text{new}, j_2^*}, \ldots, \tilde{F}_{\text{new}, j_B^*}\right] \tag{8}$$

where $j_i^* = \arg\max_j S_{ij}, \quad \forall i \in \{1, 2, \ldots, B\}$. Element-wise averaging is then performed to obtain the compensated old class features $F_{\text{old}}^{\text{com}}$:

$$F_{\text{old}}^{\text{com}} = \frac{F_{\text{old}} + F_{\text{new}}^{\text{sim}}}{2} \tag{9}$$

$F_{\text{old}}^{\text{com}}$ and the original labels y_{old} are paired as a batch of synthesized information injected into the classification model. Since the compensated old class features are similar to the new class features while retaining the characteristics of the old classes, they better enhance the model's discriminative ability during the joint learning process.

C. Optimizing Loss

New classes learning. For the rotation-augmented image $\tilde{x}_t^i$, the augmented labels are defined as:

$$\tilde{y}_t^i = \left\{y_t^i \times 4 + j\right\}_{j=0}^3 \tag{10}$$

We first optimize the two classifiers separately using cross-entropy classification loss. For the classifier $\mathcal{G}_t^s$, the cross-entropy classification loss $\mathcal{L}_{\text{new}}^{\text{aug\_cls}}$ is computed using the augmented data $\tilde{x}_t$ and labels $\tilde{y}_t$. For the standard classifier $\mathcal{G}_t^o$, the unaugmented data (0°) x_t and original labels y_t are used to compute the cross-entropy classification loss $\mathcal{L}_{\text{new}}^{\text{cls}}$.

To further enhance model generalization on arriving new classes, we employ Kullback-Leibler divergence (KL) loss to compute the aggregation loss between the outputs of these two classifiers, following PRAKA [20]:

$$\mathcal{L}_{\text{new}}^{\text{ka}} = \text{KL}(\mathcal{P}_{\text{agg}}(\tilde{y}_t^i \mid \tilde{F}_{\text{new}}) || \mathcal{P}(y_t^i \mid F_{\text{new}})) \tag{11}$$

where F_{new} represents the hidden features obtained by the feature extractor on the non-augmented image x_t. $\mathcal{P}_{\text{agg}}(\tilde{y}_t^i \mid \tilde{F}_{\text{new}})$ represents the aggregated outputs of classifier $\mathcal{G}_t^s$, and $\mathcal{P}(y_t^i \mid F_{\text{new}})$ represents the output of the classifier $\mathcal{G}_t^o$ on the current learning classes. The definitions of these two components are as follows:

$$\mathcal{P}_{\text{agg}}(\tilde{y}_t^i \mid \tilde{F}_{\text{new}}) = \frac{1}{4}\sum_{j=0}^3 \varphi_{t,s}^j \tilde{F}_{\text{new}}^{i,j} \tag{12}$$

$$\mathcal{P}(y_t^i \mid F_{\text{new}}) = \mathbb{I}_{C_t}(\varphi_{t,o} F_{\text{new}}^i) \tag{13}$$

where $\mathbb{I}_{C_t}$ is an indicator function that filters the output to only the current classes in C_t. Thus, the loss for learning new classes is summarized as follows:

$$\mathcal{L}_{\text{new}} = \mathcal{L}_{\text{new}}^{\text{cls}} + \mathcal{L}_{\text{new}}^{\text{aug\_cls}} + \mathcal{L}_{\text{new}}^{\text{ka}} \tag{14}$$

Retrospective knowledge retention. Similarly, the synthesized representations of old classes $F_{\text{old}}^{\text{com}}$ and their corresponding labels y_{old} are used to optimize the classifier $\mathcal{G}_t^o$, reconstructing the decision boundaries for old classes through cross-entropy classification loss:

$$\mathcal{L}_{\text{old}}^{\text{cls}} = \mathcal{L}_{\text{ce}}(\mathcal{G}^o(F_{\text{old}}^{\text{com},i}; \varphi_{t,o}), y_{\text{old}}^i) \tag{15}$$

Additionally, the synthesized old class representations and arriving new class features are used for knowledge distillation at both the logit output level and the deep feature level to prevent significant degradation in model stability. The feature-level knowledge distillation, as used in PASS [4], RRFE [5], and PRAKA [20], is defined as:

$$\mathcal{L}_{\text{old}}^{\text{feat\_kd}} = \|\mathcal{F}(\tilde{x}_t^i; \theta_t) - \mathcal{F}(\tilde{x}_t^i; \theta_{t-1})\|_2 \tag{16}$$

The logit-level knowledge distillation is defined as:

$$\mathcal{L}_{\text{old}}^{\text{logit\_kd}} = \text{KL}(\mathcal{G}^o(F_{\text{old}}^{\text{com},i}; \varphi_{t,o}), \mathcal{G}^o(F_{\text{old}}^{\text{com},i}; \varphi_{t-1,o})) \tag{17}$$

Thus, the loss for retaining retrospective knowledge is summarized as:

$$\mathcal{L}_{\text{old}} = \mathcal{L}_{\text{old}}^{\text{cls}} + \mathcal{L}_{\text{old}}^{\text{feat\_kd}} + \mathcal{L}_{\text{old}}^{\text{logit\_kd}} \tag{18}$$

Overall optimization loss. Combining the two parts, the overall optimization objective of our proposed method on the current task t is defined as:

$$\mathcal{L}_t = \mathcal{L}_{\text{new}} + \alpha \mathcal{L}_{\text{old}} \tag{19}$$

where α is a hyperparameter to balance the weights of these two components, and we set it to 15 in our experiments following PRAKA [20].

IV. EXPERIMENTS AND ANALYSIS

Datasets. We evaluate the performance of our method through expensive experiments on three datasets: CIFAR-100 [44], TinyImageNet [45], and ImageNet-Subset [46]. CIFAR-100 contains 100 classes, a total of 60,000 32×32 pixel images, with 500 images for training and 100 for testing in each class. TinyImageNet includes 200 classes, each containing 500 training images, 50 validation images, and 50 test images, offering a more challenging scene for performance comparison. The ImageNet-Subset is a 100-class subset of ImageNet-1k, which has larger image sizes and scales than CIFAR-100 and TinyImageNet. We employ an incremental learning setup denoted as B + C × T. Initially, the model is trained on B classes. The remaining classes are then evenly distributed across T subsequent phases (tasks), with C classes per phase. B is set to half the total number of classes, with minor adjustments made to ensure the remaining classes are evenly divided across the subsequent phases. We determine three different incremental learning settings on each dataset by varying the T value (5, 10, 20). For simplicity, these settings are referred to as 5 phases, 10 phases, and 20 phases. To fairly compare performance, the class splits for each incremental setting on all datasets strictly follow those used in [4] and [20].

Implementation details. Following [4], [5], [19], [20], we use ResNet-18 as the backbone network (feature extractor). Each new task is trained for 100 epochs, using the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 2e-4. The batch size is set to 128, and the initial learning rate is 0.001, which is reduced by a factor of 10 at the 45 and 90 epochs. Results marked with an asterisk (*)

are cited from [20] or [5]. To minimize the impact of training randomness on model accuracy, we fix the random seed to 0 during training, consistent with PRAKA [20]. Additionally, we conduct experiments using different class orders and random seeds, as shown in Figure 8, to verify the robustness of our method.

Evaluation metrics. Consistent with [4], [5], [20], we use average incremental accuracy, final accuracy, and average forgetting as evaluation metrics. Average incremental accuracy measures the average classification accuracy of the model across all incremental stages (including the initial task), reflecting the method's overall incremental performance. Final accuracy represents the top-1 classification accuracy achieved after completing all incremental learning phases. Average forgetting, as defined in [47], quantifies the average difference between the peak accuracy and final accuracy for each task after incremental learning. A smaller difference indicates better performance.

A. Overall Performance

Table I presents a quantitative comparison of average incremental accuracy between various methods. Compared to the listed methods that store previous exemplars (*E=20*), our method demonstrates significantly improved average incremental accuracy across different incremental settings on all three datasets. Moreover, our method also outperforms non-exemplar methods (*E=0*), achieving the best overall performance. Notably, compared to the state-of-the-art PRAKA method, our method achieves average accuracy improvements of 1.8%, 2.71%, and 3.53% in incremental settings of 5, 10, and 20 phases on CIFAR-100, respectively. On TinyImageNet, our method achieves accuracy gains of 1.07% and 2.85% in the 10 and 20 phases settings. On ImageNet-Subset, our method shows an average improvement of 4.89% in the 10 phases setting. Additionally, as the number of incremental phases increases, our method demonstrates greater robustness than PRAKA, effectively mitigating the rapid decline in average accuracy. These results highlight the significant superiority of our method in overall incremental learning performance.

Table II further compares the final accuracy of our method with several state-of-the-art approaches after completing different incremental learning settings. Our method significantly outperforms PASS and SSRE, and shows accuracy improvements of 2.43%, 3.77%, and 4.1% over PRAKA in the three incremental settings on CIFAR-100. In the three settings on TinyImageNet, our method also achieves accuracy gains of 0.94%, 2.04%, and 4.14%, respectively. This demonstrates that our method has superior visual recognition performance after all incremental phases. Figure 3 shows the classification accuracy curves of our method compared to these state-of-the-art methods at different incremental phases. Our method consistently maintains higher accuracy in nearly all phases, with a slower decline in accuracy as the number of incremental phases increases. To further differentiate between the methods, we compare their average forgetting at the last phase in Table III. Our method achieves lower average forgetting rates in the 5, 10, and 20 phases settings on both CIFAR-100 and TinyImageNet.

TABLE I
QUANTITATIVE COMPARISON OF AVERAGE INCREMENTAL ACCURACY (%) BETWEEN OUR METHOD AND OTHER METHODS ON
CIFAR-100, TINYIMAGENET, AND IMAGENET-SUBSET. E REPRESENTS THE NUMBER OF RETAINED EXEMPLARS. BEST RESULTS ARE
HIGHLIGHTED IN BOLD.

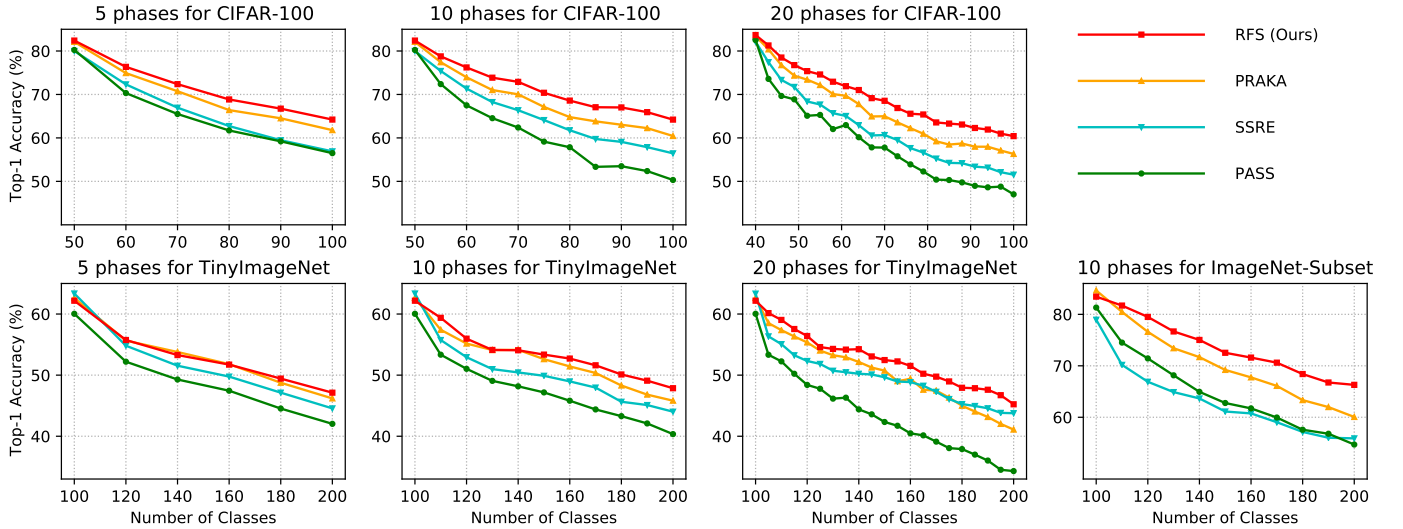| | Methods | CIFAR-100 | | | TinyImageNet | | | ImageNet-Subset | | |
		5 phases	10 phases	20 phases	5 phases	10 phases	20 phases	5 phases	10 phases	20 phases
E = 20	iCaRL-CNN*	51.07	48.66	44.43	34.64	31.15	27.90	53.62	50.53	—
	iCaRL-NCM* [10]	58.56	54.19	50.51	45.86	43.29	38.04	65.04	60.79	—
	EEIL* [48]	60.37	56.05	52.34	47.12	45.01	40.50	—	63.34	—
	UCIR* [49]	63.78	62.39	59.07	49.15	48.52	42.83	68.43	66.16	—
	WA* [50]	64.32	59.51	54.37	—	—	—	—	63.60	—
E = 0	EWC* [40]	24.48	21.20	15.89	18.80	15.77	12.39	—	20.40	—
	LwF-MC* [10]	45.93	27.43	20.07	29.12	23.10	17.43	—	31.18	—
	MUC* [51]	49.42	30.19	21.27	32.58	26.61	21.95	—	35.07	—
	SDC* [52]	56.77	57.00	58.90	—	—	—	—	61.12	—
	IL2A* [53]	66.03	60.31	57.92	47.34	44.70	40.04	—	—	—
	PASS* [4]	63.47	61.84	58.09	49.55	47.29	42.07	64.41	61.80	51.32
	SSRE* [19]	65.88	65.04	61.70	50.39	48.93	48.17	—	67.69	—
	FeTrIL* [9]	64.73	63.41	57.44	52.91	51.71	49.72	69.64	68.93	62.52
	RRFE* [5]	67.56	66.76	64.32	51.50	50.99	50.56	69.56	69.05	63.73
	PRAKA* [20]	70.02	68.86	65.86	**53.32**	52.61	49.83	—	68.98	—
	RFS (Ours)	**71.82**	**71.57**	**69.39**	53.25	**53.68**	**52.68**	**74.15**	**73.87**	**72.43**

Fig. 3. Average classification accuracy of our method and state-of-the-art methods at each phase during incremental learning on CIFAR-100 and TinyImageNet.

TABLE II
COMPARISON OF FINAL ACCURACY (%) BETWEEN OUR METHOD
AND STATE-OF-THE-ART METHODS ON CIFAR-100 AND
TINYIMAGENET.

| Methods | CIFAR-100 | | | TinyImageNet | | |
	5	10	20	5	10	20
PASS	56.49	50.32	47.02	42.03	40.35	34.30
SSRE	56.91	56.45	51.54	44.56	44.02	43.77
PRAKA	61.79	60.44	56.31	46.19	45.82	41.09
RFS (Ours)	**64.22**	**64.21**	**60.41**	**47.13**	**47.86**	**45.23**

B. Ablation Studies

In this section, we perform ablation studies to analyze the impact of the key components of our proposed method: Multivariate Gaussian Sampling (MGS) and Similarity-based Feature Compensation (SFC).

As illustrated in the first two rows of Table IV, incorporating MGS significantly enhances the average incremental accuracy across various incremental settings on CIFAR-100 and TinyImageNet, compared to the method without MGS. Similarly, the last two rows demonstrate that the combined use of MGS and SFC leads to further performance gains over using SFC alone. Figure 4 illustrates that MGS substantially

TABLE III
COMPARISON OF AVERAGE FORGETTING BETWEEN OUR METHOD
AND OTHER METHODS ON CIFAR-100 AND TINYIMAGENET.

Methods	CIFAR-100			TinyImageNet		
	5	10	20	5	10	20
iCARL-CNN*	42.13	45.69	43.54	36.89	36.70	45.12
iCaRL-NCM*	24.90	28.32	35.53	27.15	28.89	37.40
EFIL*	23.36	26.65	32.40	25.56	25.51	35.04
UCIR*	21.00	25.12	28.65	20.61	22.25	33.74
LwF_MC*	44.23	50.47	55.46	54.26	54.37	63.54
MUC*	40.28	47.56	52.65	51.46	50.21	58.00
PASS*	25.20	30.25	30.61	18.04	23.11	30.55
SSRE*	18.37	19.48	19.00	9.17	14.06	14.20
RRFE*	16.02	15.57	17.43	13.49	14.26	13.18
PRAKA*	12.59	14.65	17.39	11.84	13.95	18.51
RFS (Ours)	**5.76**	**5.04**	**7.14**	**4.06**	**3.35**	**6.58**

TABLE IV
ABLATION STUDY (IN AVERAGE INCREMENTAL ACCURACY, %) OF OUR
METHOD ON CIFAR-100 AND TINYIMAGENET.

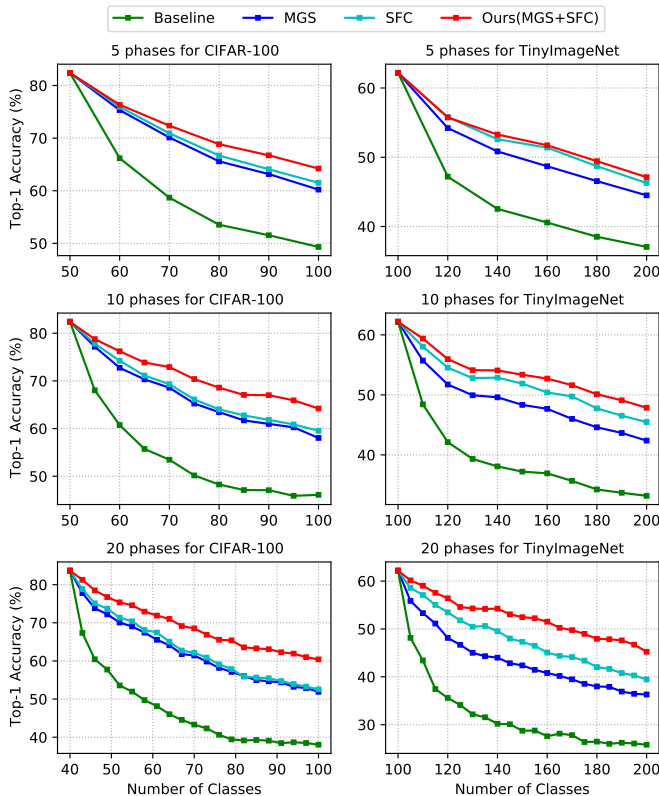| MGS	SFC	CIFAR-100			TinyImageNet		
		5	10	20	5	10	20
		60.28	55.00	47.62	44.68	40.10	32.51
✓		69.47	67.36	62.87	51.17	49.25	43.90
	✓	70.26	68.18	63.71	52.83	52.02	48.18
✓	✓	**71.82**	**71.57**	**69.39**	**53.25**	**53.68**	**52.68**

Fig. 4. Average classification accuracy for the component effectiveness analysis of our method on CIFAR-100 and TinyImageNet.

improves classification accuracy at each phase, resulting in a slower decline in accuracy compared to the baseline. These findings emphasize the pivotal role of MGS in non-exemplar class-incremental learning.

Furthermore, as indicated in the first and third rows of Table IV, incorporating SFC also leads to significant improvements in average incremental accuracy across both datasets. The second and fourth rows show that the combined use of SFC and MGS yields substantial performance improvements compared to using MGS alone. In Figure 4, SFC enhances classification accuracy at all phases of incremental learning compared to both the baseline and MGS alone, again showing a slower decline in accuracy. These results highlight the substantial impact of SFC and demonstrate that MGS and SFC complement each other, with their combined use delivering the most significant improvements in NECIL.

C. Analysis

Visualization results. To intuitively demonstrate the effectiveness of the different modules in our method, we use t-SNE [54] to visualize the 2D embeddings of deep feature vectors, as shown in Figure 5. The two columns in Figure 5(a) show that MGS effectively reduces internal distances of old classes compared to the baseline method. This enhancement improves the clustering of old class distributions and mitigates the model's bias towards new classes. In Figure 5(b), the first column illustrates that SFC increases cluster separation between old and new classes. Additionally, the second column shows that SFC reduces feature confusion between old and new classes. Together, these results demonstrate the effectiveness of SFC in improving the model's ability to distinguish between new and old classes.

Comparison with other feature generation methods. To more comprehensively demonstrate the superiority of MGS, we compare its performance with different feature generation methods in Table V. In this table, "Prototype" refers to using prototypes as representations of old classes in high-dimensional feature space. "Prototype Mixing" denotes the method employed in RREF, where prototypes of old classes are randomly and linearly combined to create hybrid prototypes with composite labels. "Gaussian Noise Aug" refers to the method used in PASS, which involves augmented prototypes with random Gaussian noise.

The results indicate that our proposed MGS achieves more substantial performance improvements than both Prototype and Prototype Mixing. Compared to Gaussian Noise Aug, MGS shows average precision gains of 0.98%, 2.39%, and 4.1% in the three incremental settings on CIFAR-100. For the 10-phase and 20-phase settings on TinyImageNet, MGS achieves average accuracy improvements of 0.86% and 2.81%, respectively. Additionally, MGS demonstrates superior performance in final accuracy after completing all incremental tasks, with more significant accuracy gains in longer-phase incremental learning settings. These results highlight the superiority of MGS, particularly in the incremental setting with longer phases.

Comparison with other feature compensation methods. To comprehensively demonstrate the superiority of SFC, we

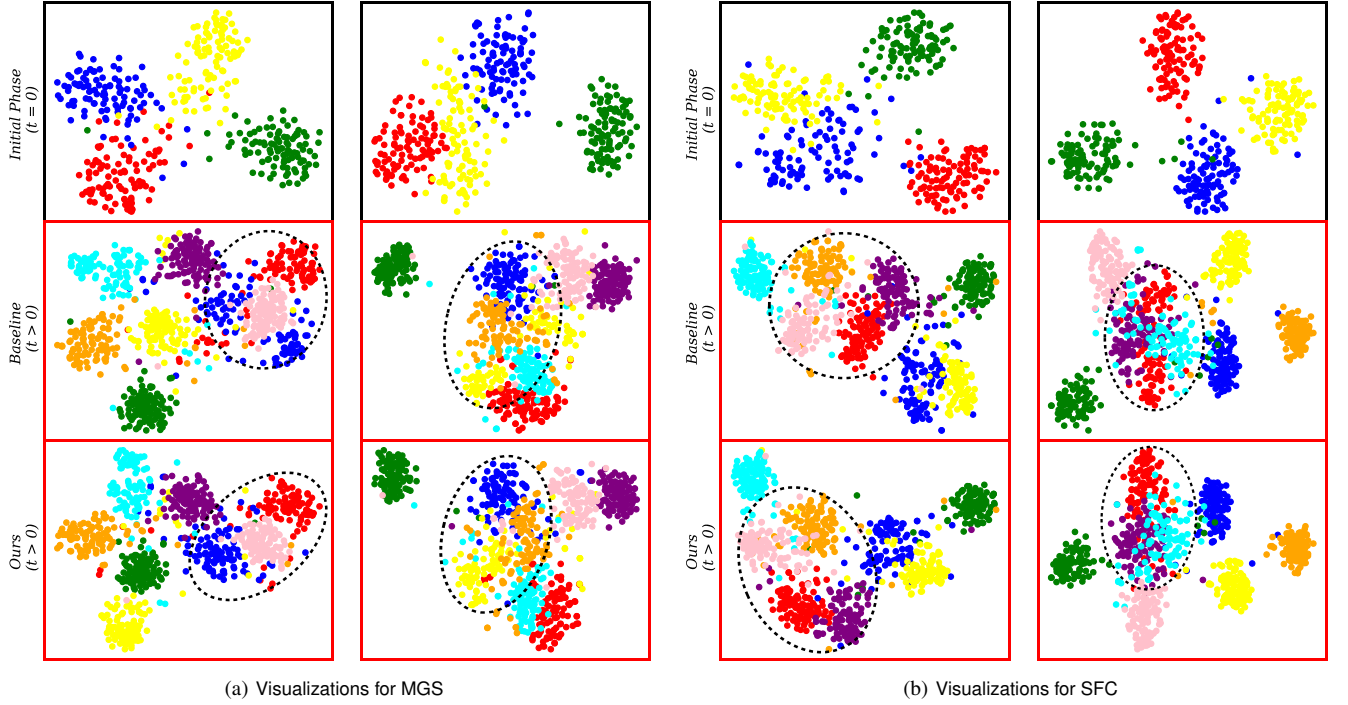(a) Visualizations for MGS   (b) Visualizations for SFC

Fig. 5. Visualization results. (a) MGS enhances the clustering of old classes in deep feature space. (b) SFC increases the discrimination between old and new classification features.

TABLE V
QUANTITATIVE COMPARISON OF AVERAGE INCREMENTAL ACCURACY (AVG,%) AND FINAL ACCURACY (FINAL, %) BETWEEN OUR METHOD AND OTHER FEATURE GENERATION METHODS.

Methods	CIFAR-100						TinyImageNet					
	5 phases		10 phases		20 phases		5 phases		10 phases		20 phases	
	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑
Prototype	70.26	61.51	68.18	59.54	63.71	52.55	52.83	46.28	52.02	45.48	48.18	39.46
Prototype Mixing [5]	68.87	59.88	67.89	58.79	63.69	51.83	51.65	44.65	50.95	43.91	47.78	37.42
Gaussian Noise Aug [4]	70.84	62.23	69.18	60.77	65.29	54.08	**53.44**	46.52	52.82	45.82	49.87	41.37
MGS (Ours)	**71.82**	**64.22**	**71.57**	**64.21**	**69.39**	**60.41**	53.25	**47.13**	**53.68**	**47.86**	**52.68**	**45.23**

TABLE VI
QUANTITATIVE COMPARISON OF AVERAGE INCREMENTAL ACCURACY (AVG,%) AND FINAL ACCURACY (FINAL, %) BETWEEN OUR METHOD AND OTHER FEATURE COMPENSATION METHODS.

Methods	CIFAR-100						TinyImageNet					
	5 phases		10 phases		20 phases		5 phases		10 phases		20 phases	
	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑
None	69.47	60.22	67.36	58.04	62.87	51.91	51.17	44.50	49.25	42.38	43.90	36.31
Rand Interp [20]	**72.19**	**64.31**	71.37	63.54	68.12	57.64	**53.58**	**47.24**	53.06	46.53	50.29	41.99
Rand Avg (Ours)	71.78	64.04	71.35	63.98	69.14	59.37	53.18	46.57	53.26	47.15	52.40	45.09
Least Sim Avg (Ours)	69.49	59.45	68.84	60.34	67.11	57.16	50.08	42.80	49.58	42.79	48.46	40.43
SFC (Ours)	71.82	64.22	**71.57**	**64.21**	**69.39**	**60.41**	53.25	47.13	**53.68**	**47.86**	**52.68**	**45.23**

TABLE VII
ABLATION STUDY OF OUR METHOD WITH AND WITHOUT LOGIT DISTILLATION ON AVERAGE INCREMENTAL ACCURACY (AVG, %) AND FINAL ACCURACY (FINAL, %).

Methods	CIFAR-100						TinyImageNet					
	5 phases		10 phases		20 phases		5 phases		10 phases		20 phases	
	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑	Avg↑	Final↑
Ours(w/o logit distill)	71.68	64.05	71.49	63.99	69.19	60.11	53.14	46.97	53.58	47.50	52.41	44.87
Ours(w/ logit distill)	**71.82**	**64.22**	**71.57**	**64.21**	**69.39**	**60.41**	**53.25**	**47.13**	**53.68**	**47.86**	**52.68**	**45.23**

compare its performance with various feature compensation methods in Table VI. "None" refers to not compensating generated old class features. "Rand Interp" refers to the random bidirectional interpolation between extracted new class features and generated old class features, as used in PRAKA to enrich old class representations. "Rand Avg" randomly selects new class features to average with generated old class features, while "Least Sim Avg" chooses new class features with the least similarity to average with generated old class features. Compared to the method without feature compensation, the various compensation methods achieve significant improvements in overall performance, verifying the critical role of feature compensation in mitigating the deviation between old class features and the updated classifier. Additionally, SFC outperforms Rand Avg and Least Sim Avg across all incremental settings and metrics on both CIFAR-100 and TinyImageNet. This superior performance is due to more similar new class features being used for compensating the generated old class features, effectively reducing the discrepancy between old class features and the updated feature extractor.

Furthermore, while Rand Interp shows slightly better performance than Rand Avg in the 5 phases setting, Rand Avg demonstrates significantly improved accuracy in the 20 phases setting on the CIFAR-100 and TinyImageNet as the phase number of incremental setting increases. This suggests that the averaging-based feature compensation method performs better than the random interpolation-based method in longer incremental sequences. Random feature interpolation can lead to unstable adjustments in old class features, either reducing the difference with new class distributions or increasing the deviation from the true class distribution excessively. In contrast, the feature averaging maintains a relatively stable distinction between compensated features and new class features, thereby enhancing the model's discriminative ability. Our method, SFC, combines similarity-based feature selection with feature averaging. It not only maintains a stable distinction between compensated and new class features through the consistent approach of feature averaging but also better alleviates the deviation between old class representations and the feature extractor by selecting the most similar new class features. As demonstrated by the experimental results, SFC outperforms other methods and achieves the best overall performance.

Confusion matrices analysis. Figure 6 illustrates the confusion matrices for PASS and our proposed method in the 10 phases setting on CIFAR-100 and TinyImageNet. The diagonal entries correspond to correct predictions, while the off-diagonal entries indicate misclassifications. In the first row, PASS displays higher classification errors in the upper right triangle region. In contrast, our method shows fewer errors in the same region. A closer examination of the diagonal entries between the two matrices reveals that our method achieves higher classification accuracy than PASS across the majority of classes. Similarly, in the second row, the confusion matrices further demonstrate that our method surpasses PASS by exhibiting better classification accuracy on the diagonal entries and fewer misclassifications in the off-diagonal entries. These findings highlight the superiority of our method from a fine-grained perspective.
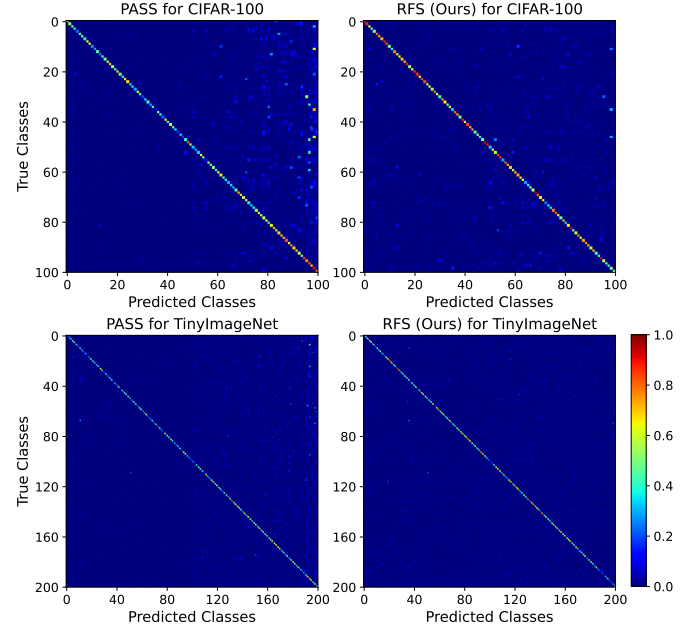

Fig. 6. Confusion matrices of PASS and our method on CIFAR-100 and TinyImageNet.
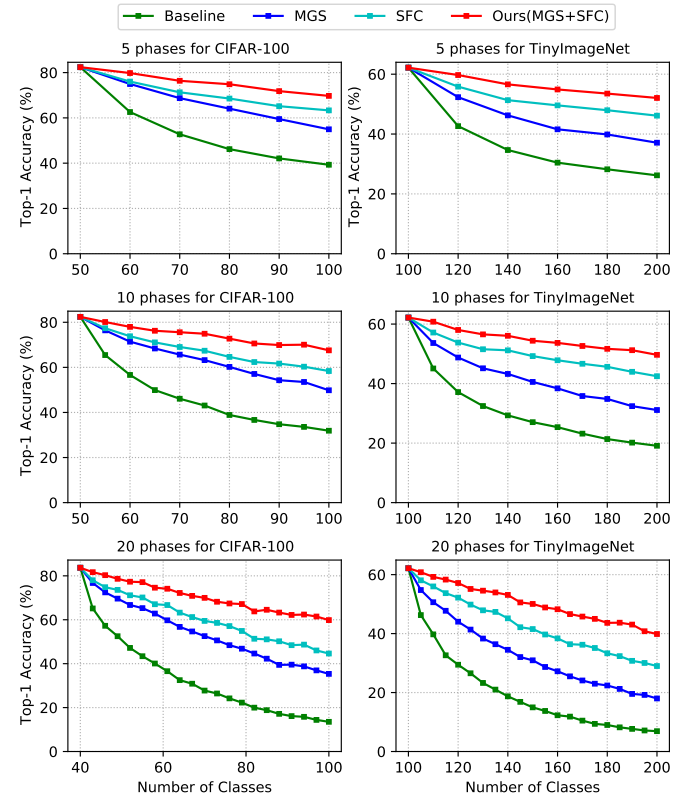

Fig. 7. Average classification accuracy of the initial classes in subsequent incremental phases on CIFAR-100 and TinyImageNet.
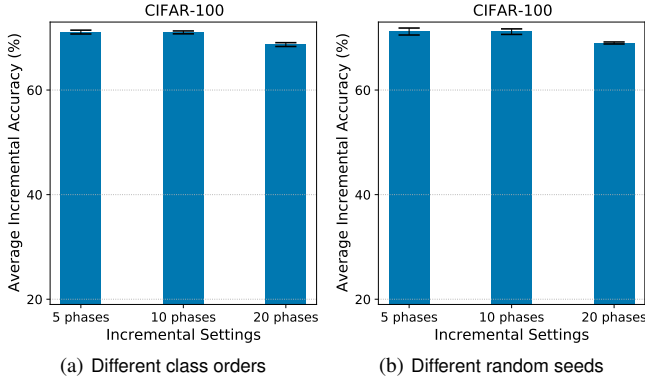
Fig. 8. Ablation study for class order and random seed. The bar chart is the middle value and the error line shows the upper and lower bound.

Mitigating catastrophic forgetting. The MGS and SFC components work together to synthesize retrospective features of previously learned classes, effectively mitigating catastrophic forgetting in non-exemplar class-incremental learning. Figure 7 presents the average classification accuracy curve of classes learned during the initial phase as they undergo subsequent incremental phases. It is clear that the use of either MGS or SFC individually helps alleviate the significant accuracy drops of the initial classes in the later phases. Furthermore, the combined use of MGS and SFC in our method results in a more pronounced improvement in the accuracy of these initial classes. These findings highlight the significant advantages of our approach in preserving the knowledge of old classes. This improvement can be attributed to the multivariate Gaussian sampling strategy, which provides diverse and accurate representations for old classes, and the similarity-based feature compensation mechanism, which minimizes the deviation between the generated old class features and the updated classifier. Together, these two components synthesize more effective retrospective features, aiding in the reconstruction of the classification decision boundaries.

Impact of class order and random seed. We investigate the impact of different class orders and random seeds on model performance. Figure 8(a) illustrates our method's performance across various class orders on CIFAR-100, where we rearrange the class order using three different random seeds. The median, upper, and lower bounds in the Figure indicate that our method shows slight performance fluctuations across different class orders, demonstrating low sensitivity to these variations. Additionally, Figure 8(b) presents our method's performance under different random seeds. We adjust the randomness of the model training process using three different seeds while keeping the class order consistent. The results show that our method is slightly sensitive to variations in random seeds. Furthermore, as the number of incremental phases increases, the model's sensitivity further decreases. This reduced sensitivity can be attributed to the fact that our method efficiently synthesizes retrospective features through a series of operations, making it more robust to resist catastrophic forgetting.

Effect of K on incremental learning performance. Different values of K simulate various high-likelihood regions
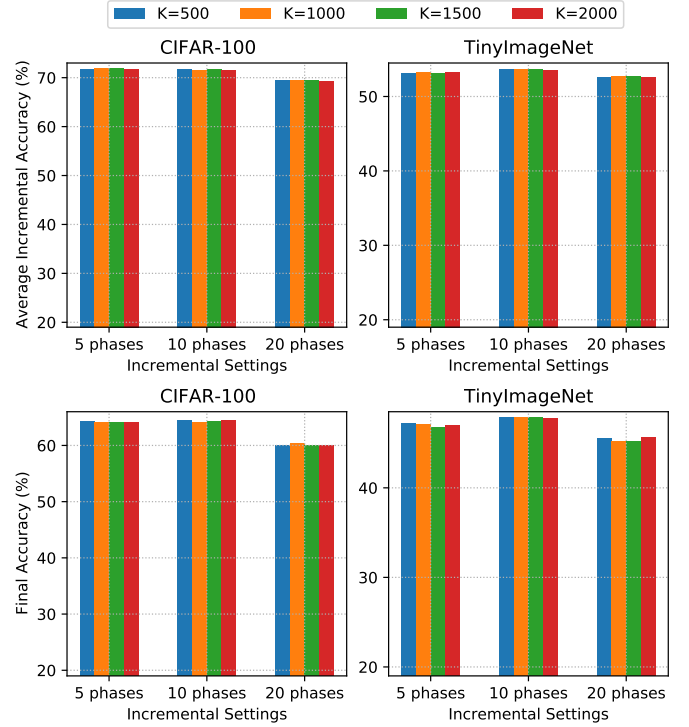

Fig. 9. Average incremental accuracy and final accuracy of our method under different K on CIFAR-100 and TinyImageNet.

for feature sampling in MGS. To evaluate the impact of K on the incremental learning performance of our model, we compared the average incremental accuracy and final classification accuracy under different K values, as shown in Figure 9. The results indicate that when K is set to 500, 1000, 1500, and 2000, our method achieves almost consistent performance across different incremental settings and datasets. These results demonstrate the robustness of our method under a range of K values.

V. DISCUSSION

Storage memory and model inference analysis. During training, our method does not store any raw samples of old classes. Instead, it retains only the mean and covariance matrix of the high-dimensional features for each class. While this approach slightly increases the storage memory compared to methods like PASS and PRAKA, which save only one prototype (class mean) per class, the increase is slight. This memory increase can be particularly negligible when compared to methods like iCaRL, which stores raw pixel-level samples directly. Additionally, our method applies self-supervised label augmentation only to newly learned classes during training, consuming fewer resources, such as GPU memory, than methods like PASS and PRAKA, which use label augmentation for all classes.

During inference, as outlined in Section III-A, our method only retains the feature extractor and the standard classifier. This allows it to perform on par with non-incremental methods that learn all classes in a single session, resulting in the smallest model size and fastest inference speed. In contrast,

methods like PASS utilize classifiers based on label augmentation through rotation transformations, significantly increasing resource consumption during inference.

Impact of knowledge distillation. Our method performs knowledge distillation at both the feature level (Equation 16) and the logit level (Equation 17). Since knowledge distillation at the feature level is commonly used to prevent drastic changes of the feature extractor during learning new data [4], [5], [20], here we focus on the impact of logit distillation. Table VII reports the performance of our method with and without logit distillation. With logit distillation, our method achieves average precision improvements of 0.14%, 0.08%, and 0.2% in the three incremental settings of CIFAR-100, and 0.11%, 0.10%, and 0.27% on TinyImageNet. Additionally, the method with logit distillation also improves final classification accuracy compared to the version without logit distillation. These gains can be attributed to logit distillation's ability to enhance model stability by reducing drastic updates biased toward new classes, thereby improving overall NECIL performance.

VI. Conclusion

In this paper, we propose a novel non-exemplar class-incremental learning method that synthesizes retrospective features for previously learned classes effectively. Our approach utilizes multivariate Gaussian distributions to model the feature space of old classes and employs a feature generation strategy that samples from high-likelihood regions to reconstruct the decision boundaries of previously learned classes. Additionally, we introduce a feature compensation mechanism that integrates the most similar new class features with generated old class representations, enhancing the model's ability to distinguish between old and new classes. Experimental results demonstrate that our method significantly improves the efficiency of non-exemplar class-incremental learning, achieving state-of-the-art performance. We believe this work provides valuable insights for advancing the field of class-incremental learning.

References

[1] X.-S. Wei, Y.-Z. Song, O. Mac Aodha, J. Wu, Y. Peng, J. Tang, J. Yang, and S. Belongie, "Fine-grained image analysis with deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 8927–8948, 2021.

[2] Y. Hao, S. Wang, P. Cao, X. Gao, T. Xu, J. Wu, and X. He, "Attention in attention: Modeling context correlation for efficient video classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 7120–7132, 2022.

[3] J. Wang, Z. Tu, J. Fu, N. Sebe, and S. Belongie, "Guest editorial: Introduction to the special section on fine-grained visual categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 560–562, 2022.

[4] F. Zhu, X.-Y. Zhang, C. Wang, F. Yin, and C.-L. Liu, "Prototype augmentation and self-supervision for incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5871–5880.

[5] Y. Luo, H. Ge, Y. Liu, and C. Wu, "Representation robustness and feature expansion for exemplar-free class-incremental learning," *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[6] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[7] S. Wang, W. Shi, S. Dong, X. Gao, X. Song, and Y. Gong, "Semantic knowledge guided class-incremental learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 10, pp. 5921–5931, 2023.

[8] H. Lin, S. Feng, X. Li, W. Li, and Y. Ye, "Anchor assisted experience replay for online class-incremental learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 5, pp. 2217–2232, 2023.

[9] G. Petit, A. Popescu, H. Schindler, D. Picard, and B. Delezoide, "Fetril: Feature translation for exemplar-free class-incremental learning," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 3911–3920.

[10] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[11] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, "Gradient based sample selection for online continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[12] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.

[13] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 831–839.

[14] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *European conference on computer vision*. Springer, 2022, pp. 398–414.

[15] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in neural information processing systems*, vol. 30, 2017.

[16] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6619–6628.

[17] Q. Jodelet, X. Liu, Y. J. Phua, and T. Murata, "Class-incremental learning using diffusion model for distillation and replay," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3425–3433.

[18] D. Yu, M. Zhang, M. Li, F. Zha, J. Zhang, L. Sun, and K. Huang, "Contrastive correlation preserving replay for online continual learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 1, pp. 124–139, 2024.

[19] K. Zhu, K. Zheng, R. Feng, D. Zhao, Y. Cao, and Z.-J. Zha, "Self-organizing pathway expansion for non-exemplar class-incremental learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 204–19 213.

[20] W. Shi and M. Ye, "Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1772–1781.

[21] X. Du, Z. Wang, M. Cai, and Y. Li, "Vos: Learning what you don't know by virtual outlier synthesis," *Proceedings of the International Conference on Learning Representations*, 2022.

[22] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations," in *International conference on machine learning*. PMLR, 2020, pp. 5714–5724.

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

[24] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[25] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

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

[27] M. Cheng, H. Wang, and Y. Long, "Meta-learning-based incremental few-shot object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2158–2169, 2021.

[28] L. Bai, H. Song, T. Feng, T. Fu, Q. Yu, and J. Yang, "Revisiting class-incremental object detection: An efficient approach via intrinsic characteristics alignment and task decoupling," *Expert Systems with Applications*, vol. 257, p. 125057, 2024.

[29] K. Joseph, J. Rajasegaran, S. Khan, F. S. Khan, and V. N. Bala-subramanian, "Incremental object detection via meta-learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 9209–9216, 2021.

[30] B. Yuan and D. Zhao, "A survey on continual semantic segmentation: Theory, challenge, method and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[31] Q. Guo, H. Song, J. Fan, D. Ai, Y. Gao, X. Yu, and J. Yang, "Portal vein and hepatic vein segmentation in multi-phase mr images using flow-guided change detection," *IEEE Transactions on Image Processing*, vol. 31, pp. 2503–2517, 2022.

[32] Q. Guo, H. Song, C. Wang, J. Fan, D. Ai, Y. Gao, X. Yu, and J. Yang, "Segmentation of 3d anatomically diffused tissues in magnetic resonance images through edge-preserving constrained center-free fuzzy c-means," *IEEE Transactions on Fuzzy Systems*, 2024.

[33] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.

[34] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.

[35] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *International conference on machine learning*. PMLR, 2018, pp. 4548–4557.

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

[37] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3366–3375.

[38] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 3, pp. 651–663, 2018.

[39] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[40] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[41] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, and C.-C. J. Kuo, "Class-incremental learning via deep model consolidation," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 1131–1140.

[42] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," *Advances in neural information processing systems*, vol. 32, 2019.

[43] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[44] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[45] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[47] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 532–547.

[48] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 233–248.

[49] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 831–839.

[50] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S.-T. Xia, "Maintaining discrimination and fairness in class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 208–13 217.

[51] Y. Liu, S. Parisot, G. Slabaugh, X. Jia, A. Leonardis, and T. Tuytelaars, "More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 699–716.

[52] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. v. d. Weijer, "Semantic drift compensation for class-incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6982–6991.

[53] F. Zhu, Z. Cheng, X.-Y. Zhang, and C.-l. Liu, "Class-incremental learning via dual augmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 306–14 318, 2021.

[54] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.