SRIL: Selective Regularization for Class-Incremental Learning

Jisu Han[1], Jaemin Na[2], and Wonjun Hwang[1]

[1] Dept. of Artificial Intelligence, Ajou University, Korea
[2] Tech. Innovation Group, KT, Korea
{jisu3709, wjhwang}@ajou.ac.kr, jaemin.na@kt.com

Abstract. Human intelligence gradually accepts new information and accumulates knowledge throughout the lifespan. However, deep learning models suffer from a catastrophic forgetting phenomenon, where they forget previous knowledge when acquiring new information. Class-Incremental Learning aims to create an integrated model that balances plasticity and stability to overcome this challenge. In this paper, we propose a selective regularization method that accepts new knowledge while maintaining previous knowledge. We first introduce an asymmetric feature distillation method for old and new classes inspired by cognitive science, using the gradient of classification and knowledge distillation losses to determine whether to perform pattern completion or pattern separation. We also propose a method to selectively interpolate the weight of the previous model for a balance between stability and plasticity, and we adjust whether to transfer through model confidence to ensure the performance of the previous class and enable exploratory learning. We validate the effectiveness of the proposed method, which surpasses the performance of existing methods through extensive experimental protocols using CIFAR-100, ImageNet-Subset, and ImageNet-Full.

Keywords: Class Incremental Learning · Knowledge Distillation · Weight Interpolation

1 Introduction

The successful development of deep learning has created many businesses and has applied to the real-world. However, many studies on deep learning have been evaluated in limited experimental settings. To achieve human-level goals, consideration of changes in real-world environments is required. In this respect, continual learning has recently been receiving a lot of attention from the artificial intelligence community. Continual learning is a methodology that can continuously learn about real-world situations where visual characteristics change, such as robot vision and autonomous driving systems [33]. Although recent studies on continual learning have shown promising results, they still suffer from the problem of losing previous knowledge in the process of learning new knowledge. This phenomenon is called catastrophic forgetting [10, 25], and this problem results

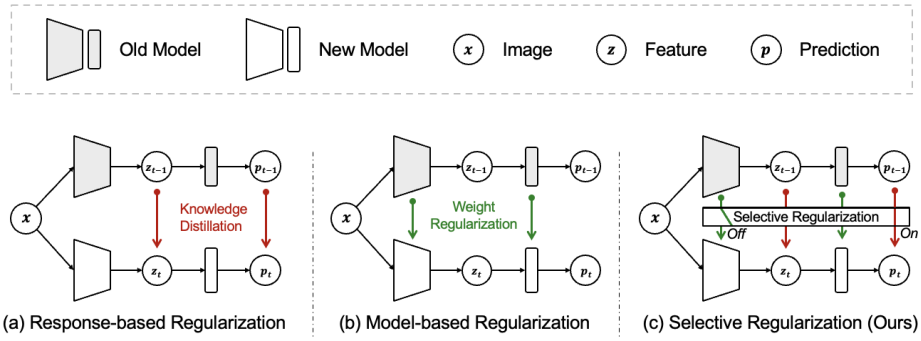

Fig. 1: Comparison of regularization methods in continual learning by location and application. **(a)** Response-based methods regularize the model's outputs through knowledge distillation. **(b)** Model-based methods regularize model's parameters by importance. **(c)** Our approach selectively applies regularization based on gradient and confidence throughout the learning process rather than maintaining constant regularization.

in performance degradation of deep learning models. One of the major concerns of recent continual learning approaches is mitigating the catastrophic forgetting problem to prevent such performance deterioration.

Class-Incremental Learning (CIL) is a representative scenario of continual learning that seeks to learn new classes that have not been learned before while preserving knowledge from old classes. To mitigate the catastrophic forgetting that occurs as new knowledge is learned, existing methods adopt regularization methods that leverage previous learning models. The major approaches for applying regularization to CIL can be divided into response-based regularization methods [7, 13, 17, 30], and model-based regularization methods [1, 4, 5, 18] (as illustrated in Fig. 1). However, both approaches maintain regularization throughout training and depend on regularization hyperparameters to strike a balance between stability and adaptability. The stability-plasticity dilemma highlights the challenge of forgetting old knowledge while acquiring new knowledge, and the risk of rejecting new knowledge to preserve old knowledge. Despite this, achieving a balance between stability and adaptability remains a topic requiring further investigation. Here, we propose a simple but effective method to selectively apply regularization, accommodating new knowledge while ensuring stability.

In this paper, we introduce a selective regularization method named **SRIL** that leverages knowledge distillation and weight interpolation. First, we propose an asymmetric feature distillation method called **Gradient-based Feature Distillation**. Our intuition begins with pattern completion and pattern separation in cognitive science [28, 32]. The pattern completion means integrating memories by judging that they are existing knowledge when they are newly readjusted by external stimuli, and pattern separation is the process of making memories distinct from existing memories by judging that they are different

from the contents in memory. We intend to achieve realignment of knowledge by adopting an asymmetric learning strategy for the previously learned class and the newly learned class in terms of pattern completion and pattern separation. In the process of learning a new class, we utilize the gradients of the knowledge distillation and classification losses to determine whether knowledge distillation is beneficial or harmful. Then, we generate a mask for the channel of each intermediate feature and exploit the mask to apply feature distillation to conflicting channels for the old and new classes. Selective feature distillation through masks generated by this process considers different characteristics by taking an asymmetric strategy for old and new classes.

Meanwhile, to deal with the stability-plasticity dilemma, we introduce a **Confid-ence-aware Weight Interpolation**, which determines whether the model retains old knowledge well and performs selective regularization according to the determined result. Here, we determine whether the new model maintains previous knowledge through the confidence of the new model on old data. The existing weight interpolation method is used to improve the generalization performance of a model as a method for approximating an ensemble [37, 38]. In contrast, we ensure that the upper bound of the loss for the old data does not increase by making the new model close to the old model in the weight space to ensure model stability. However, not moving away from the weight space can instead become a constraint on learning new knowledge. Therefore, if the new model's confidence in the old data exceeds a certain value based on the old model's confidence, we remove regularization to enable exploratory learning and balance stability and plasticity. Overall, our contributions are as follows:

- We propose a new asymmetric Gradient-based Feature Distillation (GFD) to consider feature characteristics between old and new classes.
- We propose Confidence-aware Weight Interpolation (CWI) to strike a balance between stability and plasticity. CWI ensures the stability of the model and enables exploratory learning by operating selectively.
- We achieve comparable performance to the recent state-of-the-art methods and validate our methods through various experimental settings in CIFAR-100, ImageNet-Subset, and ImageNet-Full.

2 Related Work

Class-Incremental Learning. CIL is largely classified into three types according to how to solve the problem. (i) Regularization based methods regularize the difference between the parameters of the previous network and current network to maintain a low loss area for previous tasks [1, 5, 18]. (ii) Dynamic architecture method is a method that continuously adds sub-networks or expands the network dynamically through pruning and retraining [15, 31, 40]. (iii) Replay based method allows a limited amount of memory composed of data from previous tasks. As a representative method, studies are being conducted to transfer dark knowledge of previously trained networks using knowledge distillation [3, 13, 30] or to improve performance through efficient memory management strategies [22, 23].

Knowledge Distillation. Knowledge distillation (KD) approaches have been proposed to improve the performance of lightweight models by transferring knowledge from large models to small models [12]. Similarly, many studies have been conducted in CIL to solve the catastrophic forgetting problem by transferring the knowledge of the old model through KD when learning new data. LwF [20] applies KD to CIL and uses KD so that the output of the model follows the output of the previous trained model. iCaRL [30] stores a limited amount of old data as exemplars and applies KD to new data to maintain representation. LUCIR [13] proposed feature distillation in the embedding space rather than prediction by logit. PODNet [7] applied distillation not only in the embedding space but also in the intermediate features with pooling. GeoDL [34] considers the training trajectory on sub-dimension through an additional regularization term that considers the geodesic between features in the embedding space. AFC [17] defines the gradient magnitude of each channel in intermediate features as importance and applies differentiated feature distillation for each channel. Furthermore, SnD [41] selects KD between zero-shot and continual CLIP [29] in the vision- language model. In our work, rather than selecting between two models, we choose whether to apply KD within a single model.

Weight Space Ensemble. Ensemble is the most basic method used to improve the performance of a model in deep learning. However, existing ensemble methods are accompanied by an increase in model size or additional cost in the inference process. SWA [16] showed that by performing weight averaging periodically along the trajectory of SGD in a single model, the model improves in more generalized performance and converges to a wide minima. WiSE-FT [38] demonstrated that the weight space ensemble from the zero-shot model improves the robustness of the model in the fine-tuning process. In a recent study of model soups [37], the performance of the model was improved without additional inference time by applying the weight space-ensemble from the fine-tuned multiple model. Inspired by these weight space ensemble methods, there have been attempts to apply them in CIL. It either transfer the knowledge of new classes to the old model by updating the old model via exponential moving average [35], or apply consistent weight interpolation for the entire learning process [9].

3 Method

3.1 Problem Definition

CIL aims to improve performance for all classes by starting from learning a limited number of classes and sequentially adding new classes. These sequential stages are called tasks. In CIL scenario, we have sequential training dataset $D^t = \{x_i^t, y_i^t\}_{i=1}^N$ and exemplar $\mathcal{E}^t = \left\{x_j^{1:t-1}, y_j^{1:t-1}\right\}_{j=1}^R$ containing limited data for previous tasks, where t, x and y denote task, input image and labels, respectively, and $1:t$ denotes from the 1-st task to the t-th task. Each task's classes are disjoint in CIL scenario, *i.e.*, $y^{1:t-1} \cap y^t = \varnothing$. Since our goal is to maintain the performance of previous tasks while learning new tasks, the optimization

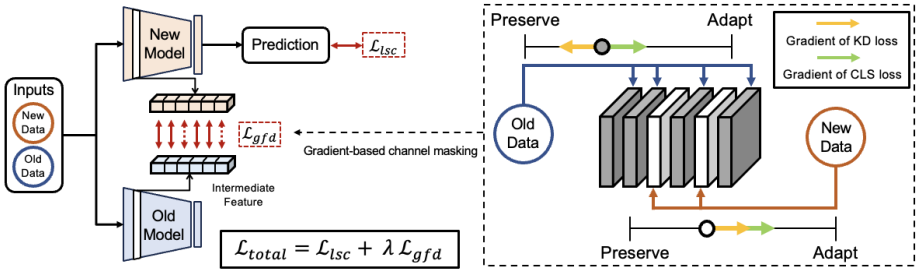

Fig. 2: Illustration of GFD. Feature distillation by new data is applied when the direction of the gradients of KD loss and classification loss are the same, and feature distillation by old data is applied when the direction is opposite.

problem is defined as follow:

$$\begin{aligned} &\arg\min_{\theta} \ \mathbb{E}_{(x,y)\sim D^{1:t}} \left[\mathcal{L}\left(x, y; \theta\right)\right] \\ &s.t. \ \mathbb{E}_{(x,y)\sim D^{t}\cup\mathcal{E}^{t}} \left[\mathcal{L}\left(x, y; \theta\right)\right] < \epsilon, \end{aligned} \tag{1}$$

where the $\mathcal{L}$ is the loss function (e.g., Cross-Entropy), and the θ is parameters of the model. Since the model converges while learning from current task data and exemplar, there exists some small number ϵ higher than loss.

3.2 Gradient-based Feature Distillation

We use the gradients of classification and KD losses for pattern integration and separation, inspired by [8,42]. We consider that each channel contains information about patterns in the features of the model, therefore we decide whether to apply feature distillation to each channel based on the similarity between gradients of the classification and the KD losses. To achieve this, we build a binary mask M where activation is determined by the cosine similarity of the two gradients.

$$M_{l,c} = \begin{cases} 1, \ if \ \ cos(\nabla_{z_{l,c}}\mathcal{L}_{kd}, \nabla_{z_{l,c}}\mathcal{L}_{lsc}) \geq 0 \\ 0, \ \text{otherwise} \end{cases}, \tag{2}$$

where $cos(\cdot)$ is a cosine similarity. The $\nabla_{z_{l,c}}\mathcal{L}_{kd}$ and $\nabla_{z_{l,c}}\mathcal{L}_{lsc}$ are the gradients of KD and classification losses for each channel c for the features $z_{l,c}$ of the l-th layer, respectively. The binary mask M is activated when directions of the two gradients for the new class data are equal. If the mask is activated, feature distillation is applied. Using the new class data, pattern completion is achieved through KD when the gradients of KD and classification losses have same directions for each channel of the features. On the other hands, applying KD can inhibit the classification performance when the two gradients have different directions [8]. Therefore, in this case, we rule out the KD to achieve the

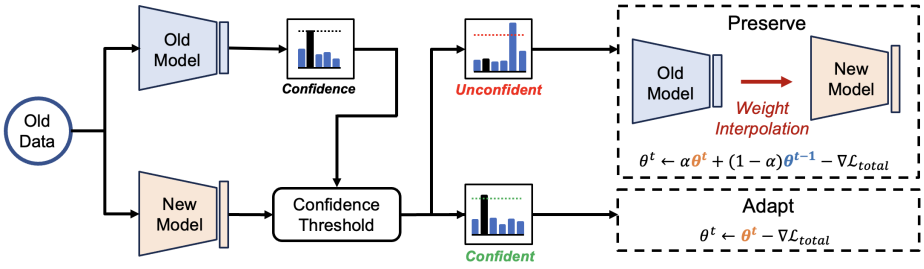

Fig. 3: Illustration of CWI. Determine the stability of the new model based on the confidence of the old model trained on old data. In the subsequent learning process, which has stable performance on previously learned data, it pursues exploratory learning on a new class by eliminating regularization by weight interpolation.

pattern separation. Nonetheless, not regularizing at all for a particular channel may cause problems with the stability of the model. Additionally, in the process of pattern separation, from the perspective of pursuing differentiation from existing knowledge, maintaining the existing pattern can lead to complete differentiation. To achieve this, we apply a mask that is opposite to the binary mask to the old classes data while applying the binary mask to the new data. Finally, the objective for our GFD is defined as follows:

$$\mathcal{L}_{gfd} = M \cdot \mathcal{L}_{fd}(x_{new}) + (1 - M) \cdot \mathcal{L}_{fd}(x_{old}), \tag{3}$$

where,

$$\mathcal{L}_{fd} = \sum_{l=1}^{L} \sum_{c=1}^{C} \left|\left| z_{l,c}^{t} - z_{l,c}^{t-1} \right|\right|_F^2, \tag{4}$$

in which x_{new} and x_{old} are new classes data and old classes data, respectively. L is the total number of layers and C is the total number of channels in each layer. $||\cdot||_F$ is the frobenius norm and $\mathcal{L}_{fd}$ is a channel-wise feature distillation loss, and we normalize each feature for learning stability.

3.3 Confidence-aware Weight Interpolation

Under the stability-plasticity dilemma [26, 36] in the CIL, simultaneously improving stability and plasticity is a challenging problem. However, continual learning considering the balance between the stability and plasticity is essential to improve overall performance. Regarding stability-plasticity, previous methods [7,17] have had difficulties in ensuring stability enough to expect satisfactory performance improvement despite performing feature distillation that directly transfer the representation. To overcome these limitations, we interpolate the weights of the old model into the new model, thereby preventing loss changes

for the old data and ensuring stability for the previous task. In addition, we pursue the plasticity through exploratory learning after the stability of the model is guaranteed enough.

Consequently, we introduce a confidence-aware weight interpolation to improve both stability and plasticity. As the model learns on the new data, we interpolate the weights of the new model through the weights of the old model to prevent the model from getting out of the low error area due to a large difference between parameters with the old model. We update the parameters of the new model θ^t through the parameters of the old model θ^{t-1}, which is expressed as:

$$\theta^t \leftarrow \beta\theta^t + (1-\beta)\theta^{t-1}, \tag{5}$$

where $\beta \in [0,1]$ is an interpolation parameter. However, weight interpolation can adversely affect plasticity from a continuous perspective. To solve this problem, we use confidence in the old data to enable exploratory learning without weight interpolation when the model has sufficient knowledge about the old data. We adaptively adjust the interpolation parameter β, to apply weight interpolation when the confidence of the new model and the old model for the old data is above the threshold, and not to apply when the confidence is below the threshold. Therefore, interpolation parameter β is defined as follows:

$$\beta = \begin{cases} \alpha, \ if \ conf(x_{old};\theta^{t-1}) - conf(x_{old};\theta^t) \geq \delta \\ 1, \ \text{otherwise} \end{cases}, \tag{6}$$

in which confidence denotes $conf(x;\theta) = \mathbb{E}\left[p_y(x;\theta)\right]$ and $\alpha \in [0,1]$ is the hyperparameter, where $p_y(x;\theta)$ is the predicted probability for the ground truth y, and δ is the threshold. Since the prediction distribution also changes as the class increases, we set $\delta = \lambda_{th}\delta^t$, where λ_{th} is the hyperparameter, and adaptive factor is $\delta^t = (n^t)^2 / n^{1:t}$ and n^t is the number of classes in task t.

Fig. 4 shows the difference in confidence between the old and new model for the previous 50 classes data in the process of learning an additional 10 classes from the model trained for 50 classes on CIFAR-100. In the initial learning process, while learning new class data, the prediction performance for old classes is destroyed, but the difference in confidence for old classes is reduced through feature distillation and weight interpolation. In addition, in the later learning process, when the difference in confidence between the two models is lower than the threshold, it is confirmed that a certain level of confidence is guaranteed by exploratory learning on new class data without weight interpolation.

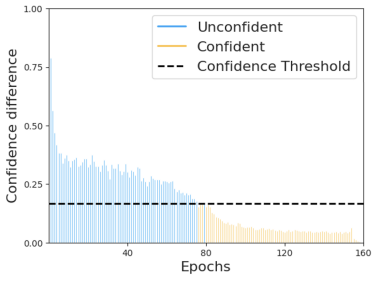

Fig. 4: Difference in confidence based on training epochs.

Algorithm 1 SRIL: Selective Regularization Algorithm

Input: training set D^t, exemplar set $\mathcal{E}^t$, old model's parameters θ^{t-1}, new model's parameters θ^t, interpolation parameter α, distill factor λ^t_{gfd}, learning rate γ
Output: θ^t
$\theta^t \leftarrow \theta^{t-1}$ // initialize new model
for e in $\{1, \cdots, E\}$ **do**
  sample a mini-batch $(x, y) \sim D^t \cup \mathcal{E}^t$
  zero initialization binary mask M
  compute gradient from new classes (x_{new}, y_{new})
  for l in $\{1, \cdots, L\}$ **do**
    if $cos(\nabla_{z_{l,c}} \mathcal{L}_{kd}, \nabla_{z_{l,c}} \mathcal{L}_{lsc}) \geq 0$ **then**
      $M_{l,c} = 1$
    end if
  end for
  $\mathcal{L}_{gfd} = M \cdot \mathcal{L}_{fd}(x_{new}) + (1 - M) \cdot \mathcal{L}_{fd}(x_{old})$
  $\mathcal{L}_{total} = \mathcal{L}_{lsc} + \lambda^t_{gfd} \mathcal{L}_{gfd}$
  compute gradient $\nabla \mathcal{L}_{total}$
  compute confidence from old classes (x_{old}, y_{old})
  if $conf(x_{old}; \theta^{t-1}) - conf(x_{old}; \theta^t) \geq \delta$ **then**
    $\theta^t \leftarrow \alpha \theta^t + (1 - \alpha)\theta^{t-1}$
  end if
  update parameters $\theta^t \leftarrow \theta^t - \gamma \nabla \mathcal{L}_{total}$
end for
update exemplar from $D^t \cup \mathcal{E}^t$

3.4 Overall Framework

Our SRIL consists of Gradient-based Feature Distillation and Confidence-aware Weight Interpolation. GFD is a KD method for considering different expressive features of the old class and new class. CWI balances between stability and plasticity. Our overall loss is follow as:

$$\mathcal{L}_{total} = \mathcal{L}_{lsc} + \lambda^t_{gfd} \mathcal{L}_{gfd}, \tag{7}$$

in which distill factor is $\lambda^t_{gfd} = \lambda_{gfd} \cdot \lambda^t$, where λ_{gfd} is a hyperparameter, and adaptation factor is $\lambda^t = \sqrt{n^{1:t}/n^t}$. λ^t is a well-used adaptation factor in CIL [7, 13, 17, 34]. We also use local similarity classifiers (LSC) following the previous works [7, 17]. LSC allows K proxies for each class. The logit for each class c is obtained as the average of K proxies.

$$s_{\mathsf{c},k} = \frac{exp\left\langle \phi_{\mathsf{c},k}, z \right\rangle}{\sum_i exp\left\langle \phi_{\mathsf{c},i}, z \right\rangle} \quad \hat{\mathbf{y}}_{\mathsf{c}} = \sum_k s_{\mathsf{c},k} \left\langle \phi_{\mathsf{c},k}, z \right\rangle, \tag{8}$$

where ϕ is a classifier weight, $s_{\mathsf{c},k}$ and $\hat{\mathbf{y}}_{\mathsf{c}}$ denote the probability by each proxy and the probability for each class. We use LSC loss [7] which is based on NCA

loss [27] as follows:

$$\mathcal{L}_{lsc} = \left[-\log \frac{exp(\eta(\hat{\mathbf{y}}_y - \varepsilon))}{\sum_{i \neq y} exp(\eta \hat{\mathbf{y}}_i)} \right]_+, \tag{9}$$

where η and ε are learnable parameters and small margin, respectively, and hinge $[\cdot]_+$ denotes $max(0, \cdot)$.

4 Experiment

To verify the experimental validity, we compare our performance with the state-of-the-art replay-based methods that address the same problem as ours and use KD. Exemplar stores 20 fixed samples for each previous class and uses herding selection [30]. We report both results of model predictions and nearest-mean-of-exemplars [30] classification, denoted as CNN and NME, respectively.

4.1 Experimental Setup

Dataset and protocol. We evaluate our method on CIFAR-100 [19], ImageNet-Subset [6, 7, 13] and ImageNet-Full [6]. CIFAR-100 consists of 60,000 images of 32×32 pixels with 100 classes. It contains 500 training data and 100 test data for each class. ImageNet is a large-scale classification dataset, including 1.28 million images and 50k test set, and consists of 1,000 classes. ImageNet-Full uses all classes of ImageNet, and ImageNet-Subset refers to a dataset that randomly extracts 100 classes out of 1,000 classes. We use the same random seed and class order as the previous methods [7, 17] for fair comparison. The mean and standard deviation of the performance are obtained as a result of three different class order. After learning a part of the entire class for CIL setting, we gradually learn a certain number of classes. Feature-based distillation methods transfer representation directly, so representation of previously learned models is important. Therefore, it shows better performance than response-based distillation in small task incremental learning settings, where it learns about half of the total class in advance and then gradually learns a small number of classes [7, 13, 17].

Implementation details. For reproducibility, we conducted an experiment based on the PODNet [7] code. All experiments are conducted on TITAN-Xp GPU. We follow the experimental setup of PODNet [7]. In the CIFAR-100 experiment, we use the ResNet-32 [11] architecture. We trained the model for 160 epochs with SGD with momentum of 0.9 and used a batch size of 128 and a weight decay of 0.0005. We use a cosine annealing learning rate scheduler with an initial learning rate of 0.1. The hyperparameters are $\alpha = 0.995$, $\lambda_{th} = 0.1$, and $\lambda_{gfd} = 2$. For ImageNet-Subset and ImageNet-Full, we use the ResNet-18 [11] architecture. We trained the model for 90 epochs with SGD with momentum of 0.9 and used a cosine annealing learning rate scheduler with an initial learning rate of 0.1 and 0.05, respectively. We use a batch size of 64 and a weight decay of 0.0001. α and λ_{th} use 0.999 and 0.1. λ_{gfd} uses 5 and 7 for ImageNet-Subset and ImageNet-Full, respectively.

Table 1: Comparison of average accuracy (%) between state-of-the-art methods using exemplar and our SRIL on CIFAR-100. The best accuracy is indicated in bold.

		CIFAR-100			
	Classifier	50 tasks	25 tasks	10 tasks	5 tasks
New classes per task		1	2	5	10
BiC [39]	CNN	47.09 ± 1.48	48.96 ± 1.03	53.21 ± 1.01	56.86 ± 0.46
LUCIR [13]	CNN	49.30 ± 0.32	57.57 ± 0.23	61.22 ± 0.69	64.01 ± 0.91
PODNet [7]	CNN	57.98 ± 0.46	60.72 ± 1.36	63.19 ± 1.16	64.83 ± 0.98
GeoDL [34]	CNN	52.28 ± 3.91	60.21 ± 0.46	63.61 ± 0.81	65.34 ± 1.05
DDE [14]	CNN	-	-	64.12 ± 1.40	65.42 ± 0.72
AANet [21]	CNN	-	62.31 ± 1.02	64.31 ± 0.90	66.31 ± 0.87
CSCCT [2]	CNN	58.80 ± 1.92	61.10 ± 1.12	63.72 ± 1.06	-
AFC [17]	CNN	62.18 ± 0.57	63.89 ± 0.93	64.98 ± 0.87	$\mathbf{66.49 \pm 0.81}$
SRIL (Ours)	CNN	$\mathbf{63.64 \pm 1.02}$	$\mathbf{64.36 \pm 1.16}$	$\mathbf{65.11 \pm 0.82}$	66.21 ± 0.89
iCaRL [30]	NME	44.20 ± 0.98	50.60 ± 1.06	53.78 ± 1.16	58.08 ± 0.59
LUCIR [13]	NME	48.57 ± 0.37	56.82 ± 0.19	60.83 ± 0.70	63.63 ± 0.87
PODNet [7]	NME	61.40 ± 0.68	62.71 ± 1.26	64.03 ± 1.30	64.48 ± 1.32
AFC [17]	NME	62.58 ± 1.02	64.06 ± 0.73	64.29 ± 0.92	65.82 ± 0.88
SRIL (Ours)	NME	$\mathbf{63.84 \pm 0.98}$	$\mathbf{64.87 \pm 0.91}$	$\mathbf{66.25 \pm 1.16}$	$\mathbf{67.13 \pm 1.03}$

4.2 Main Results

CIFAR-100. We compare the average accuracy with the state-of-the-art methods that use the exemplar and KD [2, 7, 13, 14, 17, 21, 30, 34, 39] in Table 1. For a method based on other framework, GeoDL [34], DDE [14], AANet [21] and CSCCT [2] are a performance result based on PODNet (CNN) [7]. We learn 50 classes in 0-th task and increase the classes of fixed numbers for each task. As a result of the experiment using the NME classifier, the performance improvement of 0.81–1.96 percent points in all experimental environments. Experiments using a CNN classifier show that 5 tasks had lower performance of 0.28 percent points than closest state-of-the-art, but 10 tasks, 25 tasks and 50 tasks show performance improvements of 0.13–1.46 percent points compared to the state-of-the-art method.

ImageNet-Subset/Full. In Table 2, we compare our method with the state-of-the-art methods in ImageNet-Subset/Full benchmarks. In this experiment, we employ CNN classifier following the previous evaluation protocols [7, 17] since the kNN-based NME classifier has drawbacks in dealing with a large number of classes. In the ImageNet-Subset, our method outperforms the previous methods in all experiments and achieves 0.78–1.99% higher performance than the previous state-of-the-art method, AFC. In the experiment for ImageNet-Full, we achieved the comparable results compared with the recent CIL methods. Specifically, we attain 0.07% higher accuracy than the previous state-of-the-art method, AFC, in the 5-task scenario.

Table 2: Comparison of average accuracy (%) between state-of-the-art methods and SRIL on ImageNet-Subset and ImageNet-Full. The best accuracy is represented in bold, and the second-best accuracy is underlined.

	ImageNet-Subset				ImageNet-Full	
	50 tasks	25 tasks	10 tasks	5 tasks	10 tasks	5 tasks
New classes per task	1	2	5	10	50	100
iCaRL [30]	54.97	54.56	60.90	65.56	46.72	51.36
BiC [39]	46.49	59.65	65.14	68.97	44.31	45.72
LUCIR [13]	55.44	60.81	65.83	67.07	59.92	64.34
Mnemonics [23]	–	69.74	71.37	72.58	63.01	64.54
PODNet [7]	62.48	68.31	74.33	75.54	64.13	66.95
DDE [14]	–	–	75.41	76.71	64.71	66.42
GeoDL [34]	–	71.72	73.55	73.87	64.46	65.23
AANet [21]	–	71.78	75.58	<u>76.96</u>	64.85	67.73
CSCCT [2]	–	68.91	74.35	76.41	–	–
AFC [17]	<u>72.08</u>	<u>73.34</u>	<u>75.75</u>	76.87	**67.02**	<u>68.90</u>
SRIL (Ours)	**72.86**	**75.33**	**77.28**	**78.57**	<u>66.87</u>	**68.97**

Table 3: Comparison of average accuracy (%) without class balanced fine-tuning on CIFAR-100. ∗ Reproduced by the official code.

	50steps	25steps	10steps	5steps
New classes per task	1	2	5	10
PODNet w/o CBF	58.71 ± 0.28	59.68 ± 0.37	59.75 ± 0.68	60.70 ± 1.13
AFC w/o CBF	60.71 ± 0.59	60.51 ± 0.73	59.99 ± 0.87	60.49 ± 0.77
SRIL w/o CBF	$\mathbf{63.37 \pm 0.36}$	$\mathbf{62.30 \pm 0.24}$	$\mathbf{62.59 \pm 0.16}$	$\mathbf{64.40 \pm 0.76}$

4.3 Ablation studies

Effect of class balanced finetuning Existing methods [7, 13, 17] update the classifier through an under-sampled balanced set to solve the class imbalance problem that occurs in the process of learning new data and exemplar, and this is called class balanced fine-tuning (CBF). The proposed CWI uses confidence to find a compromise between plasticity and stability of the model. Therefore, the performance changes in the absence of CBF is additionally confirmed in Table 3.

Stability-plasticity analysis. We analyze Forgetting Measure (FM) [5] and Intransigence Measure (IM) [5] to determine how the stability and plasticity of SRIL differ from other feature distillation methods in Fig. 5. FM is a measure of stability and is defined as the average difference between the maximum accuracy of the previous model and the accuracy of the current model for all tasks. The lower the FM, the higher the stability. IM is a measure of plasticity and is defined as the average difference in accuracy for new data between models learned

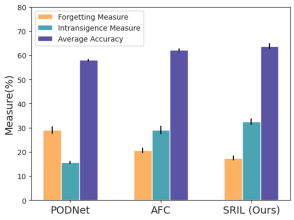

Fig. 5: Stability-plasticity analysis.

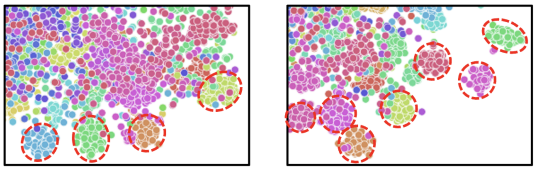

Fig. 6: t-SNE visualization.

Table 4: Effect of CWI

Method	50 tasks			5 tasks		
	Old	New	Avg.	Old	New	Avg.
Finetuning	38.71	92.79	40.02	36.71	87.13	45.76
GFD	58.61	73.32	59.21	61.27	71.60	65.34
w/ WiSE-FT [38]	63.26	63.14	63.38	**64.69**	49.58	65.39
w/ CWI (Ours)	**63.34**	**63.47**	**63.64**	63.11	**67.04**	**66.21**

Table 5: Effect of GFD

Method	50 tasks			5 tasks		
	Old	New	Avg.	Old	New	Avg.
CWI	50.17	86.57	51.22	50.91	83.55	59.62
w/ GFD (new data)	62.58	**65.44**	62.94	62.26	66.72	65.62
w/ GFD (old data)	62.72	63.10	63.03	**63.71**	65.45	66.54
w/ GFD (both)	**63.34**	63.47	**63.64**	63.11	**67.04**	**66.21**

without regularization and models with regularization of all tasks. The lower the IM, the higher the plasticity. As a result, PODNet [7] has strengths in plasticity, and AFC [17] and SRIL have strengths in stability. We explain the results of the main experiment through the results of stability-plasticity analysis. Our method has high stability by using two selective regularization methods, feature distillation and weight interpolation, and improves overall performance by maintaining performance on the old task. As a result, there was a large performance gain in small task settings, where the class increased by a small number. On ImageNet-Full, we achieved a similar level of performance to the existing method.

Feature visualization. We visualize the embedded features of 0-th task data for model trained up to the 50-th task on CIFAR-100 using t-SNE [24] in Fig. 6. Compared to previous work [7, 13, 17], our method achieves to construct more cohesive clusters for each class. These results show that our method works successfully on NME classifier and small exemplar size.

Effect of each component. We perform an ablation study to examine the effectiveness of each component of our methods in Table 4 and 5. We conduct experiments by adding each proposed method from finetuning without any regularization. In the case of GFD, the experiments demonstrate an overall performance improvement. For CWI, a significant performance gain is observed, particularly in experiments involving the learning of 50 tasks, which consist of smaller tasks. Furthermore, the highest performance is achieved when both methods are used together. The weight interpolation method [38], which maintains consistency over the learning process, contributes to stability; however, it can impair the plasticity due to the stability-plasticity trade-off in continual learning. We address this issue by achieving a balance between stability and plasticity with our distinct confidence-aware weight interpolation method. We applied KD

Table 6: Effect of memory budget on CIFAR-100 with 50 tasks. $*$ Reproduced by the official code.

Memory per class	5	10	20	50
iCaRL [30]	16.44	28.57	44.20	48.29
BiC [39]	20.84	21.97	47.09	55.01
LUCIR (NME) [13]	21.81	41.92	48.57	56.09
LUCIR (CNN) [13]	22.17	42.70	49.30	57.02
PODNet (NME) [7]	48.37	57.20	61.40	62.27
PODNet (CNN) [7]	35.59	48.57	57.98	63.69
AFC* (NME) [17]	44.33	57.27	62.33	63.83
AFC (CNN) [17]	44.66	55.78	62.18	65.07
SRIL (NME)	$\mathbf{54.09 \pm 1.45}$	$\mathbf{61.99 \pm 0.66}$	$\mathbf{63.84 \pm 0.98}$	65.31 ± 1.30
SRIL (CNN)	50.23 ± 1.05	60.47 ± 1.31	63.64 ± 1.02	$\mathbf{65.31 \pm 1.01}$

Table 7: Effect of varying initial task sizes on CIFAR-100 with 1 class per tasks.

	80 tasks	70 tasks	60 tasks	50 tasks
Initial task size	20	30	40	50
iCaRL [30]	41.28	43.38	44.35	44.20
BiC [39]	40.95	42.27	45.18	47.09
LUCIR (NME) [13]	40.81	46.80	46.71	48.57
LUCIR (CNN) [13]	41.69	47.85	47.51	49.30
PODNet (NME) [7]	49.03	55.30	57.89	61.40
PODNet (CNN) [7]	47.68	52.88	55.42	57.98
AFC (NME) [17]	51.31	57.05	60.06	62.58
AFC (CNN) [17]	52.90	57.61	60.27	62.18
SRIL (NME)	$\mathbf{53.21 \pm 1.42}$	58.30 ± 1.49	61.21 ± 1.38	$\mathbf{63.84 \pm 0.98}$
SRIL (CNN)	53.18 ± 1.11	$\mathbf{58.93 \pm 1.84}$	$\mathbf{61.29 \pm 1.64}$	63.64 ± 1.02

on new data when it does not conflict with the classification loss during the learning of new classes. This design improves plasticity, and we apply KD on old data in pursuit of stability. For clarity, Table 5 shows the impact of old and new data on GFD. In the table, 'Old' and 'New' refer to the mean accuracies of the old and new classes across all tasks.

Effect of memory budget. We conduct an experiment on exemplar that stores 5, 10, 20, and 50 samples for each class to check the dependence by memory budget in Table 6. Although our method adopts an asymmetric learning strategy for current task data and exemplar, our method achieves state-of-the-art for all memory sizes and shows robust performance even for small exemplar sizes. For the most challenging experiment with a memory size of 5 per class, we achieve significant performance improvements of 5.72 and 5.57 percent points for NME and CNN, respectively.

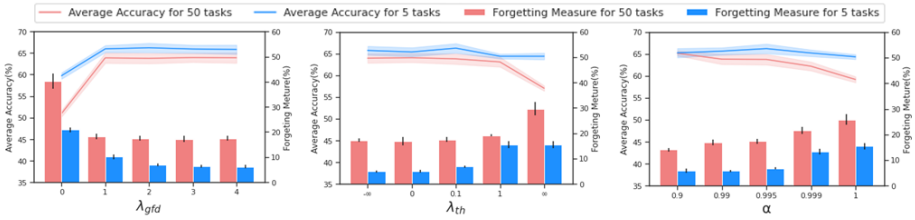

Fig. 7: Sensitivity analysis for each hyperparameter with 5 tasks and 50 tasks on CIFAR-100. The lineplot and barplot represent average accuracy and forgetting measure, respectively.

Effect of initial task size. In Table 7, we investigate the impact of the initial representation by increasing the initial task size from 20 to 50 with an interval of 10. Our method achieves competitive performance across all experiments. These results show that our method is robust to the model's initial representation, even in the lack of diversity in the initial representation.

Hyperparameter sensitivity analysis. In Fig. 7, we investigate our hyperparameters λ_{gfd} for the gradient-based feature distillation, and λ_{th} and α for the confidence-aware weight interpolation on CIFAR-100. As λ_{gfd} increases, the forgetting measure decreases, and the average accuracy remains similar. FM increases as λ_{th} and α increase. This means that stability decreases and plasticity increases as the hyperparameters of the two values increase. When the two values are infinity and 1, weight interpolation is not used at all, and at this time, the performance deteriorates the most. Our method can control the balance between plasticity and stability through these three parameters. Based on these results, we determine the final hyperparameters.

5 Conclusion

In this paper, we proposed a selective regularization method to accept new knowledge while preserving previous knowledge. We introduced an asymmetric feature distillation approach using the gradients of classification and knowledge distillation losses to decide whether to perform pattern completion and separation. Furthermore, we proposed a confidence-aware weight interpolation method to improve the balance between stability and plasticity in class-incremental learning. We achieved competitive performance to the state-of-the-art methods, and extensive experiments demonstrate the effectiveness in various scenarios.

Acknowledgments. This work was supported by IITP grant funded by the Korea government (RS-2023-00236245/Dev. of Perception/Planning AI SW for Seamless Autonomous Driving in Adverse Weather/Unstructured Env., IITP-2024-No.RS-2023-00255968/AI Convergence Innovation Human Resources Development, RS-2021-II212068/AI Innovation Hub), and Korea NRF grant (NRF-2022R1A2C1091402). W. Hwang is the corresponding author.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European conference on computer vision (ECCV). pp. 139–154 (2018)
2. Ashok, A., Joseph, K., Balasubramanian, V.N.: Class-incremental learning with cross-space clustering and controlled transfer. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII. pp. 105–122. Springer (2022)
3. Castro, F.M., Marin-Jimenez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 233–248 (2018), https://arxiv.org/abs/1807.09536
4. Cha, S., Hsu, H., Hwang, T., Calmon, F.P., Moon, T.: Cpr: classifier-projection regularization for continual learning (2021)
5. Chaudhry, A., Dokania, P.K., Ajanthan, T., Torr, P.H.: Riemannian walk for incremental learning: Understanding forgetting and intransigence. In: ECCV (2018), https://arxiv.org/abs/1801.10112
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
7. Douillard, A., Cord, M., Ollion, C., Robert, T., Valle, E.: Podnet: Pooled outputs distillation for small-tasks incremental learning. In: Proceedings of the IEEE European Conference on Computer Vision (ECCV) (2020), https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123650086.pdf
8. Du, Y., Czarnecki, W.M., Jayakumar, S.M., Farajtabar, M., Pascanu, R., Lakshminarayanan, B.: Adapting auxiliary losses using gradient similarity. arXiv preprint arXiv:1812.02224 (2018)
9. Eeckt, S.V., et al.: Weight averaging: A simple yet effective method to overcome catastrophic forgetting in automatic speech recognition. ICASSP (2023)
10. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. ArXiv e-prints (dec 2013), https://arxiv.org/abs/1312.6211
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
13. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
14. Hu, X., Tang, K., Miao, C., Hua, X.S., Zhang, H.: Distilling causal effect of data in class-incremental learning. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 3957–3966 (2021)
15. Hung, C.Y., Tu, C.H., Wu, C.E., Chen, C.H., Chan, Y.M., Chen, C.S.: Compacting, picking and growing for unforgetting continual learning. In: Advances in Neural Information Processing Systems. pp. 13647–13657 (2019), https://arxiv.org/abs/1910.06562
16. Izmailov, P., Wilson, A., Podoprikhin, D., Vetrov, D., Garipov, T.: Averaging weights leads to wider optima and better generalization. In: 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018. pp. 876–885 (2018)

17. Kang, M., Park, J., Han, B.: Class-incremental learning by knowledge distillation with adaptive feature consolidation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16071–16080 (2022)
18. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proc. of the national academy of sciences (2017)
19. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
20. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017), https://arxiv.org/abs/1606.09282
21. Liu, Y., Schiele, B., Sun, Q.: Adaptive aggregation networks for class-incremental learning. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 2544–2553 (2021)
22. Liu, Y., Schiele, B., Sun, Q.: Rmm: Reinforced memory management for class-incremental learning. Advances in Neural Information Processing Systems **34**, 3478–3490 (2021)
23. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 12245–12254 (2020)
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
25. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of learning and motivation, vol. 24, pp. 109–165. Elsevier (1989)
26. Mermillod, M., Bugaiska, A., Bonin, P.: The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects (2013)
27. Movshovitz-Attias, Y., Toshev, A., Leung, T.K., Ioffe, S., Singh, S.: No fuss distance metric learning using proxies. In: Proceedings of the IEEE international conference on computer vision. pp. 360–368 (2017)
28. O'Reilly, R.C., McClelland, J.L.: Hippocampal conjunctive encoding, storage, and recall: Avoiding a trade-off. Hippocampus **4**(6), 661–682 (1994)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
30. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017), https://arxiv.org/abs/1611.07725
31. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. ArXiv e-prints (jun 2016), https://arxiv.org/abs/1606.04671
32. Santoro, A.: Reassessing pattern separation in the dentate gyrus (2013)
33. Shaheen, K., Hanif, M.A., Hasan, O., Shafique, M.: Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. Journal of Intelligent & Robotic Systems **105**(1), 9 (2022)
34. Simon, C., Koniusz, P., Harandi, M.: On learning the geodesic path for incremental learning. In: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition. pp. 1591–1600 (2021)

35. Stojanovski, Z., Roth, K., Akata, Z.: Momentum-based weight interpolation of strong zero-shot models for continual learning. In: NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications (2022)
36. Verbeke, P., Verguts, T.: Learning to synchronize: How biological agents can couple neural task modules for dealing with the stability-plasticity dilemma. PLoS computational biology **15**(8), e1006604 (2019)
37. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International Conference on Machine Learning. pp. 23965–23998. PMLR (2022)
38. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al.: Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7959–7971 (2022)
39. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–382 (2019), https://arxiv.org/abs/1905.13260
40. Yoon, J., Yang, E., Lee, J., Hwang, S.: Lifelong learning with dynamically expandable networks. In: International Conference on Learning Representations, ICLR (2018)
41. Yu, Y.C., Huang, C.P., Chen, J.J., Chang, K.P., Lai, Y.H., Yang, F.E., Wang, Y.C.F.: Select and distill: Selective dual-teacher knowledge transfer for continual learning on vision-language models. arXiv preprint arXiv:2403.09296 (2024)
42. Zhu, Y., Wang, Y.: Student customized knowledge distillation: Bridging the gap between student and teacher. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5057–5066 (2021)