Versatile Incremental Learning: Towards Class and Domain-Agnostic Incremental Learning

Min-Yeong Park*, Jae-Ho Lee*, and Gyeong-Moon Park†

Kyung Hee University, Yongin, Republic of Korea
{pmy0792, jh.lee, gmpark}@khu.ac.kr

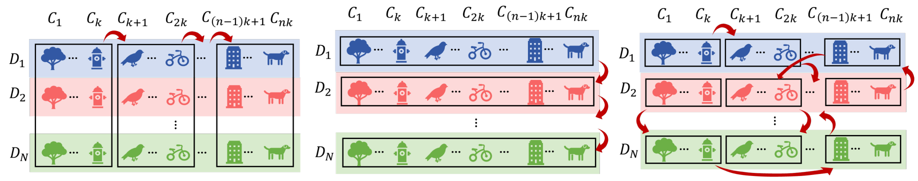
Abstract. Incremental Learning (IL) aims to accumulate knowledge from sequential input tasks while overcoming catastrophic forgetting. Existing IL methods typically assume that an incoming task has only increments of classes or domains, referred to as Class IL (CIL) or Domain IL (DIL), respectively. In this work, we consider a more challenging and realistic but under-explored IL scenario, named *Versatile Incremental Learning (VIL)*, in which a model has no prior of which of the classes or domains will increase in the next task. In the proposed VIL scenario, the model faces *intra-class domain* confusion and *inter-domain class* confusion, which makes the model fail to accumulate new knowledge without interference with learned knowledge. To address these issues, we propose a simple yet effective IL framework, named *Incremental Classifier with Adaptation Shift cONtrol (ICON)*. Based on shifts of learnable modules, we design a novel regularization method called Cluster-based Adaptation Shift conTrol (CAST) to control the model to avoid confusion with the previously learned knowledge and thereby accumulate the new knowledge more effectively. Moreover, we introduce an Incremental Classifier (IC) which expands its output nodes to address the overwriting issue from different domains corresponding to a single class while maintaining the previous knowledge. We conducted extensive experiments on three benchmarks, showcasing the effectiveness of our method across all the scenarios, particularly in cases where the next task can be randomly altered. Our implementation code is available at `https://github.com/KHU-AGI/VIL`.

Keywords: Incremental learning · Real-world scenario · Adaptation control · Incremental classifier

1 Introduction

Recently, Incremental Learning (IL) strategies [3–6, 8, 9, 11–13, 17, 19, 20, 23, 24, 26, 28, 29, 33, 34, 36, 37, 39–41, 41] have made significant progress in leveraging deep neural networks in a situation when multiple input tasks arrive sequentially. The main challenge of IL is catastrophic forgetting [18], which refers to a phenomenon

* Equally contributed
† Corresponding author



(a) Class IL (CIL). **(b)** Domain IL (DIL). **(c)** Versatile IL (VIL).

Fig. 1: Illustration of several IL scenarios, including our proposed new scenario, Versatile Incremental Learning. Each color shade indicates each domain group, and the solid box indicates each class group. The incremental step follows the red arrow.

in the model that significantly forgets what it has learned previously. The mainstream scenarios to tackle catastrophic forgetting within IL typically fall into two categories: Class IL (CIL) where tasks possess disjoint label spaces within the same domain (see Figure 1a, and Domain IL (DIL) where tasks share the same label space but exhibit distinct distributions (see Figure 1b. Most of the recent IL studies have focused on either CIL or DIL scenarios [6, 29, 34, 36, 37].

These existing settings are based on a strong assumption that sequential input tasks always share the same classes or domains, *i.e.*, only classes or domains can increase, and this assumption makes the existing methods impractical to apply to the real world. For example, the models for self-driving cars should continuously learn increasing classes of objects while the domains where a car lies continuously change by different environments (e.g., weather conditions, regions, etc.). Therefore, the models need to learn new classes or domains sequentially when they cannot expect what will increment afterward. Yet, this situation is under-explored in the existing IL settings although it is crucial for the model function well in real-world scenarios.

In this paper, to alleviate the aforementioned assumption, we introduce a new IL scenario called **Versatile Incremental Learning** (**VIL**) for the first time, which is more challenging and realistic than the existing CIL or DIL settings. VIL aims to deal with a situation where the incoming tasks can contain new classes in the same domain, the same classes in a new domain, or new classes in a new domain. In the VIL setting, the model encounters new tasks without knowing how these tasks will increase, as depicted in Figure 1c. In this class and domain-agnostic incremental scenario, the goal is for the models to learn how to accumulate task-specific knowledge continuously without forgetting, regardless of the incremental type of incoming tasks.

To investigate how the novel VIL scenario is challenging, we conducted experiments on three different datasets. As shown in Figure 2, existing CIL and DIL methods fail in the VIL scenario. We analyze that existing CIL methods face severe drift in the classifier while learning new domains that share the same classes. In the case of DIL methods, they fail on VIL due to inter-domain class confusion, which is caused by their strong assumption of increasing only the do-

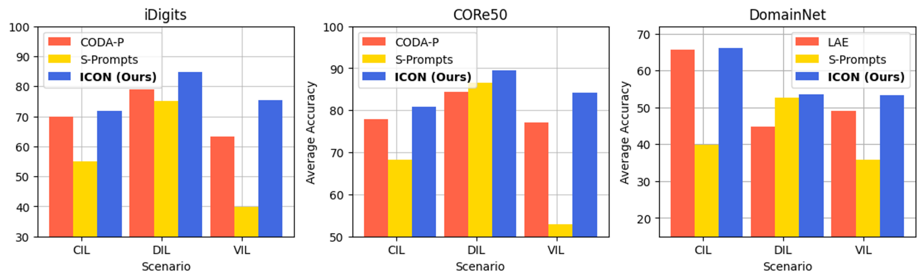


Fig. 2: Comparison of average accuracies among existing CIL and DIL methods in iDigits, CORe50, and DomainNet. In this figure, we compare the baselines that show the best performances in each benchmark, e.g., CODA-Prompt [29] and S-Prompts [34] in iDigits and CORe50, and LAE [6] and S-Prompts [34] in DomainNet. Our proposed ICON outperforms the previous state-of-the-art methods in all scenarios, including the challenging VIL setting.

mains. Therefore, VIL is a challenging scenario that causes severe catastrophic forgetting due to drift in the classifier and class distribution change.

In light of these empirical findings, we propose a new method named **I**ncremental **C**lassifier with Adaptation Shift c**ON**trol (**ICON**), which tackles the VIL setting to consider not just one kind of CIL or DIL, but both when learning incrementally. The main challenge of VIL is that since subsequent tasks can have random incremental types, it is difficult for the model to accumulate knowledge in a specific direction for each type of incremental task. To this end, we propose a simple yet effective strategy coined *Cluster-based Adaptation Shift conTrol (CAST)* to control the learning direction of the model in a stream of erratic input tasks. Specifically, we effectively regularize the direction of current learning concerning the learning directions of previous tasks. Furthermore, we propose *Incremental Classifier (IC)*, a new strategy to regulate the classifier by increasing its output nodes dynamically. It helps the model to learn knowledge of multiple domains for each class effectively while preventing severe forgetting in the classifier. Through these strategies, our proposed ICON achieves state-of-the-art performances in the VIL scenario, including existing IL scenarios across three benchmarks. Our main contributions can be summarized as follows:

- We propose a new realistic IL scenario for the first time, coined Versatile Incremental Learning (VIL), where a model has no prior knowledge of how sequential tasks possess class or domain distributions. To tackle this challenging scenario effectively, we propose a new IL framework, called Incremental Classifier with Adaptation Shift cONtrol (ICON).
- We introduce a new Cluster-based Adaptation Shift conTrol (CAST) loss to guide the learning direction of subsequent tasks to avoid colliding with those of dissimilar tasks already learned.

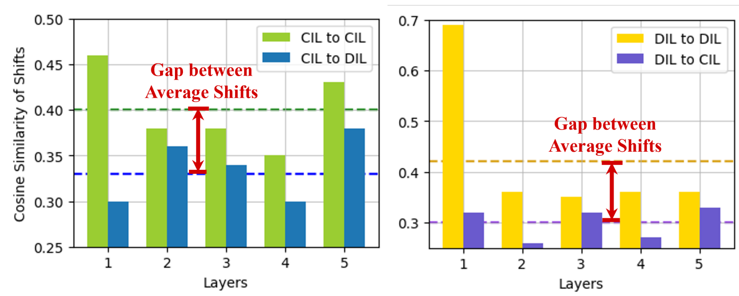


Fig. 3: Illustration of comparison of shifts in adapters when the type of IL remains the same or changes in DomainNet. Shifts are measured by subtracting the previous weights with weights after learning a task.

- To effectively learn multiple domains in the VIL scenario, we propose a new Incremental Classifier (IC) that dynamically increases the output units corresponding to a single label. This approach can alleviate severe drift in the output nodes of the classifier.
- Comprehensive experiments demonstrate that the proposed ICON outperforms the existing state-of-the-art methods significantly in the VIL setting, as well as existing IL scenarios, which shows the effectiveness of the proposed framework.

2 Related Work

More Realistic IL Scenario. Recently, cross-domain IL, which sequentially learns classes from different domains, has begun to be studied [1,27,35,38]. This is a more difficult and realistic scenario than traditional class IL or domain IL, because the large domain gap between each task, coupled with the learning of discrete classes for each task, poses challenges for knowledge transfer. However, the existing methods [1,27,35] restrict this setting in which each subsequent task always has unseen classes on different domains. Another work [38] considers the setting that early tasks have only increasing classes, and the others have only increasing domains. This setting does not consider that classes or domains can increase at any time in the real world. In this paper, we introduce a more realistic and general scenario where the model learns consecutive tasks without prior knowledge of how inputs will increase. The absence of prior knowledge regarding the increment type allows the VIL scenario to encompass not only traditional IL scenarios (CIL and DIL) but also more realistic cross-domain IL as a subset of task streams.
Regularization for IL. Regularization methods for IL have evolved to prevent catastrophic forgetting by adding regularization terms with reference to the old model. In general, the existing regularization methods can be categorized into weight regularization and function regularization [32]. EWC [9], SI [4],

and MAS [22] are the classic regularization approaches that calculate the importance of parameters and consider it to avoid severe changes of important parameters by imposing a penalty to the loss function. Function regularization methods maintain the original knowledge using the output of the model. Using only current task data, LwF [13] distills the knowledge of the previous model by matching its outputs with the current ones.

However, existing regularization methods are not optimal when the tasks increase in a versatile way. We empirically investigated how the parameters are updated when the type of IL changes compared to when the type of IL remains the same. As shown in Figure 3, the similarities of shifts of the model between the same type of IL are fairly higher than those when what increments in the following task changes. This indicates that the shifts of the model lean towards disparate directions when the type IL shifts from one to another. Therefore, when its learning direction shifts, the model cannot accumulate knowledge well, deteriorating catastrophic forgetting. Hence, it is needed to regularize the direction of learning of the model not to conflict with the direction of dissimilar learning in history. Inspired by these points, we propose a novel regularization method for this sake, which will be further described in Section 3.2.

Model Expansion for IL. Many attempts [3,7,8,11,12,33,35,39,41] have been made to dynamically expand the models, *e.g.* neurons, modules, *etc.*, and the extended capacity is utilized to acquire task-specific knowledge in IL. DER [39] increases a new feature extractor per each task to learn task-specific super features with mask layers. Moreover, DyTox [3] proposed a dynamically expandable representation model as well as a classifier in a task-dynamic manner. Recently, ESN [35] also proposes the expandable classifier method, but they leverage training samples augmented in different ways in the inference phase to vote for the classifier with low free energy (likely in-distribution) [15,30] among multiple classifiers. However, the aforementioned expandable classifiers also fail in our novel VIL setting because there is no consideration for the same class in different domains which causes *intra-class domain* confusion and *inter-domain class* confusion by weight overwriting. In contrast, we propose a novel selective node-wise expandable classifier to address the overwriting problem that leads to catastrophic forgetting in the existing classifier when learning the same classes across different domains.

3 Method

In this section, we propose a simple yet effective incremental learning framework named ICON to address the problems of VIL aforementioned. We first introduce the proposed VIL scenario and briefly review the problems that derive from the under-explored VIL scenario in Section 3.1. Next, we propose two novel methods: Cluster-based Adaptation Shift Control (CAST) and Incremental Classifier (IC) in Section 3.2 and Section 3.3, respectively. Finally, we describe the whole training scheme with an optimization objective in Section 3.4.

Table 1: Comparison of typical incremental learning scenarios and proposed VIL.

Scenario Attribute	CIL	DIL	**VIL**
Can it have disjoint label space?	✓	✗	✔
Can it have disjoint domain space?	✗	✓	✔
Can it have both disjoint label and domain space?	✗	✗	✔

3.1 Scenario Description of VIL

Previously, the types of IL can be categorized into CIL and DIL, which have limitations in that they assume that what will increase in the next task compared to the current task is fixed with class or domain. However, in the real-world, class, domain, or both can be increased at a time. Therefore, we introduce a new IL scenario named Versatile-Incremental Learning (VIL) which better suits for the real-world. In the VIL scenario, only classes, only domain or both can increase in the very following task and we describe each scenario in Table 1 and illustrate it in Figure 1. As shown in Table 1, the existing CIL setting is fixed to have only disjoint label space, and the DIL setting is fixed to have only disjoint domain space. However, in a task stream of VIL, two sequential tasks can have disjoint label space, disjoint domain space, or both. In these differences, the model in the VIL scenario suffers from the problems as follows.

In the proposed VIL scenario, the model fails on VIL due to intra-class domain confusion and inter-domain class confusion that is derived from the ever-dynamically changing input distribution. Furthermore, the model faces severe drift in the classifier while learning new domains that share the same classes. To address these problems, we propose Cluster-based Adaptation Shift conTrol (CAST) to prevent model confusion and Incremental Classifier (IC) to prevent weight drift from different domains corresponding to a single class. We precisely explain each proposed method in the following sections.

3.2 Cluster-based Adaptation Shift Control

In existing CIL and DIL scenarios, the model can accumulate knowledge steadily without any guidance about updating directions of learnable weights since domains or classes are shared for the entire tasks, which makes the accumulation of knowledge easier. However, in the VIL scenario, there is no prior knowledge of what will increment and what will remain stationary in the following task whether it is classes or domain. Consequently, this makes it difficult to accumulate knowledge in a steady direction along the entire tasks. Therefore, it is necessary to have guidance for a model about how to accumulate the knowledge preventing its current shift from moving capriciously. Here, we propose **C**luster-based **A**daptation **S**hift con**T**rol (**CAST**) loss for this sake.

In order to learn the current task without affecting the various knowledge learned previously, we regularize the direction of updates in current adapters with respect to the directions of updates in previous tasks. For this sake, the

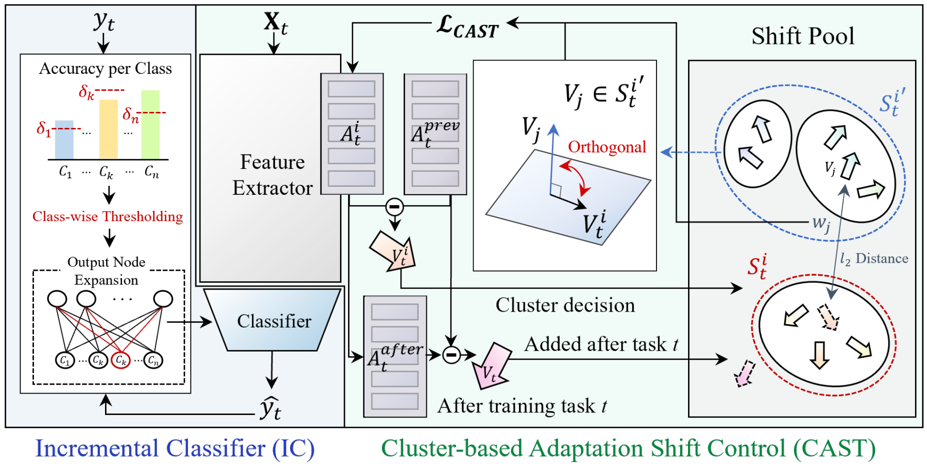


Fig. 4: Architecture overview. In training time, the model calculates the current shift V_t^i of learnable modules by subtracting them with previous ones. Then a cluster S_t^i which V_t^i belongs to is decided, and shifts in the the shift pool which belong to other clusters $S_t^{i'}$ are considered to be from disparate previous tasks. To guide the current learning toward a direction where it does not conflict with V_j, V_t^i is regularized to be orthogonal to for all V_j in $S_t^{i'}$. After learning a task, V_t is saved as a shift in the shift pool which will be used for clustering afterwards.

model saves weights of adapters A_{t-1}^{prev} before learning task $t-1$ as shown in Figure 4. After learning task $t-1$, V_{t-1} which we define as the shift in adapter while learning the task is measured by subtracting A_{t-1}^{prev} from A_{t-1}^{after}, where A_{t-1}^{after} is the adapter weights after learning task $t-1$ as follows:

$$V_{t-1} = A_{t-1}^{after} - A_{t-1}^{prev}. \tag{1}$$

Here, the subscripts of A and V indicate the task identity, while the superscripts of A indicate the status with regard to the current task. The shift V_{t-1} in the direction of task $t-1$ is then saved to a shift pool. As shown in Figure 4, the shift pool saves all previous shifts obtained after learning each task. It is followed by clustering the entire shifts in the shift pool saved until task $t-1$ using the K-Means algorithm. Then, when training the following task t, the shift of adapter for current iteration i, V_t^i is calculated using the current weights of adapter A_t^i for each iteration. Here, we define the shift in the direction of current learning in comparison with the state before the beginning of the task t as follows:

$$V_t^i = A_t^i - A_t^{prev}, \tag{2}$$

where A_t^{prev} is the adapter weights before learning task t.

The subtraction of weights implies the meaning of the direction of learning the task, and it can be derived from the formula of classic gradient descent for parameter update as follows:

$$A_t^{i+1} = A_t^i - \eta \frac{\partial \mathcal{L}}{\partial A_t^i}, \quad A_t^1 = A_t^0 - \eta \frac{\partial \mathcal{L}}{\partial A_t^0}. \tag{3}$$

From Equation 3,

$$A_t^i = A_t^0 - \eta \sum_{k=0}^{i-1} \frac{\partial \mathcal{L}}{\partial A_t^k}, \tag{4}$$

$$\therefore V_t^i = A_t^i - A_t^{prev} = A_t^i - A_t^0 = -\eta \sum_{k=0}^{i-1} \frac{\partial \mathcal{L}}{\partial A_t^k}. \tag{5}$$

We can replace A_t^i using A_t^{i-1}, and after successive replacement from A_t^i to A_t^1, the subtraction of two weights is expressed in the form of summation of gradients, which is accumulated gradients. Therefore, by simply subtracting two weights, we can utilize the shift in the direction of the current iteration with regard to the state before learning task t.

After the calculation of V_t^i, the model predicts the index of the cluster which V_t^i belongs to for each iteration among clusters established via K-Means before learning task t. The prediction is done by selecting a cluster whose center is the closest with V_t^i. The cluster that V_t^i belongs to is notated as S_t^i, and other clusters as $S_t^{i'}$. The shifts that belong to the rest of clusters $S_t^{i'}$ represent the directions of previous tasks whose directions are distinctive from current learning. Therefore, to prevent the direction of current learning V_t^i from colliding with those directions, they are used to regularize the current learning. The regularization of the direction of current learning is done by making V_t^i to be orthogonal with shifts in $S_t^{i'}$. Finally, the equation for CAST loss is defined as follows:

$$\mathcal{L}_{CAST} = \sum_j w_j \cdot \frac{V_t^i \cdot V_j}{\|V_t^i\| \, \|V_j\|}, \quad w_j = \frac{\|V_t^i - V_j\|_2}{\sum_{V_k \in S_t^{i'}} \|V_t^i - V_k\|_2}, \tag{6}$$

where $V_j \in S_t^{i'}$, $S_t^{i'} = \{V_1, V_2, \ldots, V_{t-1}\} - S_t^i$. We consider all shifts in $S_t^{i'}$ using weighted sum in the loss, with w_j being acquired using Euclidean distance between V_t^i and V_j in $S_t^{i'}$, thereby considering the discrepancy of current shift and each V_j differentially. As a result, the CAST loss leads the current shift not to affect the shifts in the shift pool while adapting to the current task. Specifically, the updates of weights in the current task are adjusted in the direction that preserves the direction of disparate tasks, while fine-tuning. Hence, the model can accumulate knowledge in a stable direction with regard to all tasks, even when the input tasks change arbitrarily. As the sequence of tasks increases, the model can further benefit from CAST by regularizing the direction of shifts and thereby accumulating knowledge in succession.

3.3 Incremental Classifier

In the proposed VIL scenario, the existing CIL methods have not considered the same class in different domains, resulting in a forgetting problem due to the

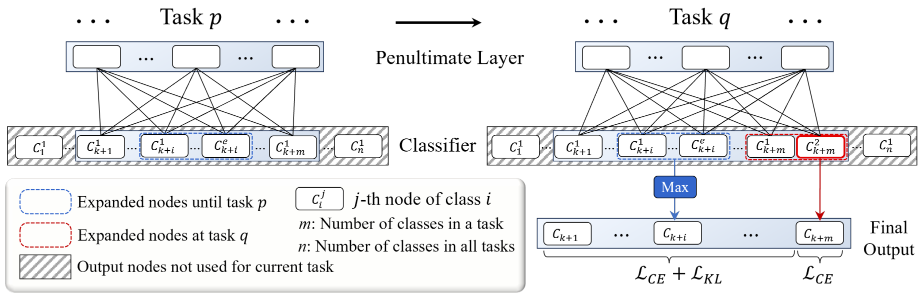


Fig. 5: Illustration of Incremental Classifier (IC) in training. The model increases the output node of classifier if needed whenever classes in current task q had already learned before, *i.e.* task p. Nodes for remaining classes included in task q are trained to preserve the knowledge via distillation. The original nodes with classes whose output nodes has been increased at task q are kept intact by omitting them from cross-entropy loss.

classifier weight overwriting while struggling with intra-class domain confusion. Also, existing DIL methods suffer from inter-domain class confusion based on their consideration of increasing only the domains. To deal with this problem, we propose a simple but effective Incremental Classifier (IC) (Figure 5) which increments the final output node of the classifier layer if needed. Unlike the existing expandable methods, the proposed IC deals with the problem through a decision process on whether to increase the node using class-wise dynamic thresholding. Detailed descriptions are as follows.

As mentioned, the fully domain-specific classifier has problems, and it is necessary to set the appropriate criteria and increase nodes accordingly. To this end, we utilize the accuracy of each class in the learned domains and the current domain to determine the class-wise threshold δ_i dynamically as follows:

$$\delta_i = tanh(p_i), \quad p_i = \gamma * \frac{\frac{1}{|D^{prev}|}\left(\sum_{d\in D^{prev}} Acc\left(C_i^d\right)\right) - Acc\left(C_i^{d_{new}}\right)}{\frac{1}{|D^{prev}|}\sum_{d\in D^{prev}} Acc\left(C_i^d\right)}, \tag{7}$$

where $D^{prev} = \{d\,|1 \le d \le D\}$ (sequential integer set), D refers to the number of previously learned tasks that contain the same classes as the newly arrived but different domains, $Acc(C_i^d)$ refers to the accuracy of i-th class in d-th domain, and γ is a scaling factor. We consider only the classes that need to be learned in the current task in the entire processes of the IC (except for the shaded nodes in Figure 5). In this context, p_i in Equation 7 which is obtained by comparing the accuracies between the new domain and the already learned domains, represents the difficulty of the corresponding class in the new domain, and the classes whose accuracies are below the thresholds are considered as challenging classes to learn using the existing classifier. Therefore, the model increments its output nodes corresponding to each challenging class. During training, for challenging classes, only the logits of the increased nodes in the current task are used, and the existing nodes are not used to prevent forgetting (red solid line box in Figure 5).

For classes that are not relatively difficult, *i.e.* whose nodes do not need to be increased in the current task, the method for handling nodes is different from the above. In the case of classes that have not been increased and therefore have only a single corresponding node (the corresponding node to class C^1_{k+1} in Figure 5), their single logits are used. In contrast, for the classes that have multiple nodes corresponding to one class which are not increased in the current task, but increased in previous tasks, a selection process is required to obtain a unique logit for learning. Our model selects the maximum logit from multiple nodes (blue dotted line box with *Max* operation in Figure 5). We adopt the reasonable design choices for simple *Max* operation as follows. A classifier learned with the cross-entropy (CE) loss can be regarded as an energy model [10], that aims to minimize the energy of it. Existing energy-based methods [14, 30] have shown that data from in-distribution usually have lower energy than data from out-of-distribution for a certain classifier. In this context, the nodes with smaller energy for a single class are better suited to current training data than the others in that they are more likely to be in-distribution [2, 25]. If the energy function is defined as $E(\boldsymbol{x}, y) = -f_y(\boldsymbol{x})$ (unnormalized negative logit of class y for input $\boldsymbol{x}$) according to [14], it can be determined that the node that produces the maximum logit (minimum energy) is the most suited for learning the current training data. Therefore, choosing the maximum logit for the final prediction is a simple but effective solution. This node selection strategy is also used in the inference.

Finally, the CE loss ($\mathcal{L}_{CE}$) is applied to all final output logits (see Figure 5). Moreover, to prevent forgetting, we distill the knowledge of the nodes learned in the $(t-1)$-th task into nodes not selected by the *Max* operation. Here, the corresponding logit and Kullback-Leibler divergence loss ($\mathcal{L}_{KL}$) are used (see Figure 5). The total loss of the proposed IC ($\mathcal{L}_{IC}$) is as follows:

$$\mathcal{L}_{IC} = \mathcal{L}_{CE}(O^t, y) + \alpha \mathcal{L}_{KL}(O^t, O^{t-1}), \tag{8}$$

where $O^t = f^{t,\mathcal{P}}_{\phi}(f_{\theta,A}(\boldsymbol{x})[CLS])$ refers to the final logits obtained by a classifier that is updated in current task t. $f^{t,\mathcal{P}}_{\phi}$ means the classifier of task t with a function $\mathcal{P}$ that decides whether to increase the node with class-wise dynamic thresholding, and $\mathcal{P}$ is applied to the final output of the classifier. $f_{\theta,A}$ and f_ϕ indicate the frozen ViT (θ) with trainable adapters (A) and the classifier, respectively. $f_{\theta,A}(\cdot)[CLS]$ is the CLS token of the output after passing all transformer layers. y is the label and α is the balancing weight between two objectives.

3.4 Training Objective

Along with $\mathcal{L}_{CAST}$ in Equation 6 and $\mathcal{L}_{IC}$ in Equation 8, our end-to-end full optimization is as follows:

$$\mathcal{L}_{Total} = \beta \mathcal{L}_{CAST}(O^t, y) + \mathcal{L}_{IC}(O^t, O^{t-1}, y, \alpha), \tag{9}$$

where $(\boldsymbol{x}, y) \in D_t$, data of current task D_t. While the parameters of the ViT are frozen, only the parameters A of the adapters and ϕ of the classifier are

Table 2: Dataset composition and configuration for each scenario. All IL scenarios were configured not deviate from the original composition of each dataset. Values with asterisk (*) refer to the number of domains on trainset pre-defined in the CORe50 [16]. N and C_t indicates that number of tasks and classes per task respectively.

Dataset	Composition		CIL		DIL		VIL	
	#Class	#Domain	N	C_t	N	C_t	N	C_t
iDigits [31]	10	4	5	2	4	10	20	2
CORe50 [16]	50	11	5	10	8*	50	40	10
DomainNet [21]	345	6	5	69	6	345	30	10

updated. The final logits $O^t = f_{\phi}^{t,\mathcal{P}}(f_{\theta,A}(\boldsymbol{x})[CLS])$ same as in Equation 8. The proposed IC can solve the forgetting problem derived from the classifier weight overwriting and inter-domain class confusion that the existing methods suffered, and its superiority is demonstrated in the next (Section 4).

4 Experiments

In this section, we compared and evaluated our approach with state-of-the-art methods on widely used datasets. First, we introduce the experimental setup including the datasets, comparison baselines, and metrics in Sec. 4.1. The details about implementation and training are described in supplementary material. Also, we show extensive results of our experiments in Sec. 4.2 to demonstrate the effectiveness of our approach. Moreover, we conducted elaborate analysis including ablation studies in Sec. 4.3 to interpret our approach in detail.

4.1 Experimental Setup

Datasets. We conducted experiments on three benchmarks, including iDigits [31], CORe50 [16] and DomainNet [21] which are possible to construct IL scenarios that can cause a large shift in distribution by clearly distinguishing both classes and domains. The composition and configuration for each scenario about the datasets are described in Table 2 and please refer to the supplementary material for a more detailed explanation of the datasets.
Comparison Baselines. We compared ICON against naive baselines and various IL methods including the latest ones. First, we set the *Lower-bound* as usual supervised sequential fine-tuning result (notated as Fine-tuning in Table 3). Then, we compared our proposed ICON with the regularization-based methods EWC [9] and LwF [13]. Moreover, we compared it with the recent prompt-based methods, including S-Prompts [34], L2P [37], DualPrompt [36], CODA-Prompt [29] and LAE [6].
Evaluation Metrics. We evaluated the methods by the widely used two IL metrics: Average Accuracy which is the higher the better (marked as Avg. Acc↑), and Forgetting which is the lower the better. For the scenarios with clear task

Table 3: Main results with all of the IL scenarios. Experiments were conducted based on the latest incremental learning models. We used the bold and the underline as brief indications of the best and the second best, respectively.

Method	CIL		DIL		VIL		Average
	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	
iDigits							
Fine-tuning	30.32±0.77	48.01±0.72	33.04±0.89	23.23±0.74	19.89±0.82	57.17±1.28	26.22±1.92
EWC [9]	34.16±0.32	38.72±0.59	68.62±0.92	25.94±0.98	21.86±1.45	53.98±1.28	37.36±1.87
LwF [13]	39.88±0.91	33.35±0.52	69.61±0.33	25.81±0.69	23.44±0.14	53.65±0.42	41.91±2.19
L2P [37]	63.17±0.88	28.53±0.81	73.83±0.26	23.43±0.65	59.07±3.01	15.82±2.64	64.43±2.44
S-Prompts [34]	55.09±3.27	25.61±1.62	75.11±2.31	25.66±6.23	39.73±3.40	15.41±1.16	54.33±4.64
DualPrompt [36]	68.82±0.97	**11.81±1.77**	76.42±0.46	26.33±0.62	60.25±2.92	23.40±3.50	67.61±3.43
CODA-P [29]	69.97±1.02	19.83±2.28	77.42±0.71	22.20±0.18	63.30±3.08	16.43±2.63	70.95±3.91
LAE [6]	65.77±0.83	28.47±0.77	79.09±1.03	21.86±0.40	59.34±0.95	29.32±1.72	68.12±3.12
ICON (Ours)	**71.53±0.68**	19.36±1.17	**84.83±0.51**	**12.67±0.61**	**75.11±2.39**	**9.13±1.88**	**77.15±1.19**
CORe50							
Fine-tuning	21.54±1.91	74.05±1.31	23.52±0.26	3.09±0.11	14.04±0.50	58.59±0.83	19.86±1.28
EWC [9]	33.89±0.83	50.18±0.30	73.86±0.38	1.09±0.12	43.20±0.71	9.56±0.46	50.62±1.94
LwF [13]	34.53±0.55	41.05±0.30	74.35±0.52	0.81±0.27	45.77±1.03	10.53±0.79	52.19±1.82
L2P [37]	70.03±0.51	6.51±0.59	80.72±0.39	0.51±0.28	64.85±0.92	6.62±0.19	70.18±0.68
S-Prompts [34]	68.27±3.92	11.79±0.24	86.50±0.46	0.92±0.31	52.88±0.85	6.18±0.83	67.51±1.67
DualPrompt [36]	71.96±0.37	5.04±0.71	81.41±0.22	0.21±0.76	66.21±1.76	7.20±0.88	71.46±1.00
CODA-P [29]	77.85±0.44	**4.78±0.37**	84.36±1.04	0.64±0.14	69.28±0.24	6.77±0.38	74.52±0.68
LAE [6]	77.11±0.31	18.38±1.67	83.09±0.71	0.17±0.51	77.11±1.37	8.23±2.59	75.89±1.00
ICON (Ours)	**80.85±0.23**	7.68±0.52	**89.01±0.33**	**0.17±0.21**	**83.18±1.21**	**4.72±0.24**	**84.34±0.59**
DomainNet							
Fine-tuning	35.43±0.58	47.79±0.28	39.52±0.32	28.81±0.64	20.35±0.72	43.22±1.14	31.66±0.57
EWC [9]	53.04±0.53	24.41±0.48	41.58±0.26	26.79±0.15	36.68±0.25	27.68±0.91	44.28±1.19
LwF [13]	53.79±0.61	19.41±0.11	43.74±0.27	18.23±0.10	38.17±0.35	21.87±0.64	44.70±1.09
L2P [37]	60.90±0.69	8.23±0.90	48.55±0.81	19.71±1.29	48.98±0.69	14.71±1.07	54.22±0.87
S-Prompts [34]	39.78±0.62	19.29±1.04	50.80±0.63	**4.20±0.53**	35.90±0.54	14.25±15.66	42.54±1.21
DualPrompt [36]	62.55±0.92	**7.62±1.07**	51.33±0.10	9.60±1.41	49.36±1.05	16.79±1.17	56.00±0.84
CODA-P [29]	65.21±0.24	15.01±0.21	49.13±0.83	25.96±1.13	49.45±1.27	17.01±2.37	58.73±0.93
LAE [6]	65.06±0.18	9.68±0.84	44.67±0.62	28.99±0.64	49.01±1.18	21.20±1.33	55.26±1.63
ICON (Ours)	**65.43±0.15**	9.72±0.46	**54.44±0.21**	13.32±0.46	**53.37±0.47**	**11.25±0.18**	**59.74±1.06**

boundaries, we reported the final test score following the general protocol [29, 34, 36, 37].

4.2 Experimental Results

Main Results. We conducted extensive experiments, including traditional IL scenarios as well as the proposed VIL scenario. The results are summarized in Table 3. As shown in the table, ICON significantly outperformed the existing methods in our proposed VIL scenario in terms of both average accuracy and forgetting (shaded column). Moreover, we demonstrated the effectiveness of the proposed methods even in existing scenarios, resulting in the best average performance for all scenarios (unshaded columns). Furthermore, the existing state-of-the-art model showed a rather unstable performance (high standard deviation) in the two aforementioned scenarios, whereas ICON showed a very stable performance (low standard deviation). This indicates that while existing methods

Table 4: Results on Cross-Domain Incremental Learning scenario.

Method	iDigits		CORe50		DomainNet		Average
	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	
Fine-tuning	21.62±5.21	51.01±6.86	20.35±2.46	33.89±1.57	31.35±0.68	56.74±3.18	24.44±2.78
EWC [9]	24.79±4.81	48.94±4.29	51.56±5.87	28.55±4.11	45.85±3.75	34.57±4.20	40.73±4.81
LwF [13]	34.71±7.38	36.34±4.91	54.12±5.18	27.10±1.41	43.12±3.14	33.13±2.77	43.98±5.23
L2P [37]	61.66±5.61	16.84±6.89	65.12±0.93	7.43±1.73	58.45±1.32	6.32±9.80	61.74±2.62
S-Prompts [34]	47.40±9.61	**10.03±3.32**	62.41±1.47	11.87±4.27	38.82±1.76	9.12±2.11	49.54±4.28
DualPrompt [36]	64.95±9.38	14.90±6.59	66.29±1.65	8.86±1.28	60.79±1.30	**5.34±1.94**	64.01±4.11
CODA-P [29]	73.09±10.85	11.41±4.13	66.59±1.03	6.08±0.95	67.56±3.44	10.47±1.71	69.08±5.10
LAE [6]	68.24±9.68	19.22±4.22	66.28±1.64	10.17±1.79	61.78±4.56	17.16±3.05	65.43±5.29
ICON (Ours)	**75.73±5.63**	10.72±2.40	**74.98±0.03**	**5.50±2.17**	**67.95±1.87**	8.18±1.80	**72.88±2.51**

Table 5: Ablation of CAST and IC in the VIL scenario.

Method		iDigits		CORe50		DomainNet		Average
CAST	IC	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	Avg. Acc↑	Forgetting↓	
		59.34±0.95	29.32±1.72	77.11±1.37	8.23±2.59	49.01±1.18	21.20±1.33	61.82±1.17
✓		68.34±2.09	18.30±6.41	79.20±0.59	**4.55±1.81**	50.56±0.51	17.50±0.96	66.03±1.06
	✓	66.97±1.03	14.32±5.10	81.13±3.01	5.32±2.11	51.60±1.32	11.90±0.77	66.57±1.79
✓	✓	**75.11±2.39**	**9.13±1.88**	**83.18±1.21**	4.72±0.24	**53.37±0.47**	**11.25±0.18**	**69.98±1.23**

have limits in their learning abilities based on the order of class and domain that consists of each task, but ICON can reliably learn in any order.

Results on Cross-Domain Incremental Learning. Moreover, we conducted experiments on cross-domain incremental learning as shown in Table 4. In cross-domain IL, both class and domain always increase at the same time in the following tasks, which is a subset of the VIL scenario. For all datasets, the number of classes in a task is the same as CIL setting. The number of tasks is 4 in iDigits, and 5 for CORe50 and DomainNet. The outstanding performance on cross-domain IL implies that our proposed CAST and IC successfully accumulate knowledge from inputs on various sequences of data distributions.

4.3 Analysis

Ablation Study. We further investigated the effectiveness of each component of ICON in Table 5. We conducted an ablation study starting from our baseline, adding CAST, IC, and both. We confirmed that using the CAST loss to regularize the direction of current learning by considering the shifts in parameters in history is effective in learning versatile tasks. Furthermore, we also confirmed that with IC, the model can accumulate knowledge of different domains within the same classes by increasing its output node when the existing nodes are decided not to be appropriate to be used in a new domain. By adopting CAST and IC for VIL, the model can leverage the effectiveness of each component with synergy. The ablation results showed that each component of ICON was effective in alleviating catastrophic forgetting in a situation where input tasks are erratic.

Moreover, we conducted an ablation study on IC by dividing it into node expansion and distillation as shown in Table 6. Implementation of simple expansion of output nodes without distillation to preserve the knowledge showed considerable performance gain. It indicates that the separation of the output nodes corresponding to a single class can accommodate disparate knowledge from different domains. As already known, a distillation of knowledge from previous classifier was also helpful in mitigating catastrophic forgetting in VIL as well.

Number of Clusters. We conducted our experiments with different numbers of clusters in our proposed CAST, varying from 0 to 6 in the VIL setting. We demonstrated the performance gain from our baseline for each dataset in Figure 6, where not using CAST is equivalent to when the number of clusters is 0, and we used all previous shifts in the history for CAST loss when the number of clusters is 1. As you can see in Figure 6, iDigits gained the best performance when the number of clusters is 2 and CORe50 when 3. Since the number of the entire tasks is 20 for iDigits and 40 for CORe50, it can be interpreted that as the number of tasks grows, clustering the history shifts into a bigger number is effective when the sequence becomes longer. Using bigger than 3 for the number of clusters did not show any noticeable performance gain.

Table 6: Ablation of IC in iDigits. NE and KD refer to node expansion and knowledge distillation respectively.

NE	KD	Avg. Acc↑	Forgetting↓
		59.34±0.95	29.32±1.72
✓		63.10±3.58	25.50±2.98
	✓	64.66±3.10	19.30±3.27
✓	✓	**66.97±1.03**	**14.32±5.10**

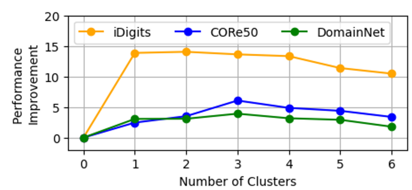


Fig. 6: Performance gain from number of clusters.

5 Conclusion

In this work, we proposed a new IL scenario named Versatile Incremental Learning (VIL), that reflects a more complex real-world derived from random incremental streams (classes, domains, or both) without any incremental prior knowledge. We defined the key challenges in VIL and proposed a novel framework, coined ICON (Incremental Classifier with Adaptation Shift cONtrol), composed of Cluster-based Adaptation Shift conTrol (CAST) loss and Incremental Classifier (IC). We demonstrated that ICON showed SOTA performance in the proposed VIL, as well as existing IL scenarios, and its effectiveness through various experiments. We look forward to our proposed VIL scenario serves as a new starting point for real-world IL field.

Nevertheless, there is still room for improvement in our work, especially regarding the scenario. A scenario that considers a varying number of classes and domains in a task can deal with a more realistic scenario, since in real-world, the distributions of classes and domains in a task can change.

Acknowledgements

This work was supported by MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2023-00258649) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and in part by the IITP grant through MSIT (Development of Semi-Supervised Learning Language Intelligence Technology and Korean Tutoring Service for Foreigners) under Grant 2019-0-00004, and in part by the IITP grant through MSIT (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068, and by the IITP grant funded by the Korea government (MSIT) (No.RS-2022-00155911, Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University)).

References

1. Buzzega, P., Boschini, M., Porrello, A., Abati, D., Calderara, S.: Dark experience for general continual learning: a strong, simple baseline. Proceedings of the Advances in Neural Information Processing Systems **33**, 15920–15930 (2020)
2. Choe, S.A., Shin, A.H., Park, K.H., Choi, J., Park, G.M.: Open-set domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23943–23953 (2024)
3. Douillard, A., Ramé, A., Couairon, G., Cord, M.: Dytox: Transformers for continual learning with dynamic token expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9285–9295 (2022)
4. Friedemann Zenke, B.P., Ganguli, S.: Continual learning through synaptic intelligence. In: Proceedings of the International Conference on Machine Learning. p. 3987–3995 (2017)
5. Gao, Q., Zhao, C., Ghanem, B., Zhang, J.: R-dfcil: Relation-guided representation learning for data-free class incremental learning. In: Proceedings of the European Conference on Computer Vision. pp. 423–439. Springer (2022)
6. Gao, Q., Zhao, C., Sun, Y., Xi, T., Zhang, G., Ghanem, B., Zhang, J.: A unified continual learning framework with general parameter-efficient tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11483–11493 (2023)
7. Hu, Z., Li, Y., Lyu, J., Gao, D., Vasconcelos, N.: Dense network expansion for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11858–11867 (2023)
8. 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. Proceedings of the Advances in Neural Information Processing Systems **32** (2019)
9. 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. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
10. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting Structured Data **1**(0) (2006)

11. Lee, S.W., Kim, J.H., Jun, J., Ha, J.W., Zhang, B.T.: Overcoming catastrophic forgetting by incremental moment matching. Proceedings of the Advances in Neural Information Processing Systems **30** (2017)
12. Li, X., Zhou, Y., Wu, T., Socher, R., Xiong, C.: Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In: Proceedings of the International Conference on Machine Learning. pp. 3925–3934. PMLR (2019)
13. Li, Z., Hoiem, D.: Learning without forgetting. In: Proceedings of the European Conference on Computer Vision. pp. 614–629. Springer (2016)
14. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Proceedings of the Advances in Neural Information Processing Systems **33**, 21464–21475 (2020)
15. Liu, Y., Hong, X., Tao, X., Dong, S., Shi, J., Gong, Y.: Model behavior preserving for class-incremental learning. IEEE Transactions on Neural Networks and Learning Systems (2022)
16. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: Conference on Robot Learning. pp. 17–26. PMLR (2017)
17. Madaan, D., Yin, H., Byeon, W., Kautz, J., Molchanov, P.: Heterogeneous continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15985–15995 (2023)
18. 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)
19. Moon, J.Y., Park, K.H., Kim, J.U., Park, G.M.: Online class incremental learning on stochastic blurry task boundary via mask and visual prompt tuning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11731–11741 (2023)
20. Park, K.H., Song, K., Park, G.M.: Pre-trained vision and language transformers are few-shot incremental learners. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23881–23890 (2024)
21. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1406–1415 (2019)
22. Rahaf Aljundi, Francesca Babiloni, M.E.M.R., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision. p. 139–154 (2018)
23. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. CoRR (2016)
24. Seo, J., Kang, J.S., Park, G.M.: Lfs-gan: Lifelong few-shot image generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11356–11366 (2023)
25. Seo, J., Lee, S.H., Lee, T.Y., Moon, S., Park, G.M.: Generative unlearning for any identity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9151–9161 (2024)
26. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. Proceedings of the Advances in Neural Information Processing Systems **30** (2017)
27. Simon, C., Faraki, M., Tsai, Y.H., Yu, X., Schulter, S., Suh, Y., Harandi, M., Chandraker, M.: On generalizing beyond domains in cross-domain continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9265–9274 (2022)

28. Smith, J., Hsu, Y.C., Balloch, J., Shen, Y., Jin, H., Kira, Z.: Always be dreaming: A new approach for data-free class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9374–9384 (2021)
29. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11909–11919 (2023)
30. Tang, K., Miao, D., Peng, W., Wu, J., Shi, Y., Gu, Z., Tian, Z., Wang, W.: Codes: Chamfer out-of-distribution examples against overconfidence issue. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1153–1162 (2021)
31. Volpi, R., Larlus, D., Rogez, G.: Continual adaptation of visual representations via domain randomization and meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4443–4453 (2021)
32. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(8), 5362–5383 (2024)
33. Wang, W., Hu, Y., Chen, Q., Zhang, Y.: Task difficulty aware parameter allocation & regularization for lifelong learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7776–7785 (2023)
34. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Proceedings of the Advances in Neural Information Processing Systems (2022)
35. Wang, Y., Ma, Z., Huang, Z., Wang, Y., Su, Z., Hong, X.: Isolation and impartial aggregation: A paradigm of incremental learning without interference. In: Proceedings of the Association for the Advancement of Artificial Intelligence. vol. 37, pp. 10209–10217 (2023)
36. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: Proceedings of the European Conference on Computer Vision. pp. 631–648. Springer (2022)
37. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)
38. Xie, J., Yan, S., He, X.: General incremental learning with domain-aware categorical representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14351–14360 (2022)
39. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021)
40. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8715–8724 (2020)
41. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: Proceedings of the International Conference on Learning Representations (2018)