S^2C^2IL: Self-Supervised Curriculum-based Class Incremental Learning

Anonymous authors
Paper under double-blind review

Abstract

This paper proposes *S^2C^2IL*: Self-Supervised Curriculum-based Class Incremental Learning algorithm to prevent catastrophic forgetting in a continual learning setting. In the proposed regularization-based class incremental learning approach, the novel pretext task utilizes stochastically-augmented labels to learn enriched feature representations without the dependency on augmented image transforms. To preclude the pretext task-specific knowledge from being transferred to downstream tasks, we leave out the final section of the pre-trained network in feature transfer. The downstream task is learned through a curriculum where the standard deviation of the filter fused with the network is varied periodically. The proposed S^2C^2IL algorithm trained with orthogonal weight modification backbone outperforms the existing state-of-the-art regularization-based, and memory-based class incremental algorithms on split-CIFAR10, split-CIFAR100, split-SVHN, and split-TinyImageNet datasets.

1 Introduction

The concept of incremental learning has been an active area of research in the deep learning community Mai et al. (2022); Delange et al. (2021). Humans have an unparalleled capability to incrementally capture information of new tasks, domains, or classes without forgetting the knowledge gained from past episodes. As shown in Fig. 1, deep learning approaches, especially CNNs, show excessive plasticity in learning new tasks though they lack the inherent tendency of incremental/continual learning. The problem of incremental learning is introducing new knowledge to an existing model calibrated with old knowledge. During the introduction of new knowledge, one of the biggest challenges is to retain the information from old knowledge as CNNs tend to lose previously acquired information leading to a phenomenon termed as 'Catastrophic Forgetting' McCloskey & Cohen (1989); Ratcliff (1990); McClelland et al. (1995); French (1999). To mitigate catastrophic forgetting, researchers have proposed different approaches to overcome catastrophic forgetting in deep learning models. These approaches include expansion-based models, in which new parameters are added to the network with the addition of each task which capacitates the model to aggregate the information of new classes subsequently. Other approaches include using memory-based methods, where a model is trained to continually relearn old information to retain memory stability with external memory banks, and regularization-based techniques to prevent the model from overwriting previously learned knowledge. Additionally, some researchers have proposed using a combination of these methods, as well as other techniques, in order to more effectively address the problem of catastrophic forgetting and enable deep learning models to continue to learn and adapt over time. Overall, developing effective approaches to overcoming catastrophic forgetting is crucial for advancing deep learning and the continued development of more intelligent and adaptive models.

This work presents a novel regularization-based continual learning algorithm focusing on class-incremental learning (CIL). In general, the performance of these approaches is inferior to memory-based. However, they are more efficient computationally, maintaining learning plasticity without needing additional memory to maintain stability. They are also easy to deploy as opposed to expansion-based and memory-based approaches where voluminous memory setup is a requisite. Without such memory reserves, regularization-based approaches depend mostly on utilizing the data available for the current incremental task. To utilize the limited data available at each incremental task, it is imperative to acquire a diverse set of features from it. One of the ways to accomplish it is through unsupervised learning. Unsupervised pre-training aids the model in learning distinct features from limited or unlabelled data, which assists the

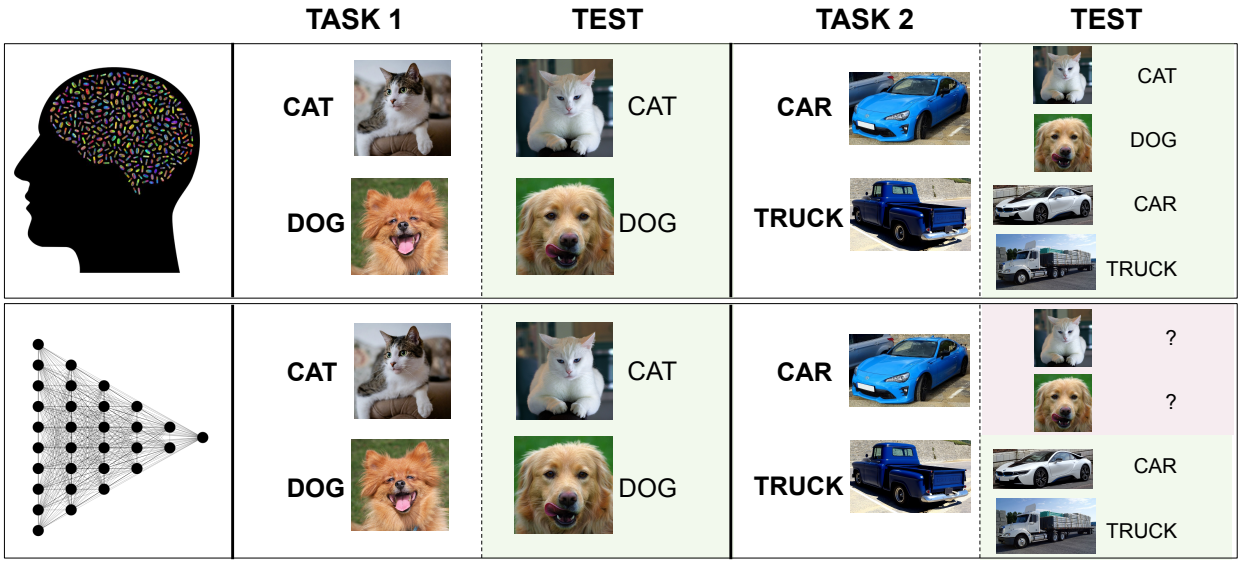

Figure 1: Humans can easily generalize over a newer set of classes/tasks, whereas neural networks suffer from the problem of catastrophic forgetting when trained for different tasks over time.

model in better generalization to newer incremental tasks. To this end, we propose a novel *Self-Supervised Learning (SSL)* task termed as *Stochastic Label Augmentation (SLA)*, which optimizes the model to extract a diverse set of features from the limited data for the current task. The task synthesizes information from different parts of an image using stochastically generated labels in a multi-task fashion, thereby providing rich and diverse feature representations to the model. Conventionally, the data is augmented in self-supervised learning to generate proxy labels that protract the training and are computationally inefficient. By fitting the pretext dataset on stochastically generated labels, we improve the computational efficiency of the model. We employ a novel task-wise weight-regularizer in the pretext task that prevents information loss from the previous to the next incremental task and keeps the model bound to the previously acquired knowledge. Further, we mitigate the transfer of pretext task-specific knowledge to the downstream task through *Penultimate Weight Sharing (PWS)* between networks.

After pre-training the network for the classification task in the CIL setting, we employ a *curriculum-based learning* technique during the downstream training. Curriculum Learning-based algorithms facilitate faster convergence and enhanced generalizability by ordered training Bengio et al. (2009). A Gaussian kernel is used to smooth the extracted features, with the smoothing effect increasing periodically. With the reduction in information through smoothing, we regulate a curriculum for the model to generalize better over incremental tasks. This curriculum is built over *Orthogonal Weights Modification (OWM)* Zeng et al. (2019) as the backbone in the downstream training. Orthogonal Weights Modification alleviates catastrophic forgetting by initiating weight updates along the orthogonal direction Zeng et al. (2019). Further, we employ a novel task-wise weight-regularizer that prevents information loss from the previous to the next incremental task. Penultimate Weight Sharing (PWS) prevents the transfer of pretext-specific knowledge to the downstream task. Through successive cycles of pre-training with SLA and downstream classification for each incremental task, we achieve state-of-the-art performance on the split-CIFAR10, split-CIFAR100, split-SVHN, and split-TinyImageNet databases.

The key highlights of the paper are summarized below:

- A novel self-supervised pretext task termed as *Stochastic Label Augmentation* (SLA) for learning rich and diverse feature representations.
- A curriculum-based learning technique for class-incremental learning through feature level smoothing.
- A novel regularization loss to constrain weight modification and prevent forgetting and Penultimate Weight Sharing (PWS) to prevent the transfer of pretext task-specific knowledge to the downstream task.

- We validate the significance of the curriculum algorithm by analyzing the shift-invariance progress and grad-cam visualizations.

- Evaluation of the proposed algorithm on split-CIFAR10, split-CIFAR 100, split-SVHN, and split-TinyImageNet dataset.

2 Related Work

The concept of incremental learning (also known as lifelong learning or continual learning) has been an active area of research in the deep learning community Mai et al. (2022); Delange et al. (2021). Within the literature, three types of incremental learning scenarios have been explored i.e. domain-incremental learning, task-incremental learning, and class-incremental learning. The addition of new classes in an existing model (considered as an incremental task for the model) is referred to as class incremental learning (CIL). In this paper, we predominantly focus on the problem of CIL. With the addition of new classes, it is imperative that previously learned classes are not forgotten. This ties Incremental learning closely to the problem of catastrophic forgetting. To mitigate catastrophic forgetting, three popular classes of techniques exist, namely: (a) expansion based, (b) memory based, and (c) regularization based.

Expansion based algorithms add new neurons (or parameters) that evolve with every task to allow the network to accumulate information of new classes sequentially. Rusu et al. Rusu et al. (2016) introduced progressive neural networks in which modules with lateral connections are added with each task while preserving the base network. Dynamically Expandable Network Yoon et al. (2017) was proposed to competently calibrate the dynamic capacity of the network for sequential tasks. Li et al. Li et al. (2019) isolated the neural architecture search framework and parameter tuning technique to actively identify the optimal structure for incremental tasks. Inspired by transfer learning, Sarwar et al. Sarwar et al. (2019) presented a clone and branch technique for efficient and dynamical adaptation in the incremental learning network. To alleviate model complexity, Yoon et al. Yoon et al. (2019) introduced additive parameter decomposition, separating and tuning the network parameters as task-specific or task-shared.

Memory-based models are either based on leveraging the subsets of the data from previous tasks (exemplars) or iteratively synthesizing the data based on the first task. Rebuffi et al. Rebuffi et al. (2017) proposed iCaRL, which utilizes the exemplars from memory for rehearsal in continual learning. Deep generative replay framework Shin et al. (2017) was introduced to sample data from the previous task and fuse it with data for the current task. Lopez et al. Lopez-Paz & Ranzato (2017) implemented a Gradient Episodic Memory (GEM) model and applied loss gradients on current tasks to preserve information from previous tasks and prevent interference with memory. Average GEM (A-GEM) Chaudhry et al. (2018) with altered loss function was presented as a more memory-efficient and better-performing variant of GEM. Reimer et al. Riemer et al. (2018) addressed the trade-off between information transfer and interference by introducing a meta-experience replay algorithm to manage the transfer and interference based on future gradients. Distillation-based techniques which preserve knowledge from old classes through storing exemplars have also been proposed recently Hou et al. (2019); Wu et al. (2019).

In ***regularization-based techniques***, catastrophic forgetting is tackled by strategic regularization to support controlled weight updates based on previously learned parameters and the significance of past tasks. Elastic Weight Modification Kirkpatrick et al. (2017) computes the importance of previous task weights and distribution of data based on the diagonal elements of the Fischer information matrix. Some work Zenke et al. (2017); Aljundi et al. (2018) use appropriate synapses to efficiently accumulate and utilize relevant information from previous tasks to prevent catastrophic forgetting while learning new tasks. Ritter et al. Ritter et al. (2018) apply Gaussian Laplace approximation of Hessian to estimate the task-based posterior. Farajtabar et al. Farajtabar et al. (2020) update the new task weights orthogonally to the gradient direction of previous tasks. Subsequently, distillation methods Hinton et al. (2015); Li & Hoiem (2017); Hu et al. (2018) have also been helpful in extracting relevant information from previous tasks and imposing regularization. The stability and plasticity dilemma was addressed by framing two types of residual blocks in Adaptive Aggregation Networks Liu et al. (2021). Combining knowledge distillation and replay, Boschini et al. Boschini et al. (2022) et al. introduced eXtended-DER (X-DER), where the model is presented with the ability to revise the replay memory. CO-transport for class Incremental Learning (COIL) adapts to new tasks by learning the class-wise semantic relationship across incremental tasks Zhou et al. (2021).

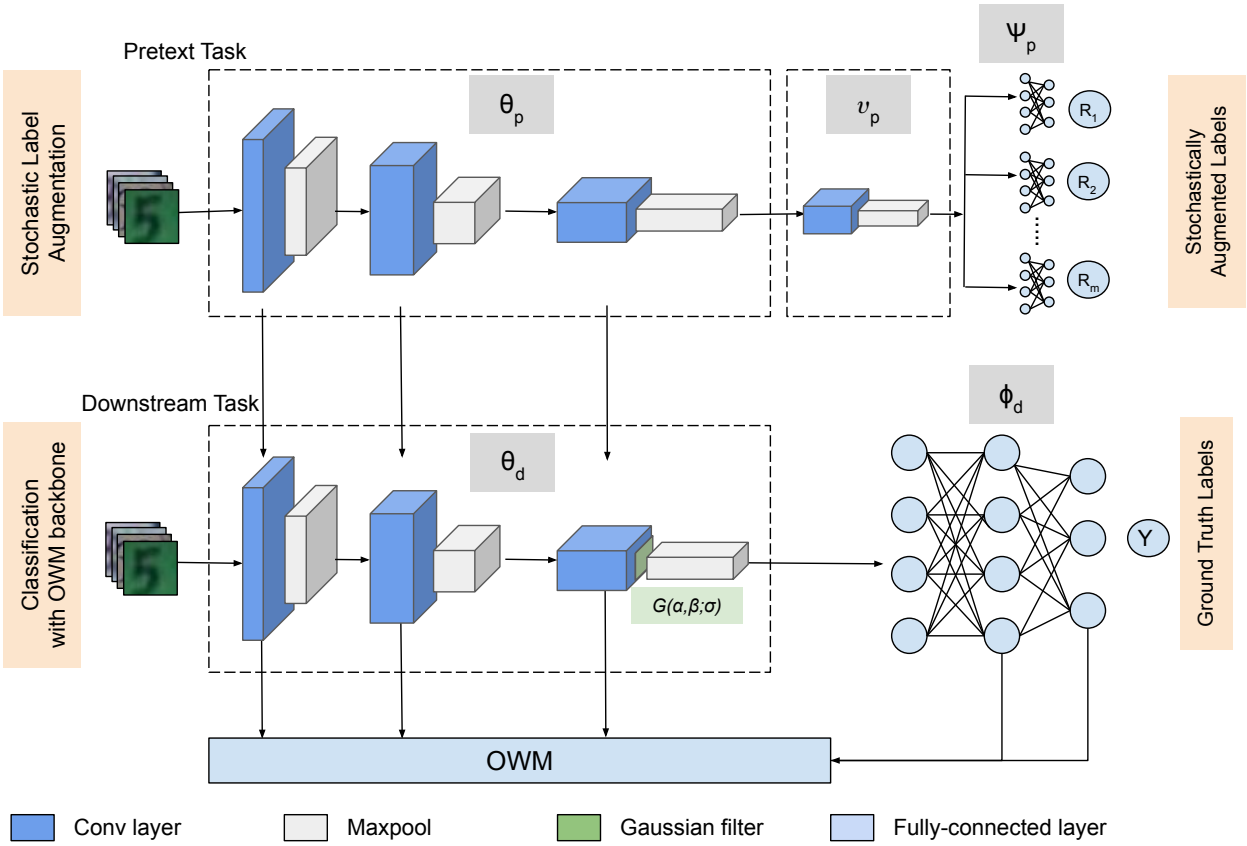

Figure 2: Block diagram of the proposed pretext and the downstream task for a particular incremental task t.

3 Methodology

Let $D = \{D^t\}_{t=1}^{T}$ be the dataset organized for continual learning. Here D^t consists of a set of N^t images for the incremental task $t \in [1, T]$. Each incremental task t corresponds to the addition of a new set of classes. The dataset D constitutes $(x_i^t, y_i^t)^t$ where $x_i^t \in X^t$ represents the set of images per task and $y_i^t \in Y^t$ as the corresponding ground truth-label for task t. During each incremental task, we define a model f with parameters θ^t and ϕ^t where θ denotes the parameters in the convolution layers and ϕ denotes the parameters in the fully-connected (FC) layers. The model is trained incrementally per task t on D^t thus culminating the final model as $f(\theta^T, \phi^T)$. Fig. 2 illustrates the block diagram describing the framework for the pretext and downstream training for a single incremental task t. The proposed algorithm consists of:

1. *Unsupervised Pre-Training by Stochastic Label Augmentation:* In this step, we propose a novel pretext task for feature extraction. The pretext task augments labels instead of images for learning feature-rich representations.

2. *Downstream Training in Class-Incremental Setting:* For the downstream task, we employ a curriculum-based smoothing mechanism in combination with Orthogonal Weight Modification (OWM).

3. *Self-Supervised CIL with Task Regularization:* The model is trained iteratively for the pretext and downstream tasks with the proposed task regularization term to prevent catastrophic forgetting.

4. *Penultimate Weight Sharing:* The weights obtained after training the pretext model is transferred to the downstream model with the exception of weights from the last layer.

Algorithm 1: Pretext training using *SLA* for task t

Input: Images X^t, Downstream model conv-layer parameters θ_d^{t-1} from previous task $t-1$
Initialize: θ_d^0 is zero-initialized.
Parameters: Number of epochs E, Number of stochastic tasks M, Number of classes per stochastic task N, Pretext model conv-layer parameters θ_p^t, Pretext model FC parameters ψ_p^t, hyperparameters a and b
Function ***train_pretext_model***(X^t, θ_d^{t-1})
  Initialize model $f(\theta_p^t, \upsilon_p^t, \psi_p^t)$
  R^t **=** $generate_stochastic_labels(X^t, m, n)$
  for *e=1 to E* **do**
    $\hat{R}^t = f(X^t; \theta_p^t, \upsilon_p^t, \psi_p^t)$
    Calculate loss terms:
    $L_1 = \sum_{i=1}^{M}(\sum_{j=1}^{N} -R^t log(\hat{R}^t))$
    $L_2 = (a/2)\left\|\theta_p^t - \theta_d^{t-1}\right\|_2^2 + (b/2)\left\|\theta_p^t\right\|_2^2$
    $L = L_1 + L_2$
    Backpropagate loss L ***and update*** θ_p^t ***and*** ψ_p^t
  end
  return θ_p^t
end

3.1 Unsupervised Pre-Training by Stochastic Label Augmentation

For unsupervised pre-training, we employ self-supervised learning where the training of model f is initiated by first training it for a novel pretext task termed as *Stochastic Label Augmentation (SLA)*. In Self-Supervised Learning (SSL), we generally augment the input images during pretext tasks, which is computationally inefficient. Consequently, we propose *SLA*, which is based on augmenting the makeshift labels instead of images. These labels are stochastically generated for M tasks, with each task having C classes. The training is performed in a multi-task fashion with different fully-connected layers for each of the M tasks (Fig. 2). The major advantage of the proposed pretext task over the existing approaches (for instance, rotation pretext) is that the training time and resource usage do not increase significantly due to an increase in data ($4\times$ in the case of rotation pretext Komodakis & Gidaris (2018)).

We integrate self-supervised pre-training in CIL setting by training the network for a pretext task p at each incremental task t. $f(\theta_p^t, \upsilon_p^t, \psi_p^t)$ represents the network for the pretext task p where θ_p^t denotes the convolution parameters except the last block, υ_p^t denotes the last convolution block, and ψ_p^t denotes the weights of the fully-connected layers for pretext task p which gives the softmax of logits as output. The convolution weights θ_p^t (excluding υ_p^t) are transferred to the downstream task d for incremental task t. The model $f(\theta_p^t, \upsilon_p^t, \psi_p^t)$ utilizes only the images $X^t \in D^t$ for pre-training. Subsequently, M branches of fully-connected layers (ψ_p) are added to the CNN corresponding to each of the M tasks. This model is trained in through multi-task learning where the features are extracted from the convolutional layers and then, for each of the M tasks (not to be confused with the incremental task t), and fully-connected layers are trained with the cumulative loss incurred from all the tasks. The loss function used for training the network in pretext task is:

$$\min_{\theta_p^t, \upsilon_p^t, \psi_p^t} \mathbb{E}_{(x^t, y^t)\sim D^t} L(f(\theta_p^t, \upsilon_p^t, \psi_p^t; x^t), y^t) = \min_{\theta_p^t, \upsilon_p^t, \psi_p^t} \mathbb{E}_{(x^t, y^t)\sim D^t} \frac{1}{N^t} \sum_{i=1}^{N^t} \sum_{j=1}^{M} \sum_{k=1}^{C} -y_{i,j,k}^t log f(\theta_p^t, \upsilon_p^t, \psi_p^t; x_{i,j}^t) \quad (1)$$

where each $y_{i,j,k}$ is the stochastic label assigned to data point $x_{i,j}$. The model takes each image as input and predicts vectorized probabilities corresponding to each task. $L(.)$ is the cross-entropy loss minimized between the target vector of stochastically generated labels and the predicted vector. The pretext training is described in Algorithm 1.

In a regularization-based class-incremental setting, the model only learns discriminative features from images that are required for a single-incremental task. These features are not discriminative enough as more tasks are introduced to

Algorithm 2: Downstream learning algorithm for task t

Input: Images X^t, Labels Y^t, Pretext model conv-layer parameters θ_p^t for current task t
Initialize: θ_d^t is initialized with θ_p^t.
Parameters: Number of epochs E, Downstream model FC parameters ϕ_d^t, Gaussian filter G with standard deviation σ, constant c
Function ***train_downstream_model***($\boldsymbol{X}^t, Y^t, \theta_d^t$)
  Initialize model $f(\theta_d^t, \phi_d^t)$
  Set σ_e **=** ***1***
  for *e=1 to E* **do**
    $\hat{a}_n = f(X^t; \theta_d^t, \phi_d^t)$
    $\sigma_e = \sigma_e . c$
    $z_{n+1} = pool(G(\sigma_e) * \hat{a}_n)$
    $\hat{Y}^t = argmax(z_{n+1})$
    Calculate loss terms:
    $L = \sum_{i=1}^{M}(\sum_{j=1}^{N} -Y^t log(\hat{Y}^t))$
    Backpropagate loss L ***and update*** θ_d^t ***and*** ϕ_d^t ***using OWM algorithm.***
  end
  return θ_d^t, ϕ_d^t
end

the model. Pre-training the model with unlabeled data drives the model to extract more and more information from each input image as the model will be forced to minimize the loss for each of the tasks. After pre-training the model $f(\theta_p^t, \upsilon_p^t, \psi_p^t)$ for the incremental task t through the pretext task, Penultimate layer Weight Sharing (PWS) is adopted to transfer the convolution weights θ_p^t to the downstream model $f(\theta_d^t, \phi_d^t)$. The proposed pretext task learns generalized, unnoticed, and discreet features from the limited data and thus, save training time and computation resources.

3.2 Downstream Training in Class-Incremental Setting

The downstream task for classification in a class-incremental setting involves learning of the downstream model $f(\theta_d^t, \phi_d^t)$ for each incremental task t. For this learning, we propose a curriculum-based learning approach with an OWM backbone Zeng et al. (2019). The downstream training of the model using these components is described in Algorithm 2.

3.2.1 Orthogonal Weight Modification (OWM)

The OWM technique has been shown to address the problem of catastrophic forgetting commonly observed in continual learning problems Zeng et al. (2019). In the OWM technique, an orthogonal projection matrix P_l is considered on the input space of layer l. This projector is defined as

$$P_l = I - A_l(A_l^T A_l + \alpha I)^{-1} A_l^T \tag{2}$$

where α is a constant to enable the calculation of the inverse of the matrix and I is a unit matrix. For each incremental task t, the weights ϕ_l^t and projector P_l are updated for each layer l in the network, such that the information learned in the previous tasks is retained. The weights ϕ_l^t are updated as:

$$\Delta\phi_l^t = \lambda P_l^{t-1} \Delta\phi_l^{t\,(BP)} \tag{3}$$

where λ is the learning rate, $\Delta\phi^{(BP)}$ is the standard weight update using the backpropagation algorithm. The projector P may be updated using an iterative or a recursive method to obtain a correlation-inverse matrix Shah et al. (1992); Haykin (2008). We use OWM as the backbone technique to address the problem of catastrophic forgetting in the proposed algorithm.

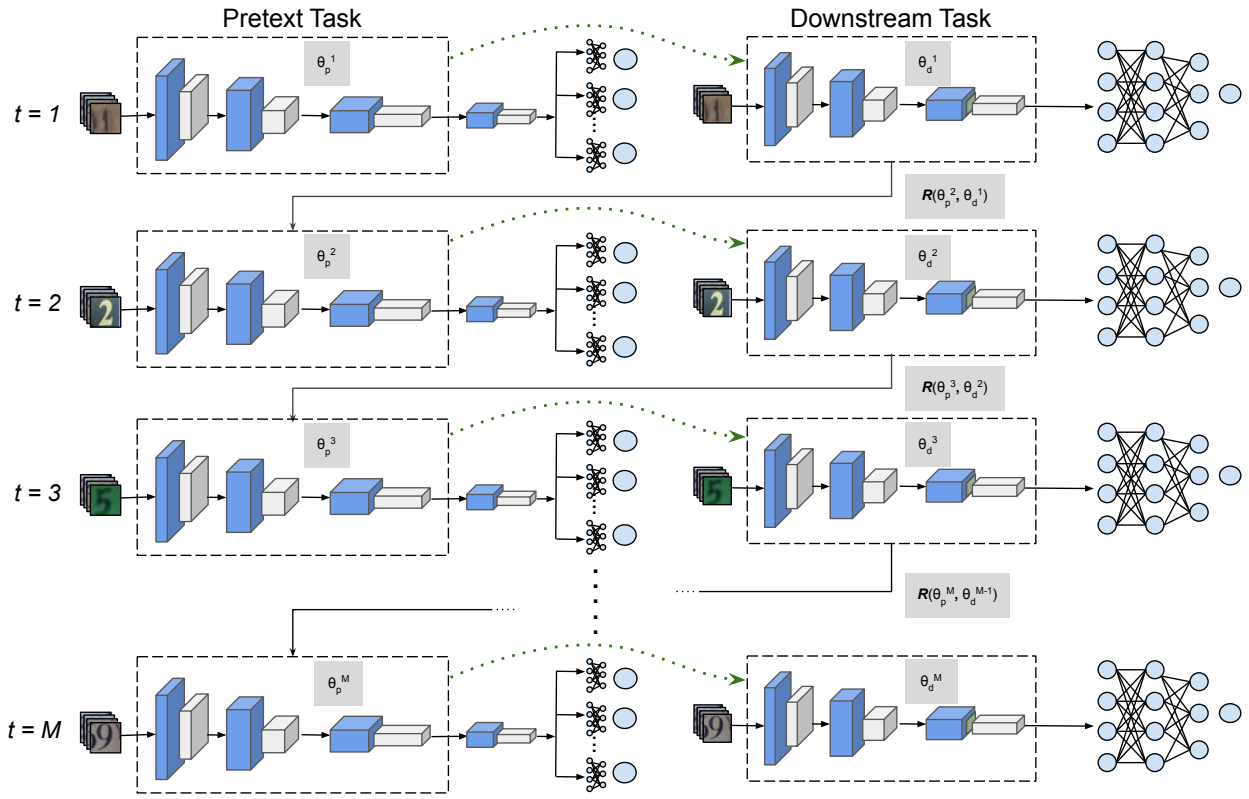

Figure 3: Block diagram to demonstrate the complete training procedure of the proposed algorithm for M incremental tasks.

3.2.2 Smoothing-based Curriculum Learning

We design a curriculum-based learning technique for training the downstream model. Recent work has shown the effectiveness of Gaussian smoothing in the context of curriculum learning Chen et al. (2019); Sinha et al. (2020). Convolution of a conventional smoothing kernel with an input signal results in a blurring effect. This means that some information in the input is lost. In other words, the smoothing kernel regulates the information that is propagated after each convolution operation.

A Gaussian filter G, parameterized with σ_e is applied to the extracted feature maps from the last convolution layers of the model. While training over e epochs, we increase the strength of the smoothing filter G simply by increasing σ_e. In a standard CNN model with weights, θ_l the following operations are performed at layer l. The activations at each layer is obtained as $\hat{a}_l = ReLU(\theta_l * z_l)$ followed by pooling, described as $z_{l+1} = pool(\hat{a}_l)$. Here $\hat{a}_l$ denotes the activated output using the rectified linear unit $ReLU$, input z_l denotes the input at layer l, $*$ is the convolution operation and $pool$ is the max-pooling layer. For an n-layer CNN with weights θ_l at each layer l, we integrate a smoothing filter after the nth convolution layer of the CNN. This can be expressed as follows:

$$z_{n+1} = pool(G(\sigma_e) * \hat{a}_n) \tag{4}$$

where z_{n+1} becomes the input to the first fully-connected layer.

The curriculum defined above is built over the observation that for each incremental task t, the feature maps obtained at the last convolutional layer have an abundance of information from which a model can learn. Plenty of information in the maps allows the model to focus on the features that are easy to extract and lead to the best optimization of the objective. This makes it an easy sample in the curriculum. Over the epochs, the difficulty of the curriculum is increased by repressing the high-frequency information from the feature maps. This is achieved by increasing the standard deviation of the smoothing kernel. The model is then forced to extract discriminative, inconspicuous, or obscured features from the smoothened feature map. This provides a curriculum-based training where the model learns to classify with

Algorithm 3: S^2C^2IL algorithm

Input: Total tasks T, Images X, Labels Y
Parameters: Pretext model conv-layer parameters θ_p, Pretext model FC parameters ψ_p, Downstream model conv-layer parameters θ_d, Downstream model FC parameters ϕ_d
for *t = 1 to T* **do**
  $\theta_p^t = train_pretext_model(X_t, \theta_d^{t-1})$ //Algorithm 1
  $\theta_d^t, \phi_d^t = train_downstream_model(X_t, Y_t, \theta_p^t)$ //Algorithm 2
end
Evaluate model $f(\theta_d^T, \phi_d^T)$ trained for T tasks.

a lesser and lesser amount of information. Furthermore, Gaussian smoothing filters have traditionally been used for anti-aliasing in image processing Gonzalez (2009). Anti-aliasing, when integrated correctly, has been shown to enhance the shift-invariance tendency of CNNs Zhang (2019). The fusion of smoothing (blur) filters with pooling/strided convolution softens the feature maps and alleviates the variance introduced by operations that predominantly ignore the Nyquist sampling theorem. Recently, the fusion of these filters has shown a boost in the performance and generalization capacity of the CNN models Zhang (2019); Zou et al. (2020). By incorporating a Gaussian smoothing-based curriculum for training, we expect improved robustness of the model towards a shift in input.

3.3 S^2C^2IL: Self-Supervised Curriculum-based Incremental Learning

Since the model is being trained in a class-incremental fashion, we only have limited data corresponding to the classes that are introduced. To make maximum use of the limited amount of data, we begin with the pretext task of self-supervision using the data at hand. This ensures that the model is able to learn good feature representations from the provided data. In the next step, the model learns to perform classification between the given set of classes using OWM combined with curriculum learning.

The cycle of pre-training and downstream classification is repeated every time a new set of classes arrive (Fig. 3). In practical instances, it is highly likely that the dataset associated with the new incremental task may belong to an entirely different distribution. Training the existing model on this new, out-of-distribution dataset may lead to excessive modification of convolution weights, and the final model may fail to generalize on the old incremental tasks. To prevent forgetting for convolutional layers at each incremental step, we incorporate a regularization term in the calculated loss Xuhong et al. (2018). The regularization term ensures that the weights are not drastically modified after each pre-training and classification cycle. The complete S^2C^2IL algorithm is presented in Algorithm 3. Mathematically put, while training the model for the pretext task of incremental task $t-1$ on dataset D^{t-1}, we transfer the convolution weights θ_p^{t-1} from pretext model $f(\theta_p^{t-1}, \psi_p^{t-1})$ to downstream model $f(\theta_d^{t-1}, \phi_d^{t-1})$. After this, on the introduction of the next incremental task t, the model is first trained for the pretext task. For this, the weights from the previous downstream task θ_d^{t-1} are transferred. We add a weight regularization term between the convolution weights between θ_d^{t-1} of the previous task and θ_p^t of the current task to mitigate forgetting at this step. In other words, we incur a regularization loss term R between the model trained on the downstream task of incremental task $t-1$ and the pretext task for the model being trained on the next incremental task i.e. incremental task t. The proposed regularization loss incurred on two consecutive incremental tasks is:

$$\boldsymbol{R}(\theta_p^t, \theta_d^{t-1}) = \frac{\boldsymbol{a}}{2} \left\|\theta_p^t - \theta_d^{t-1}\right\|_2^2 + \frac{\boldsymbol{b}}{2} \left\|\theta_p^t\right\|_2^2 \tag{5}$$

where a and b are hyperparameters, and $\boldsymbol{R(.)}$ is the regularization loss optimized with the standard multi-task cross-entropy loss. The hyperparameter a is a constant that aggravates the loss, forcing the model not to deviate much from the model trained on the previous task. The hyperparameter b handles the induced sparsity on the model being trained on the current incremental task t.

4 Experimental Setup

The proposed algorithm is evaluated on four datasets: split-CIFAR10, split-CIFAR100, split-SVHN and split-TinyImageNet. We report the average test accuracy, which is defined as the average of test accuracies achieved across all tasks. All experiments are performed using five fixed random seeds. The proposed algorithm is evaluated under two settings- (i) OWM + CL, and (ii) S^2C^2IL. In the first setting, only the curriculum-based downstream model is trained without any self-supervision. In the S^2C^2IL setting, we follow the methodology as explained in Section 3.3, and perform pre-training using the proposed SLA technique.

Datasets and Protocol: Since the focus of this work is class-incremental setting, we train and test the proposed algorithm according to the protocols defined in the works of Zeng et al. Zeng et al. (2019) and Hu et al. Hu et al. (2018). For experiments, we have used four datasets:

(i) **Split-CIFAR 10 Krizhevsky (2009)** contains 60,000 32×32 color images of 10 different classes with 50,000 images in the training set and 10,000 images in the testing set. The training and evaluation is performed for 2 classes per task.

(ii) **Split-CIFAR 100 Krizhevsky (2009)** contains 60,000 32×32 color images of 10 different classes with 50,000 images in the training set and 10,000 images in the testing set. The training and evaluation are done for 10, 20, and 50 classes per task.

(iii) **Split-SVHN Netzer et al. (2011)** contains 60,000 32×32 color images of 10 different classes with 50,000 images in the training set and 10,000 images in the testing set. The training and evaluation are performed for 2 classes per task.

(iv) **Split-TinyImageNet Le & Yang (2015)** contains 120,000 color images of size 64×64 from 200 different classes with 100,000 images in the training set, 10,000 in the validation set and 10,000 images in the testing set. The training and evaluation of the model are done for 5, 10, and 20 classes per task.

Comparison Algorithms: The results of the proposed framework are compared with various benchmark algorithms in the domain of regularization-based CIL with the exception of iCaRL. The following algorithms are used for comparison: (1) EWC Kirkpatrick et al. (2017), (2) iCaRL Rebuffi et al. (2017) with 2000 exemplars; (3) PGMA Hu et al. (2018); (4) DGM Ostapenko et al. (2019), (5) OWM Zeng et al. (2019), (6) MUC Liu et al. (2020), (7) IL2A Zhu et al. (2021a), (8) PASS Zhu et al. (2021b), (9) SSRE Zhu et al. (2022), and FeTrIL Petit et al. (2023). The EWC[1], iCaRL[1], DGM[2], OWM[3], MUC[4], IL2A[5], PASS[6], SSRE[7], and FeTrIL[8] baselines are run using open-source codes with the same network architecture as the one used in S^2C^2IL. The details of this network are described in Section 4. Further, S^2C^2IL is compared with various memory-based approaches on the Split-TinyImageNet dataset. It should be noted that the proposed S^2C^2IL algorithm uses *no* exemplars from classes of previous tasks.

Implementation Details: For all the experiments, we use a 3-layer CNN network with three fully-connected layers. The same network architecture is used by Zeng et al. Zeng et al. (2019). For each incremental task, we start the model training on the pretext task. Here, we use the 3-layer CNN architecture for feature extraction and utilize these features in multitask fashion. For our experiments, we train the model for three tasks with two classes each, i.e., the extracted features are utilized by three separate heads of fully-connected layers with two layers each. For the downstream task, the same weights from the pretext task are transferred. However, here the features are utilized by a single fully-connected layer to learn the current incremental task. The mentioned architectures used in the pretext and downstream can be better visualized in Fig. 2. We train all the models with stochastic gradient descent (SGD). For the pretext task, the multitask network is trained on stochastically generated labels for three tasks with two classes

[1]https://github.com/mmasana/FACIL
[2]https://github.com/SAP-archive/machine-learning-dgm
[3]https://github.com/beijixiong3510/OWM
[4]https://github.com/liuyudut/MUC
[5]https://github.com/Impression2805/IL2A
[6]https://github.com/Impression2805/CVPR21$_PASS$
[7]https://github.com/zhukaii/SSRE/tree/5475c9803b0143cab849b62edb7d5db76433c388
[8]https://github.com/G-U-N/PyCIL

Methods	Split-CIFAR10	Split-CIFAR100			Split-SVHN
	5 tasks	2 tasks	5 tasks	10 tasks	5 tasks
EWC* Kirkpatrick et al. (2017)	31.40 ± 2.21	27.58 ± 1.64	18.42 ± 1.53	13.28 ± 0.91	34.22 ± 3.83
iCaRL* Rebuffi et al. (2017)	50.02 ± 2.04	24.20 ± 1.60	22.16 ± 0.86	19.00 ± 0.36	71.25 ± 0.67
PGMA Hu et al. (2018)	40.47	-	-	-	-
DGM* Ostapenko et al. (2019)	50.53 ± 0.46	28.23 ± 0.75	25.43 ± 0.14	24.09 ± 0.19	73.01 ± 0.77
OWM Zeng et al. (2019)	55.71 ± 0.49	40.30 ± 0.65	33.17 ± 0.79	29.86 ± 0.33	73.50 ± 0.81
MUC* Liu et al. (2020)	-	33.86 ± 0.72	28.05 ± 1.22	22.07 ± 0.9	-
IL2A* Zhu et al. (2021a)	-	43.29 ± 0.43	32.63 ± 0.86	21.45 ± 0.67	-
PASS* Zhu et al. (2021b)	-	43.15 ± 0.31	34.89 ± 0.75	24.03 ± 0.74	-
SSRE* Zhu et al. (2022)	-	41.06 ± 0.87	**36.82 ± 0.7**	31.35 ± 1.0	-
FeTrIL* Petit et al. (2023)	-	40.88 ± 1.18	35.47 ± 1.15	**32.50 ± 1.03**	-
OWM + CL (Ours)	58.68 ± 0.37	43.10 ± 0.66	35.40 ± 0.36	31.37 ± 0.61	75.34 ± 0.64
S^2C^2IL (Ours)	**61.64 ± 0.57**	**43.98 ± 0.65**	35.59 ± 0.49	31.93 ± 0.54	**77.53 ± 0.53**

Table 1: Average test accuracy for proposed method on Split-CIFAR10, Split-CIFAR100, and Split-SVHN dataset. The best performance is depicted by **bold** and the second best by underline. All results are cited from Kirkpatrick et al. (2017); Rebuffi et al. (2017); Ostapenko et al. (2019); Hu et al. (2018); Zeng et al. (2019); Liu et al. (2020); Zhu et al. (2021a;b; 2022); Petit et al. (2023) or reproduced from their official repository for a fair comparison (* means re-run with protocols described in this paper)

Methods	5 tasks	10 tasks	20 tasks
OWM Zeng et al. (2019)	19.00 ± 0.28	16.05 ± 0.27	14.30 ± 0.32
SLA + OWM	20.59 ± 0.32	17.05 ± 0.58	15.08 ± 0.65
OWM + CL	21.12 ± 0.42	17.56 ± 0.33	15.54 ± 0.12
S^2C^2IL	**21.39 ± 0.15**	**19.00 ± 0.35**	**19.52 ± 1.56**

Table 2: Average accuracy (%) reported for the ablation experiments performed on the split-TinyImageNet dataset for 5, 10, and 20 tasks.

each. We set the learning rate to 0.001 to train it for 50 epochs. The hyperparameters a and b are fixed to 10 and 18 for split-CIFAR10 and split-CIFAR100 datasets and 5 and 12 for split-SVHN datasets, respectively. As described in section 3, the model is trained for a curriculum where the training starts with σ set to 0.9 with a decay rate of 0.95 for every 10 epochs for split-CIFAR100 and split-SVHN datasets. For the split-CIFAR10 dataset, σ is set to 1 with a decay rate of 0.9 for every ten epochs. All experiments are performed for five random seeds and the performance is reported as the average over all the seeds. The algorithm is implemented in Pytorch, and all the experiments are performed on a DGX station with 256 GB RAM and four 32 GB Nvidia V100 GPUs. For reproducibility, the source code will be released in the camera-ready version.

5 Results and Analysis

The performance of the proposed S^2C^2IL algorithm on the split-CIFAR10, split-CIFAR100, and split-SVHN datasets are reported in Table 1. From Table 1, it is observed that the proposed algorithm achieves state-of-the-art performance

Pretext Task	Average Accuracy (%)	
	split-CIFAR-10	split-SVHN
Rotation Komodakis & Gidaris (2018)	57.59 ± 0.43	76.22 ± 0.23
Colorization Larsson et al. (2017)	56.66 ± 0.92	76.10 ± 0.69
SLA (proposed)	**61.64 ± 0.57**	**77.53 ± 0.53**

Table 3: Performance of the proposed algorithm by replacing the proposed Stochastic Label Augmentation (SLA) with Rotation and Image Colorization pretext tasks on the split-CIFAR10 and split-SVHN datasets for 5 tasks.

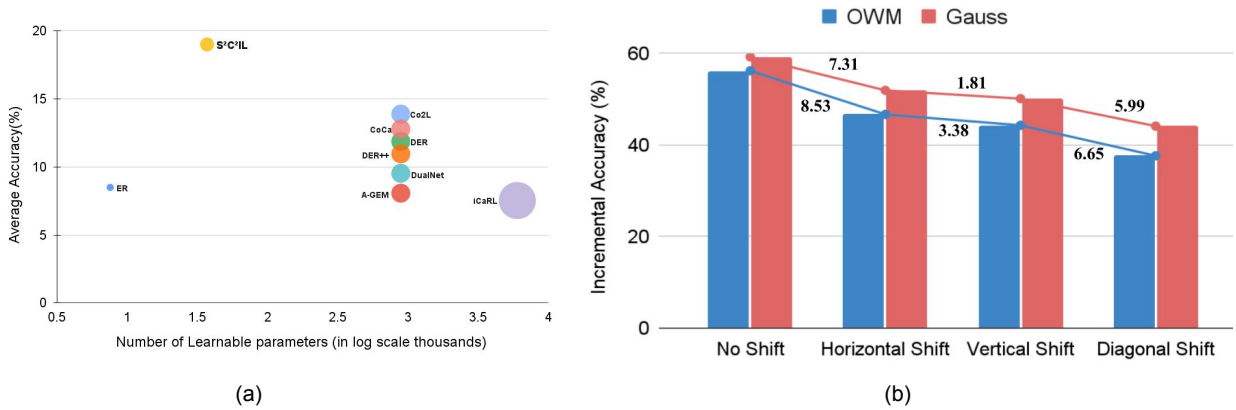

Figure 4: (a) Comparison of the proposed S^2C^2IL algorithm with memory-based continual learning algorithms Riemer et al. (2018); Chaudhry et al. (2019); Rebuffi et al. (2017); Benjamin et al. (2018); Buzzega et al. (2020); Pham et al. (2021); Cha et al. (2021); Ji et al. (2022). The accuracy achieved by each algorithm on the split-TinyImageNet dataset (for 10 incremental tasks) is plotted against the number of convolution parameters (in log scale). The size of each bubble corresponds to the network size used by the algorithm. The results obtained are summarized in Table 4. (b) Bar plot demonstrating the incremental accuracies for OWM and Gaussian-based OWM model when trained on split-CIFAR10 dataset for 5 tasks. The line graph summarizes the subsequent drop in the accuracy caused by pixel translations. Gaussian network displays far less performance drop for translated dataset than base OWM model.

on the split-CIFAR10 dataset and split-SVHN dataset when compared to the existing algorithms. When compared to the backbone algorithm (Zeng et al. (2019)), S^2C^2IL improves the average accuracy with up to 4% and 6% performance gain for the split-SVHN and split-CIFAR10 datasets, respectively. Further, we observe that training without self-supervision in S^2C^2IL (OWM + CL) also outperforms existing algorithms on both datasets. For the split-CIFAT100 dataset, our proposed algorithm achieves state-of-the-art performance closely followed by IL2A Zhu et al. (2021a) by a difference of 0.69% for 2 incremental tasks. S^2C^2IL closely follows SSRE Zhu et al. (2022) by 1.13% for 5 incremental tasks, and FeTrIL Petit et al. (2023) by 0.57% for 10 incremental tasks.

In Table 2, we perform the ablation experiments and report the performance on the split-TinyImageNet dataset. The evaluation is performed for 5, 10, and 20 tasks. From 2, it can be observed how each component contributes towards mitigating catastrophic forgetting. We perform an additional experiment comparing the proposed regularization-based S^2C^2IL algorithm with a recent and state-of-the-art memory-based algorithm. In Fig. 4 (a) the accuracy achieved by each algorithm is plotted against the number of parameters in convolution layers (in log scale of thousands). It can

Algorithm	**Accuracy (%)**	**Network Used**	**Parameters (in million)**
iCaRL Rebuffi et al. (2017)	7.53 ± 0.79	ResNet32	60
ER Riemer et al. (2018)	8.49 ± 0.16	**Four-layerd CNN**	**0.08**
A-GEM Chaudhry et al. (2018)	8.07 ± 0.08	ResNet18	8.98
DER Buzzega et al. (2020)	11.87 ± 0.78	ResNet18	8.98
DER++ Buzzega et al. (2020)	10.96 ± 1.17	ResNet18	8.98
DualNet Pham et al. (2021)	9.53 ± 0.53	ResNet18	8.98
Co2L Cha et al. (2021)	$\underline{13.88 \pm 0.42}$	ResNet18	8.98
CoCa Ji et al. (2022)	12.78 ± 0.0	ResNet18	8.98
S^2C^2IL (Proposed)	$\mathbf{19.00 \pm 0.35}$	Four-layerd CNN	0.37

Table 4: The accuracy achieved by different memory-based algorithms with backbone architecture used by each algorithm on the split-TinyImageNet dataset for 10 tasks. The best performance is depicted by **bold** and the second best by underline. The proposed S^2C^2IL achieves the highest performance with no memory and with significantly smaller backbone architectures.

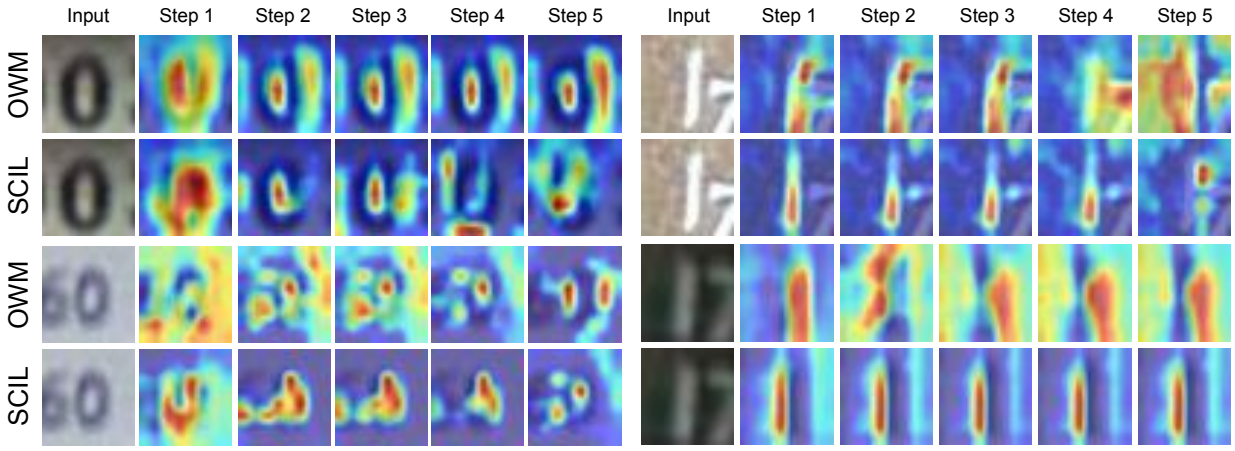

Figure 5: Task-wise GradCAM visualization (step in the figure stands for an incremental task) for the split-SVHN dataset using the OWM (row 1 and row 3) and proposed S^2C^2IL algorithm (row 2 and row 4).

be clearly visualized that the proposed S^2C^2IL algorithm outperforms existing memory-based algorithms by a large margin on the split-TinyImageNet dataset for ten tasks. Further, the S^2C^2IL framework exceeds the performance of memory-based algorithms even with a smaller backbone and *without* using any memory. The results obtained are summarized in Table 4. The results from Table 1 and Table 2 show the generalizability of the proposed S^2C^2IL on various datasets showcasing the efficacy of the proposed algorithm.

To qualitatively evaluate the performance of the proposed algorithm, we employ GradCAMs. In Fig. 5, the GradCAM visualization obtained after each incremental task using OWM algorithm and the proposed S^2C^2IL algorithm is presented. We use the images from the first incremental task of the split-SVHN dataset. From the generated maps, it is observed that after each incremental task/step, the focus of the model diverges in the case of the OWM algorithm. However, the maps generated through S^2C^2IL are better at retaining focus even after multiple incremental training steps. This highlights the stability of the proposed S^2C^2IL algorithm and its effectiveness at delaying forgetting in the network.

Efficacy of Stochastic Label Augmentation (SLA): In order to understand the effectiveness of the proposed pretext task, we perform additional experiments on the split-CIFAR10 and split-SVHN datasets for five incremental tasks each. The proposed framework is tested after replacing the SLA pretext task with two existing pretext tasks, namely Rotation Komodakis & Gidaris (2018) and Image Colorization Larsson et al. (2017). From Table 3, we observe that in comparison to Image Colorization and Rotation, SLA leads to a higher performance gain. In addition to performance gain through SLA, it should be noted that augmenting the labels instead of data leads to faster pre-training, lesser computational cycles as well as low memory usage. The reason behind successful learning through augmented labels instead of data is due to the fact that a multi-task network learns from the synergy of multiple tasks that it has to learn. Since the network is bound to minimize the loss, it will excerpt all the discriminating features to minimize it. Moreover, since the deep learning models are highly non-linear, they can reasonably achieve near-perfect accuracy on

Dataset	**Tasks**	**S^2C^2IL (w/o PWS)**	**S^2C^2IL**
split-SVHN	5	75.88 ± 0.45	77.53 ± 0.53
split-CIFAR10	5	60.75 ± 0.53	61.64 ± 0.57
split-CIFAR100	2	43.38 ± 0.20	43.98 ± 0.65
	5	35.52 ± 0.48	35.59 ± 0.49
	10	31.85 ± 0.45	31.93 ± 0.54

Table 5: Performance comparison of the proposed algorithm without and with Penultimate Weight Sharing (PWS). The proposed Stochastic Label Augmentation (SLA) task is used for unsupervised pre-training.

	Average Incremental Accuracy			
	OWM	δ_{owm}	Gauss	δ_{Gauss}
Original	56.2	0	59.23	0
Horizontal Shift	47.67	8.53	51.92	**7.31**$_{-1.22}$
Vertical Shift	44.29	11.91	50.11	**9.12**$_{-2.79}$
2D Shift	37.64	18.56	44.12	**15.11**$_{-3.45}$

Table 6: Average accuracy (in %) response of the network to the original and shifted datasets. δ_{owm} and δ_{Gauss} represent the accuracy drop in the OWM and Gaussian-based OWM models. A higher value of δ implies a more adverse effect of shift on the model performance.

the training dataset Zhang et al. (2017). The combination of these prospects leads to performance gain, which makes learning without the availability of true annotated labels possible.

Impact of Penultimate Weight Sharing (PWS): Conventionally, the optimized weights of the pretext task are utilized for training the model on the downstream task. The quality of the pre-trained features consistently improves with the position and depth of layers. Further, the task accuracy is influenced by pre-training a network only up to k-layers Misra & Maaten (2020). We hypothesize that the deeper layers of the pretext task are calibrated toward the pretext task. To alleviate the bias towards the pretext task, we transfer weights from all layers except those of the last layer for fine-tuning on the downstream task. This weight transfer algorithm is termed Penultimate Weight Sharing (PWS). PWS incorporates layers weight (θ_p^t) sharing for downstream fine-tuning (dropping out υ_p^t) and empowers the network to learn generalized feature representations for task t. The benefit of sharing weights only up till the penultimate layer prevents sharing of pretext-specific weights to the downstream model.

To evaluate the model's performance in the absence of PWS, we remove the υ_p^t convolution block from the pretext model, rendering it equivalent to the downstream model's architecture. After pre-training the model and transferring the weights to the downstream model, the performance is evaluated on different datasets. The results presented in Table 5 highlight the performance improvements obtained by transferring weights up to the penultimate layer.

Anti-aliasing Filters and Shift-Invariance: In this work, we employ a Gaussian filter for smoothing the feature maps during the downstream classification task. Since the fusion of these filters has shown improved generalization capabilities in CNNs Zhang (2019); Zou et al. (2020), we study the impact of the filtering integrated with a downstream model by evaluating the performance of S^2C^2IL using these filters. We highlight the relevance of primitive integration of the filter by stacking it after a convolution block and studying shift-invariance properties through related performance metrics. Fig. 4 (b) depicts the comparison of the incremental accuracy when a shifted/translated image is given as input to the OWM and Gaussian-based OWM network. We modify the split-CIFAR10 dataset by incorporating the random affine translation of 1% to ensure horizontal, vertical, and diagonal pixel shifts. The accuracies obtained corresponding to the shift in datasets are reported in Table 6. Attributed to the non-robustness of CNNs to shift, there is a decrease in the overall accuracy for class-incremental tasks. However, we observe the decrease in incremental accuracy to be less for Gaussian-based OWM than OWM, with an average decrease of about 2% less. This illustrates a steady response to translation in the dataset and highlights the shift-invariant tendency of the Gaussian-based network.

6 Conclusion

In this research, we focus on the problem of regularization-based class-incremental learning. We address it through unsupervised pre-training and propose a novel pretext task that augments labels instead of the data. During downstream training, we transfer the convolution weights till the penultimate layers from the pre-training and design a smoothing-based curriculum. We find that through the incorporation of self-supervised learning and curriculum learning, we are able to improve the generalizability of the model in the continual learning paradigm. The augmentation of labels instead of data in the pretext task further improves the learning for the current task and decreases the resource requirements for training the model. The utilization of the smoothing-based curriculum further enhances the model's performance. The proposed S^2C^2IL algorithm with the Orthogonal Weight Modification (OWM) backbone

achieves state-of-the-art results on split-CIFAR-10, split-CIFAR-100, split-SVHN and split-TinyImageNet datasets. The proposed algorithm can be appended with a memory component for future performance gains.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, pp. 139–154, 2018.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. *arXiv preprint arXiv:1805.08289*, 2018.

Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *arXiv preprint arXiv:2201.00766*, 2022.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9516–9525, 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019.

Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3435–3444, 2019.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3057446.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Rafael C Gonzalez. *Digital image processing*. Pearson education india, 2009.

Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2008.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Wenpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International conference on learning representations*, 2018.

Zhong Ji, Jin Li, Qiang Wang, and Zhongfei Zhang. Complementary calibration: Boosting general continual learning with collaborative distillation and self-supervision. *IEEE Transactions on Image Processing*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International conference on learning representations (ICLR)*, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. pp. 32–33, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6874–6883, 2017.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pp. 3925–3934. PMLR, 2019.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2544–2553, 2021.

Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pp. 699–716. Springer, 2020.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30:6467–6476, 2017.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11321–11329, 2019.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3911–3920, 2023.

Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

Hippolyt Ritter, Aleksandar Botev, and David Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *arXiv preprint arXiv:1805.07810*, 2018.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Syed Shakib Sarwar, Aayush Ankit, and Kaushik Roy. Incremental learning in deep convolutional neural networks using partial network sharing. *IEEE Access*, 8:4615–4628, 2019.

Samir Shah, Francesco Palmieri, and Michael Datum. Optimal filtering algorithms for fast learning in feedforward neural networks. *Neural networks*, 5(5):779–787, 1992.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *arXiv preprint arXiv:1705.08690*, 2017.

Samarth Sinha, Animesh Garg, and Hugo Larochelle. Curriculum by smoothing. *Advances in Neural Information Processing Systems*, 33:21653–21664, 2020.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

LI Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *International Conference on Machine Learning*, pp. 2825–2834. PMLR, 2018.

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Jaehong Yoon, Saehoon Kim, Eunho Yang, and Sung Ju Hwang. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv preprint arXiv:1902.09432*, 2019.

Guanxiong Zeng, Yang Chen, Bo Cui, and Shan Yu. Continual learning of context-dependent processing in neural networks. *Nature Machine Intelligence*, 1(8):364–372, 2019.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pp. 3987–3995. PMLR, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization (2016). *arXiv preprint arXiv:1611.03530*, 2017.

Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Co-transport for class-incremental learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1645–1654, 2021.

Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34:14306–14318, 2021a.

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5871–5880, 2021b.

Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9296–9305, 2022.

Xueyan Zou, Fanyi Xiao, Zhiding Yu, and Yong Jae Lee. Delving deeper into anti-aliasing in convnets. In *BMVC*, 2020.