IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 November 2025, accepted 30 November 2025, date of publication 10 December 2025,
date of current version 17 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3642464

RESEARCH ARTICLE

S^2IL: Structurally Stable Incremental Learning

**S. BALASUBRAMANIAN[1], P. YEDU KRISHNA[1], TALASU SAI SRIRAM[1],
M. SAI SUBRAMANIAM[1], MANEPALLI PRANAV PHANINDRA SAI[1],
AND RAVI MUKKAMALA[2], (Member, IEEE)**

[1]Department of Mathematics and Computer Science (DMACS), Sri Sathya Sai Institute of Higher Learning, Andhra Pradesh 515134, India
[2]Old Dominion University, Norfolk, VA 23529, USA

Corresponding author: Ravi Mukkamala (rmukkama@odu.edu)

ABSTRACT Feature Distillation (FD) strategies are proven to be effective in mitigating Catastrophic Forgetting (CF) seen in Class Incremental Learning (CIL). However, current FD approaches enforce strict alignment of feature magnitudes and directions across incremental steps, limiting the model's ability to adapt to new knowledge. In this paper, we propose Structurally Stable Incremental Learning (S^2IL), a FD method for CIL that mitigates forgetting by focusing on preserving the overall spatial patterns of features which promote flexible (plasticity) yet stable representations that preserve old knowledge (stability). We also demonstrate that our proposed method S^2IL achieves strong incremental accuracy and outperforms other FD methods on SOTA benchmark datasets CIFAR-100, ImageNet-100 and ImageNet-1K. Notably, S^2IL outperforms other methods by a significant margin in scenarios that have a large number of incremental tasks. The source code is available at `https://github.com/dlclub2311/Structurally-Stable-Incremental-Learning`

INDEX TERMS Catastrophic forgetting, class incremental learning, feature distillation, plasticity, stability, structural similarity.

I. INTRODUCTION

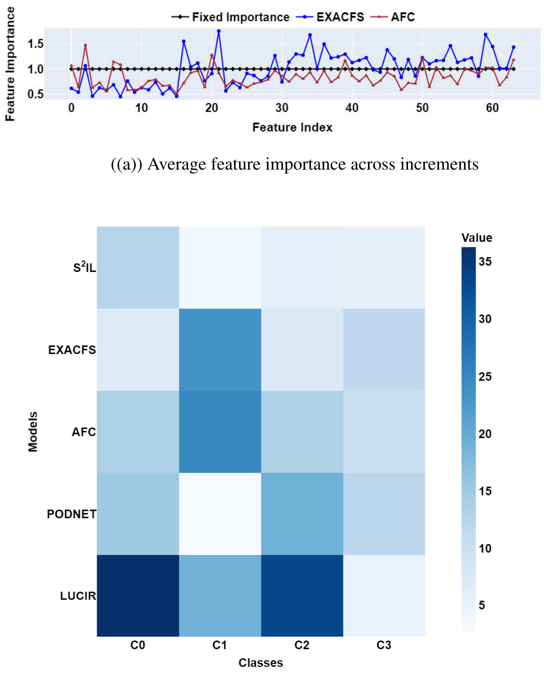
Humans learn continuously, with a unique ability to retain knowledge from past experiences. Recent efforts aim to enable machines to mimic this capability, promoting their meaningful use in solving real-world problems [1], [2]. A specific setting that has garnered significant attention over the past decade is class incremental learning (CIL) [3], [4], [5], [6], [7], [8], [9], where a machine learning model must incrementally adapt to new classes while retaining knowledge of previously learned ones. The tremendous success of the deep neural network (DNN) paradigm has spurred numerous approaches to addressing the CIL problem [7], [10], [11], [12], [13], [14]. Most approaches to CIL focus on addressing the critical challenge of catastrophic forgetting (CF), in which a model adapting to new classes tends to forget previously learned classes. A prominent one among them is based on feature distillation.

State-of-the-art (SOTA) feature distillation methods aim to enforce feature alignment across incremental tasks, thereby helping preserve representations for previously seen classes while the model is primarily trained on a new set of classes. Existing approaches typically minimize the squared-norm [6], [15] or the weighted squared-norm [13], [16] of the difference between corresponding feature maps from the current and previous tasks. The estimated weights reflect the importance of each feature map in recognizing previously seen classes [13], [16]. However, as discussed in Section IV-A and illustrated in Figure 1(a), these importance values often flatten at 1, effectively reducing these methods [13], [16] to simple squared-norm minimization in practice. Minimizing the squared-norm enforces alignment in direction and magnitude of feature maps, which stabilizes the model but restricts its flexibility to adapt to novel structures in new classes. This trade-off, favoring stability over adaptability, becomes especially problematic as the number of incremental tasks increases, ultimately constraining the model's plasticity.

We propose in this work a feature distillation approach called Structurally Stable Incremental Learning (S^2IL) that preserves structural similarity of features across incremental steps. Rather than enforcing exact matches in feature magnitudes and directions, our approach encourages the model to retain the overall spatial patterns within features, promoting flexible yet stable representations. This structural

The associate editor coordinating the review of this manuscript and approving it for publication was Ángel F. García-Fernández.

preservation, achieved using the Structural Similarity Index Measure (SSIM) [17], allows the model to adapt to new tasks while maintaining essential relationships in feature space, thereby enhancing both plasticity and stability.



((a)) Average feature importance across increments

((b)) Heatmap of Grad-CAM deviations

FIGURE 1. A motivation for exploring structure based feature distillation (FD): (a) Average feature importance ρ across increments from the last convolutional layer of two SOTA FD models, EXACFS and AFC, evaluated on CIFAR-100 with a Inc 10 setting. Both enforce feature similarity (magnitude and direction) between increments. A feature is deemed important if it significantly influences the loss. Surprisingly, ρ remains nearly constant at 1 for all the features, for both models, limiting plasticity by enforcing feature similarity of all corresponding features equally. This highlights the need for a FD idea that balances stability with plasticity. (b) Heatmap comparing class-wise deviations in Grad-CAM feature importances of various models from those of the Oracle model O. O is trained like any other CIL model except that it has access to all past and current train data. S^2IL shows significantly lower deviation from O, suggesting that accounting for feature structure results in better generalization.

To overcome these limitations, we propose the following contributions:

- We interpret the **Structural Similarity Index Measure (SSIM)** [17] in the CIL setting.
- A FD approach called Structurally Stable Incremental Learning (S^2IL) that preserves **structural similarity** of features across incremental steps instead of enforcing exact feature alignment, which encourages the model to retain the overall spatial patterns within features to promote flexible yet stable representations.
- The efficacy of S^2IL is demonstrated on benchmark datasets.

II. RELATED WORK

A. CLASS INCREMENTAL LEARNING

The CIL problem has been approached from various perspectives, including regularization, memory replay, network expansion, knowledge distillation, and feature distillation, either as stand-alone techniques or in combination. Regularization methods [12], [18], [19] estimate the importance of each model parameter, allowing less critical parameters to adapt as new classes are introduced, thereby supporting the model's flexibility.

These methods typically report lower performance compared to other approaches [20], [21]. Memory replay methods [5], [22], [23], [24], [25], [26], [27] maintain a small set of data samples (called exemplars) from previously seen classes, which are reintroduced alongside new class samples to help the model retain knowledge of earlier tasks.

Network expansion methods [7], [10], [28] increase the network's capacity during incremental tasks, separating plasticity from stability to enhance both. Most methods replicate the base extractor and then employ a pruning strategy to reduce the overall parameter count.

Knowledge distillation methods [5], [11], [23] seek to maintain consistent probability distributions for previously seen classes across incremental tasks.

Feature distillation methods focus on preserving representations across incremental tasks, with our proposed S^2IL falling within this category. We review recent key literature on feature distillation for CIL.

PODNet [15] minimizes the norm of differences between spatially aggregated features across layers, progressively aggregating first along width, then height. In the dense layer, this approach reduces to the method proposed in [6]. AFC [13] enforces feature similarity between consecutive incremental models by minimizing an upper bound on the feature loss difference, where the bound is derived using a first-order approximation. This upper bound involves a weighted norm of the feature differences, with the weights reflecting the importance of the features. EXACFS [16] enhances AFC [13] by computing class-wise feature importance values and exponentially aging these values across incremental tasks.

III. THE CIL FORMULATION

Assume $T + 1$ tasks with task 0 being the base task and the remaining tasks 1 through T arriving incrementally. The dataset for task t, $t \in \{0, 1, \ldots, T\}$, is $D^t = \{(x_i^t, y_i^t)\}_{i=1}^{n^t}$ where x_i^t is the data, y_i^t is the class label and n^t is the number of samples. Each class label y_i^t belongs to C^t where $C^t = \{c_1^t, c_2^t, \ldots, c_{m^t}^t\}$ represents the set of m^t classes associated with task t. For $i \neq j$, we assume $C^i \cap C^j = \emptyset$. Let $D^{0\sim t} = \cup_{i=0}^{t} D^i$ and $C^{0\sim t} = \cup_{i=0}^{t} C^i$. Each task t also receives a small set of exemplars E_i^t from each previously seen class c_i^t. Depending on the context, D^t ($D^{0\sim t}$) may refer to either the training or testing data for task t (for tasks 0 to t).

We assume a fixed convolutional network F appended with a global average pooling layer G and a growing classifier H across tasks. During training on task t, H has m^t nodes added to its existing nodes. The network F consists of L layers, with f_{ij} denoting the j^{th} feature map of layer i. Let i_{fm} represent the number of feature maps in convolutional layer

i. Let $f_{(L+1)}$ denote the output of global average pooling layer G. We denote the overall model by $M = H \circ G \circ F$, where '$\circ$' is the composition operation. At task t, we denote M, F, G and H by M^t, F^t, G^t and H^t, respectively. Similarly, the feature map f_{ij} at task t is denoted by f^t_{ij}. During training of M^t, the parameters of F^t and the parameters of H^t associated with the previously seen classes are initialized with parameters from the corresponding components of M^{t-1}. The new nodes of H^t are initialized using Imprinted Weights [29] as in [15]. For $t = 0$, the entire M^0 is initialized randomly.

IV. THE PROPOSED S^2IL

A. MOTIVATION

CIL can be understood like renovating a building: while the overall framework (structure) must remain stable so that the building stands strong, there should also be flexibility to remodel rooms and hallways to suit new requirements. In deep learning, this translates to maintaining the essential spatial relationships within a model's internal features for previous tasks (the structure), while allowing updates that accommodate new information (remodelling). Our motivation is to relax strict preservation of exact feature values and instead focus on safeguarding these underlying structural patterns, balancing memory of past knowledge and adaptation to new challenges.

Current SOTA FD approaches [6], [13], [15], [16], in a generic sense, minimize one of the following objectives specific to feature preservation during the training of task t.

$$\sum_{i=1}^{L}\sum_{j=1}^{i_{fm}} ||f^t_{ij}(x) - f^{t-1}_{ij}(x)||^2 + ||f^t_{L+1}(x) - f^{t-1}_{L+1}(x)||^2 \tag{1}$$

$$\sum_{i=1}^{L}\sum_{j=1}^{i_{fm}} \rho^t_{ij}||f^t_{ij}(x) - f^{t-1}_{ij}(x)||^2 + \rho^t_{L+1}||f^t_{L+1}(x) - f^{t-1}_{L+1}(x)||^2 \tag{2}$$

where x is the input sample, $||.||$ is Frobenius norm and ρ^t_{ij} is the importance of feature map j in layer i of F^t during training of task t, which helps preserve the learned model from task $t-1$. ρ^t_{L+1} is similarly defined for features from G^t. Reference [6] considers only the second term in Eqn. (1) with normalized f^t_{L+1}, which simplifies the minimization to the maximization of $< f^t_{L+1}, f^{t-1}_{L+1} >$, where $< ., . >$ denotes the standard inner product in real space. Reference [15] uses the term corresponding to layer $L+1$ as is, but for the other layers, it considers the squared norm of the difference between the corresponding spatially aggregated features (aggregated separately along width and height) instead of direct comparison. References [13] and [16] follow the formulation in Eqn. (2), with [16] estimating the importance for each layer, feature, and class, while [13] estimating the importance only for each layer and feature.

Surprisingly, the average importance factor ρ almost flattens to 1 in both [13] and [16] across the layers. This is illustrated in Figure 1(a) for the last convolutional layer, which depicts the average ρ value for each feature map, averaged across all incremental tasks on the CIFAR-100 dataset where the base task consists of half the total number of classes, with 10 new classes added per incremental task. The flattening to 1 was observed in other incremental settings also, as detailed in the supplementary material. In our experiments, the difference in average incremental accuracy (AIA) between setting ρ to 1 and explicitly estimating ρ is as small as 0.4% in both [13] and [16]. In other words, both [13] and [16] effectively minimize the squared-norm of the difference between corresponding feature maps, as in Eqn. (1), with respect to feature alignment.

A key issue with minimizing terms like $||u - v||^2$ in Eqn. (1) is that it either aligns the direction of u and v when their sizes are constrained (e.g., normalized) or aligns both their magnitude and direction, as $||u - v||^2 = ||u||^2 + ||v||^2 - 2 < u, v >$. Such alignment aims to make the feature representations for a given input nearly identical across tasks, preserving both the direction and magnitude of features in the feature space. However, this strict alignment can hinder the model's plasticity by forcing it to retain exact feature representations, which reduces its capacity to adapt to new classes and patterns. While we present the reasoning for the limitation here, [8] experimentally validates the lack of plasticity in the approaches of [6], [13], and [15].

The aforementioned limitation motivated us to frame FD as a means of preserving structural similarity between features across tasks. By preserving structure, we aim to maintain the spatial arrangement of features. This allows for more flexibility in terms of feature magnitudes and directions, providing the model room to adjust these aspects to new data, while still maintaining an overall structural similarity. This is akin to preserving the 'shape' rather than exact 'positions' in the feature space. To enforce structural similarity, we leverage SSIM [17], and refer to our proposed method as Structurally Stable Incremental Learning (S^2IL). Note that by preserving structure through SSIM, class-semantic consistency is maintained as demonstrated in Wang et al. [30] and Venkatesh et al. [31].

B. THE S^2IL LOSS

First, we adopt SSIM [17] for a pair of corresponding feature maps u and v between previous and current tasks respectively, and present our interpretations. SSIM between u and v is defined as:

$$\text{SSIM}(u, v) = l(u, v)^p \cdot c(u, v)^q \cdot s(u, v)^r \tag{3}$$

where

$$l(u, v) = \frac{2\mu_u\mu_v + C_1}{\mu_u^2 + \mu_v^2 + C_1} \quad c(u, v) = \frac{2\sigma_u\sigma_v + C_2}{\sigma_u^2 + \sigma_v^2 + C_2}$$

$$s(u, v) = \frac{\sigma_{uv} + C_3}{\sigma_u\sigma_v + C_3}$$

and p, q, r, C_1, C_2, C_3 are positive reals.

μ_u (μ_v) is the mean activation level of feature map u (v) across spatial dimensions. σ_u^2 (σ_v^2) quantifies the variance of feature activations around μ_u (μ_v). σ_{uv} quantifies the linear relationship between the activations of u and v, essentially capturing how well the structural patterns of u align with those of v. $l(u, v)$, the luminance component in SSIM, measures the similarity in the mean activations between the two feature maps across tasks. It reflects how well the average activations align spatially, representing their global similarity in terms of the feature values. $c(u, v)$, the contrast component, measures the similarity in the activation distributions between the two feature maps across the tasks. $s(u, v)$, the structure component, measures the similarity in the spatial relationships between the two feature maps across tasks. This captures how well the spatial patterns in u and v align, independent of their magnitudes.

l and c help maintain the global features and the distribution of features across tasks. This means that the model keeps a consistent representation of the features it has learned, even as it adapts to new tasks, without overfitting to old tasks. s ensures that as the model learns new tasks, the spatial structure of the features it learned from previous tasks is retained. This allows the model to reorganize and adjust its internal representations to incorporate new classes or tasks, without completely distorting the learned patterns from old tasks. Given this adoption of SSIM to CIL setting, we define the FD loss during training of task t as:

$$\mathcal{L}^t_{S^2IL} = \frac{1}{|B|}\sum_{x\in B}\sum_{j=1}^{L_{fm}} \frac{1-\text{SSIM}(f^t_{Lj}(x), f^{t-1}_{Lj}(x))}{2} \tag{4}$$

where B is the mini-batch and the inner term measures the dissimilarity between the feature maps. The inner term lies in the range $[0, 1]$ since SSIM lies in the range $[-1, 1]$. While the inner term is not a pure metric, it is a pseudo-metric. SSIM-based pseudo-metrics have been shown to be effective for clustering applications in [32], demonstrating that strict metric properties are not required for practical convergence and meaningful optimization.

Unlike [13], [15], and [16], we apply FD only on the last convolutional layer as it contains rich semantic and structural information. Additionally, applying distillation across all layers could hinder the model's plasticity, as discussed in [8] and validated by additional results presented in section V-D. The overall training of task t is governed by the loss:

$$\mathcal{L}^t = \mathcal{L}^t_{cls} + \lambda\mathcal{L}^t_{S^2IL} \tag{5}$$

where $\mathcal{L}_{cls}$ is the local similarity classifier loss adopted from [15] and λ is a hyperparameter adopted from [6] and [13] that measures the degree of need to preserve old knowledge, increasing as the ratio of new to old classes grows. The training algorithm is demonstrated in Algorithm 1.

Algorithm 1 Training at Task t

```
Input:
    $D^t$: Training data for task $t$
    $E^{0\sim t-1}$: Set of exemplars from tasks $\{0, \dots, t-1\}$
    $p, q, r$: SSIM params set to 0.1, 8, and 8, respectively
    $M^{t-1}$: Trained model from task $t-1$
    $\lambda$: Weight for $S^2IL$ loss $\mathcal{L}_{S^2IL}$
Output: Trained model $M^t$

Initialization: $F^t \leftarrow F^{t-1}$
for $(x, y) \in D^t \cup E^{0\sim t-1}$ do
    Compute $\mathcal{L}_{S^2IL}(x)$ and $\mathcal{L}_{cls}(x)$
    $\mathcal{L}(x) = \mathcal{L}_{cls}(x) + \lambda\mathcal{L}_{S^2IL}(x)$
    Backpropagate and update model parameters
end for
```

C. PROOF OF CONCEPT

As an empirical proof of concept for the efficacy of S^2IL, we conducted the following experiment. We defined an Oracle model O, which is incrementally trained like M, except that during task t, O has access to the entire training data of all previous tasks, rather than just exemplars, and does not use any distillation loss. We define the deviation measure of M from O for each class $l \in C^0$ as follows:

$$D_l(M, O) = \frac{1}{L_{fm}}\sum_{j=1}^{L_{fm}}\left(1-\frac{\alpha^{M^T}_{Lj,l}-\alpha^{M^0}_{Lj,l}}{\alpha^{O^T}_{Lj,l}-\alpha^{O^0}_{Lj,l}}\right)^2 \tag{6}$$

where $\alpha^{M^t}_{Lj,l}$ ($\alpha^{O^t}_{Lj,l}$) is the mean Grad-CAM importance [33] of the j^{th} feature map from the last convolutional layer of model $M(O)$ at task t, with the mean computed over the samples from class $l \in C^0$. This captures the influence of the j^{th} feature map on the prediction of class l. Note that Grad-CAM importance is computed post-training. The fractional term in Eqn. (6) measures the evolution of the j^{th} feature map of M relative to O across all the tasks. If M behaves similarly to O, the fractional term will be close to 1, resulting in a deviation score near 0, indicating that M maintains stability and plasticity like O. The heatmap in Figure 1(b) shows the deviations of S^2IL and other models including LUCIR [6], PODNet [15], AFC [13] and EXACFS [16], for a sample of base classes from CIFAR-100 under the same incremental setting as in Figure 1(a). Clearly, S^2IL exhibits the lowest deviations, demonstrating its superior stability and plasticity. Additional visual plots corresponding to other base classes are provided in the supplementary material.

V. RESULTS AND DISCUSSIONS

A. DATASET DETAILS

The SOTA benchmark datasets used to assess the performance of the proposed S^2IL method are CIFAR-100 [36], ImageNet-1K [37] and a subset version of ImageNet-1K named ImageNet-100 [5] which has 100 randomly selected classes from Imagenet-1K. The CIFAR-100 dataset contains 60,000 color images across 100 classes, each with a resolution of 32 × 32. It is divided into 50,000 training images and 10,000 test images. ImageNet-100 [5] is a subset of the original ImageNet-1K, consisting of 100 randomly selected classes, each containing approximately 1,300 color images. The sampling method for ImageNet-100 is detailed

TABLE 1. Performance Comparison using AIA (%) on CIFAR-100 and ImageNet-100. CIFAR-100 results are averaged over 3 runs. EXACFS is implemented by us. AANet, eTag, and MTD results are from [7], [11], and [34]. Results for other methods are based on code from [13] and [15].

Methods	CIFAR-100				ImageNet-100			
	Inc 1	**Inc 2**	**Inc 5**	**Inc 10**	**Inc 1**	**Inc 2**	**Inc 5**	**Inc 10**
iCARL [5] (2017)	44.20±0.98	50.60±1.06	53.78±1.16	58.08±0.59	54.97	54.56	60.90	65.56
LUCIR [6] (2019)	49.30±0.32	57.57±0.23	61.22±0.69	64.01±0.91	57.25	62.94	70.71	71.04
BiC [35](2019)	47.09±1.48	48.96±1.03	53.21±1.01	56.86±0.46	46.49	59.65	65.14	68.97
PODNet [15] (2020)	57.86±0.38	60.51±0.62	62.78±0.78	64.62±0.65	62.48	68.31	74.33	75.54
PODNet + AANet [7] (2021)	-	62.31±1.02	64.31±0.90	66.31±0.87	-	71.78	75.58	76.96
AFC [13] (2022)	61.39±0.86	63.85±0.3	64.53±0.56	65.89±0.85	72.08	73.34	75.75	76.87
EXACFS [16] (2024)	61.1±0.75	62.75±0.8	64.05±1.05	65.48±1.08	72.5	73.78	74.96	75.93
eTag [34] (2024)	-	61.63±0.79	65.50	**67.99**	-	71.77	75.17	76.79
MTD [11] (2024)	60.0±1.20	62.46±0.21	65.39±0.81	66.96±0.56	70.8	73.73	**76.26**	**77.82**
$S^2IL(\mathbf{O}urs)$	**62.94±1.36**	**64.23±1.24**	**65.88±0.8**	67.35±1.15	**73.15**	**74.27**	75.63	76.52

TABLE 2. Performance comparison using AIA (%) for AFC, MTD, and S2IL on ImageNet-1K.

Methods	**Inc 20**	**Inc 50**	**Inc 100**
AFC [13] (2022)	-	67.02	68.9
MTD [11] (2024)	64.11	**68.15**	**70.4**
$S^2IL(\mathbf{O}urs)$	**64.84**	67.23	68.9

in [6] and [15]. ImageNet-1K comprises over 1.2 million labelled training images across 1,000 object categories, with an additional 50,000 validation images. It serves as a key benchmark for evaluating machine learning models in image classification tasks. Image preprocessing and class order settings follow the methodology outlined in [15].

B. EXPERIMENTAL SETTINGS, HYPERPARAMETERS AND RESOURCE USAGE

We use ResNet-32 [15] for CIFAR-100 and ResNet-18 [15] for both ImageNet-100 and ImageNet-1K. For training, the base task (Task 0) uses half the number of total classes with the subsequent tasks taking increments of 1, 2, 5, or 10 new classes for CIFAR-100 and ImageNet-100, and 50 or 100 new classes for ImageNet-1K. The exemplar memory budget is fixed at 2000 for CIFAR-100 and ImageNet-100, and at 20000 for ImageNet-1K, with exemplar selection based on the herding technique [5]. p, q and r for SSIM are set at 0.1, 8.0 and 8.0.

The model is trained using the SGD optimizer with a momentum of 0.9. For CIFAR-100, training is conducted over 160 epochs with a batch size of 128, a weight decay of 0.0005, and an initial learning rate of 0.1, which is decayed using a *CosineAnnealingScheduler*. For ImageNet-100 and ImageNet-1K, training spans 90 epochs with a batch size of 64, a weight decay of 0.0001, and an initial learning rate of 0.05, also decayed using the *CosineAnnealingScheduler*. The loss function employed is the local similarity classifier loss [15], featuring a margin of 0.6, a learnable scale factor initialized to 1.0, and 10 proxies per class. The regularization coefficient (λ) in Equation (5) is configured as $4 \times \sqrt{(|C^{0 \sim t}|/|C^t|)}$ for CIFAR-100 and $10 \times \sqrt{(|C^{0 \sim t}|/|C^t|)}$ for ImageNet datasets. Following the training for each task, the model undergoes fine-tuning for 20 epochs on a balanced dataset encompassing samples from all seen classes, with learning rates of 0.05 for CIFAR-100 and 0.02 for both ImageNet-100 and ImageNet-1K.

All experiments are conducted on an NVIDIA RTX 4090 GPU (24 GB memory). For CIFAR-100 under the **Inc 10** setting, S^2IL required 2 h 22 min of training time and 1.23 GB of GPU memory, while the **Inc 1** setting extended to ≈ 12 h with a similar memory footprint (≈ 1.2 GB). The inference time per image was ≈ 0.2ms, and the computational complexity was 69.65 MMAC (Mega Multiply–Accumulate operations). For ImageNet-100, the **Inc 10** setting required 2 days 19 h 13 min of training with ≈ 8.5 GB of GPU memory, while the **Inc 1** setting took ≈ 8.5 days with comparable memory usage. The inference time per image is ≈ 0.3ms, and the final trained model required 6.91 GMAC (Giga Multiply–Accumulate operations). These results show that S^2IL introduces only moderate memory and computational overhead. The relatively low MAC (Multiply–Accumulate operations) count and inference latency ($\leq$ 0.3ms per image) indicate that the final S^2IL model for CIFAR-100 is lightweight enough for deployment on mobile or edge devices, whereas the ImageNet-100 variant is better suited for high-end embedded or desktop GPUs. Although the training phase demands moderate computational resources, the trained S^2IL model remains efficient and practical for real-time or resource-constrained inference environments.

C. COMPARING S^2IL WITH SOTA METHODS

The performance metrics used in this study are AIA [5], Backward Transfer (BT) [10] and Forgetting metric (Fgt). Forward transfer (FT) and overall accuracy i.e the accuracy of the final model are also considered. For comparative analysis, we consider memory replay, knowledge distillation, and network expansion methods, evaluated under identical conditions. Table 1 shows the results. In CIFAR-100, S^2IL outperforms others, especially in the challenging **Inc 1** and **Inc 2** settings, by around 1.5%. S^2IL also performs best in the difficult **Inc 1** and **Inc 2** settings of ImageNet-100. While other distillation strategies in difficult settings like **Inc 1** prioritize stability by aligning features in magnitude and direction, they sacrifice plasticity. In contrast, S^2IL maintains both stability and plasticity, excelling in such settings. The AIA of S^2IL on ImageNet-1K is presented in the Table 2,

TABLE 3. Backward transfer (higher is better) and Forgetting metric (lower is better) values on the CIFAR-100 dataset with Inc 10 setting.

	iCARL	LUCIR	BiC	PODNet	AFC	EXACFS	MTD	S^2IL
BT (%)	-24.14	-11.25	-6.73	-7.98	-9.0	-9.02	-6.46	**-5.3**
Fgt (%)	17.5	9.1	30.9	11.9	7.5	9.01	10.9	**7.3**
FT (%)	-4.32	-8.19	-8.73	**-1.24**	-2.58	-2.71	-7.97	-3.81
Overall Acc. (%)	46.5	54.2	39.1	55.0	56.8	55.8	**59.3**	58.0

TABLE 4. AIA (in %) of S^2IL for different values of p, q and r under Inc 10 setting in CIFAR-100.

Param	**0.1**	**0.2**	**0.4**	**4**	**8**	**16**
p=0, q=0, r	60.22	62.01	64.08	67.17	67.18	56.7
p=0, q, r=8	67.05	67.18	67.31	67.23	66.98	67.05
p, q=8, r=8	66.92	66.97	66.68	53.78	53.9	53.63

in which we notice that S^2IL performs better in the difficult INC 20 setting while MTD is does well in INC 50 and 100. Results for other methods on ImageNet-1K are presented in supplementary material. The reason for S^2IL's better AIA in low increment settings and MTD's better AIA in large increment settings can be attributed to the multiple teachers in MTD that are forced to be diverse in their correct classifications, which is feasible with many classes per increment, boosting performance in large increment settings. However, with low increment settings, the room for diversity among teachers drops significantly. With no further FD in MTD, it forgets more unlike S^2IL. This is further highlighted by better BT and Fgt percentages for S^2IL in comparison to MTD and other methods as presented in Table 3. S^2IL also achieves competitive FT while attaining one of the highest overall accuracies (58%) among all compared methods as shown in Table 3. This further supports the effectiveness of S^2IL in learning new tasks without compromising its ability to generalize to future ones, thereby complementing its already strong performance in Backward Transfer (BT) and Forgetting (Fgt).

TABLE 5. Ablation study on components of SSIM on the CIFAR-100 Inc 10 setting, averaged over 3 runs).

Our Model with	**AIA (%)**	**p-val**	**Effect size**
only l	60.82±0.65	0.0005	Large
only c	62.52±1.02	0.0028	Large
only s	67.65±0.97	0.6264	Small
l+c	64.76±0.74	0.0152	Large
l+s	67.18±1.18	0.4334	Small
c+s	67.59±1.17	0.5938	Small
l+c+s	67.35±1.15	-	-

D. ABLATION STUDIES

1) SSIM COMPONENTS

In Table 4, we present the analysis of the SSIM hyperparameters p, q, and r under the **Inc 10** setting on CIFAR-100. We fix r first, followed by q and p, because the structure component plays a prominent role compared to the contrast and luminance components, as elucidated by the subsequent ablation study. We first varied the structural weighting factor r over a logarithmic range [0.1, 16] while keeping $p = 0$ and $q = 0$. Performance improves steadily up to $r = 8$ and then drops beyond this point, indicating that moderate structural emphasis enhances knowledge preservation, whereas excessive importance may over-regularize the model. Next, we fixed $r = 8$ and varied q to study the effect of the contrast term, which yielded marginal improvements (approximately $+0.2\%$), showing that contrast contributes modestly compared to the structural component. Lastly, fixing $q = 8$ and $r = 8$, we varied the luminance weight p across the same range. The accuracy remained robust for smaller p values but degraded at higher luminance weightings. Overall, the model remains robust for $r \in [0.4, 8]$, and the configuration $(p, q, r) = (0.1, 8, 8)$ achieves a favourable balance between stability and plasticity across different datasets. Hence, we adopt this setting as the default in all experiments.

TABLE 6. Average incremental accuracy (%) of S^2IL with only structure component on ImageNet-100.

Incremental Setting	**only s**	$l+c+s$
Inc 1	70.7	73.15
Inc 2	71.7	74.27
Inc 5	75.54	75.63
Inc 10	76.18	76.52

TABLE 7. Average incremental accuracy (%) of S^2IL applied to all layers vs. last layer.

Incremental Setting	**All Layers**	**Last Layer**
Inc 1	57.69	62.93
Inc 2	59.72	64.04
Inc 5	63.02	65.56
Inc 10	65.05	66.74

Additionally, we investigated the impact of the SSIM components l, c, and s on the model performance. As shown in Table 5, the absence of the structure component leads to a sharp performance drop, highlighting its importance. Specifically, based on the one-tailed tests ($n = 3$, $\alpha = 0.05$, Cohen's d effect size measure) comparing each SSIM component configuration with the full S^2IL model that has all the three components, we observed that "only s" (structure-only) configuration achieves performance statistically equivalent to the full model ($p = 0.6264$), confirming that the structure component is the primary contributor to S^2IL's effectiveness. Removing either luminance (l) or contrast (c) components results in statistically significant performance drops with large effect sizes, demonstrating their important supporting roles. While structure alone is sufficient, the combination of all three components ($l + c + s$) provides the most stable performance across different experimental conditions. In fact,

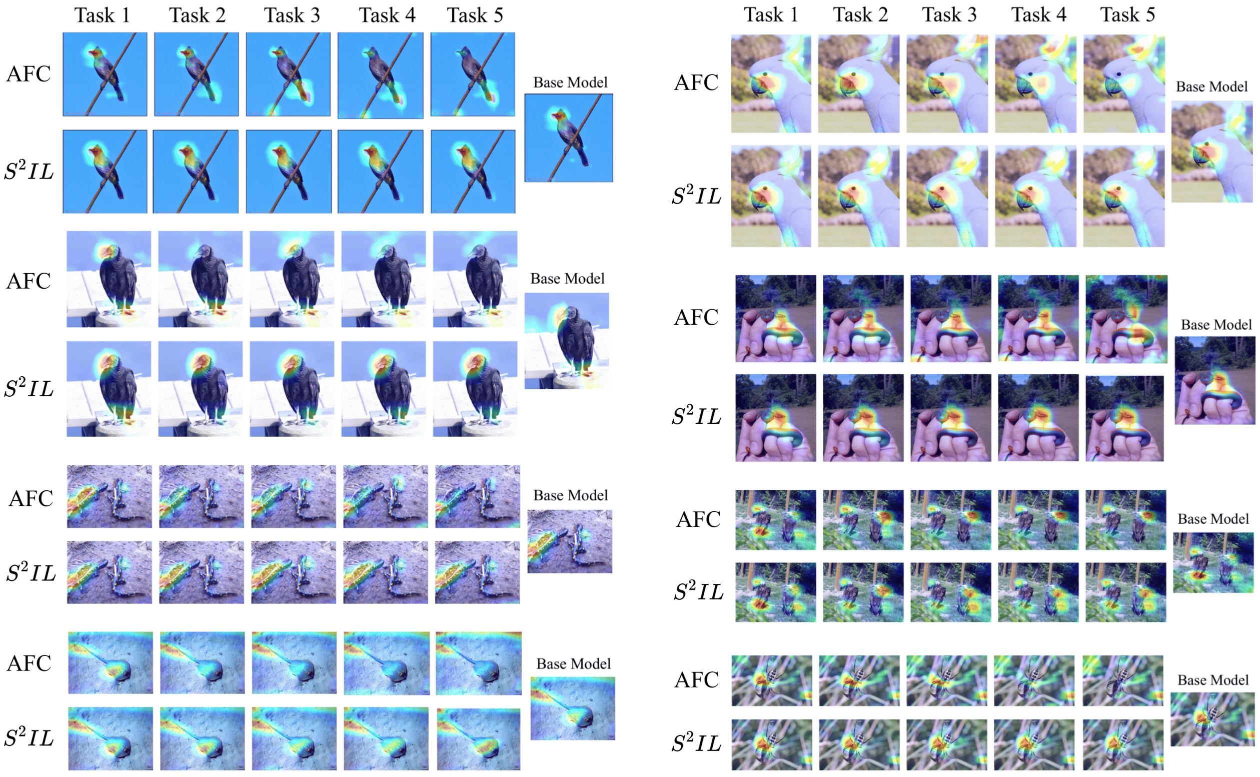


FIGURE 2. Grad-CAM attention maps for class-incremental learning on the ImageNet-100 *Inc*10 setting, shown for a subset of base task classes across incremental tasks. The rightmost column shows the attention map of the common base model. S^2IL maintains compact, class-relevant activations that closely match the base model, while AFC exhibits drifting heatmaps.

incorporating all three components (l, c, and s) provided better results on ImageNet-100 as shown in Table 6. Using only the structure component results in an AIA .3% to 2.5% lower than when all components are included. This guides us to the choice of applying all three components of SSIM with the values 0.1, 8.0 and 8.0 for the exponent components p, q and r respectively in the SSIM formula.

TABLE 8. Exemplar memory type study.

Type	Inc 1	Inc 2	Inc 5	Inc 10
M1 (CIFAR-100)	62.93	64.04	65.56	66.74
M1 (ImageNet-100)	72.9	74.25	75.55	76.05
M2 (CIFAR-100)	62.94	64.23	65.88	67.35
M2 (ImageNet-100)	73.15	74.27	75.63	76.52

2) ALL LAYERS VERSUS LAST LAYER

The Average Incremental Accuracy of S^2IL, when applied on all convolution layers and when applied on last convolution layer is presented in Table 7 for CIFAR-100 dataset. The significant performance gap demonstrates that applying distillation across all layers may impair the model's plasticity.

3) MEMORY ALLOCATION STRATEGIES

We explored two memory allocation strategies: (1) a fixed budget of 20 exemplars per class, denoted as M1, and (2) a fixed overall memory budget of 2000 exemplars, equally distributed across the previously seen classes during incremental training, denoted as M2. Table 8 presents the results, which show a clear advantage of M2 over M1. Therefore, we adopt the M2 strategy for our model.

TABLE 9. Average incremental accuracy (%) of S^2IL with MS-SSIM vs. SSIM on different datasets.

Dataset / Increment	MS-SSIM based	Proposed
CIFAR-100 / Inc 10	59.07	67.35
CIFAR-100 / Inc 1	53.50	62.94
ImageNet-100 / Inc 10	74.58	76.52
ImageNet-100 / Inc 1	70.72	73.15

4) COMPARISON WITH MS-SSIM BASED S^2IL

Wang et al. [38] introduced the multi-scale SSIM (MS-SSIM) index, an extension of the original SSIM, which incorporates multiple scales to better model the human visual system's sensitivity to structural information across different resolutions. We experimented with MS-SSIM by replacing the SSIM in Equation 4 with MS-SSIM. Table 9 presents the results. S^2IL with SSIM consistently outperforms its MS-SSIM counterpart. One key reason for MS-SSIM's inferior performance on CIFAR-100 is the dataset's low image resolution (32 × 32), which limits the benefits of its multi-scale design. This limitation is less pronounced in ImageNet-100, where images have a higher average resolution (approximately 400 × 300 pixels), leading to a noticeable performance improvement. Nonetheless, even on ImageNet-100, SSIM-based S^2IL surpasses MS-SSIM-based S^2IL by around 3.5%. This is likely because our method computes similarity using features from the final convolutional layer, where spatial resolution is already heavily reduced, diminishing the advantages of MS-SSIM's multi-scale processing.

5) STABILITY VIA GRAD-CAM HEATMAPS

Figure 2 contrasts AFC [13] and the proposed S^2IL Grad-CAM heatmaps generated on samples from the classes introduced in the base task across tasks in the $Inc10$ setting for ImageNet-100. S^2IL's activation maps remain concentrated on class-relevant structures with minimal dispersion across tasks, closely matching the base model's localization patterns. AFC progressively shifts or diffuses attention, indicating loss of stability in the long run. For example, in the heatmaps for the Bulbul and the Vulture samples (left column, rows 1–2), S^2IL consistently highlights the head–neck region and perched body across all tasks, replicating the base model's Grad-CAM. However, AFC's heatmaps drift toward other regions in the image. A similar trend of dispersion can be observed in the other samples for AFC while S^2IL continues to focus on the discriminative region learnt during it's training. This implies that S^2IL's representation learning based on structure preservation remains stable as new tasks arrive, yielding consistent attribution maps.

VI. CONCLUSION

FD methods in the literature force equality in the direction and magnitude among feature maps which impair the model's plasticity. The proposed method, S^2IL, effectively balances the stability-plasticity dilemma by enforcing structural similarity between feature maps across incremental tasks through the incorporation of SSIM. We also validated our method by testing it on SOTA benchmark datasets over the AIA, BT and Fgt metrics and find that S^2IL delivers strong and comparable performance without enforcing direct feature preservation across tasks.

Computing SSIM requires either the forward propagation of exemplars through a stored copy of the old model, or retaining feature maps of all exemplars, leading to an additional memory overhead. This can pose challenges for implementation in memory-constrained settings. However, the core mechanism of structural similarity preservation is model-agnostic and does not inherently rely on any stored data. Scalability to data-free CIL can therefore be achieved by replacing exemplar-driven SSIM computation with generative or pseudo-feature replay strategies. For example, storing feature statistics (e.g., mean and variance of feature maps) from previous tasks would allow S^2IL to estimate SSIM structurally without direct access to data. More importantly, owing to its model-agnostic nature, S^2IL can act as a direct substitute for the feature distillation losses proposed in recent work such as [39], which achieve scalability by discarding all forms of external data, including exemplars and prototypes. Furthermore, since S^2IL performs feature distillation only at the final convolutional layer, its computational footprint is scalable for scenarios like CIFAR-100–like image resolutions. For higher-resolution datasets such as ImageNet, extending this efficiency is a promising direction that will be explored in future work.

REFERENCES

[1] R. Gupta, A. Roy, C. Christensen, S. Kim, S. Gerard, M. Cincebeaux, A. Divakaran, T. Grindal, and M. Shah, "Class prototypes based contrastive learning for classifying multi-label and fine-grained educational videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 19923–19933.

[2] J. Park, M. Kang, and B. Han, "Class-incremental learning for action recognition in videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13678–13687.

[3] I. Kuzborskij, F. Orabona, and B. Caputo, "From n to N+1: Multiclass transfer incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3358–3365.

[4] Z. Li and D. Hoiem, "Learning without forgetting," in *Proc. ECCV*, vol. 40, 2017, pp. 2935–2947.

[5] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[6] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.

[7] Y. Liu, B. Schiele, and Q. Sun, "Adaptive aggregation networks for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2544–2553.

[8] D. Kim and B. Han, "On the stability-plasticity dilemma of class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 20196–20204.

[9] J. He, "Gradient reweighting: Towards imbalanced class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 16668–16677.

[10] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3013–3022.

[11] H. Wen, L. Pan, Y. Dai, H. Qiu, L. Wang, Q. Wu, and H. Li, "Class incremental learning with multi-teacher distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28443–28452.

[12] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. ICML*, 2017.

[13] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16050–16059.

[14] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. van de Weijer, "Class-incremental learning: Survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, May 2023.

[15] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PODNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. ECCV*, 2020, pp. 86–102.

[16] S. Balasubramanian, M. S. Subramaniam, S. S. Talasu, M. P. P. Sai, D. Gera, and R. Mukkamala, "EXACFS–A CIL method to mitigate catastrophic forgetting," in *Proc. 15th Indian Conf. Comput. Vis. Graph. Image Process.*, India, Dec. 2024, pp. 1–8.

[17] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[18] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[19] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. ICCV*, 2018, pp. 144–161.

[20] Y.-C. Hsu, Y. Liu, and Z. Kira, "Re-evaluating continual learning scenarios: A categorization and case for strong baselines," in *Proc. NeurIPS Continual Learn. workshop*, 2018.

[21] G. M. van de Ven and A. S. Tolias, "Three scenarios for continual learning," *Nature Mach. Intell.*, vol. 4, pp. 1185–1197, Jan. 2022.

[22] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, and G. Wayne, "Experience replay for continual learning," in *Proc. NeurIPS*, 2018.

[23] H. Ahn, J. Kwak, S. Lim, H. Bang, H. Kim, and T. Moon, "SS-IL: Separated softmax for incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 824–833.

[24] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. NeurIPS*, 2017.

[25] R. Aljundi, M. Lin, and B. Goujaud, "Gradient based sample selection for online continual learning," in *Proc. NeurIPS*, 2019.

[26] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," in *Proc. NeurIPS*, vol. 30, 2017, pp. 6467–6476.

[27] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. ICLR*, 2018.

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

[29] H. Qi, M. Brown, and D. G. Lowe, "Low-shot learning with imprinted weights," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5822–5830.

[30] Z. Wang, E. Yang, L. Shen, and H. Huang, "A comprehensive survey of forgetting in deep learning beyond continual learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 3, pp. 1464–1483, Mar. 2025.

[31] D. K. Venkatesh, D. Rivoir, M. Pfeiffer, F. Kolbinger, M. Distler, J. Weitz, and S. Speidel, "Exploring semantic consistency in unpaired image translation to generate data for surgical applications," *Int. J. Comput. Assist. Radiol. Surgery*, vol. 19, no. 6, pp. 985–993, Feb. 2024.

[32] K. Winderlich, C. Dalelane, and A. Walter, "Classification of synoptic circulation patterns with a two-stage clustering approach using a modified structural similarity index metric," *Earth Syst. Dyn.*, vol. 15, pp. 607–634, Jun. 2024.

[33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[34] L. Huang, Y. Zeng, C. Yang, Z. An, B. Diao, and Y. Xu, "ETag: Class-incremental learning via embedding distillation and task-oriented generation," in *Proc. AAAI*, 2024, vol. 38, no. 11, pp. 12591–12599.

[35] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–374.

[36] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[38] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc.37th Asilomar Conf. Signals, Syst. Comput.*, 2003.

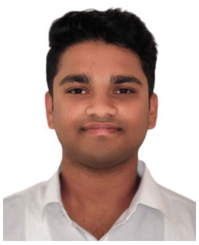
[39] M. M. Hasan, S. M. Sami, and N. Nasrabadi, "CLFace: A scalable and resource-efficient continual learning framework for lifelong face recognition," 2024, *arXiv:2411.13886*.

S. BALASUBRAMANIAN is currently an Associate Professor with the Department of Mathematics and Computer Science, Sri Sathya Sai Institute of Higher Learning. He has more than 50 publications in both mathematics and computer science, including articles in top avenues like ICLR, WACV, and Annali di Matematica Pura ed Applicata. His research interests include deep learning, mathematics for machine learning, and differential geometry.

P. YEDU KRISHNA received the bachelor's and master's degrees in mathematics, with a specialization in computer science from the Sri Sathya Sai Institute of Higher Learning. He is currently pursuing the Ph.D. degree with Indian Institute of Technology, Hyderabad (IITH). His primary research interests include deep learning for computer vision and continual learning.

TALASU SAI SRIRAM received the bachelor's and master's degrees in mathematics and the Master of Technology degree in computer science from the Sri Sathya Sai Institute of Higher Learning. His primary research interest includes deep learning for computer vision.

M. SAI SUBRAMANIAM received the bachelor's and master's degrees in mathematics and the Master of Technology degree in computer science from the Sri Sathya Sai Institute of Higher Learning. His primary research interest includes deep learning for computer vision.



MANEPALLI PRANAV PHANINDRA SAI received the bachelor's and master's degrees in mathematics and the Master of Technology degree in computer science from the Sri Sathya Sai Institute of Higher Learning. His primary research interest includes deep learning for computer vision.

RAVI MUKKAMALA (Member, IEEE) received the Ph.D. degree from The University of Iowa, Iowa City, IA, USA, in 1987, and the M.B.A. degree from Old Dominion University (ODU), Norfolk, VA, USA, in 1993. In 1987, he joined ODU as an Assistant Professor, where he is currently a Professor of computer science and an Associate Dean with the College of Sciences. He has published more than 175 research papers in refereed journals and conference proceedings. He has received more than $3 million in research grants as a PI or Co-PI from agencies, including NASA, Jefferson Laboratory, and private industries. His current research interests include computer security, privacy, data mining, and modeling. He received the Most Inspirational Faculty Award from ODU, in 1994. He has won several best paper awards at national and international conferences over the years learning for computer vision.