Task-Adaptive Saliency Guidance for Exemplar-free Class Incremental Learning

Xialei Liu [2,1,*] Jiang-Tian Zhai [1, *] Andrew D. Bagdanov [3] Ke Li [4] Ming-Ming Cheng [2,1, †]
[1] VCIP, CS, Nankai University [2]NKIARI, Shenzhen Futian [3] MICC, University of Florence [4] Tencent Youtu Lab
{xialei,cmm}@nankai.edu.cn, {jtzhai30,tristanli.sh}@gmail.com, andrew.bagdanov@unifi.it

Abstract

Exemplar-free Class Incremental Learning (EFCIL) aims to sequentially learn tasks with access only to data from the current one. EFCIL is of interest because it mitigates concerns about privacy and long-term storage of data, while at the same time alleviating the problem of catastrophic forgetting in incremental learning. In this work, we introduce task-adaptive saliency for EFCIL and propose a new framework, which we call Task-Adaptive Saliency Supervision (TASS), for mitigating the negative effects of saliency drift between different tasks. We first apply boundary-guided saliency to maintain task adaptivity and plasticity on model attention. Besides, we introduce task-agnostic low-level signals as auxiliary supervision to increase the stability of model attention. Finally, we introduce a module for injecting and recovering saliency noise to increase the robustness of saliency preservation. Our experiments demonstrate that our method can better preserve saliency maps across tasks and achieve state-of-the-art results on the CIFAR-100, Tiny-ImageNet, and ImageNet-Subset EFCIL benchmarks. Code is available at https://github.com/scok30/tass.

1. Introduction

Deep neural networks achieve state-of-the-art performance on many computer vision tasks. However, most of these tasks consider a static world in which tasks are well-defined, stationary, and all training data is available in a single training session. The real world consists of dynamically changing environments and data distributions, which – especially given the computational burden of training large CNNs – has led to renewed interest in learning new tasks incrementally while avoiding catastrophic forgetting [14, 37].

Class Incremental Learning (CIL) [1, 36] is a scenario that considers the possibility of adding new classes to already-trained models. Most CIL methods rely on a memory buffer to store exemplars from past tasks [2, 10, 40, 45].

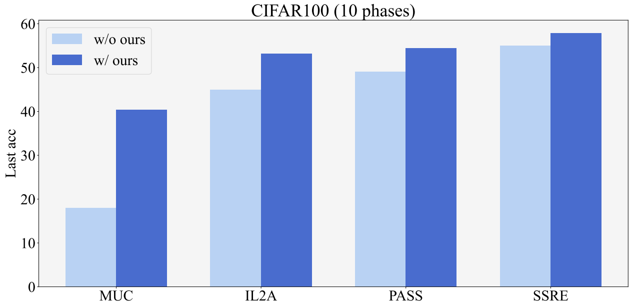

Figure 1. We propose the TASS method, which can be directly applied to many recent exemplar-free class incremental learning methods, resulting in a significant improvement in EFCIL classification accuracy and a reduction in catastrophic forgetting.

In this paper, we consider Exemplar-Free Class Incremental Learning (EFCIL), which is a more challenging setting in which *no data from previous tasks is retained.* This is a realistic scenario and of great interest due to privacy concerns or restrictions on the long-term storage of data. The inability to retain examples from past tasks, however, significantly exacerbates the problem of catastrophic forgetting.

There are several recent works that consider the EFCIL problem. DeepInversion [50] inverts trained networks from random noise to generate images as exemplars and mixes them with current task samples for training. SDC [51] updates prototypes of each learned class by hypothesizing that semantic drift of classes from previous tasks can be approximated and estimated using new data. Other previous works propose representation learning methods for overcoming catastrophic forgetting [56, 57]. As pointed out in IL2A [56], learning better representations can reduce representation bias when transferred to new tasks. Incorporating self-supervised learning tasks, such as Barlow Twins [39] and rotation prediction [57], has also been proposed to achieve more stable representations and alleviate forgetting.

CNNs naturally learn to *attend* to features that are discriminative for the tasks they are trained to solve. Catastrophic forgetting also occurs in EFCIL due to the model's attention to *salient* features drifting to features specific to the new task. Standard regularization approaches do little

[*]The first two authors contribute equally.
[†]Corresponding author

to prevent this saliency drift when learning new tasks. One direct method of regularizing saliency is to apply distillation on saliency maps of old samples [11]. However, this is complicated by the inability to save samples from previous tasks in the EFCIL setting. Another method is to apply saliency distillation between current task samples and previous task attention [9]. This method however suffers from the semantic gap between current and old classes when enforcing saliency consistency.

A lack of saliency regularization may lead to attention drifting toward the background in future tasks causing forgetting. Besides, simply applying distillation on attention [9] fails to offer plasticity and is susceptible to attention forgetting, which is a crucial factor of knowledge forgetting. In comparison, our Task-Adaptive Saliency Supervision (TASS) approach aims at keeping saliency focused on incrementally learned tasks (for more details, see Section 4), while maintaining its plasticity and stability. It improves many previous EFCIL methods with higher performance by supervising their attention, as shown in Figure 1.

Specifically, TASS integrates three components to address this issue. Firstly, we apply dilated boundary maps to prevent saliency drift across object boundaries at intermediate layers. Since saliency drift typically occurs across tasks, encouraging the model to focus on significant foreground regions through dilated boundary supervision reduces the likelihood of saliency shifting toward the background, allowing the model to adaptively select attention areas within the foreground that are relevant to the task. Secondly, to simultaneously enhance the stability of the model's attention across tasks, we add a task-agnostic low-level auxiliary supervision task to the class-incremental framework, which is closely related to our core EFCIL task since image classification has been shown to help models localize the most salient areas within an image. Finally, we propose a module to inject saliency noise into certain feature channels and train the network to denoise them, helping the network further resist attention drift across tasks.

The main contributions of this work are: **(i)** We provide new insight into task-adaptive saliency supervision under EFCIL settings; we also show the negative effect of methods with no or trivial saliency supervision, which illustrates the superiority of our method and motivates the need for saliency drift mitigation in EFCIL. **(ii)** We propose the Task-Adaptive Saliency Supervision (TASS) with three components that combine to mitigate the saliency drift problem. **(iii)** We show that TASS can be easily integrated into other state-of-the-art methods, such as MUC [31], IL2A [56], PASS [57], SSRE [58], leading to significant performance gains. **(iv)** Our experiments demonstrate that TASS outperforms all existing EFCIL methods and even several exemplar-based methods on the CIFAR-100, Tiny-ImageNet, and ImageNet-Subset EFCIL benchmarks.

2. Related Work

We first discuss previous work on incremental learning from the recent literature and then describe work on EFCIL.

2.1. Incremental Learning

A variety of methods have been proposed for incremental learning in the past few years [1, 7]. Recent works can be coarsely grouped into three categories: replay-based, regularization-based, and parameter-isolation methods. Replay-based methods mitigate the task-recency bias by retaining training samples from previous tasks [40, 53]. In addition to replaying samples, BiC [45], PODNet [10], iCaRL [40] and LTCIL [30] apply a distillation loss to prevent forgetting and enhance model stability. GEM [33], AGEM [4], and MER [41] exploit past-task exemplars by modifying gradients on current training samples to match old samples. Rehearsal may cause models to overfit to stored samples. Regularization-based approaches such as LwF [24], EWC, R-EWC [18, 29], and DMC [54] offer ways to learn better representations while leaving enough plasticity for adaptation to new tasks. Parameter-isolation methods [35, 47] use models with different computational graphs for each task. With the help of growing models, new model branches mitigate catastrophic forgetting at the cost of more parameters and computational cost. It is also widely studied in other fields, such as semantic segmentation [3, 26, 46, 52] and object detection [13, 32].

As for saliency-guided incremental learning, LwM [9] regularizes the saliency activations of previous classes on current task data. However, there exist semantic gaps between current classes and old ones, which results in an inaccurate distillation target for preserving saliency activations on old samples. RRR [11] directly saves the Grad-CAM saliency activations of each sample in a replay buffer and applies distillation to memorize this old knowledge, which requires storing additional samples during incremental learning. Despite these initial works on saliency-guided incremental learning, saliency drift still remains problematic and leads to catastrophic forgetting.

2.2. Exemplar-free Class Incremental Learning

Compared to conventional class incremental learning, exemplar-free class incremental learning is more appropriate for applications where training data is sensitive and may not be stored in perpetuity. DAFL [5] uses a GAN to generate synthetic samples from past tasks as an alternative to storing actual data. DeepInversion, which inverts trained networks using random noise to generate images, is another popular EFCIL method [50]. Always Be Dreaming further improves on DeepInversion for EFCIL [44]. SDC attempts to overcome the problems caused by semantic drift when training new tasks on old class samples [51]. It directly estimates prototypes of each learned class to use in

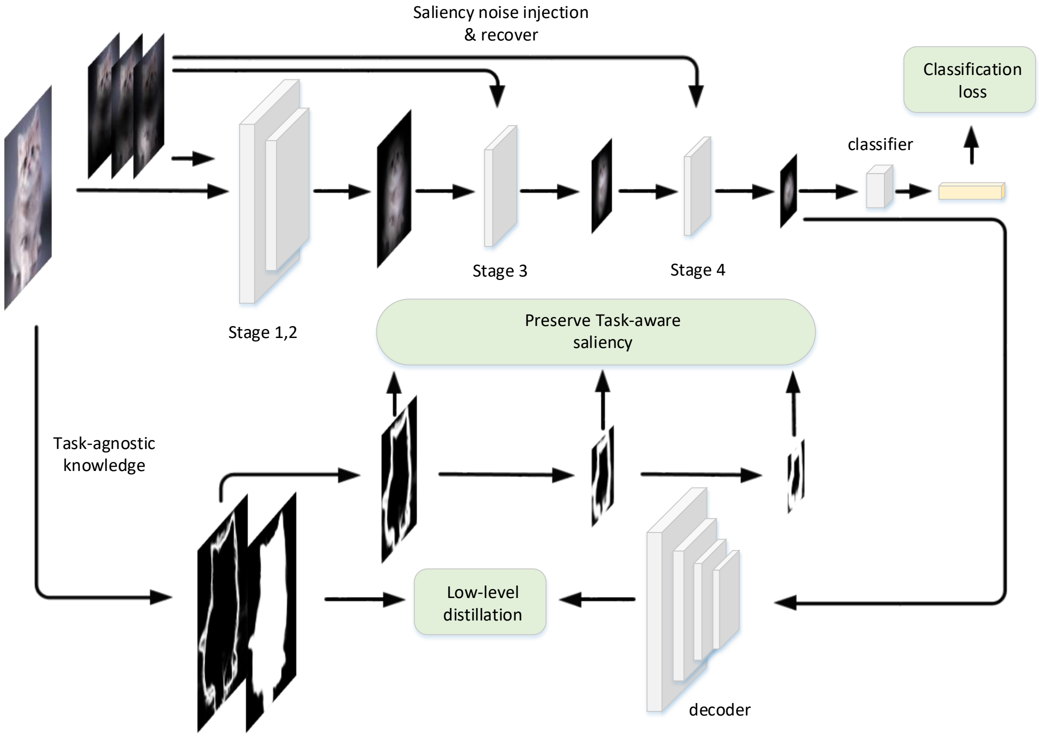

Figure 2. Overall framework of Task-Adaptive Saliency Supervision (TASS). We apply a low-level model to generate saliency and boundary maps. The boundary map is dilated and downsampled to provide supervision at different stages of the encoder. A decoder is attached after the encoder for low-level distillation, which serves as stationary task-agnostic saliency guidance. To prevent saliency drift in later training phases, we introduce saliency noise into each encoder stage. The model is trained to denoise and reduce the saliency drift on current data in future phases. TASS can be integrated into an EFCIL approach to provide robust saliency guidance across incremental tasks.

a nearest class mean classifier. PASS [57] and IL2A [56] are prototype-based replay methods for efficient and effective EFCIL. Since these prototypes are *features* computed from past training samples, image exemplars need not be retained. SSRE [58] introduces a re-parameterization method to trade-off between old and new knowledge. Our task-Adaptive Saliency Supervision (TASS) method uses three new components to reduce saliency drift in EFCIL and is complementary to several of the approaches mentioned above.

3. Task-Adaptive Saliency Supervision

We first define the Exemplar-Free Class Incremental Learning (EFCIL) scenario. Then we describe our TASS approach including dilated boundary supervision, auxiliary low-level supervision, and saliency noise injection. Our overall framework is illustrated in Figure 2.

3.1. Exemplar-free Class-Incremental Learning

Class-incremental learning aims to sequentially learn tasks consisting of disjoint classes of samples. Let $t \in \{1, 2, ..T\}$ denote the incremental learning tasks. The training data D_t for each task contains classes C_t with N_t training samples $\{(x_t^i, y_t^i)\}_{i=1}^{N_t}$, where x_t^i are images and $y_t^i \in C_t$ are their labels.

Most deep networks applied to class-incremental learning can be split into two components: a feature extractor F_θ and a common classifier G_ϕ which grows with each new task $t+1$ to include classes C_{t+1}. The feature extractor F_θ first maps the input x to a deep feature vector $z = F_\theta(x) \in \mathbb{R}^d$, and then the unified classifier $G_\phi(z) \in \mathbb{R}^{|C_t|}$ is a probability distribution over classes C_t used to make predictions on input x.

Class-incremental learning requires that the model be capable of correctly classifying all samples from previous tasks at *any* training task – that is, when learning task t, the model must not forget how to classify samples from classes from tasks $t' < t$. *Exemplar-free* class-incremental learning additionally restricts models to learn each new task without access to samples from previous ones. This typically leads to learning objectives that minimize a loss function $\mathcal{L}$ defined on current training data D_t:

$$\mathcal{L}_t^{\text{CIL}}(x, y) = \mathcal{L}_{\text{ce}}(G_{\phi_t}(F_{\theta_t}(x)), y) + \mathcal{L}_t^{\text{M}}, \tag{1}$$

where $\mathcal{L}_{\text{ce}}$ is the standard cross-entropy classification loss

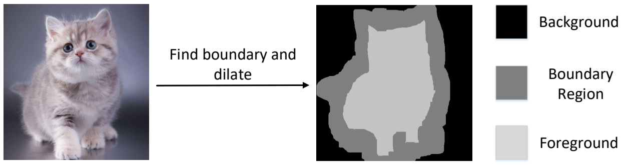

Figure 3. We dilate the boundary map and apply a binary cross entropy loss at three stages in the CNN backbone to prevent mid-level attention from drifting into boundary regions.

and $\mathcal{L}_t^{\mathrm{M}}$ is a method-specific loss that mitigates forgetting during incremental learning. Note that *without* $\mathcal{L}_t^{\mathrm{M}}$, Eq. 1 reduces to fine-tuning on task t.

3.2. Boundary-guided Mid-level Saliency Drift Regularization

Simply distilling attention between tasks does not consider task-adaptive attention. A model generates low-level representations of each input image x (saliency and boundary maps in our experiments). We use CSNet [6] to generate salient regions and boundary maps, as it is lightweight and efficient, but any model producing saliency maps can be used in our framework (we explore other options in the Supplementary Material). To introduce plasticity on attention at these intermediate layers in the backbone, we use the generated boundary maps as a type of adaptive supervision as shown in Figure 3. We add a penalty term on object boundaries to avoid drift into the background. We first use 0.5 as the threshold to binarize the generated boundary map and dilate the maps by:

$$B_d(x) = \mathrm{Dilate}(A_b(x), d), \tag{2}$$

where $A_b(x)$ is the generated boundary map of image x, which is converted from the saliency map with a Laplacian filter, and d denotes the dilation radius applied on the boundary map for controlling the strictness of boundary-guided saliency.

Rather than use a decoder at each layer as done for the low-level saliency maps described above, the mid-level saliency maps of our model are generated using Grad-CAM [42][1] at three stages of the CNN backbone (see Figure 2). We also experiment with several other methods for generating student saliency maps and report on them in the Supplementary Material. The dilated generated boundary map $B_d(x)$ is downsampled to match the feature map dimensions at these three stages in order to compare with the Grad-CAM generated saliency boundary maps. We use the binary cross entropy loss for supervision on dilated boundary regions. This loss is defined as:

$$\mathcal{L}_t^{\mathrm{dbs}}(x) = -\frac{\sum_{j=1}^{N} B_d(x,j)\log(1 - S(x,j))}{\sum_{j=1}^{N} B_d(x,j)}, \tag{3}$$

where $S(x,j)$ denotes the saliency map of our model on image x at pixel j, $B_d(x,j)$ is the dilated generated boundary map at pixel j, and N is the number of pixels in x. We compute this loss only within dilated boundary regions (i.e. where $B_d(x,j) = 1$). This loss helps the student saliency map avoid intersecting with the dilated teacher boundary region.

3.3. Auxiliary Low-level Supervisions for EFCIL

We propose to learn stable features from low-level stationary tasks shared across all incremental classification tasks during class-incremental learning. Low-level vision tasks like salient object detection require useful representations of input images. By learning these feature representations across tasks, the model can focus on key areas of input images and exploit learned, stable features with less representation drift since the low-level features change very little between tasks.

Saliency map prediction is relevant to image classification since the foreground largely determines the results, while the background is comparatively less important. When learning new tasks with new classes, the background of images of new classes may contain new visual concepts that introduce undesirable noise and lead to forgetting of essential previous knowledge. The effectiveness of saliency features for learning classification tasks was demonstrated by Saliency Guided Training [17]. Additional supervision of salient region boundaries can aid salient object detection tasks for both segmentation and localization [12, 21, 25, 34, 55]. The positive interaction between these two tasks brings richer attention to features relevant to the main classification task. It can provide positive guidance in the form of stationary knowledge across class-incremental tasks. Some examples are illustrated in Figure 5.

We incorporate low-level vision tasks into the network as an auxiliary supervision for enriching task-agnostic attention. The boundary map is computed with a Laplacian filter over the estimated saliency map. We add a decoder D_ψ [22] after the backbone F_θ to predict low-level saliency and boundary maps for input images. The average L2 distance between the prediction and target is used as a low-level saliency distillation loss:

$$\mathcal{L}_t^{\mathrm{lms}}(x) = \frac{||D_\psi(F_\theta(x)) - A(x)||_2}{\sqrt{N}}, \tag{4}$$

where $A(x)$ denotes the target low-level maps on input x, consisting of a saliency map $A_s(x)$ and a boundary map $A_b(x)$. $D_\psi(F_\theta(x))$ are combined saliency and boundary maps produced by the decoder, and N is the number of pixels in the saliency maps.

[1] https://github.com/jacobgil/pytorch-grad-cam

Algorithm 1 TASS Pseudocode

```
Input: The number of tasks $T$, training samples $D_t = \{(x_i, y_i)\}_{i=1}^{N}$ of task $t$, initial parameters $\Theta^0 = \{\theta_0, \phi_0, \psi_0\}$ containing parameters of feature extractor $F_\theta$, classifier $G_\phi$, and low-level decoder $D_\psi$.
Output: Model $\Theta^T$
for $t \in \{1, 2, ..., T\}$ do
    $\Theta^t \leftarrow \Theta^{t-1}$
    while not converged do
        Sample $(x, y)$ from $D_t$
        $\mathcal{L}_t^{\text{CIL}} \leftarrow$ SaliencyNoiseInjection$(x, y)$
        $\mathcal{L}_t^{\text{lms}} \leftarrow$ LowLevelMultitask$(x, A(x))$
        $S \leftarrow$ ComputeGradCAMSaliency$(x, y)$
        $\mathcal{L}_t^{\text{dbs}} \leftarrow$ DilatedBoundarySupervision$(S, A(x))$
        update $\Theta^t$ by minimizing $\mathcal{L}_t^{\text{all}}$ from Eq. 5
    end while
end for
```

3.4. Saliency Noise Injection

Although we apply low-level distillation and dilated boundary supervision to maintain saliency representations across tasks, the model can still forget saliency on samples from previous tasks. To address this, we force the model to be able to recover correct saliency maps from injected saliency noise.

At each task there is no available training data from previous or future tasks, and therefore we cannot directly know saliency drift on these samples. Instead of supervising the model with ground-truth saliency drift signals, we introduce saliency noise on random feature channels. We use a random ellipse to approximate the potential saliency drift in future tasks and the model is trained to denoise within each stage. Therefore the model can learn to effectively reduce real saliency drift.

We generate elliptical noise using a very simple approach. There are six parameter dimensions: the center coordinate (x, y), the major and minor axis lengths (a, b), the rotation angle α, and the mask weight w. A detailed explanation of this process is given in the Supplementary Material. With the help of dilated boundary supervision, each stage learns to eliminate this additional saliency noise and this aids generalization for future tasks and mitigates saliency forgetting in previous ones.

3.5. Learning Objective and Training Algorithm

The overall learning objective combines the low-level multi-task learning, dilated boundary supervision, and random saliency noise injection modules:

$$\mathcal{L}_t^{\text{all}} = \mathcal{L}_t^{\text{CIL}} + \mathcal{L}_t^{\text{lms}} + \mathcal{L}_t^{\text{dbs}}. \tag{5}$$

Comparing this loss with Eq. 1, for TASS $\mathcal{L}_t^{\text{M}} = \mathcal{L}_t^{\text{CIL}} + \mathcal{L}_t^{\text{lms}}$, thus incorporating saliency-aware supervision with the cross-entropy loss. The entire process is detailed in Algorithm 1.

4. Experimental Results

In this section we first describe our experimental setup and then compare TASS to state-of-the-art methods on several EFCIL benchmarks. In Section 4.3 we give further analysis of the various components of TASS.

4.1. Experimental Setup

We follow standard experimental protocols for EFCIL on three benchmark datasets.

Datasets. We perform experiments on CIFAR-100 [19], Tiny-ImageNet [20], and ImageNet-Subset [8]. For most experiments, we train the model on half of the classes for the first task, and then equally distribute the remaining classes across each of the subsequent tasks. The convention we use is: $F + C \times T$ means that the first task contains F classes, and the next T tasks each contain C classes. This is a common configuration for EFCIL used in both PASS [57] and SSRE [58].

State-of-the-art methods. Since we focus on EFCIL, we mainly compare with exemplar-free state-of-the-art approaches: SSRE [58], PASS [57], IL2A [56], EWC [18], LwF-MC [40], and MUC [31]. To demonstrate the effectiveness of our method, we also compare its performance with several exemplar-based methods like iCaRL (both nearest-mean and CNN) [40], EEIL [2], and LUCIR [16]. We also compare with RRR [11] integrated with SSRE, which focuses on preserving saliency using exemplar replay.

Implementation details and metrics. We use ResNet-18 [15] as a feature extraction backbone. This is the same base network used in SSRE [58] and PASS [57], two state-of-the-art EFCIL approaches. We use the decoder in [22] to estimate low-level student saliency maps. All experiments are trained from scratch using Adam for 100 epochs with an initial learning rate 0.001. The learning rate is reduced by a factor 10 at epochs 45 and 90. For exemplar-based approaches, we use herding [40] to select and store 20 samples per class following common settings [16, 40]. We implement RRR [11] with SSRE to fairly compare it with TASS. For dilated boundary supervision, we set d of the three mid-level boundary dilation stages to be 5%, 10% and 15% of the image size.

We report three common metrics for class incremental learning: the average and last top-1 accuracy, as well as average forgetting for all classes learned up to task t. Denoting by Acc_i the accuracy over all learned classes up to and including task i, the average accuracy is defined as $Avg = \frac{\sum_{i=1}^{T} Acc_i}{T}$, and the last accuracy is Acc_T. Letting $a_{m,n}$ denotes the accuracy of task n after learning task m,

Dataset	CIFAR-100			Tiny-ImageNet		
Method	**5 tasks**	**10 tasks**	**20 tasks**	**5 tasks**	**10 tasks**	**20 tasks**
MUC	38.45	19.57	15.65	18.95	15.47	9.14
+TASS	49.17 (+10.72)	40.34 (+20.77)	37.86 (+22.21)	32.47 (+13.46)	30.13 (+14.66)	27.70 (+18.56)
IL2A	55.13	45.32	45.24	36.77	34.53	28.68
+TASS	58.74 (+3.61)	53.24 (+7.92)	53.07 (+7.83)	42.49 (+5.72)	41.34 (+6.81)	40.59 (+11.91)
PASS	55.67	49.03	48.48	41.58	39.28	32.78
+TASS	59.10 (+3.43)	54.45 (+5.42)	52.37 (+3.89)	44.05 (+2.47)	43.06 (+3.78)	42.57 (+9.79)
SSRE	56.33	55.01	50.47	41.45	40.07	39.25
+TASS	59.26 (+2.93)	57.93 (+2.92)	53.78 (+3.31)	44.13 (+2.68)	43.86 (+3.79)	43.55 (+4.30)

Table 1. Performance gain in top-1 accuracy by applying TASS to other EFCIL methods in a plug-and-play way. Absolute gains are indicated in (red).

Dataset		**CIFAR100**									**TinyImageNet**								
Setting		**5 tasks**			**10 tasks**			**20 tasks**			**5 tasks**			**10 tasks**			**20 tasks**		
Method		Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$
E=20	iCaRL-CNN†	51.07	40.12	42.13	48.66	39.65	45.69	44.43	35.47	43.54	34.64	22.31	36.89	31.15	21.10	36.70	27.90	20.46	45.12
	iCaRL-NCM†	58.56	49.74	24.90	54.19	45.13	28.32	50.51	40.68	35.53	45.86	33.45	27.15	43.29	33.75	28.89	38.04	28.89	37.40
	LUCIR†	63.78	55.06	21.00	62.39	50.14	25.12	59.07	48.78	28.65	49.15	37.09	20.61	48.52	36.80	22.25	42.83	32.55	33.74
	EEIL†	60.37	52.35	23.36	56.05	47.67	26.65	52.34	41.59	32.40	47.12	34.24	25.56	45.01	34.26	25.91	40.50	30.14	35.04
	RRR†	66.43	57.22	18.05	65.78	55.74	18.59	62.43	51.35	18.40	51.20	42.23	16.67	49.54	40.12	21.64	47.46	35.54	29.10
E=0	LwF_MC	45.93	36.17	44.23	27.43	50.47	17.04	20.07	15.88	55.46	29.12	17.12	54.26	23.10	12.33	54.37	17.43	8.75	63.54
	EWC	16.04	9.32	60.17	14.70	8.47	62.53	14.12	8.23	63.89	18.80	12.71	67.55	15.77	10.12	70.23	12.39	8.42	75.54
	MUC	49.42	38.45	40.28	30.19	19.57	47.56	21.27	15.65	52.65	32.58	17.98	51.46	26.61	14.54	50.21	21.95	12.70	58.00
	IL2A	63.22	55.13	23.78	57.65	45.32	30.41	54.90	45.24	30.84	48.17	36.14	25.43	42.10	35.23	28.32	36.79	28.74	35.46
	PASS	63.47	55.67	25.20	61.84	49.03	30.25	58.09	48.48	30.61	49.55	41.58	18.04	47.29	39.28	23.11	42.07	32.78	30.55
	SSRE	65.88	56.33	18.37	65.04	55.01	19.48	61.70	50.47	18.37	50.39	41.67	17.25	48.93	39.89	22.50	48.17	39.76	26.74
	TASS (Ours)	**68.75**	**59.26**	**16.42**	**67.42**	**57.93**	**17.78**	**62.76**	**53.78**	**17.78**	**55.12**	**44.13**	**15.40**	**54.21**	**43.86**	**18.47**	**52.79**	**43.55**	**22.51**

Table 2. Average, last top-1 accuracy, and forgetting on CIFAR-100 with different numbers of tasks. Replay-based methods storing 20 exemplars from each previous class are indicate with †. The best overall results are in **bold**. We run all experiments three times and report the mean for all metrics.

Dataset	**ImageNet-Subset**								
Setting	**5 tasks**			**10 tasks**			**20 tasks**		
Method	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$	Avg↑	Last↑	$F\downarrow$
LwF_MC	34.86	24.10	49.36	31.18	20.01	53.04	27.54	17.42	56.07
MUC	40.65	27.89	47.13	35.07	22.65	52.10	31.44	20.12	53.85
PASS	63.12	52.61	22.47	61.80	50.44	23.57	55.23	46.07	26.73
SSRE	69.54	58.46	17.22	67.69	57.51	18.60	61.23	50.05	23.22
TASS (Ours)	**74.32**	**63.14**	**14.37**	**72.60**	**57.93**	**16.09**	**68.79**	**57.60**	**18.41**

Table 3. Average, last top-1 accuracy, and forgetting on ImageNet-Subset for different numbers of tasks. We run all experiments three times and report means for all metrics.

the forgetting measure f_k^i of task i after learning task k is computed as $f_k^i = \max_{t \in 1,2,\ldots,k-1}(a_{t,i} - a_{k,i})$. The average forgetting F_k is defined as $F_k = \frac{1}{k-1}\sum_{i=1}^{k-1} f_k^i$.

4.2. Comparison with the State-of-the-art

We compare TASS with the state-of-the-art on CIFAR-100 in Table 2. TASS outperforms all exemplar-free approaches. For exemplar-based methods like iCaRL [40], EEIL [2], and LUCIR [16], our method still has significantly better performance. On longer sequences (i.e. 10 and 20 tasks), our method significantly reduces forgetting when learning new classes compared to other EFCIL methods. TASS outperforms the best method SSRE by about 3% on the last task. This performance improvement can be also observed in terms of average forgetting.

As we see in Table 3 and Figure 4 for Tiny-ImageNet and ImageNet-Subset, although our method has similar top-1 accuracy on the first task in Figure 4, it has better performance at most intermediate tasks and also the final one. For longer sequences in Figure 4, the gap between our method and the best baseline is largely consistent, showing the effectiveness of our method at mitigating forgetting. The performance gain in Table 3 is larger on Tiny-ImageNet and ImageNet-Subset compared to CIFAR100, and this demonstrates that our method generalizes to datasets with larger images and object scales. It is worth mentioning that TASS also produces results with smaller variance. We believe this to be due to TASS reducing saliency drift to background regions, which may include random noise.

Plug-and-play with other EFCIL methods. Some existing EFCIL methods, like PASS [57], IL2A [56] and

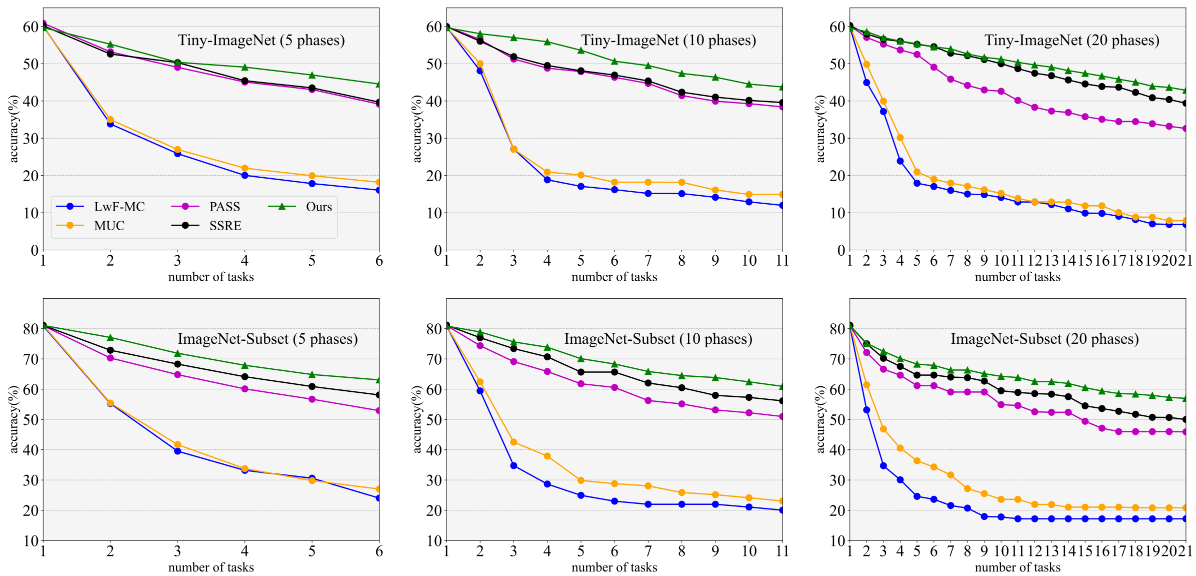

Figure 4. Results on Tiny-ImageNet and ImageNet-Subset for different numbers of tasks. Our method outperforms others, especially on longer task sequences (i.e. more, but smaller, tasks).

SSRE [58], focus on reducing forgetting via embedding regularization. Considering the importance of saliency to image classification, it is natural to consider whether TASS can be integrated into these methods. The results in Table 1 show the performance gain brought by this integration. Adding TASS doubles the performance for MUC in many cases and significantly improves IL2A and PASS. When we incorporate it into the best baseline SSRE, it yields a consistent gain of about 3%. These results clearly show that TASS, by explicitly mitigating saliency drift, is complementary to other methods in relieving forgetting. They additionally demonstrate the significance of saliency drift as a cause of catastrophic forgetting in EFCIL.

4.3. Additional Analysis

In this section we take a deeper look at the method we propose. If not specified, the results are produced using TASS integrated into SSRE [58].

Ablation Study. We performed ablations using the 10-task setting on CIFAR-100 (see Table 4). We ablate on both PASS [57] and SSRE [58]. Low-level multi-task supervision is crucial and improves by 2.2% (PASS) and 1.2% (SSRE). Dilated boundary supervision further boosts performance by about 1-2%. Saliency noise injection is also helpful for both methods and improves by 1.5% for PASS. In total, TASS improves baselines by 5.5% and 2.9% points, respectively. Note that SSRE is the previous state-of-the-art method and TASS outperforms it by a large margin.

Low-level Multi-task Supervision. To analyze the effect of our proposed low-level saliency supervision, we performed experiments on ImageNet-Subset in the 5 and 10 task settings. We first plot the loss across tasks in Figure 5(c). After learning to predict boundary and saliency maps in the first task, the network maintains good performance for the rest of the 5 task sequence. This shows that the low-level tasks are stable during continual learning. Furthermore, we visualize the results of saliency and boundary map prediction during incremental learning in Figure 5(a-b). We give some examples of predicted boundary and saliency maps after learning different tasks. Although CIL involves samples of different classes, we see that the low-level outputs are relatively stable and class agnostic. Since the model is able to stably predict these low-level features across tasks, it therefore can preserve useful prior knowl-

Method & Tasks	$\mathcal{L}_{\text{lms}}$	$\mathcal{L}_{\text{dbs}}$	SNI	Accuracy
Baseline (PASS)				49.0
Variants	✓			51.2
	✓	✓		53.0
	✓	✓	✓	54.5
Baseline (SSRE)				55.0
Variants	✓			56.2
	✓	✓		57.3
	✓	✓	✓	57.9

Table 4. Ablations on each TASS component. Experiments are on CIFAR-100 in the 10-task setting and we report the top-1 accuracy in % for TASS integrated into PASS and SSRE. $\mathcal{L}_{\text{dbs}}$ (Eq. 3), $\mathcal{L}_{\text{lms}}$ (Eq. 4), and SNI denote the three TASS components: Dilated Boundary Supervision, Low-level Multi-task Supervision, and Saliency Noise Injection.

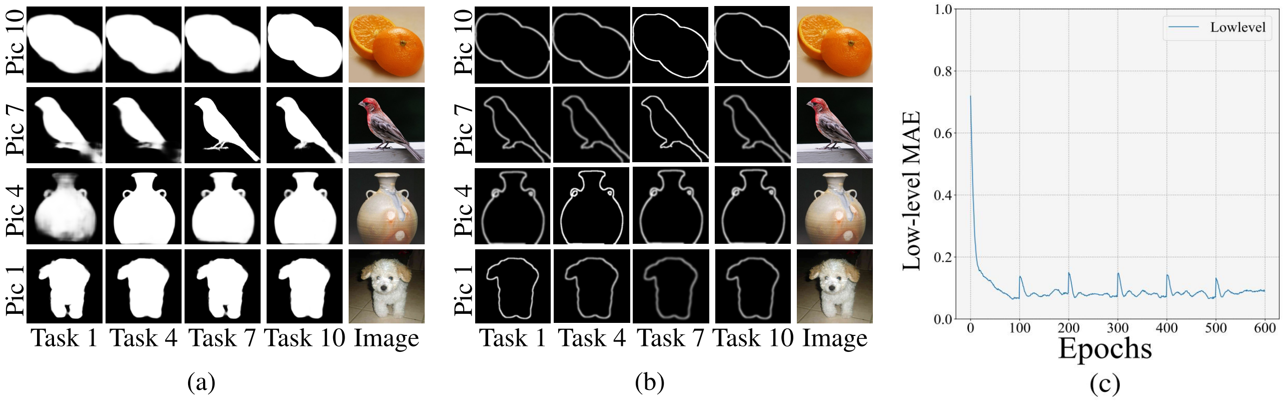

Figure 5. Visualization of the saliency (a) and boundary (b) maps from our student encoder-decoder network with original images from different tasks at different stages of incremental learning. Our method produces stable low-level results while reducing forgetting in classification. (c) The MAE loss between the student and teacher network across tasks.

edge for continual learning.

Metric & Method	Avg↑	Last↑
FeTrIL [38]	65.20	56.34
SOPE [59]	65.84	56.80
PRAKA [43]	68.86	59.20
TASS (SSRE)	67.42	57.93
TASS (PRAKA)	**69.70**	**60.04**

Table 5. Average and last accuracy on CIFAR-100 10-task setting.

Metric & Method	Avg↑
SOPE [59]	60.20
FeTrIL [38]	65.00
TASS (FeTrIL)	**66.03**

Table 6. Average accuracy on the ImageNet-Full 10-task setting.

TASS with more methods and benchmarks. Due to the strong generalization capability of our method, we have also applied our TASS paradigm to PRAKA [43], resulting in further performance improvements. We give specific experimental comparisons in Table 5. We also compare with FeTrIL [38] in Table 5 and Table 6 above. Experiments on ImageNet-Full show consistent improvement in Table 6 above.

Quantitative analysis of saliency drift. We measure the intersection over union (IoU) between the self-attention map of the last layer in DINO and the Grad-CAM saliency maps for both SSRE and SSRE+TASS. As shown in Table 7, TASS significantly reduces saliency drift in the student model, which again shows the effectiveness of our method for maintaining saliency during CIL.

Task	1	4	7	10
SSRE	47.4	50.1	56.6	78.5
SSRE+TASS	75.2	82.3	88.5	90.1

Table 7. Average IoU (%) between DINO self-attention maps and student model saliency in the CIFAR-100 10-task.

5. Conclusions

In this paper, we propose an approach of task-adaptive saliency guidance for EFCIL. The insight behind TASS is to guide the model to focus on salient regions and inhibit saliency drift. We show that robust saliency guidance is crucial to mitigating forgetting across tasks. Experiments demonstrate that TASS is effective and surpasses the state-of-the-art. TASS can be easily combined with other methods, leading to large performance gains over baselines. Qualitative results also show that low-level tasks are stable across different tasks, resulting in less forgetting.

Limitations and Future Work. In this work, we introduce the auxiliary saliency knowledge across tasks for better CIL performance, which may raise the concern of unfair comparison. However, we believe it is important to explore external knowledge to make the CIL system applicable in real applications. Additionally, our method can be easily integrated with other methods and the low-level auxiliary knowledge takes negligible cost and is easy to obtain. We will explore leveraging other forms of knowledge in future work, such as pre-trained visual and large language models.

Acknowledgments This work is funded by NSFC (NO. 62206135, 62225604), Young Elite Scientists Sponsorship Program by CAST (2023QNRC001), and the Fundamental Research Funds for the Central Universities (Nankai Universitiy, 070-63233085). Computation is supported by the Supercomputing Center of Nankai University.

References

[1] Eden Belouadah, Adrian Popescu, and Ioannis Kanellos. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135:38–54, 2021. 1, 2

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018. 1, 5, 6

[3] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 2

[4] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 2

[5] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, 2019. 2

[6] Ming-Ming Cheng, Shanghua Gao, Ali Borji, Yong-Qiang Tan, Zheng Lin, and Meng Wang. A highly efficient model to study the semantics of salient object detection. *IEEE TPAMI*, 2021. 4, 11

[7] 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 TPAMI*, 2021. 2

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[9] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, 2019. 2

[10] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 1, 2

[11] Sayna Ebrahimi, Suzanne Petryk, Akash Gokul, William Gan, Joseph E. Gonzalez, Marcus Rohrbach, and trevor darrell. Remembering for the right reasons: Explanations reduce catastrophic forgetting. In *ICLR*, 2021. 2, 5

[12] Deng-Ping Fan, Ge-Peng Ji, Peng Xu, Ming-Ming Cheng, Christos Sakaridis, and Luc Van Gool. Advances in deep concealed scene understanding. *Visual Intelligence*, 1(1):16, 2023. 4

[13] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9427–9436, 2022. 2

[14] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 1

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 5, 6

[17] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. *NeurIPS*, 2021. 4

[18] 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*, 2017. 2, 5

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 5

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[21] Hao Li, Dingwen Zhang, Nian Liu, Lechao Cheng, Yalun Dai, Chao Zhang, Xinggang Wang, and Junwei Han. Boosting low-data instance segmentation by unsupervised pre-training with saliency prompt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15485–15494, 2023. 4

[22] Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *ECCV*, 2020. 4, 5

[23] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 11

[24] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, 2016. 2

[25] Liqiang Lin, Pengdi Huang, Chi-Wing Fu, Kai Xu, Hao Zhang, and Hui Huang. On learning the right attention point for feature enhancement. *Science China Information Sciences*, 66(1):112107, 2023. 4

[26] Zheng Lin, Zhao Zhang, Zi-Yue Zhu, Deng-Ping Fan, and Xia-Lei Liu. Sequential interactive image segmentation. *Computational Visual Media*, 9(4):753–765, 2023. 2

[27] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2019. 11

[28] Jiang-Jiang Liu, Qibin Hou, and Ming-Ming Cheng. Dynamic feature integration for simultaneous detection of salient object, edge and skeleton. *IEEE TIP*, 2020. 11

[29] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *ICPR*, 2018. 2

[30] Xialei Liu, Yu-Song Hu, Xu-Sheng Cao, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Long-tailed class incremental learning. In *European Conference on Computer Vision*, pages 495–512. Springer, 2022. 2

[31] 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 *ECCV*. Springer, 2020. 2, 5

[32] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost van de Weijer. Augmented box replay: Overcoming foreground shift for incremental object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11367–11377, 2023. 2

[33] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *NeurIPS*, 2017. 2

[34] Zhouzhou Ma, Guanghua Gu, and Wenrui Zhao. Self-attention guidance based crowd localization and counting. *Machine Intelligence Research*, pages 1–17, 2024. 4

[35] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018. 2

[36] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE TPAMI*, 2022. 1

[37] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 1

[38] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *WACV*, 2023. 8

[39] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *NeurIPS*, 2021. 1

[40] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 2, 5, 6

[41] 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. In *ICLR*, 2019. 2

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 4

[43] Wuxuan Shi and Mang Ye. Prototype reminiscence and augmented asymmetric knowledge aggregation for non-exemplar class-incremental learning. In *ICCV*, 2023. 8

[44] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *ICCV*, 2021. 2

[45] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 1, 2

[46] Jia-Wen Xiao, Chang-Bin Zhang, Jiekang Feng, Xialei Liu, Joost van de Weijer, and Ming-Ming Cheng. Endpoints weight fusion for class incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7204–7213, 2023. 2

[47] Ju Xu and Zhanxing Zhu. Reinforced continual learning. *NeurIPS*, 2018. 2

[48] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, 2013. 11

[49] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 11

[50] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *CVPR*, 2020. 1, 2

[51] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, 2020. 1, 2

[52] Lu Yu, Xialei Liu, and Joost Van de Weijer. Self-training for class-incremental semantic segmentation. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 2

[53] Jiang-Tian Zhai, Xialei Liu, Andrew D Bagdanov, Ke Li, and Ming-Ming Cheng. Masked autoencoders are efficient class incremental learners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19104–19113, 2023. 2

[54] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C-C Jay Kuo. Class-incremental learning via deep model consolidation. In *WACV*, 2020. 2

[55] Jia-Xing Zhao, Jiang-Jiang Liu, Deng-Ping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, 2019. 4

[56] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *NeurIPS*, 2021. 1, 2, 3, 5, 6

[57] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7, 11

[58] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *CVPR*, 2022. 2, 3, 5, 7

[59] Kai Zhu, Kecheng Zheng, Ruili Feng, Deli Zhao, Yang Cao, and Zheng-Jun Zha. Self-organizing pathway expansion for non-exemplar class-incremental learning. In *ICCV*, 2023. 8

Method	5 Tasks	10 Tasks	20 Tasks
baseline (SSRE)	40.2	40.0	39.3
CAM	41.2	40.7	40.4
SmoothGrad	42.1	41.0	40.4
Grad-CAM	44.1	43.9	43.5

Table 8. Ablation on methods for generating saliency maps on Tiny-Imagenet.

Model	Parameter(M)	FLOPS(G)
Ours	17.9	0.78
Pre-trained salient model	0.0941	0.012

Table 9. Parameters and FLOPs of the pre-trained salient model. FLOPs are computed using 3 × 32 × 32 images.

A. Further Ablation Studies

Ablation on saliency methods. To show the generalization of TASS, we use several methods to compute saliency maps and report the results in Table 8. Grad-CAM performs the best, although other methods yield performance gains, demonstrating the effectiveness of TASS.

Ablation on low-level target maps. In the manuscript we use CSNet [6] to compute all the pre-trained saliency and boundary maps because it is very lightweight. Compared to our main model, the pre-trained model has fewer than 1% parameters and requires 1.5% of the FLOPs (as shown in Table 9). Note that we compute all low-level maps offline before new tasks, and so the extra FLOPs should be amortized over the number of epochs. Therefore, the additional FLOPs required by the low-level model is only about 0.015% of the main model, which is negligible in practice.

To show the effectiveness of TASS, we perform an ablation on the low-level maps. We replace them with the Grad-CAM generated from a ResNet-152 network. To avoid information leakage, ResNet-152 was trained from scratch. Before each new task, we first train it only on task data and use the Grad-CAM output to supervise saliency in our incremental model. From Table 10 we see that TASS still outperforms other methods. Moreover, TASS is applicable to other models for generating saliency maps, (e.g. DFI [28] or PoolNet [27]) with more parameters, and produces even better performance with larger networks.

Ablation on method architecture and salient model pretraining. We select PASS [57] as our baseline method to apply TASS to (as shown in Table 11 and Table 12). Experiments in these two tables are conducted on ImageNet-Subset with 5 tasks. Since some methods use ImageNet pretrained weights for better saliency map estimation, we train CSNet [6] from scratch on the dataset (with and without pretraining) for salient object detection [23, 48, 49]. This allows us to verify that no information leakage happens due to pretraining the saliency network on ImageNet. The low-level network without pretraining works almost as well as pretraining the saliency network on ImageNet. We also compare the number of parameters of different methods in Table 11. This shows that adding network capacity for PASS from ResNet-18 to ResNet-32 with more parameters only improves the performance marginally. Ours with ResNet-18 based on PASS achieves a significant gain surpassing SSRE which has more parameters.

Low-level source (Method)	Accuracy (%)
PASS	39.3
SSRE	40.0
ResNet152 (Ours)	42.1
CSNet (Ours)	43.9
PoolNet (Ours)	44.2
DFI (Ours)	44.4

Table 10. Ablation on low-level saliency maps on Tiny-ImageNet with 10 tasks.

Method	Parameter (M)	Accuracy (%)
PASS-Res18	14.5	50.4
PASS-Res32	21.7	51.2
SSRE-Res18	19.4	58.7
Ours-Res18	17.9	61.5

Table 11. Comparison of different method network architectures. Method-Res18 denotes applying Method with ResNet18 as its backbone.

Method	Accuracy (%)
No pretraining	61.5
Pretrained salient detection model	62.0

Table 12. Ablation on salient detection network pretraining.

Our approach in non-DFCIL scenarios. We apply our saliency supervision in a non-DFCIL scenario using PASS in Table 13 by including 20 exemplars per class. TASS boosts performance significantly here as well.

Method	buffer size	Acc (%)
PASS	20	52.36
PASS+TASS	20	**55.75**

Table 13. TASS on a non-DFCIL scenario.

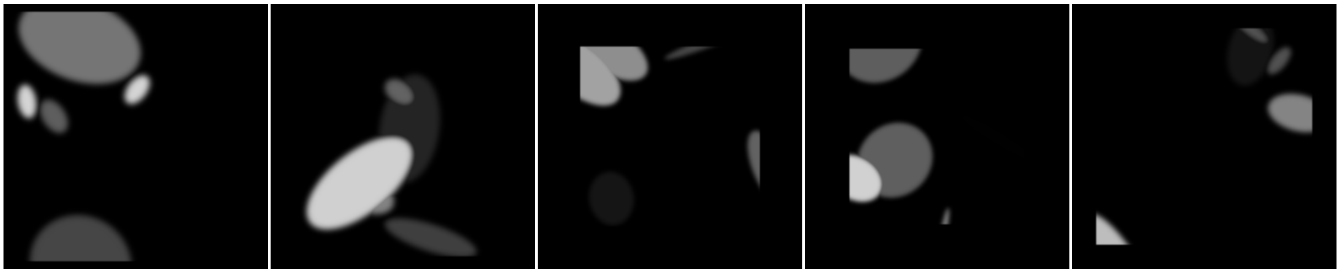

Figure 6. Visualization of some generated saliency noise maps.

Hyper-parameters of multiple losses. In Eq. 5 we weight all loss terms equally. As suggested, we explore more options in Table 14. Tuning further improves the performance slightly, but we stick with $\lambda_{\text{CIL}} = \lambda_{\text{lm}} = \lambda_{\text{dbs}} = 1.0$ for convenience. $\sqrt{N}$ in Eq. 2, where N is the number of pixels, is used to normalize the L2 distance.

λ_{CIL}	λ_{lm}	λ_{dbs}	Acc(%)
1	1	1	55.01
0.1	1	1	**55.27**
1	0.1	1	54.22
1	1	0.1	54.31

Table 14. Hyper-parameters of multiple losses for SSRE+TASS.

Method		L	D	S	LD	LS	DS	LDS
PASS	49.0	51.2	50.6	51.4	53.0	52.6	53.7	54.5
SSRE	55.0	56.2	55.8	56.7	57.3	57.0	57.6	57.9

Table 15. LDS represent Low-level multi-task supervision, Dilated boundary supervision, and Saliency noise injection, respectively.

All loss permutations ablation. We give all possible combinations of all three loss terms in Table 15. These results show that each component contributes to the final performance and that a combination of them performs best.

Class division in experimental protocol. We follow conventional experimental setups from previous works like PASS and SSRE to divide the classes of the dataset as $F + C \times T$ with $F = 50$. As suggested, we evaluate different options for F in Table 16. TASS shows consistent gain compared to the baseline under all settings.

B. More Visualizaions on TASS

Saliency Noise. For each ellipse there are 6 dimensions: the center coordinate (x, y), the rotation angle α, the mask weight w, and the major and minor axes (a, b). x, y, α and w are sampled from a uniform distribution over ranges: $x \in [0, H)$, $y \in [0, W)$, $\alpha \in [0, 2\pi)$, $w \in [0, 1]$. H and W denote the height and width of input images. To generate ellipses of appropriate size, we draw the major and minor axes from a Gaussian distribution with $\mu_a = \max(H, W)/2$, $\sigma_a = \max(H, W)/6$, $\mu_b = \min(H, W)/2$, $\sigma_b = \min(H, W)/6$. The sampled a, b is clipped to$[0, max(H, W)/2]$ and $[0, min(H, W)/2]$, respectively. For each ellipse, we create a saliency map S_i. We repeat this random generation process 3-5 times and apply an element-wise max operation on the S_i to obtain a single saliency map S. Then we crop and resize S to the original image size, with crop size sampled from a uniform distribution in $[\min(H, W)/2, \min(H, W)]$, introducing center-aware saliency noise to the network for training. Finally, we apply a Gaussian blur on S to better simulate a realistic saliency map. The kernel size for Gaussian blurring is the closest odd integer to $\min(H, W)/20$. For each encoder feature map, 10% of randomly selected channels are directly masked with S, where each selected channel will have an independent S. We visualize several generated samples in Figure 6.

F & Acc (%)	50	30	10	0
PASS	49.03±0.9	46.78±0.9	44.65±1.0	40.27±1.0
PASS+TASS	54.45±0.4	51.22±0.5	48.58±0.5	44.30±0.5

Table 16. Ablation on F with $T = 10$.

Embedding Visualization. Since our method helps the model focus on the foreground, more class-specific pixels contribute to the embedding. Thus embeddings are more discriminative and contain less distracting background information. In Figure 7 we use t-SNE to visualize embeddings of five initial classes after learning the base and last task in the 10-task setting on ImageNet-Subset. At the base task, both Baseline (SSRE) and Ours (SSRE+TASS) perform well. After the last task, it is clear that TASS helps maintain discriminative features between tasks while the Baseline has overlapping embeddings.

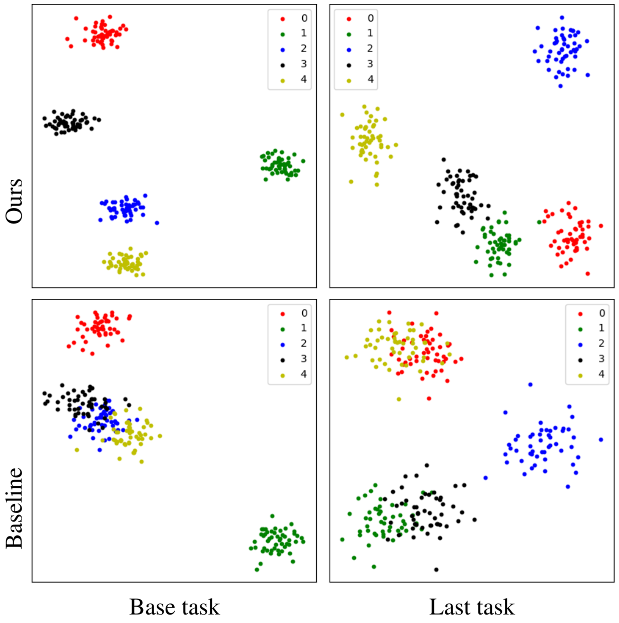

Figure 7. Visualization of the embedding $F_\phi(x)$ with and without TASS. Compared to the baseline, our method preserves more discriminative representations.