AAKR: Adversarial Attack-based Knowledge Retention for Continual Semantic Segmentation

Zhidong Yu[1,2], **Xiaoman Liu**[1,3], **Jiajun Hu**[1,3], **Zhenbo Shi**[1,2,3,4,*], **Wei Yang**[1,2,3,*]
[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei, China
[2]Hefei National Laboratory, University of Science and Technology of China, Hefei, China
[3]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou, China
[4]Laboratory for Advanced Computing and Intelligence Engineering, Wuxi, China
zbshi@mail.ustc.edu.cn, qubit@ustc.edu.cn

Abstract

In the context of Continual Semantic Segmentation (CSS), replay-based methods tend to achieve better performance than knowledge distillation-based ones, as the former utilizes additional data to transfer old knowledge. However, this advantage is at the cost of necessitating additional space for storing the generative model and extra time for continual training. To address this predicament, we propose a novel CSS framework, namely **A**dversarial **A**ttack-based **K**nowledge **R**etention (AAKR). The AKKR framework generates specific adversarial samples by adding images, and uses them to retain old knowledge. Specifically, we leverage adversarial attacks to generate adversarial images for incremental samples. By imposing additional constraints within these attacks, we enhance the transfer of old knowledge, thereby reinforcing the understanding of previously learned information. Furthermore, we design an attack probability module that adjusts adversarial attack directions based on training feedback. This module effectively encourages the new model to learn old knowledge from poorly protected classes, significantly improving knowledge transfer effectiveness. Our comprehensive experiments demonstrate the efficacy of AAKR, and showcase that AAKR surpasses state-of-the-art competitors on benchmark datasets.

Introduction

Semantic segmentation is a fundamental task in computer vision. With the continuous advancement of deep learning, sophisticated models have exhibited exceptional proficiency in this task. However, these deep learning models face a significant challenge known as catastrophic forgetting within the realm of Continual Semantic Segmentation (CSS) (Michieli and Zanuttigh 2019). Catastrophic forgetting refers to the susceptibility of neural networks to rapidly overwrite their prior knowledge when learning new classes. As the model accommodates new categories, the risk of erasing previously learned ones increases, which adversely affects the overall performance.

Existing approaches usually use two strategies to solve catastrophic forgetting: knowledge distillation-based and replay-based. The former (Cermelli et al. 2020; Douillard et al. 2021; Phan et al. 2022; Michieli and Zanuttigh 2021; Cha et al. 2021; Zhang et al. 2022) aims to distill knowledge from the old model into a new one. In contrast, the latter (Maracani et al. 2021; Zhu et al. 2023) relies on retaining past data or generating relevant data to prevent forgetting. The latter tends to yield better results, especially in longer learning steps. However, additional space is required to save the generative model as well as additional time to learn the generated and added images together.

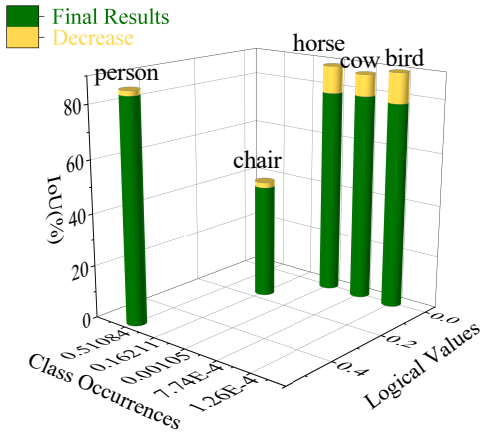

Figure 1: Relationship between performance decay and the proportions of class occurrences and logical values of added images.

To circumvent these limitations, we conduct an in-depth study. As shown in Fig. 1, during incremental training, the performance decay of each old class correlates with both the frequency of its pseudo-label appearances in the added samples and its predicted values from the old model trained by PLOP (Douillard et al. 2021). The old class appearing more frequently in the added samples exhibits improved knowledge transfer. Therefore, we can enhance their knowledge transfer by boosting the predictions of old models on old classes with much degraded performance. Based on this, we introduce a new CSS method: **A**dversarial **A**ttack-based **K**nowledge **R**etention (AAKR). AAKR employs adversarial attacks to perturb added samples in a finite number of steps, enhancing old classes that are prone to forgetting. This targeted knowledge retention at a minimal cost leads to signif-

*Corresponding Authors.

icant performance improvements.

AAKR includes an attack probability module that calculates the percentage of attack targets based on continual training feedback. Due to the limited inclusion of old classes in the added samples, accurately reflecting the extent of forgetfulness for each class is challenging. This module initially constructs test samples on the added data through untargeted attacks and calculates the forgetfulness degree for each class based on predictions from both old and new models. Using these results, it determines the proportion of each class in the attack target. Higher forgetfulness corresponds to greater importance for knowledge retention, requiring a higher proportion to induce the old model to predict the class. Ultimately, the attack goal label is computed from the proportions of each class.

Extensive experiments conducted on the benchmark dataset demonstrate that AAKR not only surpasses distillation-based approaches but also achieves superior results compared to replay-based methods, without a substantial increase in computational and storage requirements.

Our main contributions can be summarized as follows:

- We present AAKR, an approach that strategically employs adversarial attacks to enhance the protection of old knowledge in forgettable categories.
- We implement an attack probability module that constructs attack targets using continual training feedback to improve knowledge transfer effectiveness.
- Extensive experiments show that AAKR performs better than state-of-the-art methods with minimal computational cost, eliminating the need for external memory storage.

Related Work

Continual Learning

There are gradually increasing concerns about continual learning. Previous works are divided into three main categories: replay-based, regularization-based, and parameter isolation-based. Replay-based methods (Rebuffi et al. 2017; Castro et al. 2018; Wu et al. 2018; Hou et al. 2019; Iscen et al. 2020; Meng et al. 2025) select or generate examples of previous tasks in some way. Then, the model employs these examples along with the new data to learn the new classes. Regularization-based methods (Zenke, Poole, and Ganguli 2017; Dhar et al. 2019; He et al. 2020; Kanakis et al. 2020; Douillard et al. 2020; Kang, Park, and Han 2022) in continual learning aim to preserve knowledge by introducing additional loss terms with regularization constraints to control parameter variations. Parameter isolation-based methods (Mallya, Davis, and Lazebnik 2018; Liu et al. 2020; Hu et al. 2023) allocate an independent set of model parameters to each task to prevent forgetting.

Continual Semantic Segmentation

Michieli et al. (Michieli and Zanuttigh 2019) first proposed continual learning for semantic segmentation and put forward a distillation-based framework to address the catastrophic forgetting. Subsequently, several distillation-based works (Cermelli et al. 2020; Douillard et al. 2021; Michieli and Zanuttigh 2021; Yang et al. 2022; Phan et al. 2022; Xiao et al. 2023; Cong et al. 2023; Goswami et al. 2023; Yuan, Zhao, and Shi 2024; Park et al. 2025) are proposed. MiB (Cermelli et al. 2020) models the background to address the background shift problem. PLOP (Douillard et al. 2021) proposes Local POD that preserves long and short-distance spatial relationships at the feature level. SDR (Michieli and Zanuttigh 2021) uses prototype matching and contrast learning to construct robust features. REMINDER (Phan et al. 2022) adjusts the distillation weights of each class based on the similarity between objects. While MBS(Park et al. 2025) proposes a background-class separation framework that distills only trustworthy past knowledge and separates between the background and new classes.

In addition, the replay-based method (Maracani et al. 2021; Zhu et al. 2023) retains the seen classes using additional images. Maracani et al. (Maracani et al. 2021) use the images generated by GAN or crawled from the Web to retain the old knowledge. Afterwards, Zhu et al. (Zhu et al. 2023) propose a reinforcement-based learning method for choosing valuable past samples. Some other approaches (Cha et al. 2021; Zhang et al. 2022) achieve promising results with additional models or structures. Toldo et al. (Toldo, Michieli, and Zanuttigh 2024) put forward a new incremental learning setup, i.e., variable distributions in the input and labeling space. In this paper, we propose an adversarial attack-based approach that achieves better performance at a lower cost compared to replay-based approaches.

Adversarial Attack

Adversarial attacks are commonly used to generate images that produce false predictions when these images are fed into a model. Initially, Goodfellow et al. (Goodfellow, Shlens, and Szegedy 2015) address the effect of adversarial attacks on image classification by introducing a Fast Gradient Symbol Method (FGSM). Subsequently, the Projective Gradient Descent (PGD) (Madry et al. 2018) is proposed. As a multi-step technique, PGD tailors adversarial instances to classification models. In semantic segmentation, the SegPGD (Gu et al. 2022) can effectively utilize adversarial strategies to distort the prediction results. CosPGD (Agnihotri and Keuper 2023) exploits the cosine similarity between the prediction and ground truth to realize dedicated attacks for pixel predictions. Rony et al. (Rony, Pesquet, and Ben Ayed 2023) handle large numbers of constraints within a nonconvex minimization framework via an Augmented Lagrangian approach. In this paper, we introduce a novel adversarial attack that constructs attack goals based on knowledge transfer during continual training, resulting in adversarial samples that enhance knowledge transfer.

Methodology

Overview

Before formulating the framework, we first introduce some related notations. The purpose of CSS is to train a segmentation model over T steps to incorporate new classes while retaining the knowledge of existing ones. We denote the class

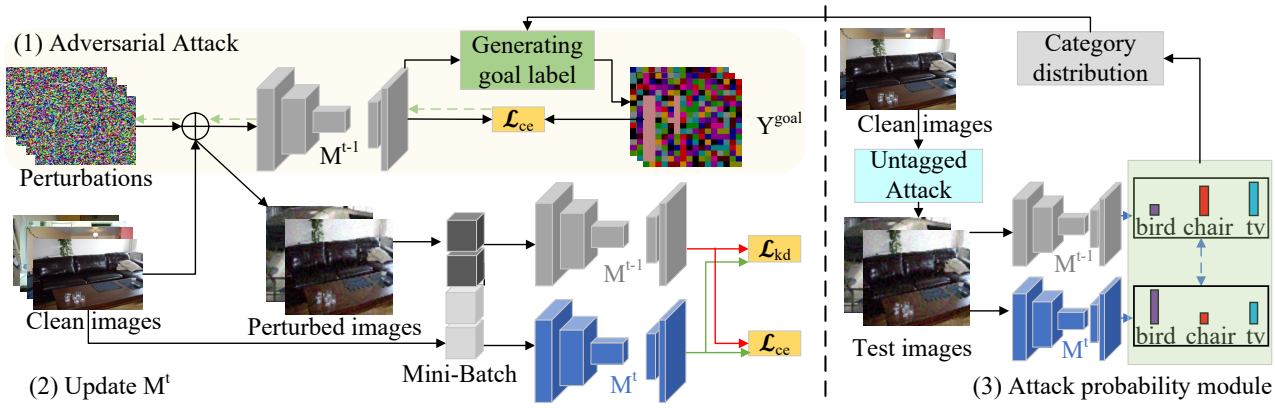

Figure 2: Overview of AAKR. At each batch, the added images are attacked to generate adversarial samples, which enables M^{t-1} to generate predictions containing more old classes. Afterward, the clean and adversarial samples form a mini-batch at the same time, which is optimized by $\mathcal{L}_{ce}$ and $\mathcal{L}_{kd}$ for the model M^t. The model M^{t-1} in gray indicates being frozen.

learned at step t as C^t, and $C^{1:t-1}$ encompasses all encountered classes from step 1 to step $t-1$. At step t, we introduce a dataset D_t consisting of pairs $(\mathbf{X}^t, \mathbf{Y}^t)$, where $\mathbf{X}^t$ represents an image with dimensions $H \times W$, and $\mathbf{Y}^t$ stands as the corresponding label, exclusively containing the class C^t acquired in the present step. Conventionally, the segmentation model at step t is denoted as M^t. And, the pseudo-label of all classes $\tilde{\mathbf{Y}}^{1:t}$ are combined with $\mathbf{Y}^t$ and the pseudo-label of old classes $\tilde{\mathbf{Y}}^{1:t-1}$ outputted by M^{t-1} of $\mathbf{X}^t$.

Fig. 2 illustrates the framework of AAKR, which initially attacks the added images within each batch. The perturbations incites the old model to generate additional information about old classes for the adversarial images. This targeted perturbation is an integral component of the attack objective, as realized by our designed $\mathbf{Y}^{goal}$, which signifies the target for the attack. Following this, the adversarial images are amalgamated with the original ones, creating a mini-batch. Then, on this mini-batch, the model is optimized by two different losses, i.e., the cross-entropy loss ($\mathcal{L}_{ce}$) and the knowledge distillation loss ($\mathcal{L}_{kd}$). Additionally, an attack probability module is used to calculate the importance of each class and assign sampling weights to each class.

Analysis for Retaining Knowledge

In general, the model M^t is updated through $\mathcal{L}_{ce}$ and $\mathcal{L}_{kd}$ during the continual learning phase. The former facilitates the incorporation of new classes learned by M^t while retaining the knowledge from the addition of old classes through new data. The formulation of $\mathcal{L}_{ce}$ is:

$$\mathcal{L}_{ce} = \frac{-1}{|D_t|} \sum_{\mathbf{X}^t \in D_t} \tilde{\mathbf{Y}}^{1:t} log(M^t(\mathbf{X}^t)) \tag{1}$$

where $M^t(\mathbf{X}^t)$ means the prediction maps of $\mathbf{X}^t$.

The latter is utilized to ensure the retention of old knowledge by aligning the predictive distribution of M^t with that of M^{t-1} for the same input data. The formulation of $\mathcal{L}_{kd}$ is:

$$\mathcal{L}_{kd} = \frac{-1}{|D_t|} \sum_{\mathbf{X}^t \in D_t} M^{t-1}(\mathbf{X}^t) log(M^t(\mathbf{X}^t)) \tag{2}$$

Through the optimization of these two losses, the model becomes capable of achieving CSS on the added data. Next, we use $\mathcal{L}_{ce}$ and $\mathcal{L}_{kd}$ to analyze the effect of predictions of the old model on the added data.

Firstly, considering $\mathcal{L}_{ce}$, its derivative is expressed as:

$$\frac{\partial \mathcal{L}_{ce}}{\partial \theta_{M^t}} = \frac{-1}{|D_t|} \sum_{\mathbf{X}^t \in D_t} \sum_{c}^{C^{1:t-1}} \frac{\tilde{\mathbf{Y}}^{1:t}_{[c]}}{M^t(\mathbf{X}^t)_{[c]}} \cdot \frac{\partial M^t(\mathbf{X}^t)_{[c]}}{\partial \theta_{M^t}} \tag{3}$$

where θ_{M^t} means the parameters of M^t, $\tilde{\mathbf{Y}}^{1:t}_{[c]}$ denotes the c-channel of $\tilde{\mathbf{Y}}^{1:t}$, and $M^t(\mathbf{X}^t)_{[c]}$ is the c-channel of $M^t(\mathbf{X}^t)$.

In Eq. (3), it is evident that within the cross-entropy loss, $\tilde{\mathbf{Y}}^{1:t}$ is presented as a one-hot label. The impact of class c on the parameters is proportional to its frequency within $\tilde{\mathbf{Y}}^{1:t}$. As a result, older classes that are either absent from the label or occur infrequently exert diminished influence on θ_{M^t}. Besides, these classes are more prone to being forgotten during the process of learning new classes.

Similarly, for $\mathcal{L}_{kd}$, its derivative can be expressed as:

$$\frac{\partial \mathcal{L}_{kd}}{\partial \theta_{M^t}} = \frac{-1}{|D_t|} \sum_{\mathbf{X}^t \in D_t} \sum_{c}^{C^{1:t-1}} \frac{M^{t-1}(\mathbf{X}^t)_{[c]}}{M^t(\mathbf{X}^t)_{[c]}} \cdot \frac{\partial M^t(\mathbf{X}^t)_{[c]}}{\partial \theta_{M^t}} \tag{4}$$

It is important to note that $M^{t-1}(\mathbf{X}^t)$ remains constant, as M^{t-1} is held fixed. According to Eq. (4), the influence of class c in the knowledge distillation loss during backpropagation is proportional to the predicted value of class c in $M^{t-1}(\mathbf{X}^t)$. Old classes with lower predictions exert weaker constraints on the model parameters θ_{M^t}. As a result, these classes are more prone to being forgotten as the model learns new classes.

Building on the preceding analysis, it is theoretically more challenging to retain the old knowledge for additional data featuring infrequent occurrences of class c and smaller predicted values, which is consistent with Fig. 1.

Adversarial Attack for Added Images

Target attack. Contrary to traditional adversarial attacks, our approach diverges by aiming to prompt the old model

to produce predictions containing more forgettable classes through image perturbations. To achieve this, we employ multiple iterations to generate adversarial examples $\mathbf{A}_k^t$:

$$\mathbf{A}_k^t = \phi^\epsilon(\mathbf{A}_{k-1}^t - \alpha \cdot \text{sign}(\nabla_{\mathbf{A}_{k-1}^t} \mathcal{L}_{ce}(\mathbf{Z}_{k-1}^t, \mathbf{Y}^{goal}))) \quad (5)$$

where α and ϵ represent the step size and perturbation range, respectively. $\mathbf{Z}_{k-1}^t$ is the prediction map of the old model, i.e., $\mathbf{Z}_{k-1}^t = M^{t-1}(\mathbf{A}_{k-1}^t)$. The function ϕ^ϵ is used to clip the generated example within the ϵ-ball. $\mathbf{A}_k^t$ denotes the adversarial example at the k-th attack step. The initial value is set as $\mathbf{A}_0^t = \mathbf{X}^t + \mathbf{U}(-\epsilon, +\epsilon)$, where $\mathbf{U}(-\epsilon, +\epsilon)$ denotes the random initialization of the perturbation for each pixel, ranging from $-\epsilon$ to $+\epsilon$.

Attack probability module. Subsequently, to define the attack goal, we use the attack probability module to generate test images via an untargeted attack. This module assesses the class distribution in the attack target by computing the predicted difference between the old and new models.

Specifically, we need to define the degree of forgetfulness d_c of each class in training to keep adjusting this value. The knowledge distillation loss can only compute the difference between the two model outputs as a whole. Therefore, we utilize each channel of the prediction to compute the difference. The added images may contain only some of the old classes, making it difficult to fully assess the degree of forgetting for all classes. Therefore, we attack the added images to serve as test images $\mathbf{I}_k^t$ by the untargeted attack:

$$\mathbf{I}_k^t = \phi^\epsilon(\mathbf{I}_{k-1}^t + \alpha \text{sign}(\nabla_{\mathbf{I}_{k-1}^t} \mathcal{L}_{ce}(\mathbf{F}_{k-1}^t, \tilde{\mathbf{Y}}^{1:t-1}))) \quad (6)$$

where $\mathbf{F}_{k-1}^t = M^{t-1}(\mathbf{I}_{k-1}^t)$, $\tilde{\mathbf{Y}}^{1:t-1}$ is the pseudo-labels of $\mathbf{X}^t$ output by M^{t-1}, and α is the step size.

The degree of forgetfulness d_c of each class is captured by taking the square root of the predictions made by both the old and new models for each class:

$$d_c = \sqrt{M^t(\mathbf{I}_k^t)_{[c]} - M^{t-1}(\mathbf{I}_k^t)_{[c]}} \quad (7)$$

where $M^t(\mathbf{I}_k^t)_{[c]}$ means the prediction of the c-th channel.

The frequency of attacks for each class is:

$$\hat{d}_c = \frac{d_c}{\sum_{c=1}^{C} d_c} \quad (8)$$

Then, we set the label $\mathbf{Y}^{goal}$ to contain all observed categories, with the number of occurrences of each category assigned the value N_c according to $\hat{d}_c$.

$$N_c = B * H * W * \hat{d}_c \quad (9)$$

where B is the batch size.

Generating goal label. Specifically, the pseudo-label $\tilde{\mathbf{Y}}^{1:t-1}$ of the original data $\mathbf{X}^t$ is obtained according to the model M^{t-1}. Then, we calculate the counter of occurrence for all old classes, denoted as S_c:

$$S_c = \sum \mathbb{1}(\tilde{\mathbf{Y}}^{1:t-1} = c) \quad (10)$$

Based on the relationship between S_c and N_c, we distribute pixels by the following strategy:

- When S_c exceeds N_c, we randomly select N_c pixels from the pool of S_c, and allocate them the class c label, while the remaining $(S_c - N_c)$ pixels are placed in set A.
- When S_c is less than N_c, we initially utilize all S_c pixels. Following that, we select the additional $(N_c - S_c)$ pixels from set A, and allocate them the class c label.

The pixels selected for each class form the set P_c. And, $\mathbf{Y}^{goal}$ is defined as:

$$\forall c \in C^{1:t-1}, \forall (i,j) \in P_c : \mathbf{Y}_{[i,j]}^{goal} = c \quad (11)$$

where $\mathbf{Y}_{[i,j]}^{goal}$ is the label of the pixel (i,j).

In this process, we derive the goal label $\mathbf{Y}^{goal}$, adjusting the number of pixels per class according to the degree of forgetfulness of each class.

Training with Clean and Adversarial Data

To mitigate the background shift, the previous model trained at step $t-1$ is employed to generate the pseudo-label $\tilde{\mathbf{Y}}^{1:t-1}$, which contains all seen classes. It is combined with the label $\mathbf{Y}^t$ to generate a new pseudo-label $\tilde{\mathbf{Y}}^{1:t}$, which includes the current class and all seen classes. The pseudo-label $\tilde{\mathbf{Y}}^{1:t}$ is:

$$\tilde{\mathbf{Y}}_{[i,j]}^{1:t} = \begin{cases} \mathbf{Y}_{[i,j]}^t, & \text{if } \mathbf{Y}_{[i,j]}^t \neq c_b \\ \tilde{\mathbf{Y}}_{[i,j]}^{1:t-1}, & \text{if } \mathbf{Y}_{[i,j]}^t = c_b \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

where c_b is the background, and $\mathbf{Y}_{[i,j]}^t$ is the ground truth of the pixel (i,j) at step t.

For adversarial data, we utilize M^{t-1} to obtain the corresponding labels $\tilde{\mathbf{Y}}_{adv}^{1:t-1}$.

The final loss function is:

$$\begin{aligned} \mathcal{L}_{obj} =& (\mathcal{L}_{ce}(\mathbf{X}^t, \tilde{\mathbf{Y}}^{1:t}) + \mathcal{L}_{kd}(\mathbf{X}^t)) \\ &+ \lambda_{adv}(\mathcal{L}_{ce}(\mathbf{A}_k^t, \tilde{\mathbf{Y}}_{adv}^{1:t-1}) + \mathcal{L}_{kd}(\mathbf{A}_k^t)) \end{aligned} \quad (13)$$

where $[\mathbf{X}^t, \mathbf{A}_k^t]$ is a mini-batch, $[\tilde{\mathbf{Y}}^{1:t}, \tilde{\mathbf{Y}}_{adv}^{1:t-1}]$ means the pseudo-labels, and λ_{adv} is the weight of adversarial samples.

Experiments

Experimental Setup

Datasets. We validate our method on benchmark datasets Pascal VOC2012 (Everingham et al. 2010) and ADE20k (Zhou et al. 2017). The Pascal VOC2012 dataset contains 20 object classes and one background. Its training and validation sets include 10,582 and 1,449 images, respectively. The ADE20k dataset contains 150 objects with 20,210 training images and 2,000 test images.

Experimental setting. Initially, MiB (Cermelli et al. 2020) sets two different experimental settings, namely disjoint and overlapped. Following previous works (Douillard et al. 2021; Phan et al. 2022), we evaluate the performance of the model in the overlapped setting, as it is more realistic and challenging. For the Pascal VOC2012 dataset, we perform experiments in six settings, including adding 1 class

Method	19-1 (2 tasks)				15-5 (2 tasks)				15-1s (6 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT†	67.75	10.88	65.05	71.23	67.08	39.23	60.45	70.37	8.75	7.99	8.56	40.16
MiB†	70.57	22.82	68.30	72.95	75.30	48.68	68.96	75.07	39.47	14.50	33.53	54.44
REMINDER	76.48	32.34	74.38	76.22	76.11	50.74	70.07	75.36	68.30	27.23	58.52	68.27
RCIL‡	76.48	35.36	74.52	76.35	**78.66**	52.12	72.35	76.57	71.53	22.53	59.87	69.82
RECALL	67.90	**53.50**	68.40	-	66.60	50.90	64.00	-	65.70	**47.80**	62.70	-
LAG	-	-	-	-	77.33	51.76	71.24	-	**75.00**	37.52	66.08	-
PLOP†	75.50	30.22	73.35	75.43	75.44	49.65	69.30	74.82	63.41	26.76	54.68	66.96
AAKR	**77.95**	53.40	**76.78**	**77.34**	78.00	**54.52**	**72.41**	**76.67**	74.33	40.88	**66.37**	**72.06**

Table 1: mIoU for different continual learning settings on Pascal VOC2012. Herein, best results are marked in **boldface**, and second best results are underlined. †: results excerpted from (Phan et al. 2022). ‡: results comes from re-implementation.

Method	10-10 (2 tasks)				10-5s (3 tasks)				10-1s (11 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT‡	70.82	63.52	67.34	73.94	55.59	47.67	51.82	66.45	16.98	7.27	3.77	5.60
MiB‡	70.51	63.73	67.28	73.91	56.99	51.47	54.36	68.28	20.02	20.11	20.06	39.14
RCIL‡	73.98	65.34	69.87	75.22	61.11	**55.74**	58.55	**71.36**	55.44	15.03	36.20	47.37
RECALL	65.00	58.40	63.10	-	60.80	52.90	58.40	-	59.50	**46.70**	54.80	-
LAG	-	-	-	-	-	-	-	-	**69.56**	42.62	**56.73**	-
PLOP‡	73.82	63.55	68.93	74.81	58.58	53.66	56.24	69.89	44.95	15.43	30.89	44.77
AAKR	**74.57**	**65.58**	**70.29**	**75.41**	**66.52**	51.97	**59.59**	70.24	58.39	36.19	47.82	**58.22**

Table 2: mIoU for different continual learning settings on Pascal VOC2012.

after training 19 classes (19-1), adding 5 classes after training 15 classes (15-5), adding 5 classes *sequentially* after training 15 classes (15-1s), and more challenging settings of 10-10, 10-5s, and 10-1s. For the ADE20k dataset, we perform experiments in four settings, which are 100-50, 50-50s, 100-10s and 100-5. We compare the experimental results of AAKR with state-of-the-art methods, including distillation-based methods (ILT (Michieli and Zanuttigh 2019), MiB (Cermelli et al. 2020), PLOP (Douillard et al. 2021), REMINDER (Phan et al. 2022), RCIL (Zhang et al. 2022), LAG(Yuan, Zhao, and Shi 2024)) and replay-based method (RECALL (Maracani et al. 2021)). Note that all methods do not use any past samples during training.

Metrics. For semantic segmentation, the mean Intersection over Union (mIoU) metric is frequently used to measure the performance. In CSS, we report four different mIoUs. First, the mIoU of all initial classes (old) is used to indicate the ability of the model to retain the old knowledge. Second, the mIoU of all incremental classes (new) is used to indicate the ability of the model to learn new knowledge. Then, the mIoU of all classes (all) shows the combination performance of the model. Finally, the average value of mIoU (avg) evaluates the performance of the model throughout the continual training.

Implementation details. We use the same distillation and cross-entropy strategies as PLOP. As in previous work, we use Deeplabv3 (Chen et al. 2017) as the segmentation network with ResNet-101 (He et al. 2016) as the backbone, which is pre-trained on ImageNet (Deng et al. 2009). For Pascal VOC2012 and ADE20k datasets, the model is trained with a crop size of 512×512. The model is trained for 30 epochs on Pascal VOC2012 and 60 epochs on ADE20k, respectively. Moreover, λ_{adv} is 0.5. For the adversarial attack of added images, ϵ is 64, attack step k is 3, and α is ϵ/k.

Quantitative Evaluation

For the Pascal VOC2012 dataset, Tab. 1 shows the results for the 19-1, 15-5, and 15-1s settings. In the 19-1 setting, AAKR obtains advanced results for all classes (76.78%) and achieves a significant improvement on the new classes (+23.18%) compared with the baseline, PLOP. Moreover, for the 15-5 setting, AAKR improves the mIoU on both new and all classes (+4.87% and +3.11%, respectively) compared with PLOP. For the longer setting (15-1s), there is a significant performance degradation for each method. Compared with the baseline, AAKR achieves a larger improvement (+14.12%) in the new classes and retains the old knowledge well (74.33%). Besides, AAKR also obtains a better result (66.37%) for all classes than the replay-based method (RECALL). In addition, Tab. 2 shows the results for the 10-10, 10-5s, and 10-1s settings. Compared with PLOP, the mIoU of all classes is improved by +1.36%, +3.35%, and +16.93% at 10-10, 10-5s and 10-1s settings, respectively. RECALL excels in the longer setting (10-1s) but falls short in others, requiring extra model storage. In contrast, AAKR achieves remarkable performances across various settings.

For the ADE20k dataset, Tab. 3 shows the results for the 100-50, 50-50s, and 100-10s settings. For the 100-50 setting, AAKR improves the mIoU of all classes (+0.66%), compared with the advanced method RCIL. Besides, AAKR improves the baseline by +2.42% in the all classes. For the 50-50s setting, AAKR obtains an mIoU of 33.84% for all

Method	100-50 (2 tasks)				50-50s (3 tasks)				100-10s (6 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT†	18.29	14.40	17.00	29.42	3.53	12.85	9.70	30.12	0.11	3.06	1.09	12.56
MiB†	40.52	17.17	32.79	37.31	45.57	21.01	29.31	38.98	38.21	11.12	29.24	35.12
REMINDER	41.55	19.16	34.14	38.43	47.11	20.35	29.39	39.26	38.96	21.28	33.11	37.47
RCIL	**42.30**	18.80	34.50	-	48.30	24.40	32.50	-	39.30	17.50	32.10	-
LAG	41.64	19.73	34.34	-	47.69	26.12	33.31	-	**41.00**	18.69	33.56	-
PLOP†	41.76	14.52	32.74	37.73	47.33	20.27	29.41	38.75	38.59	14.21	30.52	34.48
AAKR	41.98	**21.39**	**35.16**	**38.78**	**48.73**	**26.25**	**33.84**	**40.97**	40.05	**21.66**	**33.96**	**37.92**

Table 3: mIoU for different continual learning settings on ADE20k.

Method	100-5s (11 tasks)			
	old	*new*	*all*	*avg*
MiB†	36.01	5.66	25.96	32.69
REMINDER	36.06	16.38	29.54	36.49
RCIL	38.50	11.50	29.60	-
LAG	**39.96**	17.22	**32.38**	-
PLOP†	35.72	12.18	27.93	35.10
AAKR	37.92	**18.27**	31.41	**37.01**

Table 4: mIoU for the setting 100-5s (11 tasks) on ADE20k.

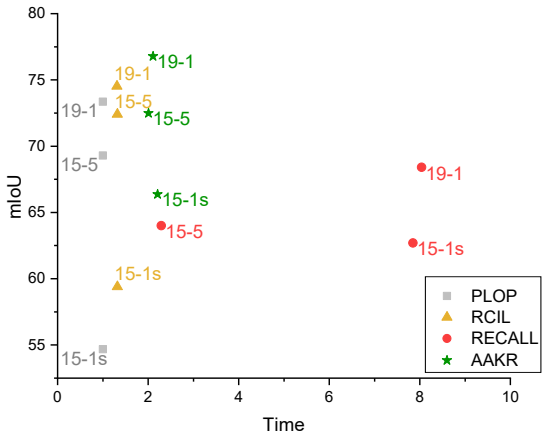

Figure 3: Comparisons of the training time on Pascal VOC2012.

classes, which is 4.43% higher than PLOP. Moreover, for the 100-10s setting, AAKR preserves the old knowledge well, improving +7.45% on new classes, compared with PLOP, and again obtaining a state-of-the-art result of 33.96% for all classes. Tab. 4 compares the performance of models in the longer setup of 100-5s. In such a setting, all models are highly susceptible to forgetting the old knowledge due to more learning steps. As shown in the table, AAKR is more effective in learning new classes with an mIoU of 18.27%.

Comparisons of Training Time

To demonstrate that AAKR reduces computational overhead compared to replay-based techniques, we conduct experiments on the training time, as shown in Fig. 3. The training time of PLOP is taken as a reference, normalized to 1, and used to evaluate the training time of other methods. Notably, AAKR outperforms knowledge distillation-based methods (PLOP and RCIL) with only a slight increase in training time. In the more challenging setting (15-1s), the replay-based technique (RECALL) outperforms both PLOP and RCIL, but with a substantial increase in training time (approximately 8-fold). In contrast, AAKR achieves superior outcomes compared to them with minor time investment. Experimental results show that AAKR accomplishes higher performance with little cost compared to replay-based strategy, while eliminating the need for additional model storage.

Method	*old*	*new*	*all*
Baseline	63.41	26.76	54.68
PGD	73.02	20.72	60.57
SegPGD	73.21	21.14	60.81
AAKR	**74.33**	**40.88**	**66.37**

Table 5: Ablation study of different adversarial attacks on the 15-1s setting of the Pascal VOC2012 dataset.

Method	*old*	*new*	*all*
Baseline	63.41	26.76	54.68
+attack	72.06	33.87	62.97
+APM	**74.33**	**40.88**	**66.37**

Table 6: Ablation study of different components for adversarial images on the 15-1s setting of the Pascal VOC2012 dataset. APM means the attack probability module.

Ablation Study

To validate the effectiveness of AAKR in generating adversarial images, we conduct experiments on the 15-1s setting of the Pascal VOC2012 dataset, as shown in Tab. 5. PLOP is used as the baseline. Additionally, we compare AAKR with PGD and SegPGD, which utilize adversarial attacks to manipulate the added data. The competitors primarily aim to induce differences in model predictions without yielding substantial improvements. In contrast, AAKR strategically compels the old model to produce comprehensive predictions, leading to the most promising results.

To verify the effect of the proposed components, we conduct experiments on the 15-1s setting of the Pascal VOC2012 dataset, as shown in Tab. 6. We begin with

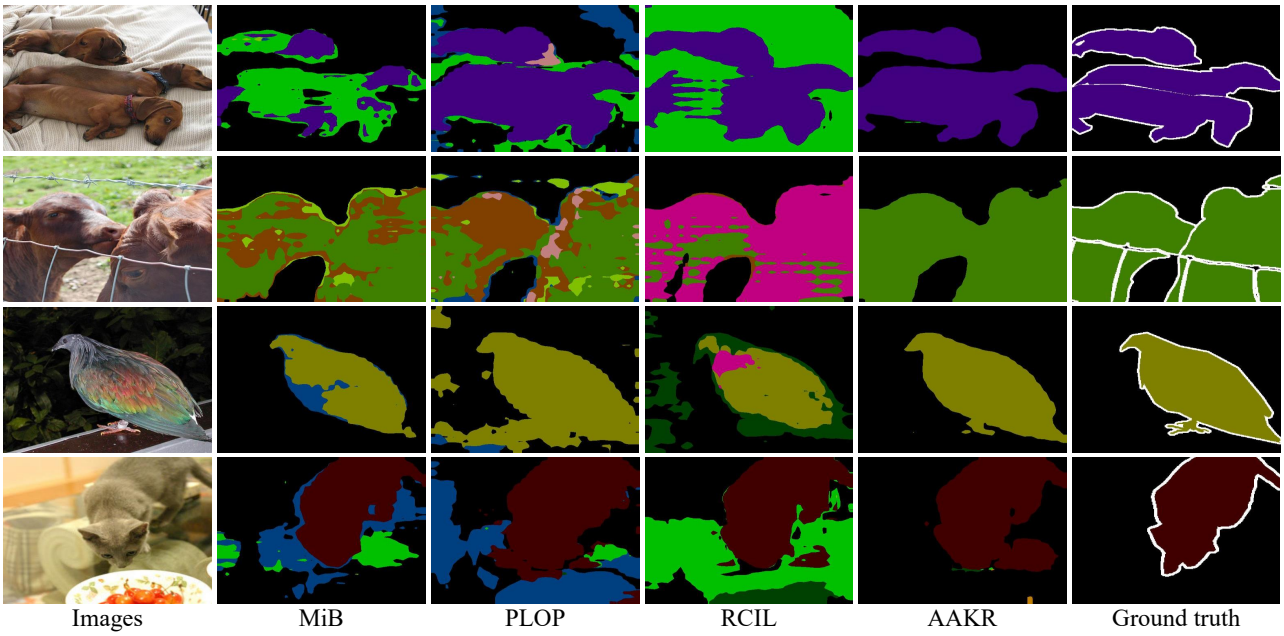

Figure 4: Visualization results of MiB, PLOP, RCIL and AAKR for some test images on Pascal VOC2012 in the 15-1s setting. AAKR has less confusion between the background and foreground classes, compared with MiB, PLOP and RCIL.

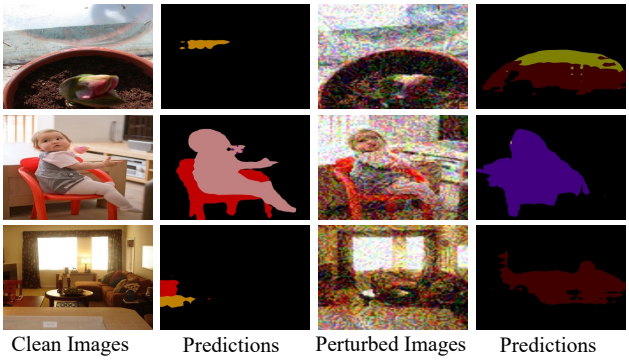

Figure 5: Visualization results of the clean and adversarial images and the corresponding predictions on Pascal VOC2012 in 15-5 setting.

the baseline by using PLOP. Subsequently, we enhance the knowledge transfer by generating adversarial samples through the proposed attack. The goal label $\mathbf{Y}^{goal}$ for the attack is randomly selected. Building upon the baseline, this approach significantly improves results (+8.29%) for the all classes. Furthermore, we introduce the attack probability module (APM) to construct $\mathbf{Y}^{goal}$ that constitutes the AAKR, further refining the impact (+3.40%). These experiments demonstrate the effectiveness superiority of the proposed components.

Qualitative Evaluation

Fig. 4 illustrates the predictions of MiB, PLOP, RCIL, and AAKR on the 15-1s setting. MiB, PLOP and RCIL exhibit instances of unreasonable predictions for specific pixels, frequently resulting in confusion between background and foreground classes, as well as among different foreground classes. For instance, in rows 1 and 2, there is evident confusion between various foreground classes, while rows 3 and 4 demonstrate confusion between background and foreground classes. In contrast, AAKR mitigates this issue by transferring the prior knowledge of the adversarial images.

As depicted in Fig. 5, we present the predictions of the old model for both clean and adversarial images on the 15-5 setting of the Pascal VOC2012 dataset. Notably, the old model generates predictions encompassing a broader spectrum of categories for the adversarial image as opposed to the original image. This expansion in predicted categories proves beneficial in facilitating the transfer of more comprehensive knowledge to the new model.

Conclusions

In this paper, we explored the impact of old model predictions on continual learning within the CSS framework, emphasizing the crucial role of higher occurrences and predictions in facilitating the knowledge retention. Building on these insights, we proposed the Adversarial Attack-based Knowledge Retention (AAKR) method. AAKR assigns larger weights to classes forgotten faster, achieved through an attack probability module. This strategic weighting prompts more attacks on these classes, leading the old model to generate additional relevant pseudo-labels or higher predictions. Accordingly, this augmentation enhances the transfer of old knowledge. Extensive experiments on benchmark datasets consistently validate the superior performance of AAKR over existing methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62172385), the Natural Science Foundation of Jiangsu Province (BK20241819), and the Innovation Program for Quantum Science and Technology (No. 2021ZD0302900), the Jiangsu Province Science Foundation for Youths (BK20240463), in part by the Laboratory for Advanced Computing and Intelligence Engineering Fund, in part by the Xiaomi Young Talents Program, and in part by the China Postdoctoral Science Foundation (2024M753115).

References

Agnihotri, S.; and Keuper, M. 2023. CosPGD: a unified white-box adversarial attack for pixel-wise prediction tasks. *arXiv preprint arXiv:2302.02213*.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, 233–248.

Cermelli, F.; Mancini, M.; Bulo, S. R.; Ricci, E.; and Caputo, B. 2020. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9233–9242.

Cha, S.; Yoo, Y.; Moon, T.; et al. 2021. SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning. *Advances in Neural Information Processing Systems*, 34: 10919–10930.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Cong, W.; Cong, Y.; Dong, J.; Sun, G.; and Ding, H. 2023. Gradient-semantic compensation for incremental semantic segmentation. *IEEE Transactions on Multimedia*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dhar, P.; Singh, R. V.; Peng, K.-C.; Wu, Z.; and Chellappa, R. 2019. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5138–5146.

Douillard, A.; Chen, Y.; Dapogny, A.; and Cord, M. 2021. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4040–4050.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 86–102. Springer.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Goswami, D.; Schuster, R.; van de Weijer, J.; and Stricker, D. 2023. Attribution-aware weight transfer: A warm-start initialization for class-incremental semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 3195–3204.

Gu, J.; Zhao, H.; Tresp, V.; and Torr, P. H. 2022. Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In *European Conference on Computer Vision*, 308–325. Springer.

He, J.; Mao, R.; Shao, Z.; and Zhu, F. 2020. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13926–13935.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 831–839.

Hu, Z.; Li, Y.; Lyu, J.; Gao, D.; and Vasconcelos, N. 2023. Dense network expansion for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11858–11867.

Iscen, A.; Zhang, J.; Lazebnik, S.; and Schmid, C. 2020. Memory-efficient incremental learning through feature adaptation. In *European conference on computer vision*, 699–715. Springer.

Kanakis, M.; Bruggemann, D.; Saha, S.; Georgoulis, S.; Obukhov, A.; and Gool, L. V. 2020. Reparameterizing convolutions for incremental multi-task learning without task interference. In *European Conference on Computer Vision*, 689–707. Springer.

Kang, M.; Park, J.; and Han, B. 2022. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16071–16080.

Liu, Y.; Parisot, S.; Slabaugh, G.; Jia, X.; Leonardis, A.; and Tuytelaars, T. 2020. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, 699–716. Springer.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 67–82.

Maracani, A.; Michieli, U.; Toldo, M.; and Zanuttigh, P. 2021. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7026–7035.

Meng, Z.; Zhang, J.; Yang, C.; Zhan, Z.; Zhao, P.; and Wang, Y. 2025. Diffclass: Diffusion-based class incremental learning. In *European Conference on Computer Vision*, 142–159. Springer.

Michieli, U.; and Zanuttigh, P. 2019. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Michieli, U.; and Zanuttigh, P. 2021. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1114–1124.

Park, G.; Moon, W.; Lee, S.; Kim, T.-Y.; and Heo, J.-P. 2025. Mitigating Background Shift in Class-Incremental Semantic Segmentation. In *European Conference on Computer Vision*, 71–88. Springer.

Phan, M. H.; Phung, S. L.; Tran-Thanh, L.; Bouzerdoum, A.; et al. 2022. Class Similarity Weighted Knowledge Distillation for Continual Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16866–16875.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Rony, J.; Pesquet, J.-C.; and Ben Ayed, I. 2023. Proximal splitting adversarial attack for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20524–20533.

Toldo, M.; Michieli, U.; and Zanuttigh, P. 2024. Learning with style: Continual semantic segmentation across tasks and domains. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wu, C.; Herranz, L.; Liu, X.; van de Weijer, J.; Raducanu, B.; et al. 2018. Memory replay gans: Learning to generate new categories without forgetting. *Advances in Neural Information Processing Systems*, 31.

Xiao, J.-W.; Zhang, C.-B.; Feng, J.; Liu, X.; van de Weijer, J.; and Cheng, M.-M. 2023. Endpoints weight fusion for class incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7204–7213.

Yang, G.; Fini, E.; Xu, D.; Rota, P.; Ding, M.; Tang, H.; Alameda-Pineda, X.; and Ricci, E. 2022. Continual attentive fusion for incremental learning in semantic segmentation. *IEEE Transactions on Multimedia*, 25: 3841–3854.

Yuan, B.; Zhao, D.; and Shi, Z. 2024. Learning At a Glance: Towards Interpretable Data-Limited Continual Semantic Segmentation Via Semantic-Invariance Modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 3987–3995. PMLR.

Zhang, C.-B.; Xiao, J.-W.; Liu, X.; Chen, Y.-C.; and Cheng, M.-M. 2022. Representation Compensation Networks for Continual Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7053–7064.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhu, L.; Chen, T.; Yin, J.; See, S.; and Liu, J. 2023. Continual semantic segmentation with automatic memory sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3082–3092.