Attacks on Continual Semantic Segmentation by Perturbing Incremental Samples

Zhidong Yu[1], Wei Yang[1,2*], Xike Xie[1,3*], Zhenbo Shi[1,3]

[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
[2]Hefei National Laboratory, Hefei 230088, China
[3]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China
qubit@ustc.edu.cn

Abstract

As an essential computer vision task, Continual Semantic Segmentation (CSS) has received a lot of attention. However, security issues regarding this task have not been fully studied. To bridge this gap, we study the problem of attacks in CSS in this paper. We first propose a new task, namely, attacks on incremental samples in CSS, and reveal that the attacks on incremental samples corrupt the performance of CSS in both old and new classes. Moreover, we present an adversarial sample generation method based on class shift, namely Class Shift Attack (CS-Attack), which is an offline and easy-to-implement approach for CSS. CS-Attack is able to significantly degrade the performance of models on both old and new classes without knowledge of the incremental learning approach, which undermines the original purpose of the incremental learning, i.e., learning new classes while retaining old knowledge. Experiments show that on the popular datasets Pascal VOC, ADE20k, and Cityscapes, our approach easily degrades the performance of currently popular CSS methods, which reveals the importance of security in CSS.

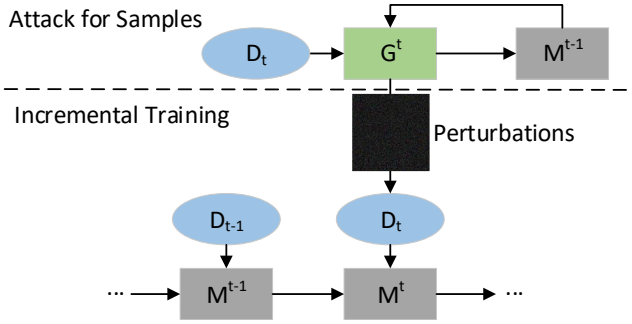

Figure 1: Overview of adversarial attacks on incremental samples. In this case, the attack on the incremental samples is separate from the incremental training. The attack phase occurs before the incremental training, using only the old training model and the new data D_t. The incremental model forgets the old knowledge due to the interference of the training data.

Introduction

Semantic segmentation is a crucial computer vision task extensively applied in diverse real-world scenarios (Siam et al. 2018; Milioto, Lottes, and Stachniss 2018; Asgari Taghanaki et al. 2021). Recently, numerous models (Shelhamer, Long, and Darrell 2017; Chen et al. 2017; Cheng, Schwing, and Kirillov 2021) have been developed to tackle this task, displaying encouraging outcomes. However, these models encounter a significant hurdle known as the catastrophic forgetting (Michieli and Zanuttigh 2019) in the scenario of continual learning. In other words, the network learns new classes while rapidly forgetting those it has already acquired.

The continual semantic segmentation (CSS) task was originally proposed by Michieli et al. (Michieli and Zanuttigh 2019). After that, some methods are proposed to solve this task with better results. A number of works (Cermelli et al. 2020; Douillard et al. 2021; Michieli and Zanuttigh 2021a,b; Phan et al. 2022) address the catastrophic forgetting of this task by distilling the knowledge of the old model to the new one. For example, MiB (Cermelli et al. 2020) takes into account the problem of background bias in CSS and models it to alleviate the confusion of new classes and the old knowledge. Moreover, there are works (Cha et al. 2021; Zhang et al. 2022) that utilize other techniques to retain the old classes, such as saliency detection and model compression.

However, with the rapid development of CSS, a potential risk in this task remains neglected. It is well known that the standard CSS task is to update the model by incremental samples to learn new classes and retain old knowledge. These incremental samples may be provided by third parties, and thus they can be exploited to corrupt the old knowledge of the model for attack purposes. Specifically, it is feasible to corrupt the performance of the model by perturbing the incremental samples so that the predictions of certain pixels deviate from the previous model.

Adversarial attacks against incremental classification were first proposed by Han et al. (Han et al. 2022). It assumes that the losses of the model in incremental training can be obtained. Based on this, adversarial samples are generated in real-time to attack the model, which is an online attack. Unlike it, we first consider the adversarial attack in CSS. Furthermore, we propose a more difficult and practical

*Corresponding Authors.

setting. Specifically, the training process of the incremental model is difficult to obtain, so we consider the CSS attack method in an offline mode. The settings are as follows: 1) given incremental data D_t without any old data D_{t-1}, 2) the prediction map of the trained model at step $t-1$ on the sample is available, and 3) the specific structure and parameters of the model and the incremental learning paradigm are unknown.

Under the proposed settings, we propose an attack method for incremental data in CSS, which aims to make the incremental model forget the learned knowledge quickly when trained on the perturbed incremental data. Fig. 1 illustrates the attack at step t without knowing the incremental learning paradigm and the model structure. The attack model G_t is first trained on the data D_t and the old segmentation model. Then, G_t generates perturbations for D_t. The disturbed D_t crashes the incremental training. In addition, we propose a loss based on class shift. It uses adversarial attacks to apply perturbations to the incremental data so that the old model produces the same prediction for all pixels on the image, limiting the exploitation of old class knowledge.

Extensive experiments demonstrate the destruction of CS-Attack on CSS, proving that the learned knowledge is quickly forgotten due to incremental data being attacked. This reminds us that it is crucial to consider the attack on incremental data streams when designing CSS schemes.

Our main contributions can be summarized as follows:

- We reveal for the first time the potential risk in CSS and propose a novel task, namely, the attack against incremental samples in CSS.
- We propose an attack method, namely CS-Attack, for incremental data that uses class shift to guide the generation of samples.
- We conduct extensive experiments at multiple incremental settings on the standard benchmarks, and the proposed method substantially reduces the performance of CSS on old and new classes.

Related Work

Class Incremental Learning

Concerns surrounding continual learning, also termed incremental or lifelong learning, have been steadily growing. Previous works are divided into three main categories: regularization-based, replay-based, and parameter isolation-based. Regularization-based methods (Zenke, Poole, and Ganguli 2017; Dhar et al. 2019; Douillard et al. 2020) can be subdivided into two categories: data-focused and prior-focused. The former utilizes techniques like distillation (Hinton et al. 2015) to generate an additional loss that acts as a regularization constraint to prevent forgetting. The latter preserves acquired knowledge by controlling the variation of parameters with differing levels of importance. Replay-based methods (Rebuffi et al. 2017; Castro et al. 2018; Hou et al. 2019; Iscen et al. 2020) select or generate examples of previous steps, which the model incorporates alongside new data to learn the updated classes. Then, the model employs these examples along with the new data to learn the new classes. Parameter isolation-based methods (Mallya, Davis, and Lazebnik 2018; Liu et al. 2020) allocate an independent set of model parameters to each task, aiming to forestall forgetting.

Class Incremental Semantic Segmentation

Michieli et al. (Michieli and Zanuttigh 2019) propose continual semantic segmentation and put forward a general framework to retain old knowledge through knowledge distillation. Subsequently, MiB (Cermelli et al. 2020) initially highlights the background shift in CSS, addressing it by modeling the background to alleviate transfer issues. PLOP (Douillard et al. 2021) introduces Local POD, preserving both long and short-distance spatial relationships at the feature level. SDR (Michieli and Zanuttigh 2021a) uses prototype matching and contrast learning to construct robust features. The REMINDER (Phan et al. 2022) designs CSW-KD, adjusting the distillation weight of each class based on the similarity between new and old classes. Rong et al. (Rong et al. 2022) focus on utilizing historical information to guide class-incremental semantic segmentation in remote sensing images. Furthermore, several other approaches (Cha et al. 2021; Zhang et al. 2022) achieve promising results with additional models or structures. For instance, SSUL (Cha et al. 2021) relies on the saliency detection model to discover potential objects, which requires models trained on other datasets. On the other hand, RCIL (Zhang et al. 2022) utilizes parallel convolutions to enhance performance.

In this paper, we focus on the risk of attacks against incremental samples that are overlooked by the CSS schemes, and design CSS attacks to verify the feasibility of attacks against them.

Adversarial Attack

The adversarial attack (Goodfellow, Shlens, and Szegedy 2015) usually refers to the use of perturbed samples to cause machine learning models to make incorrect predictions. It is a way to generate such examples through various techniques. It can be divided into test-time adversarial attack methods (Goodfellow, Shlens, and Szegedy 2015; Carlini and Wagner 2017; Madry et al. 2018; Gu et al. 2022; Agnihotri and Keuper 2023) and training-time ones (Feng, Cai, and Zhou 2019). The former generates adversarial images during inference and confuses the model to produce false predictions. The latter generates images at training time to deviate the training of the model from expectations. In addition, (Han et al. 2022) explores an online attack method for incremental learning in classification for the first time. Unlike them, we explore the possibility of offline style attacks against training samples in CSS, i.e., attacks without knowledge of the incremental learning paradigm and the information in training.

Methodology

Problem Definition for CSS

Before introducing the problem, we first introduce some related concepts. The purpose of CSS is to train a segmentation model based on the data stream D_1 to D_t in T steps to

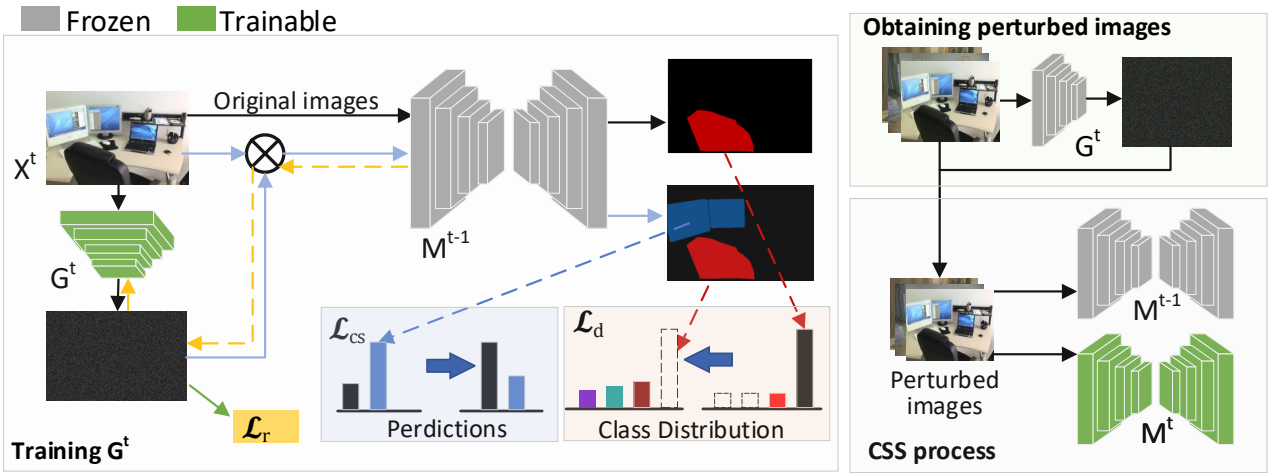

Figure 2: Overview of the CS-Attack. G_t is trainable, while the $t-1$ step trained model M_{t-1} is frozen. The attack model outputs the relevant perturbations based on input images, and M_{t-1} outputs the predictions of the original and perturbed samples, using the constructed losses to optimize G_t. The objective function contains three parts, $\mathcal{L}_{cs}$ and $\mathcal{L}_d$ are to destroy the prediction map of M_{t-1}, and $\mathcal{L}_r$ is to constrain the perturbed image to be close to the original image. After that the perturbed images are generated and passed to the model for CSS.

learn new classes without forgetting the old ones. We define that C^t is the class learned at step t, and $C^{1:t-1}$ denotes all the seen classes from step 1 to step $t-1$. For step t, we present a dataset D_t, which comprises a set of pairs (X^t, Y^t), where X^t is an image with a size of $H \times W$, and Y^t is the ground truth segmentation map, which contains only the background and the class C^t learned in the current step. Catastrophic forgetting means that the performance of the model on $C^{1:t-1}$ degrades rapidly while learning C^t.

Typically, the segmentation model at step $t-1$ is defined as M_{t-1}, which generates corresponding semantic segmentation prediction map with $|C^{1:t-1}|$ channels.

Adversarial Attack for Incremental Samples

As mentioned earlier, CSS is constantly learns new classes and retains old knowledge in an incremental data stream. Therefore, one possible attack is to add perturbations to the incremental data so that the model quickly forgets what it has learned during incremental training. We define this as a new problem, i.e., the adversarial attack on incremental samples. Attacks against incremental samples require us to generate adversarial samples to interfere with the training of the model, so that the incremental learning method fails on these samples, i.e., it cannot do the job of learning new classes while retaining old ones.

For this task, we consider a more difficult but easier to implement setup, i.e., generating perturbations for the current step of data without knowing the incremental training process, and thus the model quickly forgets the existing knowledge in the training of them. Specifically, the settings of this task are as follows:

1) *The incremental training process and the incremental model are agnostic, and information such as the loss generated by the model during training is not available.*

2) *Following the standard CSS task, only the data from the current step is available, all old data is not available.*

3) *The model trained at step $t-1$ is available, and its prediction map for the sample is available.*

In the following, we illustrate the feasibility of training the model with incremental samples to make it forget the old knowledge.

For the CSS approach, the model M_{t-1} from the previous step is usually used to provide the old knowledge contained in the picture and M_t is supervised using two losses, namely $\mathcal{L}_{learn}$ and $\mathcal{L}_{retain}$. $\mathcal{L}_{learn}$ is used to learn new classes from those regions that are labeled as new classes, and $\mathcal{L}_{retain}$ relies on M_{t-1} to retain knowledge from those regions that are labeled as background. Therefore, attacking the regions in the background so that M_{t-1} produces results that contradict the original prediction will destroy the old knowledge, i.e., it is feasible to train the model with incremental samples in order to make it forget the old knowledge. Note that we do not specify the type of losses, i.e., any loss function can be employed.

The Proposed Attack Paradigm

The proposed attack paradigm is shown in Fig. 2. Note that this process is independent of incremental training. In step t, the perturbation attack model G_t is trained using M_{t-1} and D_t. M_{t-1} is the segmentation model. M_{t-1} is frozen and does not update the parameters, but computes the gradient to train the model G_t.

$$r = G(X^t) \tag{1}$$

where X^t is an incremental sample of step t, whose label Y^t contains only the classes of C^t and the background.

The image being attacked $\hat{X}^t$ is defined as:

$$\hat{X}^t = X^t + r \tag{2}$$

where all attacked images $\hat{X}^t$ constitute the adversarial data $\hat{D}_t$ at step t.

We impose a constraint loss $\mathcal{L}_r$ between the original with perturbed images to avoid excessive image modification, which is calculated as:

$$\mathcal{L}_r = ||\hat{X}^t - X^t||_2 \tag{3}$$

where $|| * ||_2$ denotes the $L2$-norm.

Then, we train the generative model by creating constraints between the prediction maps of the perturbed image and the original image, and these maps are obtained by M_{t-1}. The goal is to make the generated disturbance aggressive and significantly reduce the performance of the model in incremental training.

For semantic segmentation, X^t contains rich information that can be categorized into two types: the information that has already been learned, and the information of all new classes. For the former, the CSS method uses them to mine old class information as a way to retain old knowledge. Therefore, we first construct a class shift loss. This loss, denoted as $\mathcal{L}_{cs}$, forces G_t to generate a perturbation r to confuse M_{t-1} and predict these locations as the background. The $\mathcal{L}_{cs}$ erases any useful information that might be contained in the original image, especially about the old classes. The prediction map of $\hat{X}^t$ generated by M_{t-1} is $P_{\hat{X}^t}$. The definition of $\mathcal{L}_{cs}$ is:

$$\mathcal{L}_{cs} = -\sum_{Y^t_{i,j} \notin C^t} y_b log(\hat{p}_{i,j}) \tag{4}$$

where $\hat{p}_{i,j}$ is the prediction vector of $P_{\hat{X}^t}$ at pixel (i,j), y_b is the one-hot label of the background, and $Y^t_{i,j}$ is the label of pixel (i,j). y_b is a vector of length $|C^{1:t-1}|$ with the dimension of the background being 1 and the other values being 0.

For the latter, the model needs to learn these classes. This process corrects the learned knowledge of the old model (which may be old or new classes) to the new classes. This part is usually supervised by the cross-entropy loss after softmax processing:

$$\mathcal{L}_{ce} = -\sum_{Y^t_{i,j} \in C^t} y_c log(p_{i,j}) \tag{5}$$

where y_c is the one-hot label of the new class in current step, and $p_{i,j}$ is the prediction vector of P_{X^t} at pixel (i,j).

According to the softmax function, the output of other old classes is suppressed when learning new classes. For these regions containing new classes, we wish to utilize the attack to change the output of the old model for it to be predicted as a non-background old class.

Therefore, we construct a disorder loss $\mathcal{L}_d$. It drives the prediction map outputted by M_{t-1} of perturbed images to deviate from the prediction map of the original images, thus corrupting the learning of new classes in incremental learning. $\mathcal{L}_d$ is defined as:

$$\mathcal{L}_d = \frac{1}{\sum_{Y^t_{i,j} \in C^t \cap \hat{Y}_{i,j} \neq bk} \mathcal{L}_{KL}(p_{i,j}, \hat{p}_{i,j})} \tag{6}$$

Algorithm 1: Attack process at step t

Input: Segmentation model M_{t-1} and incremental data D_t
Parameter: Attack model G_t and segmentation model M_t at step t
Output: Segmentation model M_t

while Current epochs less than total epochs **do**
  while $\{(X^t, Y^t)\} \in D_t$ **do**
    Get the Perturbations: $r = G(X^t)$
    Get the perturbed image with Eq. (2)
    Get original and perturbed predictions by M_{t-1}.
    Update G_t with Eq. (7)
  end while
end while
Generate the Perturbations r for $X^t \in D_t$ and construct Adversarial Data $\hat{D}_t$
Incremental Training for M_t with $\hat{D}_t$
return M_t

where $p_{i,j}$ is the prediction vector of P_{X^t} at pixel (i,j), $\hat{Y}_{i,j}$ is the pseudo labels of $P_{\hat{X}^t}$, and $\hat{Y}_{i,j} \neq bk$ means the predictions of the old model for $\hat{X}^t$ is not the background. $\mathcal{L}_{KL}$ is the KL-divergence loss.

Finally, the objective loss used to train the model G_t can be obtained:

$$\mathcal{L}_{obj} = \mathcal{L}_r + \mathcal{L}_d + \lambda \mathcal{L}_{cs} \tag{7}$$

where λ is the weighting factor, and the weights of the other two terms are 1.

Algorithm 1 shows the pseudocode for the attack process at step t. The proposed method uses the model from step $t-1$ and the data from step t to train G_t. After that, the trained G_t generates attack samples and submits them to the incremental model for learning. The algorithm is an offline and easy to implement method, which does not require real-time generation of adversarial data during incremental learning.

Experiments

Experimental Setup

Datasets. We validate CS-Attack on different standard semantic segmentation datasets: Pascal VOC2012 (Everingham et al. 2010), ADE20k (Zhou et al. 2017) and Cityscapes (Cordts et al. 2016). The Pascal VOC2012 dataset contains 20 object classes and the background. It includes 10,582 images for training and 1,449 images for validation, respectively. The ADE20k dataset contains 150 objects and includes 20,210 training images and 2,000 test images. The Cityscapes dataset contains 19 classes from 21 cities with 2,975 training images, 500 validation images and 1,525 test images.

Setting. MIB (Cermelli et al. 2020) sets two experimental protocols: disjoint and overlap. We consider the latter more realistic and challenging, and recent works mainly report their results in the overlapping setting. Therefore, we evaluate the performance in the overlapped setting for each dataset. We conduct experiments on Pascal VOC2012 in three settings: adding 1 class after training 19 classes (19-1), adding 5 classes after training 15 classes (15-5), and

Method	19-1 (2 tasks)			15-5 (2 tasks)			15-5s (6 tasks)		
	0-19 ↓	20 ↓	all ↓	0-15 ↓	16-20 ↓	all ↓	0-15 ↓	16-20 ↓	all ↓
ILT	67.71	11.65	65.04	67.14	39.24	60.50	8.74	7.94	8.55
+Noise	67.43	9.86	64.69	66.77	38.63	60.07	8.20	8.49	8.27
CS-Attack (ours)	**42.75**	**4.63**	**40.93**	**45.23**	**9.74**	**36.78**	**2.75**	**1.03**	**2.34**
MiB	70.29	33.28	68.53	75.32	48.79	69.00	39.41	14.71	33.53
+Noise	70.77	23.82	68.53	75.76	48.32	69.23	39.27	14.81	33.45
CS-Attack (ours)	**46.56**	**12.36**	**44.93**	**47.51**	**18.54**	**40.61**	**11.48**	**8.32**	**10.73**
PLOP	72.78	28.12	70.65	74.54	47.58	68.12	60.00	18.46	50.11
+Noise	72.77	30.39	70.75	73.71	48.45	67.70	60.00	17.18	49.80
CS-Attack (ours)	**55.02**	**21.74**	**53.43**	**61.07**	**30.30**	**53.74**	**36.21**	**4.35**	**28.62**
RCIL	77.00	25.11	74.53	78.95	50.66	72.21	69.91	22.82	58.70
+Noise	75.34	23.62	72.88	76.44	48.65	69.82	68.14	25.76	58.05
CS-Attack (ours)	**58.99**	**8.70**	**56.60**	**65.15**	**28.36**	**56.39**	**37.23**	**7.82**	**30.22**
Joint	77.45	77.94	77.47	78.88	72.63	77.39	78.88	72.63	77.39

Table 1: mIoU for different incremental learning settings on the dataset Pascal VOC2012. For each CSS method, noise is added to the samples (+Noise) and adversarial samples are generated (CS-Attack) in incremental learning. Best results for each CSS method are marked in boldface.

Method	11-5 all ↓	11-1s all ↓	1-1s all ↓
PLOP	61.52	58.46	45.14
+Noise	58.59	52.18	42.76
CS-Attack (ours)	**45.15**	**30.22**	**17.59**

Table 2: mIoU for different incremental learning settings on the dataset Cityscapes. mIoU of all classes after incremental training is reported.

adding 5 classes sequentially after training 15 classes (15-5s). For ADE20k, we perform experiments with four settings: adding 50 classes after training 100 classes (100-50), adding 50 classes each time after training 50 classes (50-50), and adding 10 classes each time sequentially after training 100 classes (100-10s). For Cityscapes, as in the previous work (Douillard et al. 2021), we treat the training data for each city as a class and apply three settings: adding 5 classes each time after training 11 classes (11-5), adding 5 classes each time sequentially after training 11 classes (11-1s), and adding one class at a time (1-1s).

Evaluation metrics. The mean Intersection over Union (mIoU) metric is frequently used to measure the performance of the model in semantic segmentation. And for a comprehensive evaluation, we report different mIoUs in CSS. Initially, the mIoU of all initial classes assesses the ability of the model to retain the old knowledge. Subsequently, the mIoU of all new classes indicates the ability of the model to acquire novel knowledge. Finally, the mIoU of all classes (all) evaluates the performance of the model.

Details. We validate the attack effectiveness of CS-Attack on several state-of-the-art CSS methods RCIL (Zhang et al. 2022), PLOP (Douillard et al. 2021), MIB (Cermelli et al. 2020) and ILT (Michieli and Zanuttigh 2019). All results are from the Deeplabv3 (Chen et al. 2017) architecture, which is pre-trained on ImageNet (Deng et al. 2009). There are no special requirements for the specific structure of the generative model G_t, and herein a simple encoder-decoder structure is used in our experiments. The encoder contains a 7×7 convolution with 64 channels and three 3×3 convolutions with 128, 256 and 512 channels, respectively. The decoder is composed of four 3×3 convolutions with 512, 256, 128 and 64 channels, respectively. The final output is a perturbation map r of size $H \times W$. It is worth noting that there is no attack method specifically set up for this task, hence we imposed additional Gaussian noise (+Noise) on the images as a comparison. Moreover, we use CS-Attack as a new baseline for this task. Our experiments are conducted on 4 NVIDIA 2080Ti GPUs. To train the model G_t, we use the stochastic gradient descent (SGD) optimizer, where the base learning rate is 0.01 for all datasets, and λ is 1 in our experiments. G_t is trained for 30 epochs on PASCAL VOC2012 and ADE20k datasets and 50 epochs on Cityscapes dataset. As for the incremental learning process, we follow the setup of the original works.

Quantitative Results

Results on the Pascal VOC2012 dataset. Tab. 1 shows the attack results of CS-Attack on the Pascal VOC dataset with different incremental settings. First, the existing CSS methods achieve satisfactory results on the PASCAL VOC2012 dataset, especially PLOP, and RCIL. Second, the application of Gaussian noise to the original image does not lead to a significant decrease in the effectiveness of these methods. This suggests that these methods have some robustness as they only learn the probability distribution of old knowledge instead of labels. In contrast, the proposed method has a significant impact on almost all of the CSS methods, and their performance decreases significantly. In particular, in the long-term incremental process, we considerably destroy their results on the old classes.

Results on the Cityscapes dataset. For Cityscapes, we report the results of the proposed attack on PLOP in different settings. As shown in Tab. 2, PLOP is a structure change-based approach that achieves promising results in various

Method	100-50 (2 tasks)			50-50 (3 tasks)			100-10s (6 tasks)		
	0-100 ↓	101-150 ↓	all ↓	0-50 ↓	51-150 ↓	all ↓	0-100 ↓	101-150 ↓	all ↓
ILT	18.46	17.02	17.99	2.93	11.82	8.82	0.45	0.98	0.63
+Noise	18.88	14.95	17.57	3.25	12.99	9.74	0.44	2.19	1.03
CS-Attack (ours)	**6.86**	**6.70**	**6.81**	**1.45**	**4.89**	**3.73**	**0.24**	**1.36**	**0.61**
MiB	40.81	18.96	33.57	45.90	21.64	29.84	37.97	12.40	29.50
+Noise	40.53	17.56	32.87	46.16	21.46	29.80	37.25	10.21	28.30
CS-Attack (ours)	**19.81**	**9.36**	**16.35**	**20.90**	**12.54**	**15.36**	**19.35**	**5.14**	**14.64**
PLOP	41.27	16.65	33.12	47.64	21.21	30.13	39.28	11.63	30.13
+Noise	41.63	14.32	32.59	46.74	21.07	29.74	39.49	13.75	30.97
CS-Attack (ours)	**27.84**	**4.28**	**20.04**	**23.81**	**12.49**	**16.31**	**21.04**	**7.45**	**16.54**
RCIL	40.68	19.97	33.82	47.23	18.93	28.49	38.05	22.33	32.84
+Noise	41.81	18.24	34.01	47.74	20.67	29.69	38.21	21.03	32.52
CS-Attack (ours)	**29.19**	**10.61**	**23.04**	**23.30**	**13.97**	**17.12**	**21.05**	**10.93**	**17.70**
Joint	44.34	28.21	39.00	51.21	32.77	39.00	44.34	28.21	39.00

Table 3: mIoU for different incremental learning settings on the dataset ADE20k.

$\mathcal{L}_d$	$\mathcal{L}_{cs}$	0-15 ↓	16-20 ↓	all ↓
		73.91	48.45	67.70
	✓	70.98	37.81	63.08
✓	✓	**61.07**	**30.30**	**53.74**

Table 4: Ablation study of different components on the 15-5 (2 tasks) setting of the Pascal VOC dataset. The CSS method is PLOP.

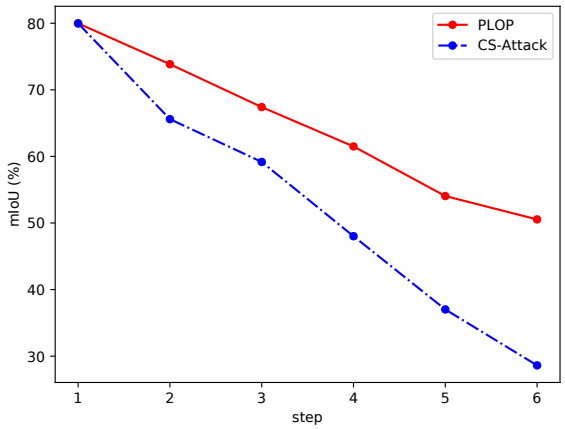

Figure 3: The mIoU (%) at each step in the PASCAL VOC2012 dataset with the 15-5s setting.

settings. Random noise causes some attack effects. In contrast, CS-Attack is still useful and substantially waves the performance of the model, especially for the training setting with more incremental steps (1-1s).

Results on the ADE20k dataset. Tab. 3 shows the results of attacking various CSS methods using CS-Attack under different settings on the ADE20k dataset. Firstly, the recently proposed methods perform well on this data. Similar to that on PASCAL VOC2012 dataset, the performance of the CSS methods is not significantly affected after imposing noise. This suggests that simple random noise does not affect performance. In contrast, our method plays a destructive role in all three different settings, and several CSS methods show drastic performance degradation. Moreover, the degradation is proportionally greater for the ADE20k dataset. This is due to the significantly larger incremental sample, which makes the attacked sample increase and the model corrupted to a greater extent.

Through the above experiments on multiple incremental methods on multiple datasets, we demonstrate that the existing CSS methods are vulnerable and can be easily corrupted by attacks targeting incremental samples.

Ablation Study

Effectiveness of different components. We evaluate the impact of the proposed modules, and the performance analysis is shown in Tab. 4. For a fair comparison, these experiments are performed on Pascal VOC2012 with the setting 15-5. The baseline is PLOP without other attacks, and the method achieves promising incremental learning. Then, we add adversarial attacks, and use $\mathcal{L}_d$ with $\mathcal{L}_r$ as supervision to generate adversarial samples. These samples are used for incremental learning, where the effect of PLOP drops significantly (-4.62%). After that, we introduce the class shift loss ($\mathcal{L}_{cs}$) in the adversarial phase, and the performance of PLOP decreases significantly (-9.34 %) after learning with perturbations. Integrating all the modules, PLOP finally decreases to 53.74%. This shows that CS-Attack is effective for the incremental sample attack of CSS, which defeats the original purpose of CSS.

Attack effect in different steps. To show the performance of CS-Attack at each step of the attack, we show in Fig. 3 the results of PLOP at each step on both the original incremental data and the data of the attack. They have the same performance before incremental learning. Then CS-Attack starts attacking the model at each incremental step. The introduction of CS-Attack corrupts the incremental results. As the incremental steps are gradually added, our method further destroys the performance of the model and

Figure 4: Visualization results of PLOP, with imposed noise (+Noise) and proposed adversarial attack (CS-Attack) on the dataset PASCAL VOC2012 with 15-5s settings on several test images. (a) Input image. (b) The baseline is PLOP. (c) The results of PLOP for incremental training using the attack samples generated by CS-Attack. (d) Hand-annotated labels.

S_1	S_2	S_3	S_4	S_5	all ↓
					54.68
✓					37.00
	✓				38.62
		✓			35.67
			✓		36.35
				✓	37.14
✓	✓	✓	✓	✓	**28.62** (-26.06)

Table 5: Ablation experiment for attacks on different incremental steps on the PASCAL VOC2012 dataset with 15-5s (6 tasks) settings.

the effect degrades even more.

Experimentation of different attack points. To visualize the impact of the attack on the incremental results in CSS, we show the results of the proposed attack on different incremental steps in Tab. 5. The experiments are conducted using the CSS method PLOP with the PASCAL VOC2012 dataset in the 15-5s setting. The S_i denotes the attack at step i. First, this attack is valid for any incremental step. Similar results are achieved for any step of the attack. This is because once the model is attacked, its learned knowledge is corrupted and cannot be recovered again in subsequent increments. Moreover, the effect is most effective when all incremental steps are attacked.

Qualitative Evaluation

Fig. 4 shows the original predictions of PLOP and the prediction results after training with adversarial samples in the PASCAL VOC2012 dataset with the 15-5s setting. PLOP is able to retain the old knowledge better during the incremental process, which produces clear results. However, its performance drops dramatically after training with adversarial samples, and the original knowledge is induced as background in the incremental training.

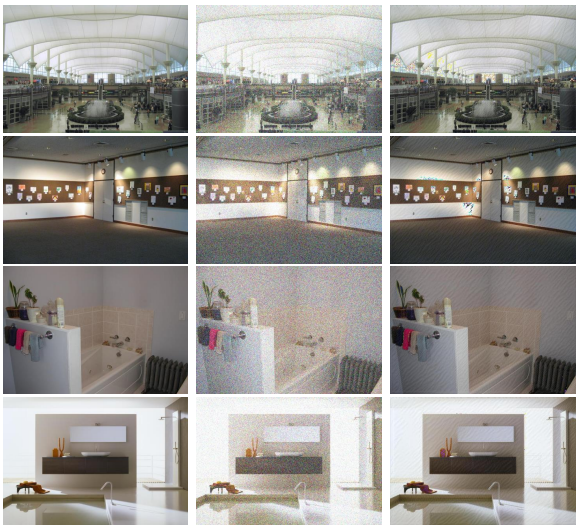

Figure 5: Visualization results of some adversarial samples on ADE20k in the 100-50 setting. (a) Original images. (b) Images of adding Gaussian noise. (c) The adversarial samples generated in the second incremental learning step.

Fig. 5 shows some original and perturbed images after incremental training under ADE20k dataset with the 100-50 setting. These adversarial samples are generated in the second step of incremental learning, i.e., 50 classes are added. The CSS method is PLOP. Thanks to the loss term $\mathcal{L}_r$, there is a gap between the adversarial samples and the original images, but the difference is almost imperceptible.

Conclusions

In this paper, we focused on the potential risk in continual semantic segmentation. We showed for the first time that the attack for incremental samples in CSS can substantially disrupt the performance of incremental models and reduce the retention of old knowledge. Moreover, we proposed a new task, namely, an adversarial attack on incremental samples in CSS. Specifically, perturbing pictures by adversarial attacks on incremental data streams leads to the failure of incremental learning methods, thus defeating the purpose of incremental learning, i.e., the retention of the learned knowledge. On this basis, we proposed a class shift-based attack method to disrupt the incremental process by changing the predictive distribution of the old model over the incremental data stream and obfuscating the old knowledge generated by the model. We validated the proposed approach in several classic CSS methods and experimentally demonstrated that CSS methods greatly forget the old knowledge due to the attack on incremental samples.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62172385 and 62072428), and the Innovation Program for Quantum Science and Technology (No. 2021ZD0302900).

References

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

Asgari Taghanaki, S.; Abhishek, K.; Cohen, J. P.; Cohen-Adad, J.; and Hamarneh, G. 2021. Deep semantic segmentation of natural and medical images: a review. *Artificial Intelligence Review*, 54: 137–178.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, 39–57. Ieee.

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*.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34: 17864–17875.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

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.

Feng, J.; Cai, Q.-Z.; and Zhou, Z.-H. 2019. Learning to confuse: generating training time adversarial data with auto-encoder. *Advances in Neural Information Processing Systems*, 32.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*.

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.

Han, G.; Choi, J.; Hong, H.; and Kim, J. 2022. Training Time Adversarial Attack Aiming the Vulnerability of Continual Learning. In *NeurIPS ML Safety Workshop*.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

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.

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.

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.

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

Michieli, U.; and Zanuttigh, P. 2021a. 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.

Michieli, U.; and Zanuttigh, P. 2021b. Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding*, 205: 103167.

Milioto, A.; Lottes, P.; and Stachniss, C. 2018. Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs. In *2018 IEEE international conference on robotics and automation (ICRA)*, 2229–2235. IEEE.

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.

Rong, X.; Sun, X.; Diao, W.; Wang, P.; Yuan, Z.; and Wang, H. 2022. Historical information-guided class-incremental semantic segmentation in remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–18.

Shelhamer, E.; Long, J.; and Darrell, T. 2017. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4): 640–651.

Siam, M.; Gamal, M.; Abdel-Razek, M.; Yogamani, S.; Jagersand, M.; and Zhang, H. 2018. A comparative study of real-time semantic segmentation for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 587–597.

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.