Attribution-aware Weight Transfer: A Warm-Start Initialization for Class-Incremental Semantic Segmentation

Dipam Goswami[†§] René Schuster[†] Joost van de Weijer[‡] Didier Stricker[†]
dipamgoswami01@gmail.com rene.schuster@dfki.de joost@cvc.uab.es didier.stricker@dfki.de
[†] DFKI - German Research Center for Artificial Intelligence, Kaiserslautern
[§] Birla Institute of Technology and Science, Pilani [‡] Computer Vision Center, Barcelona

Abstract

In class-incremental semantic segmentation (CISS), deep learning architectures suffer from the critical problems of catastrophic forgetting and semantic background shift. Although recent works focused on these issues, existing classifier initialization methods do not address the background shift problem and assign the same initialization weights to both background and new foreground class classifiers. We propose to address the background shift with a novel classifier initialization method which employs gradient-based attribution to identify the most relevant weights for new classes from the classifier's weights for the previous background and transfers these weights to the new classifier. This warm-start weight initialization provides a general solution applicable to several CISS methods. Furthermore, it accelerates learning of new classes while mitigating forgetting. Our experiments demonstrate significant improvement in mIoU compared to the state-of-the-art CISS methods on the Pascal-VOC 2012, ADE20K and Cityscapes datasets.

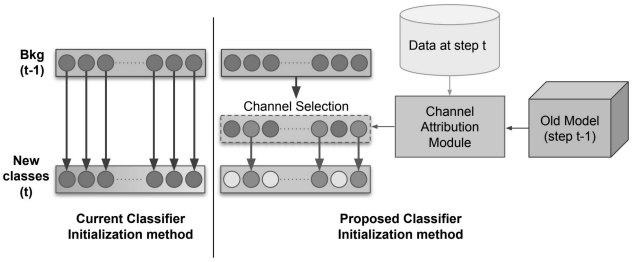

Figure 1: Comparison of classifier initialization methods. Classifier initialization was previously found to be crucial to obtain good plasticity [6, 7, 17, 48, 53] of an incremental learner. However, the method did not address the *background shift*. Previous methods (left) copy all filter weights from previous background (bkg) to initialize *new classes*, our attribution-based weight selection (right) explicitly addresses the *semantic background shift* by selecting only the relevant weights for the classification of *new classes*. This allows us to combine the plasticity of the previous initialization method with a further reduction of catastrophic forgetting.

1. Introduction

Semantic segmentation assigns a class label to every pixel of an image. The emergence of deep neural networks as well as the availability of pixel-level annotated datasets [13, 20, 56] has achieved state-of-the-art performance on semantic segmentation tasks [12,32,54]. The majority of papers in the field considers that all classes are labelled in all training data, and that all training data is jointly available. However, for many applications this is an unrealistic scenario, and the algorithm has to learn to segment all classes from partially labelled data, and every moment (called *step* in CISS) only has access to a limited batch of training data. This restriction is imposed either by data storage limitations or data privacy and data security considerations [14]. Incremental learning [14, 35] proposes algorithms for this setting where the main challenge is to prevent *catastrophic forgetting* [36] which refers to a drop in performance on classes learned in previous steps.

Another critical challenge faced by CISS approaches is the *semantic background shift*. This challenge does not exist for incremental image classification, and is a result of the multi-class nature of image segmentation. The ground truth at any step provides labels for pixels belonging to current classes only and considers all other pixels as background while the model should correctly classify the old and current class pixels to their original labels and the future class pixels to the background. So, the background class includes the real background class, the previously seen classes, and the future classes. As a result, there exists ambiguity due to changing definition of the background class from step to step.

In this paper, we highlight the importance of the ini-

tialization of the classifier's weights for the new classes. Since at every step, the final classifier layer has to predict the new classes in addition to the past classes, the classifier's weights for the new classes should be well aligned with the features extracted by the model. Thus, the initialization of classifier's weights is critical for stable training of the model and faster convergence on the new classes resulting in less forgetting of past classes. MiB [6] adapted weight imprinting [40] for segmentation and initialized the classifier's weights for the new classes and the background with the classifier's weights for the previous background. This initialization has been followed in most of the recent approaches [7, 17, 48, 53] but it does not address the semantic background shift problem. Instead, we propose a new warm-start classifier initialization method that explicitly tackles the background shift, differentiating the current foreground classes from the previous background at the classifier level as depicted in Figure 1.

We propose to transfer the learning of the future classes from the previous background to the new classes by weight transfer from relevant classifier input channels. The proposed method follows the strict incremental setting [35], which does not need access to any data from past or future classes. Our method can be used with various CISS approaches. We validate our proposed approach using multiple CISS settings, class orders and ablation experiments. The main contributions can be summarized as follows:

- To better address the background shift, we apply an attribution method to identify the most relevant classifier channels for predicting the new classes as previous background and transfer only those channel weights.
- Our method reduces catastrophic forgetting on old classes while improving plasticity on new classes, owing to quicker convergence on new classes.
- We experimentally show that our method significantly outperforms the state-of-the-art approaches on several incremental settings and datasets.

2. Related Work

Semantic Segmentation: Commonly used segmentation approaches are based on Fully Convolutional Networks (FCNs) [32]. These methods improve the accuracy by using multiscale representations [31], retaining more spatial information by using atrous convolution [9] or convolution with upsampled filters, modelling contextual cues [9], or by using attention mechanisms [11, 55]. Recent approaches used strip pooling [22] along the width or height dimensions to capture both global and local statistics. In our work, we use the Deeplabv3 [10] architecture which employs atrous convolution in parallel manner in order to capture multi-scale context to segment objects at multiple scales.

Incremental Learning: Most studies in incremental learning have focused on object detection and classification problems [5, 30, 41, 43, 47]. Some of these works use replay-based approaches, which store samples from previous tasks [5, 41] or generate training data [25, 42]. Some methods extend the initial architecture to learn new classes [29,49] or use embedding networks [51] or use classifier drift correction to account for changing class distributions [3, 4]. Distillation-based methods constrain the learning of the model on new tasks by penalizing updates on the weights [1, 26] or the gradients [8, 33] or the intermediate features [16, 19, 23]. Our work focuses on the distillation-based approaches for semantic segmentation.

Class-Incremental Semantic Segmentation: Recently, incremental learning has been studied for semantic segmentation [6, 7, 17, 18, 24, 27, 37, 38, 50]. Initial approaches use relevant examples from old tasks and perform rehearsal for segmentation in medical imaging [39] and remote sensing data [46]. Michieli *et al*. [37] considered an incremental setting where labels for old classes are available when learning new tasks. Cermelli *et al*. [6] was the first to highlight the semantic background shift and proposed a novel distillation method to tackle the shift. Douillard *et al*. [17, 18] proposed using multi-scale spatial distillation loss to preserve short and long range dependencies. Cha *et al*. [7] proposed SSUL which considers a separate class apart from the semantic background class for old and future classes in addition to freezing the backbone and past classifiers. UCD [48] proposed to enforce similarity between features for pixels of same classes and reduce feature similarity for pixels of different classes. RCIL [53] decoupled the learning of both old and new classes and introduced pooled cube knowledge distillation on channel and spatial dimensions.

Replay of samples from previous classes has also been used for CISS either by storing images from old tasks [7] or by recreating them using generative networks [34]. Self-training approach using unlabelled data [50] has also been proposed. We propose to model the semantic background shift for the classifier initialization used in [6, 7, 17, 48, 53] without using any data from the previous steps.

Attribution Methods: Feature attribution methods assign importance scores to the features for a given input which are responsible for the class prediction. Existing attribution methods are based on perturbation or backpropagation. Perturbation methods [52] compute the attributions of input features by removing or masking them and then do a forward pass to measure the difference in outputs. Backpropagation methods compute the attributions for the input features by doing one forward and backward pass. Some of these methods are DeepLIFT [44], Integrated Gradients [45] and Layer-wise Relevance Propagation (LRP) [2]. We use the popular Integrated Gradients [45] which requires no modification to the network and is simple to implement.

3. Proposed Weight Transfer Method

3.1. Class-Incremental Semantic Segmentation

Consider an image x and label space $\mathcal{C}$, semantic segmentation aims to assign a label $c_i \in \mathcal{C}$ to every pixel in x. Provided with a training set $\mathcal{T}$, a model f_θ having parameters θ is learned which maps the input image to the pixel-wise class probabilities. In an incremental setup, the model is learned in $t = 1...T$ steps. The training set at incremental step t is $\mathcal{T}^t = \{(x_1^t, y_1^t), ..., (x_{n^t}^t, y_{n^t}^t)\}$ where $x_i^t \in X^t$ is the set of images, $y_i^t \in Y^t$ is the set of corresponding ground truth maps and a new set of classes $\mathcal{C}^t$ is added to the existing set of classes $\mathcal{C}^{1:t-1}$. Since the background class is present in all the incremental steps, we denote it as b^t at step t. The model at step t is denoted as f_{θ^t} which learns the parameters θ^t.

For an image $x_i^t \in X^t$, the ground truth segmentation map $y_i^t \in Y^t$ only provides the labels of current classes $\mathcal{C}^t$ while collapsing all other labels (old classes $\mathcal{C}^{1:t-1}$ and future classes $\mathcal{C}^{t+1:T}$) into the background class b^t. The model after step t is expected to predict all classes learned over time $\mathcal{C}^{1:t}$. Here, both the real background class pixels and the future class $\mathcal{C}^{t+1:T}$ pixels should be predicted as background b^t. Hence, the future classes classified as background after the first step gradually become the foreground during the incremental steps. During the inference of the final step, only the real background class should be classified as the background.

3.2. Problems with Existing Initialization Method

We discuss the existing initialization approach and the problems that are yet to be addressed. Since the pixels of $\mathcal{C}^t$ are learned as background b^{t-1} at the previous step, the old model $f_{\theta^{t-1}}$ will most likely assign these pixels to class b^{t-1}. To account for this initial bias on predictions of f_{θ^t} for pixels of $\mathcal{C}^t$, Cermelli *et al.*, [6] proposed to initialize the classifier's weights for the classes in $\mathcal{C}^t$ (including background) with the weights for the previous background class so that the background class probability is uniformly spread among the classes in $\mathcal{C}^t$ ($b^t \in \mathcal{C}^t$). It is important to note, that this classifier initialization was found to be crucial to achieve good plasticity. For several settings, classifier initialization more than doubles performance on the classes learned after the first step (see for example Table 3 in [6]).

However, this direct transfer of classifier's weights from background to new classes does not directly address the shift of classes from background to foreground across time, which is one of the main challenge for CISS problems. The background classifier weights are learned for the real background and future classes $\mathcal{C}^{t+1:T}$ but the direct transfer guides the model to initially assign high probabilities for pixels of $\mathcal{C}^{t+1:T}$ and real background class to $\mathcal{C}^t$ instead of b^t.

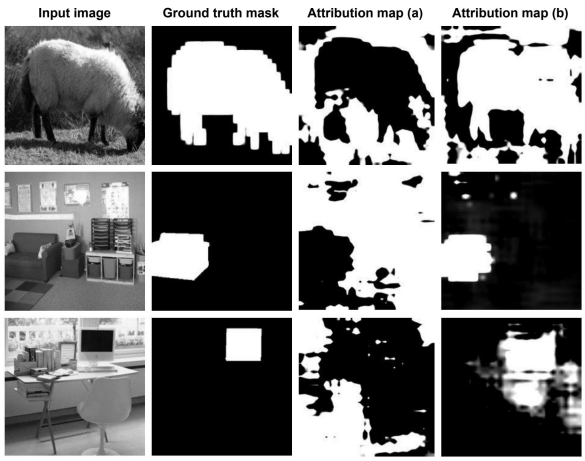

Figure 2: Attribution maps for the background (*bkg*) class corresponding to different channels of the classifier layer. The new classes (sheep, sofa, monitor) belong to the *bkg* of the previous step. Attribution map (a) has high contribution towards predicting the real *bkg* pixels and does not predict the new class while the attribution map (b) contributes more towards predicting the new class pixels as *bkg*.

3.3. Novel Warm-Start Classifier Initialization

To explicitly address the background shift at the initialization stage, we propose Attribution-aware Weight Transfer (AWT) as illustrated in Figure 3. AWT aims to transfer only the significant classifier channel weights from the background b^{t-1} to the new classes in $\mathcal{C}^t$. We use attribution methods to select the classifier weights for the background at the previous step, which have significant contributions in predicting the pixels of new classes in $\mathcal{C}^t$ as b^{t-1} (as depicted in Figure 2). Here, we exploit the ability of the background classifier to learn different representations using different channels. This selection separates the classifier level weights for the new classes in $\mathcal{C}^t$ and the future classes. AWT aims not to transfer the significant classifier weights for the future classes $\mathcal{C}^{t+1:T}$ to the new classes thereby maintaining the stability of the model and accelerating learning of the new classes.

3.3.1 Attribution-aware Channel Selection

Integrated Gradients [45] approximates the integral of the gradients of the model's output with respect to the inputs along a straight-line path from baselines to inputs. Here, baseline refers to the starting point from which integral is computed and is taken as blank (zero) input. We employ the Integrated Gradients attribution method, which assigns importance scores to the inputs of the classifier layer for predicting the background b^{t-1}. More details of Integrated Gradients are provided in the supplementary material.

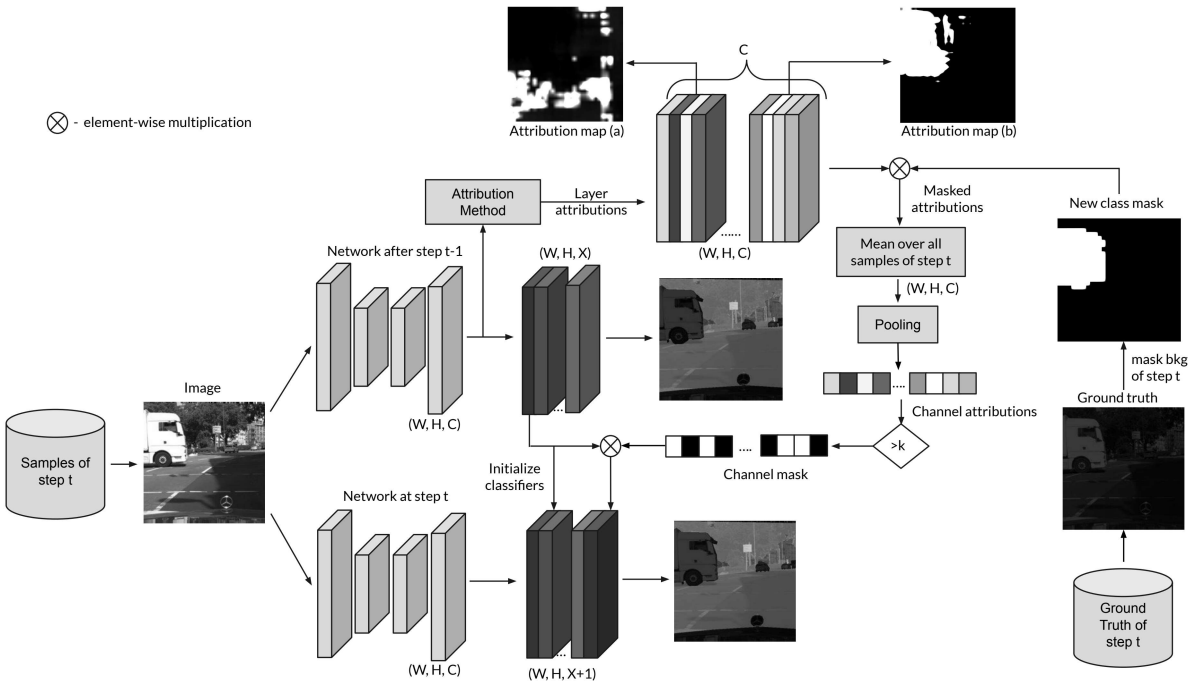

Figure 3: Overview of AWT: Images from the current step are given to the old model at step $t-1$. The inputs to the classifier layer are used to generate the layer attributions. Here, the attribution map (b) is more significant for new class pixels than map (a). We mask the attribution maps for the background pixels using the ground truth. Masked attributions from all images of step t are averaged and max-pooled to obtain channel attributions. The significant channels are then selected using a threshold k and these channel weights are transferred to the classifier weights for the new classes.

We use the images from the current training set X^t and the old model $f_{\theta^{t-1}}$ for computing the attribution maps for each of the input channels to the classifier layer. Since, the old and future class pixels are labelled as b^t in the ground truth maps at step t, we obtain a ground truth mask $\bar{y}_i^t$ for new classes only by masking the pixels m belonging to b^t as follows:

$$\bar{y}_i^t[m] = \begin{cases} 0 & \text{if } y_i^t[m] = b^t \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

We use the ground truth masks $\bar{y}_i^t$ to obtain the attributions corresponding to the pixels of new classes in $\mathcal{C}^t$ only. Since we consider only one set of channel mask for each new class in $\mathcal{C}^t$, we take the mean of the masked attributions from all the images in X^t. Let A be the set of classifier layer attribution maps, each of size $W \times H \times C$ for all images in X^t obtained using Integrated Gradients, then we obtain the mean attribution as follows:

$$A_{avg} = \frac{1}{n^t} \sum_{i=1}^{n^t} A(x_i^t) \odot \bar{y}_i^t \tag{2}$$

where $\odot$ refers to an element-wise multiplication along the spatial dimensions.

Max-Pooling is performed on the mean attribution A_{avg} to get an attribution value for each of the channels of classifier's weights for background b^{t-1}. We transform the mean attribution map of size $W \times H \times C$ to channel attribution A_c of size C with pooling along the channel axis as follows:

$$A_c[c] = \max_{w \in [1,W], h \in [1,H]} (A_{avg}[w, h, c]) \tag{3}$$

where $c \in [1, C]$. Note that we choose max-pooling over average-pooling based on experiments in Section 4.3.

3.3.2 Classifier Initialization

A threshold k is applied on the channel attribution to obtain a channel mask c_{mask} to select the most contributing channels as follows:

$$c_{mask}[c] = \begin{cases} 1 & \text{if } A_c[c] > k \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where A_c denotes the channel attribution and c refers to the channels. Let the classifier's weights for the class c at step t be w_c^t and the classifier's weights for the background at step $t-1$ be w_b^{t-1}. We propose to initialize the classifier's weights for the new classes with the selected channel's

weights as follows:

$$w_c^t = \begin{cases} w_c^{t*} + w_b^{t-1} \cdot c_{mask} & \text{if } c \in \mathcal{C}^t \setminus b^t \\ w_c^{t-1} & \text{otherwise} \end{cases} \quad (5)$$

where w_c^{t*} refers to the default initialized weights. We transfer the masked weights by adding them on top of the default weights and thus we avoid having zero weights for the remaining channels. We show in Section 4.3 that adding the weights is beneficial compared to copying.

4. Experiments

4.1. Experimental settings

Datasets: We conduct experiments on the segmentation datasets namely Pascal VOC 2012 [20], ADE20K [56] and Cityscapes [13] using different incremental splits. Pascal VOC 2012 [20] covers 20 object (or *things*) classes and one background class. ADE20K [56] is a large scale dataset containing 150 classes of both *things* and *stuff* (uncountable or amorphous regions like sky or grass). Cityscapes [13] has 19 classes having both *things* and *stuff* and covering scenes from 21 different urban cities.

CISS Protocols: Two different CISS settings introduced by [6] are *disjoint* and *overlapped*. While the *disjoint* setting assumes that the future classes are known and removes images with future classes from the current step, the *overlapped* setting is more realistic and has no such assumption. Similar to [7, 17], we also follow the *overlapped* setting in our experiments. We denote the different settings as X-Y where X is the number of classes in the initial step followed by Y number of classes at every step until all the classes are seen. We train 15-5 (15 classes followed by 5 classes), 15-1 (15 then 1 class in each step), 5-3 and 10-1 settings on VOC. Similarly, we train 100-50, 100-10, 100-5 and 50-50 on ADE20K and 14-1 and 10-1 settings on Cityscapes.

Metrics: The mean Intersection over Union (mIoU) metric is calculated after the last step for the initial set of classes, the incremental classes, and for all the classes. The mIoU for the initial classes reflects the stability of model to catastrophic forgetting. The mIoU for the incremental classes reflects the plasticity of the model to learn new classes and the overall mIoU metric signifies the overall performance.

Implementation Details: Deeplab-v3 [10] with a ResNet-101 [21] backbone pretrained on ImageNet [15] having output stride of 16 is used for the experiments. Similar to [53], we use a higher initial learning rate and obtain an improved baseline for MiB. We train MiB and MiB+AWT models with SGD and a learning rate of 2×10^{-2} for the first step only and 10^{-3} for the incremental steps. The models are trained with a batch size of 24 using 2 GPUs for 30 epochs per step for VOC and Cityscapes and 60 epochs for ADE20K. Specific to SSUL models, we follow the same training settings as [7] since it performs freezing of weights and requires different training hyperparameters. The final results are reported on the validation set of the datasets. Since Cityscapes does not have a real background class, we merge the unlabeled classes into a virtual background class.

We use layer integrated gradients from [28] for obtaining the attributions and a threshold k to select the 25% most significant channels for new classes based on experiments provided in the supplementary material. We obtain a unique set of channels mask for each new class for all settings having 5 or lesser class increments. For settings like 100-10, 100-50 and 50-50 on ADE20K, we use a single channel mask for all the new classes. Code is publicly available[1].

Baselines: We compare our approach with the recent state-of-the-art methods ILT [37], MiB [6], SDR [38], PLOP [17], SSUL [7], RCIL [53] and UCD [48]. We apply AWT on two methods, MiB [6] and SSUL [7]. We also compare with the upper bound (Joint model learned in non-incremental manner). We do not consider approaches using data from past steps [34] or auxiliary unlabeled data [50].

4.2. Quantitative Evaluation

Pascal VOC 2012: We show the quantitative experiments on VOC 15-5, 15-1, 5-3 and 10-1 settings in Table 1. We observe that while ILT struggle on all settings, other methods show significant improvements. Pooling-based distillation methods like PLOP and RCIL do better in 15-5, 15-1 and 10-1 settings but these methods perform poorly on the 5-3 setting where the number of classes is less in the initial step.

AWT with MiB outperforms MiB significantly on all the settings. On 15-5, our model outperforms MiB by 1.5 percentage point ($p.p$) on the overall mIoU metric. On the 15-1 setting, our model reduces the forgetting of the initial classes by 11 $p.p$ while the overall performance improves by 8.7 $p.p$. On the 5-3 setting having multiple class increments, AWT improves the overall mIoU by 4.3 $p.p$ over MiB. On the most challenging setting of 10-1 having 11 steps, AWT reduces the forgetting of the initial classes by 19.1 $p.p$ and improves the learning of new classes by 4.2 $p.p$.

AWT with SSUL [7] performs similar to SSUL for the 15-5, 15-1 and 5-3 settings, while for the challenging 10-1 setting, it reduces forgetting of the initial classes by 1.8 $p.p$ and improves the performance on new classes by 1.0 $p.p$. SSUL makes use of saliency maps targeted for *things* or objects and moves them from background to an unknown class for representing the future classes. This label augmentation improves performance on all settings of VOC since this dataset has only object classes. On the contrary, this saliency-based modelling is not applicable for ADE20K, Cityscapes and other datasets which have both *things* and *stuff* classes, and SSUL suffers from high forgetting as we observe in Tables 2 and 4.

[1] https://github.com/dfki-av/AWT-for-CISS

Table 1: Experimental results on Pascal VOC 2012. Improvements using AWT underlined. Best among columns in **bold**. †: results excerpted from [53]. * implies results come from re-implementation. Other results come from the respective papers.

	15-5 (2 steps)			15-1 (6 steps)			5-3 (6 steps)			10-1 (11 steps)		
Method	0-15	16-20	all	0-15	16-20	all	0-5	6-20	all	0-10	11-20	all
ILT† [37]	67.8	40.6	61.3	9.6	7.8	9.2	22.5	31.7	29.0	7.2	3.7	5.5
SDR† [38]	76.3	50.2	70.1	47.3	14.7	39.5	-	-	-	32.4	17.1	25.1
PLOP [17]	75.7	51.7	70.1	65.1	21.1	54.6	17.5	19.2	18.7	44.0	15.5	30.5
MiB+UCD [48]	78.5	50.7	71.5	51.9	13.1	42.2	-	-	-	33.7	26.5	31.1
RCIL† [53]	**78.8**	52.0	**72.4**	70.6	23.7	59.4	59.3	33.8	41.1	55.4	15.1	34.3
MiB [6]	75.5	49.4	69.0	35.1	13.5	29.7	-	-	-	12.3	13.1	12.7
MiB* [6]	76.4	49.4	70.0	48.1	15.8	40.4	58.2	41.3	46.1	14.1	13.8	13.9
MiB+AWT (Ours)	77.3	**52.9**	71.5	59.1	17.2	49.1	61.8	45.9	50.4	33.2	18.0	26.0
SSUL [7]	77.8	50.1	71.2	**77.3**	36.6	**67.6**	**72.4**	50.7	56.9	71.3	46.0	59.3
SSUL+AWT (Ours)	78.0	50.2	71.4	77.0	**37.6**	**67.6**	71.6	**51.4**	**57.1**	**73.1**	**47.0**	**60.7**
Joint	79.8	72.4	77.4	79.8	72.4	77.4	76.9	77.6	77.4	78.4	76.4	77.4

Table 2: Experimental results on ADE20K. Improvements using AWT underlined. Best among columns in **bold**. †: results excerpted from [53]. * implies results come from re-implementation. Other results come from the respective papers.

	100-50 (2 steps)			100-10 (6 steps)							50-50 (3 steps)			
Method	1-100	101-150	all	1-100	101-110	111-120	121-130	131-140	141-150	all	1-50	51-100	101-150	all
ILT† [37]	18.3	14.8	17.0	0.1	0.0	0.1	0.9	4.1	9.3	1.1	13.6	12.3	0.0	9.7
PLOP [17]	41.9	14.9	32.9	40.6	15.2	16.9	18.7	11.9	7.9	31.6	**48.6**	30.0	13.1	30.4
PLOP+UCD [48]	42.1	15.8	33.3	**40.8**	-	-	-	-	-	32.3	47.1	-	-	31.8
SSUL* [7]	38.0	20.5	32.2	36.5	**16.5**	29.0	21.7	16.4	13.5	30.8	44.1	23.0	18.6	28.7
RCIL† [53]	**42.3**	18.8	34.5	39.3	14.6	26.3	23.2	12.1	11.8	32.1	48.3	**31.3**	18.7	32.5
MiB† [6]	40.5	17.7	32.8	38.3	12.6	10.6	8.7	9.5	15.1	29.2	45.3	26.1	17.1	29.3
MiB* [6]	41.5	22.9	35.3	38.9	10.3	13.8	12.3	5.1	13.0	29.6	46.1	27.1	21.8	31.8
MiB+AWT (Ours)	40.9	**24.7**	**35.6**	39.1	14.3	**31.9**	**24.4**	**20.6**	**15.2**	**33.2**	46.6	30.1	**23.6**	**33.5**
Joint	44.3	28.2	38.9	44.3	26.1	42.8	26.7	28.1	17.3	38.9	51.1	38.3	28.2	38.9

ADE20K: ADE20K [56] is a difficult dataset with 150 classes and has the joint model mIoU of only 38.9%. We report the experimental results on ADE20K 100-50, 100-10 and 50-50 in Table 2 with analysis of performance on the incremental sets of classes. We also consider a long setting of 100-5 (11 tasks) in Table 3.

On 100-50, our model improves the overall performance over MiB by 0.3 $p.p$. On 50-50 setting, our model achieves an overall improvement of 1.7 $p.p$ over MiB and 1.0 $p.p$ over RCIL. Moving to the longer sequence of 100-10 with 6 steps, our model improves MiB by 3.6 $p.p$ and PLOP+UCD by 0.9 $p.p$. On the 11 step setting of 100-5, AWT improves MiB by 4.6 $p.p$ and its nearest contender SSUL by 1.0 $p.p$. MiB+AWT achieves state-of-the-art results on all settings of ADE20K indicating the robustness towards predicting both *things* and *stuff* classes.

Cityscapes: We perform CISS experiments on two long sequence settings of 14-1 (6 tasks) and 10-1 (10 tasks) of Cityscapes [13] dataset. We introduce the 10-1 setting where we initially train on 10 classes (road, sidewalk, building, wall, fence, pole, light, sign, vegetation, terrain) and add each of the 9 classes (sky, person, rider, car, truck, bus, train, motorcycle, bicycle) one at a time. We evaluate naive fine-tuning (FT), PLOP, RCIL, SSUL, MiB, and AWT with both SSUL and MiB, and report the mIoU results in Table 4.

Table 3: Experimental results on the 100-5 setting on ADE20K. Improvements using AWT underlined. Best among columns in **bold**. †: results excerpted from [53]. * implies results come from re-implementation.

	100-5 (11 steps)		
Method	1-100	101-150	all
ILT† [37]	0.1	1.3	0.5
PLOP [17]	**39.1**	7.8	28.8
RCIL† [53]	38.5	11.5	29.6
SSUL* [7]	36.0	**18.2**	30.1
MiB† [6]	36.0	5.6	25.9
MiB* [6]	36.9	5.4	26.5
MiB+AWT (Ours)	38.6	16.0	**31.1**
Joint	44.3	28.2	38.9

Table 4: Experimental results on Cityscapes. Improvements using AWT underlined. Best among columns in **bold**. All results come from our implementation.

Method	14-1 (6 steps)			10-1 (10 steps)		
	1-14	15-19	all	1-10	11-19	all
FT	0.0	10.1	2.5	0.0	4.8	2.2
PLOP [17]	55.7	12.3	44.8	**52.2**	24.1	39.6
RCIL [53]	55.7	7.1	43.6	51.0	17.4	35.9
SSUL [7]	43.2	33.0	40.7	38.6	38.1	38.3
SSUL+AWT	43.9	**35.1**	41.5	38.6	**39.0**	38.8
MiB [6]	**56.3**	12.5	45.4	51.6	30.1	41.9
MiB+AWT	55.9	19.8	**46.9**	51.2	37.2	**44.9**
Joint	56.7	54.3	56.1	51.7	61.4	56.1

We observe that while FT has very low overall mIoU, PLOP, RCIL and MiB have improved overall performance on both settings. SSUL shows higher performance on incremental classes but with very high forgetting on the initial classes compared to others. On the 14-1 setting, AWT with SSUL improves the overall mIoU over SSUL by 0.8 $p.p$ and MiB+AWT outperforms MiB by 1.5 $p.p$ with a significant improvement of 7.3 $p.p$ on the performance of incremental classes (15-19). On the longer 10-1 setting, SSUL+AWT increases the overall mIoU by 0.5 $p.p$ over SSUL while MiB+AWT improves over MiB by 3.0 $p.p$ with a good margin of 7.1 $p.p$ improvement on the incremental classes (11-19). AWT significantly improves the plasticity of the models to better learn the new classes in both settings.

4.3. Ablation Study

We analyze the effectiveness of our approach with ablation experiments on Pascal-VOC 2012 for the 15-1 setting.

Selective weight transfer: We analyze the importance of the selective weight transfer approach in Table 5. The weight transfer proposed by MiB [6] is a better choice compared to the case when no weights are transferred. We show that our proposed AWT ensures the selection of the most significant channels for new classes by performing experiments with random selection of channels without using attributions. We observe that randomly selecting the same number of channels (25% of total channels) and transferring their weights in the same way as AWT performs poorly on both initial and incremental sets of classes.

Design choices: We consider the alternative ways of selecting the significant channels and analyze them in Table 6. In AWT, we take the mean of the attribution maps from all images of the current step and then perform max-pooling. Here, we consider the alternative of pooling the channels first for all the images and then take the mean of the pooled values. We also consider using average-pooling instead of max-pooling. We experimentally show that mean followed by max-pool is the best choice for channel selection.

Table 5: Ablation study for selective channel weights transfer on Pascal-VOC 2012.

Strategy	% of filters	copy	add	VOC (15-1) 0-15	16-20	all
No transfer	0	×	×	45.7	5.3	36.1
MiB [6]	100	✓	×	48.1	15.8	40.4
Random	25	×	✓	46.3	6.1	36.8
AWT	25	✓	×	58.3	14.8	47.9
AWT	25	×	✓	**59.1**	**17.2**	**49.1**

Table 6: Ablation study for different design choices using MiB [6] + AWT on Pascal-VOC 2012.

MiB+AWT	VOC (15-1) 0-15	16-20	all
Max-Pool $\Longrightarrow$ Mean	55.2	14.5	45.5
Avg-Pool $\Longrightarrow$ Mean	58.3	14.1	47.8
Mean $\Longrightarrow$ Avg-Pool	57.6	14.2	47.2
Mean $\Longrightarrow$ Max-Pool	**59.1**	**17.2**	**49.1**

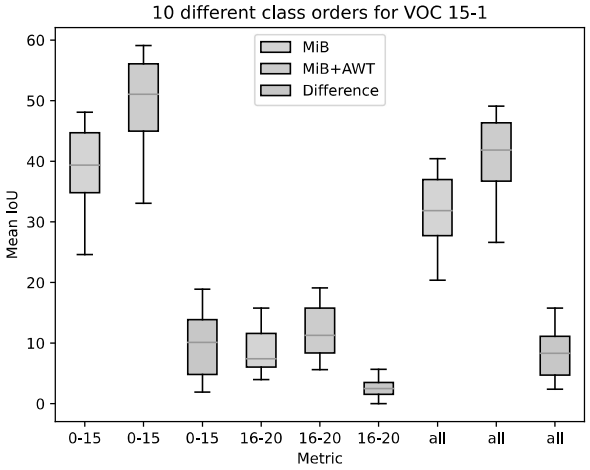

Figure 4: Boxplots of the mIoU of initial, new, and all classes for 10 random class orders.

Random class ordering: The order of classes plays an important role in CISS settings. We experiment with 10 different class orderings on VOC 15-1 setting. We show the average performance for MiB and MiB+AWT in Figure 4. We also plot the difference between MiB+AWT and MiB for every class order to demonstrate the robustness of our method using random class sequences.

Computational Complexity: The time taken for the attribution module depends on the number of new-class images. We use two Nvidia RTXA6000 GPUs for training the models. For each image, it takes approximately 0.68 seconds to

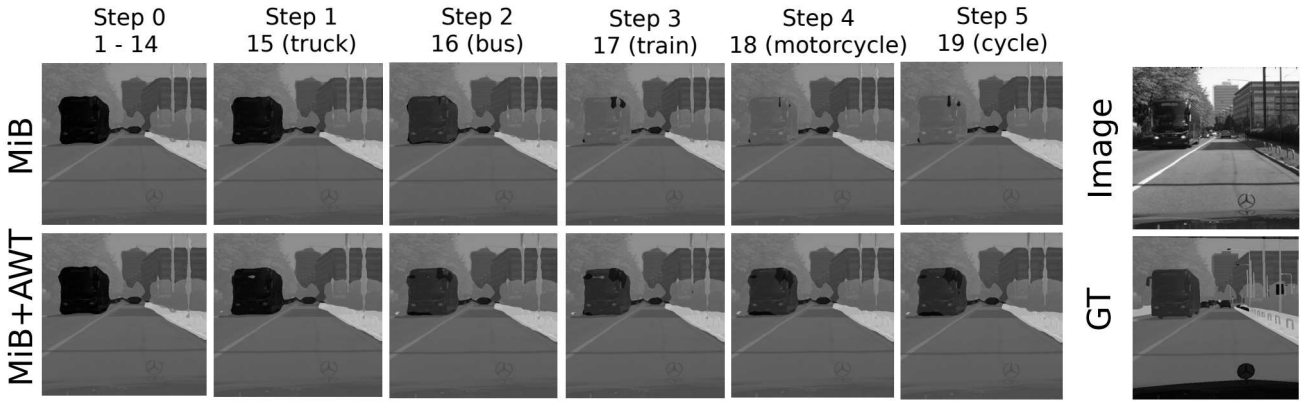

Figure 5: Visualization of predictions using MiB and MiB+AWT in 14-1 setting for Cityscapes. MiB is highly biased towards the new classes and classifies the *bus* as *train* (row 1) while MiB+AWT correctly classifies the *bus* (row 2).

compute the attributions. For VOC 15-1, MiB+AWT takes 10.32 hours for training while the attribution module for all steps only takes 37 minutes (6% of the total training time). Thus, the computational time for the attribution module is considerably less compared to the entire training process.

For further analysis, refer to the supplementary paper.

4.4. Qualitative Evaluation

Figure 5 shows the predictions of MiB and MiB+AWT across time on Cityscapes 14-1 setting. MiB is biased towards the new classes and forgets the class (*bus*) learned in step 2 and classifies the *bus* as *train* from step 3 onward. MiB+AWT still classifies the *bus* correctly till step 5.

Figure 6 shows the predictions for both MiB and MiB+AWT models trained in 100-5 setting on test images of ADE20K. We show that MiB+AWT improves predictions of classes like *fan* (row 1), *wardrobe* (row 2) and *chair, chandelier* (row 3) compared to MiB.

(a) Image (b) MiB [6] (c) MiB+AWT (d) GT

Figure 6: Visualization of predictions using MiB and MiB+AWT in 100-5 setting on test images of ADE20K.

5. Conclusion and Limitations

In this paper, we addressed the issue of semantic background shift during the initialization of the new foreground classifiers at each step of CISS. We discussed the problems with the existing initialization method, and then proposed an attribution-based approach to identify weights that are of interest for the new foreground classes and transfer these weights. This selective initialization takes into account the gradual transition of classes from background to foreground across time. Experimental results on multiple datasets showed that our approach reduces the forgetting of old classes and further improves the plasticity. Our weight transfer approach generalizes well with both *things* and *stuff* classes and outperforms the existing CISS methods. This work lays the foundation for attribution-aware weight initialization that could be more generally used for incremental learning problems involving multi-class classification

Although our method works well with most incremental settings, the strategy of selecting multiple set of channels for multiple new classes would involve a significant increase in computational complexity as a function of the number of new classes at every step, especially for 10, 50 or more class increments at a step. We believe future work can be done to address this limitation. We hope that our attribution-based channel selection approach would be adapted beyond semantic segmentation to other computer vision applications.

Acknowledgement. This work was partially funded by the Federal Ministry of Education and Research Germany under the project DECODE (01IW21001) and partially by the Spanish Government funded project PID2019-104174GB-I00/AEI/10.13039/501100011033.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision (ECCV)*, 2018.

[2] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS One*, 2015.

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *International Conference on Computer Vision (ICCV)*, 2019.

[4] Eden Belouadah and Adrian Popescu. Scail: Classifier weights scaling for class incremental learning. In *Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[5] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Conference on Computer Vision (ECCV)*, 2018.

[6] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[8] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations (ICLR)*, 2019.

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2017.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[11] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[13] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2021.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[16] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[18] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Tackling catastrophic forgetting and background shift in continual semantic segmentation. *arXiv preprint arXiv:2106.15287*, 2021.

[19] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[20] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision (IJCV)*, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[24] Tobias Kalb, Masoud Roschani, Miriam Ruf, and Jürgen Beyerer. Continual learning for class-and domain-incremental semantic segmentation. In *Intelligent Vehicles Symposium (IV)*, 2021.

[25] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[26] 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 (PNAS)*, 2017.

[27] Marvin Klingner, Andreas Bär, Philipp Donn, and Tim Fingscheidt. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[28] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, et al. Captum: A unified and generic model interpretability library for pytorch. *arXiv preprint arXiv:2009.07896*, 2020.

[29] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning (ICML)*, 2019.

[30] Zhizhong Li and Derek Hoiem. Learning without forgetting. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2017.

[31] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[34] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *International Conference on Computer Vision (ICCV)*, 2021.

[35] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022.

[36] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989.

[37] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *International Conference on Computer Vision Workshops (ICCVW)*, 2019.

[38] Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[39] Firat Ozdemir and Orcun Goksel. Extending pretrained segmentation networks with additional anatomical structures. *International Journal of Computer Assisted Radiology and Surgery (IJCARS)*, 2019.

[40] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[42] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[43] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *International Conference on Computer Vision (ICCV)*, 2017.

[44] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning (ICML)*, 2017.

[45] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[46] Onur Tasar, Yuliya Tarabalka, and Pierre Alliez. Incremental learning for semantic segmentation of large-scale remote sensing data. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS)*, 2019.

[47] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018.

[48] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022.

[49] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations (ICLR)*, 2018.

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

[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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[52] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.

[53] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[54] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[55] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *European Conference on Computer Vision (ECCV)*, 2018.

[56] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.