Incrementer: Transformer for Class-Incremental Semantic Segmentation with Knowledge Distillation Focusing on Old Class

Chao Shang, Hongliang Li*, Fanman Meng, Qingbo Wu, Heqian Qiu*, Lanxiao Wang
University of Electronic Science and Technology of China
shangc@std.uestc.edu.cn, hlli@uestc.edu.cn, fmmeng@uestc.edu.cn, qbwu@uestc.edu.cn,
hqqiu@std.uestc.edu.cn, lanxiao.wang@std.uestc.edu.cn

Abstract

Class-incremental semantic segmentation aims to incrementally learn new classes while maintaining the capability to segment old ones, and suffers catastrophic forgetting since the old-class labels are unavailable. Most existing methods are based on convolutional networks and prevent forgetting through knowledge distillation, which (1) need to add additional convolutional layers to predict new classes, and (2) ignore to distinguish different regions corresponding to old and new classes during knowledge distillation and roughly distill all the features, thus limiting the learning of new classes. Based on the above observations, we propose a new transformer framework for class-incremental semantic segmentation, dubbed Incrementer, which only needs to add new class tokens to the transformer decoder for new-class learning. Based on the Incrementer, we propose a new knowledge distillation scheme that focuses on the distillation in the old-class regions, which reduces the constraints of the old model on the new-class learning, thus improving the plasticity. Moreover, we propose a class deconfusion strategy to alleviate the overfitting to new classes and the confusion of similar classes. Our method is simple and effective, and extensive experiments show that our method outperforms the SOTAs by a large margin (5∼15 absolute points boosts on both Pascal VOC and ADE20k). We hope that our Incrementer can serve as a new strong pipeline for class-incremental semantic segmentation.

1. Introduction

Semantic segmentation [4, 5, 36, 41] is one of the fundamental tasks in the field of computer vision, which aims to classify each pixel in an image and assign a class label. Traditional semantic segmentation networks are trained on datasets where labels for all classes are available simultaneously. However, in practical applications, a more realistic situation is that the network needs to continuously learn new classes, while the data containing labels of old classes is not available due to privacy or legal reasons. If the old model is fine-tuned directly on the new data, the network will fit the new data and forget the old knowledge learned before, resulting in catastrophic forgetting [14, 24]. Thus, this problem leads to a challenging task referred as Class-Incremental Semantic Segmentation [2, 8, 28]. The existing methods [2, 8, 32, 48] for this task typically use fully-convolutional network [4, 5] as the basic framework, which uses a CNN backbone as an encoder to extract image features, and then generates segmentation predictions through a convolutional decoder. To prevent catastrophic forgetting, most existing methods [8, 32, 45, 48] preserve the learned old knowledge through knowledge distillation.

*Corresponding authors.

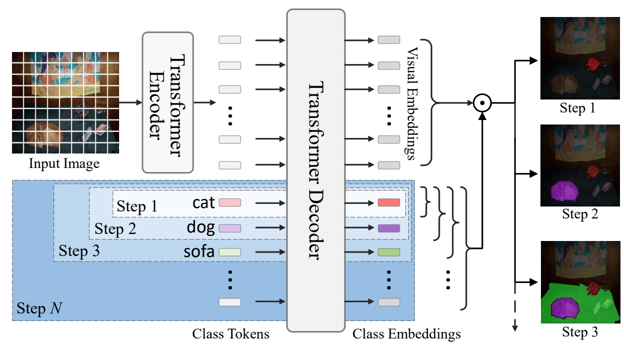

Figure 1. Class-incremental learning process based on transformer, assuming that each step contains one new class, so each step adds one new class token to the decoder.

Although these methods [8, 25, 48] have achieved remarkable performance, there are still some limitations. Firstly, global context information is critical for accurate semantic segmentation, while convolution kernel with the local view is difficult to capture global information. More, in the incremental learning, existing convolution-based methods need to add additional convolution layers into the decoder to predict new classes, and generate segmentations of classes in different steps through different decoding layers, which is inefficient with the increasing number of learning tasks. Secondly, for knowledge distillation, existing methods [8,25,28,45] regard the features generated by the model as a whole and neglect to distinguish the different regions corresponding to the old and new classes. The old model has no ability to discriminate the new-class regions, so it classifies the new classes as background. If the features of all regions are distilled using the old model without discrimination, it will be difficult for the current model to learn more discriminative feature representations for new classes, thus limiting the plasticity of the model.

To address the above problems, we propose a new transformer-based framework for class-incremental semantic segmentation, dubbed Incrementer, which is a new structural paradigm with both high performance and high efficiency. Specifically, we first adopt a vision transformer [7, 39] as the encoder to extract visual features that capture more global contextual information based on the self-attention mechanism. Next, for the decoder, inspired by [36], we assign a class token to each class, and then input the class tokens into a transformer decoder jointly with the patch-wise visual features generated from the encoder, so as to generate corresponding visual embeddings and class embeddings for final segmentation predictions. In incremental learning, our method only needs to add new class tokens for the new classes to the transformer decoder. As shown in Fig. 1, assuming that each step of incremental learning contains one new class, we add a new class token in each step, and the segmentation predictions of both old and new classes can be output in parallel, without adding additional network structures like the CNN-based methods, which improves the efficiency of incremental learning.

Based on the above transformer framework, we propose a new knowledge distillation scheme that only focuses on old classes (FOD). Different from the previous distillation methods that do not distinguish different regions of old and new classes, we separate the image into old and new class regions, and only distill the features corresponding to the old-class regions at both local and global levels. Our distillation scheme not only preserves the capability of the model on the old classes, but also reduces the constraints of the old model on the current model to learn new-class features, thereby improving both the stability and plasticity.

Moreover, we observe that the model overfits new classes when the incremental data contains a small number of new classes and confuses similar old and new classes. Therefore, we further propose a class deconfusion strategy (CDS) to balance the learning of the old and new classes, which reduces the learning weight of the new class, and uses an old-new binary mask to aggregate the scattered old classes in the process of learning new classes, thus alleviating the overfitting to new classes and improving the model's ability to distinguish similar classes.

Our contributions are summarized as follows:

- Structurally, we propose a new transformer-based pipeline for class-incremental semantic segmentation named Incrementer, which is a simple and efficient framework that not only achieves higher accuracy, but also is convenient to implement incremental learning.
- Methodologically, we propose a novel knowledge distillation scheme FOD that focuses on the distillation of the old-class features to improve both stability and plasticity. And we further propose a class deconfusion strategy CDS to alleviate the model's overfitting to new classes and the confusion of similar classes.
- We conduct extensive experiments on Pascal VOC and ADE20k, and the results show that our Incrementer significantly outperforms the state-of-the-art methods.

2. Related Works

Incremental Learning. Deep neural networks suffer from catastrophic forgetting [14, 24] in the process of incrementally learning new classes, and numerous researchers spare no efforts to address this problem in a variety of ways: Replay-based methods either store a small number of old samples [16, 33, 37, 47] or use an additional generator to synthesize fake samples [15, 26, 35] with a similar distribution to the old data, and then train them jointly with the current data. Architectural-based methods [13, 17, 22, 34, 42] dynamically create new architecture branches or provide a subnet for new tasks. Regularization-based methods [3,18,20] measure the importance of parameters to old tasks and design losses to avoid the shift of important parameters. Distillation-based methods [9, 19, 24, 33, 46, 51] utilize the old model from the last step to supervise the current model, such as the supervision of logits [24,33] or intermediate features [9, 50]. Recently, incremental learning is extended to more vision tasks, like object detection [12, 43, 44], semantic segmentation [2, 8, 21, 28], instance segmentation [31].

Class-Incremental Semantic Segmentation. Class-incremental semantic segmentation is also known as continual semantic segmentation, which is first proposed by ILT [28] and builds the method on the fully-convolutional network Deeplab [4], while most subsequent methods follow this framework. MiB [2] further raises the issue of background shift and proposes unbiased knowledge distillation. SDR [29] improves the model's ability to learn new classes by learning discriminative prototypes for different classes. Replay-based RECALL [27] obtains more additional data via GAN or web. For Distillation-based methods, PLOP [8] alleviates forgetting of old knowledge by distilling multi-scale features, and REMINDER [32] assigns different weights to distillation based on class similarities. Architectural-based RC [45] utilizes two parallel networks to store old knowledge and learn new classes respectively. More recently, RBC [48] proposes the effect of context on incremental learning segmentation and decouples different classes through context-rectified image-duplet learning. SPPA [25] preserves the class structure and reduces forgetting by constraining inter and intra class relationships.

Transformers. [39] first proposes transformers for natural language processing (NLP). Since the transformer can obtain more global information by capturing long-distance dependencies, which is also required in computer vision. In recent years, the transformer has been widely used in computer vision tasks, such as image classification [7, 23, 38], semantic segmentation [36, 41], object detection [1], and achieved remarkable improvement. Further, transformers are also used in incremental learning [10, 40]. However, in the field of class-incremental semantic segmentation, existing methods [8, 25, 48] are still based on CNN, which limits the overall performance of this task. In this paper, we apply transformer to class-incremental semantic segmentation, which significantly improves performance while simplifying the structural paradigm of incremental learning.

3. Method

3.1. Problem Formulation

The task of class-incremental semantic segmentation is to perform semantic segmentation in multiple steps, and we assume that there are T steps. In step t, the model is trained on data D^t that only has labels for new classes C^t, where data D^t contains a set of samples and each sample contains an image $X^t \in \mathbb{R}^{3\times H\times W}$ and corresponding ground truth Y^t. Y^t only contains the labels of C^t, and does not contain labels of old class $C^{1:t-1}$. The number of new classes is denoted as $|C^t|$. If any old classes $C^{1:t-1}$ appear in the image X^t, they are classified as background class c^0 in the ground truth Y^t. Each class is only learned once by the model (i.e. $C^{1:t-1} \cap C^t = \emptyset$), so the model needs to keep the ability to segment the old classes $C^{1:t-1}$ while learning to segment the new ones C^t. However, in the process of learning new classes, the model will forget the old classes $C^{1:t-1}$ and fit the new classes C^t due to the lack of the old class labels in the new data, resulting in catastrophic forgetting. And we propose Incrementer to address this problem.

3.2. Incrementer Structure

We present an overview of our proposed Incrementer in Fig. 2. In this section, we first introduce the overall transformer framework employed in our method, and then elaborate the incremental learning process of Incrementer.

Framework. The network of our method can be divided into an encoder and a decoder, both composed of transformers. Given an input image $X \in \mathbb{R}^{3\times H\times W}$, we first split the image into a series of patches with size $P \times P$, the number of patches is $N = HW/P^2$. Then we flatten each patch and project it into a D-dimensional feature vector, and obtain a feature sequence $\mathbf{f} = \{f^1, f^2, ..., f^N\} \in \mathbb{R}^{N\times D}$ with length N, each f^i represents the feature of the corresponding image patch. Next, $\mathbf{f}$ added with the spatial embeddings is input into a vision transformer encoder, through multiple layers of transformers with self-attention, each patch feature in the feature sequence captures rich long-range contextual information, and outputs the final visual feature sequence $\mathbf{f}_v = \{f_v^1, f_v^2, ..., f_v^N\} \in \mathbb{R}^{N\times D}$ for subsequent decoding.

For the decoder, to achieve that the decoder can cope with future incremental classes without adding additional network structure, inspired by Segmenter [36], we assign each class to be predicted a learnable class token and get class token sequence $\boldsymbol{\tau} = \{\tau^0, \tau^1, \tau^2, ..., \tau^M\} \in \mathbb{R}^{(M+1)\times D}$, where τ^i represents the learnable token vector corresponding to class c^i, M represents the number of classes, in incremental learning, $M = |C^{1:t}|$, and τ^0 denotes the token of background c^0. Then we concatenate the tokens $\boldsymbol{\tau}$ with the patch-wise visual features $\mathbf{f}_v \in \mathbb{R}^{N\times D}$ from the encoder, and input the concatenated sequence into a transformer decoder to generate the corresponding visual embeddings $\mathbf{e}_v = \{e_v^1, e_v^2, ..., e_v^N\} \in \mathbb{R}^{N\times D}$ and class embeddings $\mathbf{e}_c = \{e_c^0, e_c^1, ..., e_c^M\} \in \mathbb{R}^{(M+1)\times D}$.

Finally, the segmentation mask of each class c^i is obtained by calculating the similarity between each class embedding e_c^i and the visual embeddings $\mathbf{e}_v$. In incremental learning, to prevent the similarity scores from being biased towards new classes, we use cosine similarity for mask generation. So we first l_2-normalize each embedding in $\mathbf{e}_v$ and $\mathbf{e}_c$, and get $\bar{\mathbf{e}}_v = \{\frac{e_v^1}{||e_v^1||_2}, \frac{e_v^2}{||e_v^2||_2}, ..., \frac{e_v^N}{||e_v^N||_2}\}$ and $\bar{\mathbf{e}}_c = \{\frac{e_c^0}{||e_c^0||_2}, \frac{e_c^1}{||e_c^1||_2}, ..., \frac{e_c^M}{||e_c^M||_2}\}$ and then generate the segmentation masks S' by:

$$S' = \gamma\, \bar{\mathbf{e}}_c \bar{\mathbf{e}}_v^T \tag{1}$$

where $S' \in \mathbb{R}^{(M+1)\times N}$, γ is a hyperparameter used to amplify the peak value after softmax due to cosine similarity in the range of $[-1, 1]$ [16]. We reshape S' back to $(M+1) \times H/P \times W/P$, and then upsample to the size of the input image and use the softmax operation to get the final segmentation prediction $S \in \mathbb{R}^{(M+1)\times H\times W}$.

Class-incremental learning. Based on the above transformer framework, we can flexibly add new class tokens to predict new classes in the incremental learning, and the old and new classes can be predicted in parallel, which is simpler and more efficient and does not need to add additional network structure for new classes like the previous convolution-based methods. In incremental learning step t, for new classes C^t to be predicted, we fix the old class tokens $\boldsymbol{\tau}^{1:t-1}$ as shown in the bottom of Fig. 2 and assign a new class token to each class in C^t, we denote $\boldsymbol{\tau}^t = \{\tau^{|C^{0:t-1}|+1}, \tau^{|C^{0:t-1}|+2}, ..., \tau^{|C^{0:t}|}\} \in \mathbb{R}^{|C^t|\times D}$ as the new class tokens, where $C^{0:t-1}$ includes learned classes $C^{1:t-1}$ and a background class c^0. Then feed $\boldsymbol{\tau}^t$ into the decoder

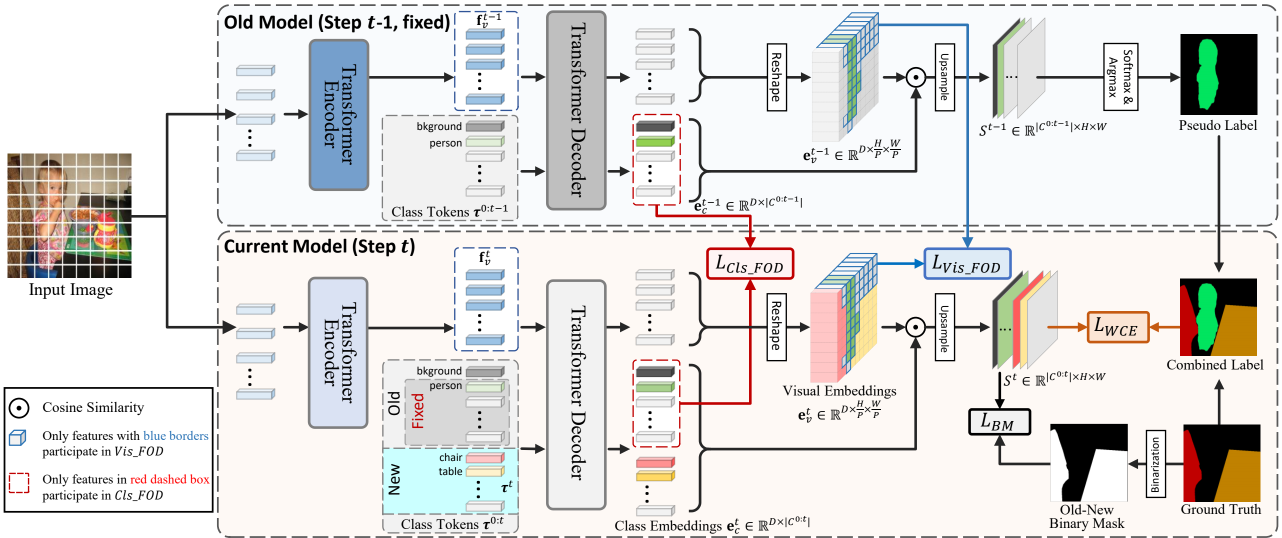

Figure 2. The overview of the proposed **Incrementer**. At each step t, the model only needs to feed the new class tokens into the transformer decoder to generate segmentation predictions for the new classes. The image is first fed into the old model of the last step $t-1$ to generate the pseudo label of the old classes and combined with the current ground truth, and use L_{WCE} to supervise the current segmentation masks. Then use L_{Vis_FOD} and L_{Cls_FOD} to perform knowledge distillation (FOD) on the visual and class embeddings corresponding to the old classes, respectively. Finally, use ground truth to generate the old-new binary mask to deconfuse similar old and new classes.

jointly with the previous token $\boldsymbol{\tau}^{0:t-1}$ and visual feature $\mathbf{f}_v$, and generate segmentation prediction $S^t \in \mathbb{R}^{(|C^{0:t}|) \times H \times W}$.

Since the dataset D^t in step t only contains the labels of the new classes C^t, if the model is fine-tuned directly on D^t, the model will forget the segmentation ability of the old classes, resulting in catastrophic forgetting. Moreover, the learned old classes are treated as background in D^t, which shifts the semantics of the background [2], thus aggravating catastrophic forgetting. Therefore, to address this problem, we adopt the pseudo-labeling [8] w/o entropy thresholds. Specifically, given a sample pair (X^t, Y^t), we first generate the segmentation prediction S^{t-1} for the learned old classes through the old model in the last step. Then, we re-label the background area in the ground truth of current sample Y^t according to the predicted foreground classes in S^{t-1}, and obtain the combined label $\hat{Y}^t$. We use $\hat{Y}^t$ to supervise the current prediction S^t with the weighted cross-entropy loss:

$$L_{WCE} = \frac{1}{HW} \sum_{i=1}^{HW} \sum_{c \in C^{0:t}} \omega_i \hat{Y}^t_{c,i} log S^t_{c,i} \tag{2}$$

where $S^t_{c,i}$ represents the probability score that the model predicts the pixel at position i as class c, and ω_i is a weight used to alleviate model overfitting to new classes, which will be introduced in Section 3.4. Based on the transformer framework, we improve the performance and simplify the network structure paradigm of incremental learning.

3.3. Knowledge Distillation Focusing on Old Class

To further alleviate catastrophic forgetting, existing works propose a variety of knowledge distillation methods [2, 8, 25, 28] to preserve learned knowledge. However, existing feature distillation methods, whether coarse [8, 9] or fine-grained [25, 28], treat the feature map as a whole and neglect to distinguish different class regions. While in semantic segmentation, the old and new classes are corresponding to different regions, and the old model lacks the ability to recognize the new classes and regards the new classes as the background. If all features are distilled by the old model, the new-class features generated by the current model will also be constrained to be similar to the old model, which makes it difficult for the current model to generate more discriminative feature representation for the new classes, thus limiting the plasticity of the model.

Therefore, we argue that not all features in the current model must be distilled by the old model, and we propose a novel knowledge distillation scheme (FOD) that only focuses on distilling the features in the old-class (non-new-class) regions. Specifically, we first perform FOD on the visual embedding features $\mathbf{e}^t_v$. As shown in Fig. 2, we get the old-class regions through the current ground truth and perform knowledge distillation only on the features in the old-class regions (features with blue borders). Since we are using cosine similarity in segmentation generation, we still use cosine similarity as a constraint in the distillation loss to maintain the consistency of similarity measurement and avoid sacrificing plasticity by using hard knowledge distillation loss such as l_2-distance. The knowledge distillation loss of visual embedding based FOD is formulated as:

$$L_{Vis\_FOD} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i (1 - \langle e^t_{vi}, e^{t-1}_{vi} \rangle) \tag{3}$$

where

$$\alpha_i = \begin{cases} 0, & if \ \ argmax\, \hat{Y}_i^t \in C^t \\ 1, & if \ \ argmax\, \hat{Y}_i^t \in C^{1:t-1} \\ \frac{|C^{0:t-1}|}{|C^{0:t}|}, & if \ \ argmax\, \hat{Y}_i^t = c^0 \end{cases} \tag{4}$$

That is, we set the distillation loss weight α to 0 in the new-class regions, set the weight to 1 in the old-class regions obtained according to the refined label $\hat{Y}^t$, and set the weight of the background class to $\frac{|C^{0:t-1}|}{|C^{0:t}|}$ due to the semantics of the background are not completely consistent in the old and new data. $\langle , \rangle$ denotes cosine similarity.

Each embedding e_{vi}^t in $\mathbf{e}_v^t \in \mathbb{R}^{N \times D}$ represents the visual feature of a local patch, so the distillation L_{Vis_FOD} on $\mathbf{e}_v^t$ is the local-level distillation. While class embeddings $\mathbf{e}_c^t \in \mathbb{R}^{|C^{0:t}| \times D}$ captures the global features of each class through the transformer decoder. Thus we further perform knowledge distillation on $\mathbf{e}_c^t$. As shown in Fig. 2, we also only focus on the old classes in $\mathbf{e}_c^t$ (in the red dashed box), and the distillation loss of class embedding based FOD is:

$$L_{Cls\_FOD} = \frac{1}{|C^{0:t-1}|} \sum_{i=0}^{|C^{0:t-1}|} \beta_i (1 - \langle e_{ci}^t, e_{ci}^{t-1} \rangle) \tag{5}$$

where we set the weight of the background class β_0 to $\frac{|C^{0:t-1}|}{|C^{0:t}|}$, and set the weight of other old classes $C^{1:t-1}$ to 1. Our total distillation loss of our proposed FOD is:

$$L_{FOD} = L_{Vis\_FOD} + L_{Cls\_FOD} \tag{6}$$

With the above distillation method at the local and global levels, we preserve the stability of the model for old classes, while reducing the constraints on plasticity for new ones.

3.4. Class Deconfusion Strategy

In the process of incremental learning, we observed that when the new data contains a small number of samples or new classes, especially when learning multiple steps and only one new class per step, which leads to a small number of new classes with a high probability of occurrence and a high concentration, while a large number of old classes has a low probability of occurrence and is scattered. The imbalance between the old and new classes causes the model to overfit the new classes and incorrectly predict non-new-class regions as new class, resulting in false positives. Moreover, if the new data contains similar classes to the old classes, the model will confuse them. We will demonstrate the above phenomenon in Section 4.3.

We propose a class deconfusion strategy (CDS) for this problem. First, when training on data with only a small number of new classes, the new classes occupy a large proportion of the segmentation loss, which reduces the network's attention to the old classes. Thus we reduce the weight of the segmentation loss for the new classes to alleviate overfitting. As shown in Eq. 2, we set the segmentation loss weight $\omega_i = \lambda \sqrt{\frac{|C^t|}{|C^{0:t}|}}$ for new-class ($argmax\, \hat{Y}_i^t \in C^t$), and otherwise to 1, and λ is a hyperparameter.

Second, to solve the confusion of similar new and old classes, we need to improve the network's ability to discriminate between old and new classes, and treat the old and new classes more balanced in the training process. Therefore, we propose to classify all the old classes into one class to improve the concentration of the old classes, and generate an old-new binary mask B^t:

$$B_i^t = \begin{cases} 1, & if \ \ argmax\, \hat{Y}_i^t \in C^t \\ 0, & otherwise \end{cases} \tag{7}$$

And we sum the masks of the new and old classes in the segmentation prediction $S^t \in \mathbb{R}^{|C^{0:t}| \times H \times W}$ respectively along the channel dimension, and obtain two masks predicted for the old $S_O^t \in \mathbb{R}^{1 \times H \times W}$ and new $S_N^t \in \mathbb{R}^{1 \times H \times W}$ class:

$$S_{Oi}^t = \sum\nolimits_{c=0}^{|C^{0:t-1}|} S_{c,i}^t \ ; \quad S_{Ni}^t = \sum\nolimits_{c=|C^{0:t-1}|+1}^{|C^t|} S_{c,i}^t \tag{8}$$

Then, we add a new loss to use the binary mask B^t as the supervision for the old and new class masks S_O^t and S_N^t. We use Dice loss as the objective function, where Dice loss is proposed by [30] to solve the imbalance of the foreground and background in binary segmentation, and our binary mask loss is formulated as:

$$\begin{aligned} L_{BM} = &(1 - \frac{2 \sum_{i=1}^{HW} B_i^t S_{Ni}^t}{\sum_{i=1}^{HW} {B_i^t}^2 + \sum_{i=1}^{HW} {S_{Ni}^t}^2}) \\ &+(1 - \frac{2 \sum_{i=1}^{HW} \tilde{B}_i^t S_{Oi}^t}{\sum_{i=1}^{HW} {\tilde{B}_i^t}^2 + \sum_{i=1}^{HW} {S_{Oi}^t}^2}) \end{aligned} \tag{9}$$

where $\tilde{B}_i^t$ is the non of B_i^t. More effectiveness analyses of our deconfusion strategy will be demonstrated in Section 4.3. The total loss of our method is:

$$L = L_{WCE} + L_{FOD} + L_{BM} \tag{10}$$

4. Experiments

4.1. Experimental Setup

Datasets. We conduct extensive experiments on Pascal VOC [11] and ADE20k [49]. Pascal VOC contains 20 foreground classes with 10,582 images for training and 1,449 images for testing. ADE20k contains 150 classes with 20,210 images for training and 2,000 images for testing.

Incremental Protocols. To evaluate the incremental learning ability, the dataset is divided into different subsets

Table 1. Comparison of class-incremental semantic segmentation results on Pascal VOC under different settings. † denotes results from [8, 48], and ∗ denotes the results re-implemented on our transformer framework.

Method	Frame	19-1 (2 steps)						15-5 (2 steps)						15-1 (6 steps)					
		Disjoint			Overlapped			Disjoint			Overlapped			Disjoint			Overlapped		
		1-19	20	all	1-19	20	all	1-15	16-20	all	1-15	16-20	all	1-15	16-20	all	1-15	16-20	all
EWC† [20]	CNN	23.20	16.00	22.90	26.90	14.00	26.30	26.70	37.70	29.40	24.30	35.50	27.10	0.30	4.30	1.30	0.30	4.30	1.30
ILT† [28]	CNN	69.10	16.40	66.40	67.75	10.88	65.05	63.20	39.50	57.30	67.08	39.23	60.45	3.70	5.70	4.20	8.75	7.99	8.56
MiB† [2]	CNN	69.60	25.60	67.40	71.43	23.59	69.15	71.80	43.30	64.70	76.37	49.97	70.08	46.20	12.90	37.90	34.22	13.50	29.29
SDR† [29]	CNN	69.90	37.30	68.40	69.10	32.60	67.40	73.50	47.30	67.20	75.40	52.60	69.90	59.20	12.90	48.10	44.70	21.80	39.20
PLOP† [8]	CNN	75.37	38.89	73.64	75.35	37.35	73.54	71.00	42.82	64.29	75.73	51.71	70.09	57.86	13.67	46.48	65.12	21.11	54.64
RECALL [27]	CNN	65.20	50.10	65.80	67.90	53.50	68.40	66.30	49.80	63.50	66.60	50.90	64.00	66.00	44.90	62.10	65.70	47.80	62.70
REMIND [32]	CNN	-	-	-	76.48	32.34	74.38	-	-	-	76.11	50.74	70.07	-	-	-	68.30	27.23	58.52
RC [45]	CNN	-	-	-	-	-	-	75.00	42.80	67.30	78.80	52.00	72.40	66.10	18.20	54.70	70.60	23.70	59.40
SPPA [25]	CNN	75.50	38.00	73.70	76.50	36.20	74.60	75.30	48.70	69.00	78.10	52.90	72.10	59.60	15.60	49.10	66.20	23.30	56.00
RBC† [48]	CNN	76.43	45.79	75.01	77.26	55.60	76.23	75.12	49.71	69.89	76.59	52.78	70.92	61.68	19.52	51.60	69.54	38.44	62.14
Joint		77.40	78.00	77.40	77.40	78.00	77.40	79.10	72.56	77.39	79.10	72.56	77.39	79.10	72.56	77.39	79.10	72.56	77.39
MiB*	TransF	80.61	45.17	79.61	79.91	47.70	79.10	74.98	59.90	72.27	78.62	63.10	75.62	66.74	26.32	58.28	72.55	23.14	61.73
RBC*	TransF	80.94	42.05	79.68	80.24	38.79	78.99	77.70	59.06	74.05	78.86	62.01	75.53	69.03	28.37	60.54	75.90	40.15	68.24
Ours	TransF	**82.39**	**64.18**	**82.15**	**82.54**	**60.95**	**82.14**	**81.59**	**62.17**	**77.60**	**82.53**	**69.25**	**79.93**	**81.42**	**57.05**	**76.25**	**79.60**	**59.56**	**75.55**
Joint		83.03	73.56	82.58	83.03	73.56	82.58	83.26	77.97	82.58	83.26	77.97	82.58	83.26	77.97	82.58	83.26	77.97	82.58

Table 2. Comparison on Pascal VOC 10-1 *overlapped* setting.

Method	1-10	11-20	all
MiB [2]	20.0 (-59.8)	20.1 (-52.5)	20.1 (-58.1)
SDR [29]	32.4 (-47.4)	17.1 (-55.5)	25.1 (-53.1)
PLOP [8]	44.0 (-35.8)	15.5 (-57.1)	30.5 (47.7)
RECALL [27]	59.5 (-20.3)	46.7 (-25.4)	54.8 (-23.4)
RC [45]	55.4 (-24.4)	15.1 (-57.5)	34.3 (-43.9)
Joint (CNN)	79.8	72.6	78.2
Ours	**77.62** (-3.36)	**60.33** (-22.58)	**70.16** (-12.42)
Joint (TransF)	80.98	82.91	82.58

for multi-step learning according to the classes. And [2] further proposes different division settings: disjoint and overlap. In the disjoint setting, the data in each step only contains the old classes $C^{0:t-1}$ learned in the previous steps and the current classes C^t, without the future classes, and the old classes are labeled as the background. In the overlap setting, the data of each step further contains future classes, which is more consistent with realistic scenes.

Following the common protocols [2, 8], for Pascal VOC, we evaluate our method on multiple division benchmarks, including: 15-5 (2 steps, first training on 15 classes, then on the 5 new classes), 19-1 (2 steps), 15-1 (6 steps), and more challenging 10-1 (11 steps). For ADE20k, we evaluate on: 100-50 (2 steps), 50-50 (3 steps), 100-10 (6 steps) and 100-5 (11 steps). For metrics, we use mean Intersection over Union (mIoU). Specifically, after retraining T steps, we compute the mIoU of the initial classes C^1 to evaluate the stability, the mIoU of the following steps $C^{2:T}$ to evaluate the plasticity, and the mIoU of all classes to evaluate the overall performance. We further use mean False Positive (mFP, where in each class, FP is the proportion of the area of wrong prediction to the total area predicted to this class, mFP is to average FP of all classes) in ablation studies to evaluate the degree of model overfitting to new classes.

Implementation Details. We build our method on the transformer framework [36], the vision encoder adopts ViT-B/16 [7] pre-trained on ImageNet [6], and the decoder contains two layers of transformer. The input image is cropped to 512×512 following common setting [2, 48]. In the initial step, we train our method on Pascal VOC with learning rate 1e-4 for 32 epochs and ADE20k with 1e-3 for 64 epochs, and the learning rate is half of the initial value in the following steps. For the single-class per step protocols, we reduce the learning rate and iterations in part of steps to prevent overfitting. At step t of incremental learning, we initialize the current model with the old model parameters from step $t-1$, and the old class tokens (except the background) are fixed and the new class tokens are randomly initialized.

4.2. Comparisons with the state-of-the-arts

Pascal VOC. We first perform experiments on Pascal VOC. As reported in Table 1, our method significantly outperforms the previous state-of-the-art methods by a large margin on all three protocols (about 6∼14 absolute points on *all* mIoU). For short-step learning, our method outperforms the previous best by 7.14 (*disjoint*) and 5.91 (*overlapped*) points on *all* mIoU in 19-1 setting, and 7.71 and 7.53 points in 15-5 setting. For the long-step, the superiority of our method is more obvious. In 6 steps setting 15-1, our method outperforms the previous best by 14.15 and 13.85 absolute points on *all* mIoU, and outperforms the previous methods by a large margin on both old and new classes.

Further, we evaluate our method on a longer setting 10-1 *overlapped* (11 steps), which is shown in Table 2. In longer learning steps, our method has a stronger ability to learn new classes, while forgetting much fewer old classes than the previous. Our method outperforms the previous best by 15.36 points on *all* mIoU, and even though the previous best

Table 3. Comparison of class-incremental semantic segmentation results on ADE20k under the *overlapped* setting .

Method	100-50 (2 steps)			50-50 (3 steps)			100-10 (6 steps)			100-5 (11 steps)		
	1-100	101-150	all	1-50	51-150	all	1-100	101-150	all	1-100	101-150	all
MiB [2]	40.52	17.17	32.79	45.57	21.01	29.31	38.21	11.12	29.24	36.01	5.66	25.96
SDR [29]	37.40	24.80	33.20	40.90	23.80	29.50	28.90	7.40	21.70	-	-	-
PLOP [8]	41.66	15.42	32.97	47.75	21.60	30.43	39.42	13.63	30.88	39.11	7.81	28.75
REMINDER [32]	41.55	19.16	34.14	47.11	20.35	29.39	38.96	21.28	33.11	-	-	-
RC [45]	42.30	18.80	34.50	48.30	25.00	32.50	39.30	17.60	38.90	38.50	11.50	29.60
SPPA [25]	42.90	19.90	35.20	49.80	23.90	32.50	41.00	12.50	31.50	-	-	-
RBC [48]	42.90	21.49	35.81	49.59	26.32	34.18	39.01	21.67	33.27	-	-	-
Joint (CNN)	43.90	27.20	38.30	50.90	32.10	38.30	43.90	27.20	38.30	43.90	27.20	38.30
MiB*	46.40	34.95	42.58	52.21	35.56	41.11	42.95	30.80	38.90	40.21	26.59	35.67
Ours	**49.42**	**35.62**	**44.82**	**56.15**	**37.81**	**43.92**	**48.47**	**34.62**	**43.85**	**46.93**	**31.31**	**41.72**
Joint (TransF)	49.79	37.09	45.56	56.43	40.12	45.56	49.79	37.09	45.56	49.79	37.09	45.56

Table 4. Component ablation results on Pascal VOC 15-1 *overlapped* setting. We use the pseudo-labeling based transformer framework in Sec. 3.2 plus VKD as the baseline, gradually add our proposed FOD (including Vis_FOD and Cls_FOD) in Sec. 3.3 and CDS in Sec. 3.4 to get our complete Incrementer.

VKD	Vis_FOD	Cls_FOD	CDS	1-15	16-20	all
✓				73.73	24.49	62.96
	✓			75.27	31.77	65.82
✓		✓		74.88	32.27	65.62
	✓	✓		75.96	39.93	68.23
	✓	✓	✓	**79.60**	**59.56**	**75.55**

method RECALL [27] utilizes additional synthetic data, our method still achieves much higher performance on all three metrics. Even compared with previous methods based on respective joint training, ours only forgets -3.36 points on the initial classes after 11 steps, compared to the previous best of -20.3 points. Meanwhile, our method has a stronger learning ability for new classes with a gap of -22.58 points from joint training, which is much better than the previous with a gap of more than -50 points. This proves that our method not only preserves the stability of the old knowledge, but also improves the plasticity of the new knowledge.

For a more fair comparison, we re-implemented the previous methods on our transformer framework, including the classic method MiB [2] and the previous best method RBC [48], and the comparison is shown at the bottom of Table 1. Based on the transformer, the above two methods achieve performance improvements, but our method still outperforms them by 2$\sim$15 points in the three protocols, especially in long-step learning, which demonstrates the effectiveness of our proposed new knowledge distillation scheme (FOD) and class deconfusion strategy (CDS).

ADE20k. We further perform experiments on the more challenging ADE20k dataset. As shown in Table 3, in the short-step settings (100-50 and 50-50), our method outperforms existing methods on *all* mIoU by more than 9 absolute points. More importantly, in the long-step setting, aside from much higher performance, based on the joint training results of the respective frameworks, our method achieves performance close to the short-step setting on new classes while maintaining less forgetting on old classes, where ours forgets -1.32 points in 100-10 and -2.86 points in 100-5, while the previous best forgets -2.9 and -3.79 points. All above experimental results on the above datasets verify that our proposed transformer framework Incrementer is a powerful and robust pipeline for incremental learning, and our proposed FOD and CDS can balance the stability and plasticity of the model better, especially in long-step learning.

4.3. Ablation Studies

To verify the effectiveness of our proposed knowledge distillation scheme focusing on old classes (FOD) and class deconfusion strategy (CDS), we perform ablation experiments on the 15-1 *overlapped* setting of Pascal VOC.

Component Ablations. Our distillation scheme FOD consists of two parts, knowledge distillation focusing on old-class of visual embedding features (Vis_FOD) and class embedding features (Cls_FOD). To verify the effectiveness of the two distillation schemes, we take the pseudo-labeling based transformer framework introduced in Section 3.2 as the basis, and add vanilla feature knowledge distillation (VKD) as the baseline, where VKD distills all visual embeddings, and the baseline do not reduce the weight of new classes in the segmentation loss. As shown in the first two rows of Table 4, we first compare VKD and our proposed Vis_FOD. Our Vis_FOD not only outperforms VKD by 7.28 points on the new classes, but also 1.54 points higher the on old ones, which shows that compared to distilling all features, distilling only the old-class features is more conducive to stability-plasticity balance. Next, we compare the performance after adding Cls_FOD, as shown in the third and fourth rows of Table 4, we combine Cls_FOD with VKD and Vis_FOD respectively, where Cls_FOD+Vis_FOD is the complete FOD, and Cls_FOD further improves the *all* mIoU by $\sim$2.5 points. Our full FOD improves the method's performance for new classes by over 15 points. Finally, we add CDS to get our full method, and CDS further improves the performance of our method for new classes by nearly 20 points. The above results verify that our proposed FOD and CDS greatly improve the plasticity of the model for new

Table 5. Ablation results of CDS on Pascal VOC 15-1 *overlapped*. We use our Incrementer w/o CDS as the baseline, and add the two parts of CDS (ω and L_{BM}) respectively, to show the impact of ω and L_{BM} on the mIoU of each class and the mFP of the old and new classes.

	S0															S1	S2	S3	S4	S5	mIoU	mFP(↓)		
Class	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	pot	sheep	sofa	train	tv	all	1-15	16-20	all
w/o CDS	81.20	43.15	90.19	66.73	86.00	46.31	86.18	95.33	42.82	71.81	66.69	93.48	89.27	90.45	89.82	50.43	31.86	45.47	38.93	32.99	68.23	9.16	38.76	16.56
w/ ω	92.04	44.16	92.33	75.65	86.35	87.06	89.58	95.98	48.39	78.54	68.60	94.22	89.84	90.46	89.56	29.47	0.75	53.95	44.36	50.73	71.27	9.30	**22.45**	**12.59**
w/ L_{BM}	81.09	43.47	89.64	67.85	86.12	60.72	86.96	95.39	43.88	77.82	65.09	93.24	90.63	90.35	89.76	62.31	83.45	46.24	50.62	48.21	73.67	9.44	32.58	15.22
w/ CDS	85.86	43.42	90.32	70.98	86.37	76.44	88.79	95.77	45.62	79.55	66.53	93.34	91.36	90.14	89.58	62.64	74.11	50.83	56.36	53.86	**75.55**	**9.15**	26.80	13.03

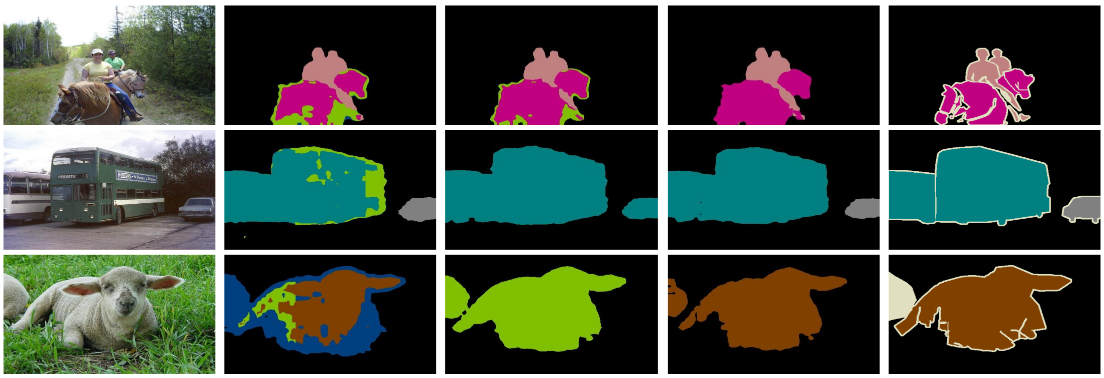

Figure 3. Comparison of visualization results on Pascal VOC 15-1 *overlapped* setting.

classes while improving the stability for old ones.

Analysis of Class Deconfusion Strategy. In the incremental learning process, the performance of new classes is often much lower than joint training. Our proposed FOD significantly alleviates this problem, but there is still a large room for improvement, especially in the setting of long-step with few new classes. To analyze this issue, we first observed the relationship between the old and new classes, and found that if the new class has similarities with an old class, such as 'sheep' and 'cow', 'train' and 'bus', the performance of these similar classes will drop significantly, as shown on the left side of Table 5. Further, we calculated the proportion of false positives (mFP) in the old and new classes, as shown on the right side of Table 5, it can be observed that the mFP of the new classes is much higher than that of the old ones, which means that the model overfits the new classes and incorrectly identifies non-new-class regions as new class. In summary, the model overfits the new classes in the incremental learning process, and generates a large number of false positives, which reduces the mIoU of the new classes, and also confuses the similar old and new classes, resulting in the performance drop of the old classes.

To solve this problem, we propose DCS, including reducing the segmentation loss weight ω for new classes to alleviate overfitting, and using the old-new binary mask loss L_{BM} to improve the ability of the model to distinguish old and new classes. In Table 5, we use our method without CDS as the baseline and add ω and L_{BM} respectively. First, with ω in segmentation loss, the loss weights of the new classes are reduced, so the model's overfitting to the new classes is alleviated, and the mFP of the new classes drops significantly, but it limits the ability of the model to learn new classes, especially similar ones. Second, with L_{BM}, the ability of the model to distinguish similar classes is improved, but it still overfits the new classes with high mFP. Therefore, we combine the above two and get our CDS, which takes their advantages, not only reduces the overfitting of new classes, but also improves the discrimination of similar classes, thus obtaining better overall performance.

Qualitative Results. Fig. 3 shows the visualized segmentation results of the three methods based on transformer. For some classes that are easy to be confused, such as 'bus', 'seep', MiB [2] and RCB [48] either forget or have difficulty to distinguish these classes, resulting in inaccurate segmentation predictions. In contrast, our method can distinguish confusing classes without forgetting old classes and generate more accurate segmentation results.

5. Conclusion

In this paper, we propose a new transformer-based framework, Incrementer, for class-incremental semantic segmentation. Based on this framework, we further propose FOD, a knowledge distillation scheme focusing on old classes, to balance model stability and plasticity. And a class deconfusion strategy (CDS) is proposed to alleviate the model's overfitting to new classes and the confusion of similar classes. Our method outperforms the SOTAs by a large margin on both Pascal VOC and ADE20k datasets.

Acknowledgement. This work was supported in part by National Key R&D Program of China (2021ZD0112001) and National Natural Science Foundation of China (No. 61831005, 62271119 and 61971095).

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[2] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 1, 2, 4, 6, 7, 8

[3] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 2

[4] 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 6

[8] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4050, 2021. 1, 2, 3, 4, 6, 7

[9] 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*, pages 86–102. Springer, 2020. 2, 4

[10] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022. 3

[11] 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*, 111(1):98–136, 2015. 5

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

[13] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017. 2

[14] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1, 2

[15] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. *arXiv preprint arXiv:2203.13104*, 2022. 2

[16] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2, 3

[17] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[18] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, Rao Muhammad Anwer, and Vineeth N Balasubramanian. Energy-based latent aligner for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7452–7461, 2022. 2

[19] Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16071–16080, 2022. 2

[20] 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*, 114(13):3521–3526, 2017. 2, 6

[21] Marvin Klingner, Andreas Bär, Philipp Donn, and Tim Fingscheidt. Class-incremental learning for semantic segmentation re-using neither old data nor old labels. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020. 2

[22] 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*, pages 3925–3934. PMLR, 2019. 2

[23] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 4804–4814, 2022. 3

[24] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2

[25] Zihan Lin, Zilei Wang, and Yixin Zhang. Continual semantic segmentation via structure preserving and projected feature alignment. In *European Conference on Computer Vision*, pages 345–361. Springer, 2022. 1, 2, 3, 4, 6, 7

[26] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. *arXiv preprint arXiv:2207.11213*, 2022. 2

[27] Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7026–7035, 2021. 2, 6, 7

[28] Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1, 2, 4, 6

[29] Umberto Michieli and Pietro Zanuttigh. 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*, pages 1114–1124, 2021. 2, 6, 7

[30] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. IEEE, 2016. 5

[31] Khoi Nguyen and Sinisa Todorovic. ifs-rcnn: An incremental few-shot instance segmenter. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7010–7019, 2022. 2

[32] Minh Hieu Phan, Son Lam Phung, Long Tran-Thanh, Abdesselam Bouzerdoum, et al. Class similarity weighted knowledge distillation for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16866–16875, 2022. 1, 2, 6, 7

[33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2

[34] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 2

[35] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017. 2

[36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1, 2, 3, 6

[37] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2022. 2

[38] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 3

[40] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 3

[41] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1, 3

[42] Qingsen Yan, Dong Gong, Yuhang Liu, Anton van den Hengel, and Javen Qinfeng Shi. Learning bayesian sparse networks with full experience replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 109–118, 2022. 2

[43] Binbin Yang, Xinchi Deng, Han Shi, Changlin Li, Gengwei Zhang, Hang Xu, Shen Zhao, Liang Lin, and Xiaodan Liang. Continual object detection via prototypical task correlation guided gating mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9255–9264, 2022. 2

[44] Li Yin, Juan M Perez-Rua, and Kevin J Liang. Sylph: A hypernetwork framework for incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9035–9045, 2022. 2

[45] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7053–7064, 2022. 1, 2, 6, 7

[46] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13208–13217, 2020. 2

[47] Hanbin Zhao, Hui Wang, Yongjian Fu, Fei Wu, and Xi Li. Memory efficient class-incremental learning for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. 2

[48] Hanbin Zhao, Fengyu Yang, Xinghe Fu, and Xi Li. Rbc: Rectifying the biased context in continual semantic segmentation. *arXiv preprint arXiv:2203.08404*, 2022. 1, 2, 3, 6, 7, 8

[49] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 5

[50] Peng Zhou, Long Mai, Jianming Zhang, Ning Xu, Zuxuan Wu, and Larry S Davis. M2kd: Multi-model and multi-level knowledge distillation for incremental learning. *arXiv preprint arXiv:1904.01769*, 2019. 2

[51] Kai Zhu, Wei Zhai, Yang Cao, Jiebo Luo, and Zheng-Jun Zha. Self-sustaining representation expansion for non-exemplar class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9296–9305, 2022. 2