**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

**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 de-confusion 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 realiz-
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 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 Dt that only has labels for new classes Ct, where data Dt contains a set of samples and each sample contains an image Xt ∈ R3×H×W and corresponding ground truth Yt. Yt only contains the labels of Ct, and does not contain labels of old class C1:t−1. The number of new classes is denoted as |Ct|. If any old classes C1:t−1 appear in the image Xt, they are classified as background class c0 in the ground truth Yt. Each class is only learned once by the model (i.e. C1:t−1 ∩ Ct = ∅), so the model needs to keep the ability to segment the old classes C1:t−1 while learning to segment the new ones Ct. However, in the process of learning new classes, the model will forget the old classes C1:t−1 and fit the new classes Ct 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 ∈ R3×H×W, we first split the image into a series of patches with size P × P, the number of patches is N = HW/P2. Then we flatten each patch and project it into a D-dimensional feature vector, and obtain a feature sequence f = {f1, f2, ..., fN} ∈ RN×D with length N, each ft represents the feature of the corresponding image patch. Next, 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 ft = {f1, f2, ..., fN} ∈ RN×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 τ = {τ0, τ1, τ2, ..., τM} ∈ RM+1×D, where τi represents the learnable token vector corresponding to class ci. M represents the number of classes, in incremental learning, M = |C1:t|, and τ0 denotes the token of background c0. Then we concatenate the tokens τ with the patch-wise visual features ft ∈ RN×D from the encoder, and input the concatenated sequence into a transformer decoder to generate the corresponding visual embeddings et = {e1, e2, ..., eN} ∈ RN×D and class embeddings ec = {e1, e2, ..., eM} ∈ RM+1×D.

Finally, the segmentation mask of each class ci is obtained by calculating the similarity between each class embedding ei and the visual embeddings et, and cosine similarity for mask generation. So we first l2-normalize each embedding in et and ec, and get e̅t = {e̅1, e̅2, ..., e̅N} = {ei/||ei||2, ..., eN/||eN||2} and e̅c = {e̅1, e̅2, ..., e̅M} = {ei/||ei||2, ..., eM/||eM||2} and then generate the segmentation masks St by:

\[ S_t = \gamma \bar{e_t} \bar{e_c}^T \] (1)

where \( S_t \) ∈ R(M+1)×N, \( \gamma \) is a hyperparameter used to amplify the peak value after softmax due to cosine similarity in the range of [-1, 1] [16]. We reshape St back to (M + 1) × H/P × W/P, and then upsample to the size of the input image and use the softmax operation to get the final segmentation prediction \( S \) ∈ R(M+1)×H×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 Ct to be predicted, we fix the old class tokens τ1:t−1 as shown in the bottom of Fig. 2 and assign a new class token to each class in Ct, we denote τt = {τC0:t−1+1, τC0:t−1+2, ..., τC0:t} ∈ R|C|×D as the new class tokens, where C0:t−1 includes learned classes C1:t−1 and a background class c0. Then feed τt into the decoder.
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 \( e'_v \). 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:

\[
L_{V, FOD} = \frac{1}{N} \sum_{t} \alpha_t (1 - \langle e'_v(t), e'_v(t-1) \rangle)
\]
where

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

(4)

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

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

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

(5)

where we set the weight of the background class $\beta_0$ to $\frac{|C^{0:t-1}|}{|C^t|}$, and set the weight of other old classes $C^{1:t-1}$ to 1. Our total distillation loss of our proposed FOD is:

$$
L_{FOD} = L_{Vis,FOD} + L_{Cls,FOD}
$$

(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 $\omega_i = \lambda \frac{|C^t|}{|C^{0:t-1}|}$ for new-class ($\argmax \hat{Y}_i^t \in C^t$), and otherwise to 1, and $\lambda$ 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 $B^t$:

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

(7)

And we sum the masks of the new and old classes in the segmentation prediction $S^t_i \in \mathbb{R}^{C^0|t| \times H \times W}$ 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:

$$
S_O^t = \sum_{i=0}^{|C^{0:t-1}|} S_{c,i}^t; \quad S_N^t = \sum_{i=|C^{0:t-1}|+1}^{|C^t|} S_{c,i}^t
$$

(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:

$$
L_{BM} = (1 - \frac{2 \sum_{i=1}^{HW} B_{i} S_{O_i}^t}{\sum_{i=1}^{HW} B_{i}^2 + \sum_{i=1}^{HW} S_{O_i}^t})
+ (1 - \frac{2 \sum_{i=1}^{HW} B_{i} S_{N_i}^t}{\sum_{i=1}^{HW} B_{i}^2 + \sum_{i=1}^{HW} S_{N_i}^t})
$$

(9)

where $B_{i}$ 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:

$$
L = L_{WCE} + L_{FOD} + L_{BM}
$$

(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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame</th>
<th>19-1 (2 steps)</th>
<th>15-1 (2 steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Disjoint</td>
<td>Overlapped</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-19</td>
<td>20 all</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-15</td>
<td>16-20 all</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1-15</td>
<td>16-20 all</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>1-10 (all)</th>
<th>11-20 (all)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MiB* [2]</td>
<td>20.0 (-39.8)</td>
<td>20.1 (-52.5)</td>
</tr>
<tr>
<td>SDR [29]</td>
<td>32.4 (-47.4)</td>
<td>17.1 (-55.5)</td>
</tr>
<tr>
<td>PLOP [8]</td>
<td>44.0 (-35.8)</td>
<td>15.7 (-57.1)</td>
</tr>
<tr>
<td>RECALL [27]</td>
<td>59.5 (-20.3)</td>
<td>46.7 (-25.4)</td>
</tr>
<tr>
<td>RC [45]</td>
<td>55.4 (-24.4)</td>
<td>15.1 (-57.5)</td>
</tr>
<tr>
<td>Joint (CNN)</td>
<td>77.62 (-3.36)</td>
<td>60.33 (-22.58)</td>
</tr>
<tr>
<td>Ours</td>
<td>83.03 77.97 77.98</td>
<td></td>
</tr>
<tr>
<td>Joint (TransF)</td>
<td>80.98 82.91 82.58</td>
<td></td>
</tr>
</tbody>
</table>

for multi-step learning according to the classes. And [2] further proposes different division settings: disjoint and overlapped. In the disjoint setting, the data in each step only contains the old classes \( C^{t-1} \) 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 \( 512 \times 512 \) following common setting [2, 48]. In the initial step, we train our method on Pascal VOC with learning rate 1e-4 for 30 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 \((\text{disjoint})\) and 5.91 \((\text{overlapped})\) points on all mIoU in 19-1 setting, and 7.71 and 5.73 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 \((\text{overlapped})\) (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.

<table>
<thead>
<tr>
<th>Method</th>
<th>100-50 (2 steps)</th>
<th>50-50 (3 steps)</th>
<th>100-10 (6 steps)</th>
<th>100-5 (11 steps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-100</td>
<td>101-150</td>
<td>all</td>
<td>1-100</td>
</tr>
<tr>
<td>MiB [2]</td>
<td>40.52</td>
<td>17.17</td>
<td>32.79</td>
<td>45.57</td>
</tr>
<tr>
<td>SDR [29]</td>
<td>37.40</td>
<td>24.80</td>
<td>33.20</td>
<td>40.90</td>
</tr>
<tr>
<td>REMINDER [32]</td>
<td>41.55</td>
<td>19.16</td>
<td>34.14</td>
<td>47.11</td>
</tr>
<tr>
<td>RC [45]</td>
<td>42.30</td>
<td>18.80</td>
<td>34.50</td>
<td>48.30</td>
</tr>
<tr>
<td>SPPA [25]</td>
<td>42.90</td>
<td>19.90</td>
<td>35.20</td>
<td>49.80</td>
</tr>
<tr>
<td>RBC [48]</td>
<td>42.90</td>
<td>21.49</td>
<td>35.81</td>
<td>49.59</td>
</tr>
</tbody>
</table>

Joint (CNN) | 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 |

Table 4. Component ablations 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.

<table>
<thead>
<tr>
<th>VKD</th>
<th>Vis_FOD</th>
<th>Cls_FOD</th>
<th>CDS</th>
<th>1-15</th>
<th>16-20</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>73.73</td>
<td>24.94</td>
<td>62.96</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.27</td>
<td>31.77</td>
<td>65.82</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>74.88</td>
<td>32.27</td>
<td>63.62</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>75.96</td>
<td>39.93</td>
<td>68.23</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>79.60</td>
<td>59.56</td>
<td>75.55</td>
</tr>
</tbody>
</table>

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 2~15 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 ~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
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. 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.

![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.](image)

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

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


[23] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvt2: Improved multiscale vision transformers for classification and detection. In Proceedings of the IEEE/CVF.
Zhizhong Li and Derek Hoiem. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947, 2017. 1, 2


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


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


Habib 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

