Self-Training for Class-Incremental Semantic Segmentation

Lu Yu, Xialei Liu, and Joost van de Weijer

Abstract—In class-incremental semantic segmentation, we have no access to the labeled data of previous tasks. Therefore, when incrementally learning new classes, deep neural networks suffer from catastrophic forgetting of previously learned knowledge. To address this problem, we propose to apply a self-training approach that leverages unlabeled data, which is used for rehearsal of previous knowledge. Specifically, we first learn a temporary model for the current task, and then, pseudo labels for the unlabeled data are computed by fusing information from the old model of the previous task and the current temporary model. In addition, conflict reduction is proposed to resolve the conflicts of pseudo labels generated from both the old and temporary models. We show that maximizing self-entropy can further improve results by smoothing the overconfident predictions. Interestingly, in the experiments, we show that the auxiliary data can be different from the training data and that even general-purpose, but diverse auxiliary data can lead to large performance gains. The experiments demonstrate the state-of-the-art results: obtaining a relative gain of up to 114% on Pascal-VOC 2012 and 8.5% on the more challenging ADE20K compared to previous state-of-the-art methods.

Index Terms—Class-incremental learning, self-training, semantic segmentation.

I. INTRODUCTION

SEMANTIC segmentation is a fundamental research field in computer vision. It aims to predict the class of each pixel in a given image. The availability of large labeled datasets [1], [2] and the development of deep neural networks [3] have resulted in significant advancements. The vast majority of semantic segmentation research focuses on the scenario where all data are jointly available for training. However, for many real-world applications, this might not be the case, and it would be necessary to incrementally learn the semantic segmentation model. Examples include scenarios where the learner has only limited memory and cannot store all data (common in robotics) or where privacy policies might prevent the sharing of data (common in health care) [4], [5].

Incremental learning aims to mitigate catastrophic forgetting [6], which occurs when neural networks update their parameters for new tasks. Most works have focused on image classification [4], [5], [7], while less attention has been dedicated to other applications, such as object detection [8] and semantic segmentation [9]–[11]. Recently, MiB [11] achieved the state-of-the-art results on incremental semantic segmentation. Its main novelty is to model the background distribution shift during each incremental training session by reformulating the conventional distillation loss. However, forgetting of learned knowledge is still severe due to the lack of previously labeled data.

An illustration of class-incremental semantic segmentation is shown in Fig. 1. In this example, at task $t$, we have ground-truth annotation (bike and background), where the background contains person (from the previous task $t-1$). The model is required to learn continually and segment all seen objects during inference time, including person, bike, and bus. The main challenge of incremental semantic segmentation is the inaccessibility of previous data. We here consider the scenario where no data of previous tasks can be stored, as is
common in incremental semantic segmentation [10], [11]. Storing of data could be prohibited due to privacy concerns or government regulations and is one of the settings studied in class-incremental learning [5]. To address this problem, we propose to use self-training to exploit unlabeled auxiliary data. Self-training [12]–[14] has been successfully applied to semisupervised learning and domain adaptation. It aims to first predict pseudo labels of the unlabeled data and then learn from them iteratively. To the best of our knowledge, self-training has not yet been explored for incremental learning.

In this work, we propose a self-training framework for incremental semantic segmentation, as shown in Fig. 2. The idea is to introduce self-training of unlabeled data to aid the incremental semantic segmentation task. Specifically, we first train a temporary model with current labeled data, and then, pseudo labels are predicted and fused for auxiliary unlabeled data by both the old and temporary models. We retrain a new model on auxiliary data to mitigate catastrophic forgetting. Simply fusing the pseudo labels from two models causes problems due to conflicting predictions. We show some challenges in Fig. 3 for generating pseudo labels for unlabeled data. Fusing the label information from both models is not straightforward. It is clear that neither the prediction from the old model (second column) nor the temporary model (third column) is ideal. Therefore, we further propose a conflict reduction mechanism in Section III to fuse pseudo labels (last column) to learn a new model. In addition, predicted pseudo labels from neural networks are often overconfident [14], [15], which might mislead the training. We, therefore, propose to maximize self-entropy to smooth the overconfident predictions and reduce the confidence of predictions.

Our main contributions are as follows.

1) We are the first to apply self-training for class-incremental semantic segmentation to mitigate forgetting by rehearsal of previous knowledge using auxiliary unlabeled data.

2) We propose a conflict reduction mechanism to tackle the conflict problem when fusing the pseudo labels from the old and temporary models for auxiliary data.

3) We show that maximizing the self-entropy loss can smooth the overconfident predictions and further improve the performance.

4) We demonstrate the state-of-the-art results, obtaining up to 114% relative gain on the Pascal-VOC 2012 dataset and 8.5% on the challenging ADE20K dataset compared to the MiB method.

II. RELATED WORK

A. Semantic Segmentation

Image segmentation has achieved significant improvements with the advance of deep neural networks [3]. Fully convolutional networks (FCNs) [16] is one of the first works for semantic segmentation to use only convolutional layers and can take any arbitrary size of input images and output segmentation maps. Encoder–decoder-based architectures are popular for semantic segmentation. The deconvolutional (transposed convolutional) layer [17] is proposed to generate accurate segmentation maps. SegNet [18] proposes to use indices of encoder max pooling to upsample the corresponding decoder. In addition, multiresolution information [19], [20], attention mechanism [21], [22], and dilated convolution (atrous convolution) [23]–[25] are further developed to improve the performance. Apart from single-modal data-based semantic segmentation methods, multimodal data fusion-based methods were also proposed by incorporating other input modalities leading to further improvements. RTFNet [26] fuses both RGB and thermal information to perform semantic segmentation.
for autonomous vehicles, DFM [27] developed a benchmark of existing data fusion networks evaluating the fusion of different types of visual features. References [28]–[30] incorporate depth into semantic segmentation via a fusion-based architecture. RoadSeg [31] fuses features from both RGB images and the inferred surface normal information for free-space detection. However, these works assume a static world to learn semantic segmentation with all data available, while our method considers a more realistic setting, where a model has to adapt continually to new tasks.

B. Incremental Learning

The problem of catastrophic forgetting [6] has been studied extensively in recent years when neural networks are required to adapt to new tasks. Most works have been focused on image classification. It can be roughly divided into three categories according to [4], [5], [7]: regularization-focused on image classification. It can be roughly divided into three categories according to [4], [5], [7]: regularization-based [32]–[36], rehearsal-based [37], [38], and architecture-based methods [39]–[41]. Regularization-based methods alleviate the forgetting of previously learned knowledge by introducing additional regularization term to constraint the output embeddings or parameters while training the current task. Knowledge distillation has been very popular for several methods [32], [42], [43]. Rehearsal-based methods usually need to store exemplars (small amounts of data) from the previous tasks that are used to be replayed. Some approaches propose alternative ways of replay to avoid storing exemplars, including using a generative model to do rehearsal. Architecture-based methods dynamically grow the network to increase capacity to learn new tasks, while the old part of the network can be protected from forgetting.

Due to privacy issues or memory limits, it is not always possible to access data from previous tasks, which causes catastrophic forgetting of previous knowledge. In this article, we consider this more difficult setting of exemplar-free CIL, in which the storing of previous task data is prohibited. Unlabeled data are seen as an alternative to secure privacy and mitigate forgetting. There are some works that employ unlabeled data in the context of the continual learning of an image classification system. Zhang et al. [44] proposed to train a separate model with only new data and then use auxiliary data to train a student model using a distillation loss with both new and old models. Lee et al. [45] proposed confidence-based sampling to build an external dataset, while there is no existing work on pixel-wise incremental task leveraging unlabeled data.

Recently, the attention of continual learning has also moved to other applications, such as object detection [8] and semantic segmentation [10], [11]. Shmelkov et al. [8] proposed to use a distillation loss on both bounding box regression and classification outputs for object detection. An incremental few-shot detection (iFSD) setting is proposed in [46], where new classes must be learned incrementally (without revisiting base classes) and with few examples. Michieli et al. [10] proposed to use distillation both on the output logits and intermediate features for incremental semantic segmentation. Recently, MiB (Cermelli et al. [11]) achieves the state-of-the-art performance by considering previous classes as background for the current task and current classes as background for distillation. It is also investigated in remote sensing [9] and medical data [47] for incremental semantic segmentation. In knowledge distillation, one trains the new task while distilling the knowledge of the previous model; as a consequence, the model might have suboptimal performance on the current task. In our approach, we allow the new model to adapt totally to the new task, and then in the second phase, we aim to join the knowledge of both the old model and the current temporary model in a new model.

C. Self-Training

Self-training [12], [48], [49] aims to leverage unlabeled data by computing pseudo labels for it with a teacher model trained on the existing labeled data. Self-training iteratively generates one-hot pseudo labels corresponding to the prediction confidence of a teacher model and then retrains a network based on these pseudo labels. Wang et al. [50] designed a traditional method to generate coarse labels (pseudo labels) and then used the coarse labels to train the existing semantic segmentation networks to achieve results better than traditional methods. Recently, it has achieved significant success on semisupervised learning [48], [51] and domain adaptation [13], [52]. However, the predicted pseudo labels tend to be over-confident, which might mislead the training process and hurt the learning behavior [53]. Different methods to solve noisy label learning, such as label smoothing [15] and confidence regularization [13], [14], are proposed to mitigate this phenomenon. In this work, we explore self-training for learning semantic segmentation sequentially with confidence regularization. Moreover, a conflict reduction mechanism is proposed to fuse the pseudo labels specifically for incremental learning, which is different from ensemble networks [54], [55], where different models are complementary to each other and there are no conflicts between them.

III. PROPOSED METHOD

A. Class-Incremental Semantic Segmentation

Semantic segmentation aims to assign each pixel \((x_i, x_j)(1 \leq i \leq h, 1 \leq j \leq w)\) of image \(x\) a label \(y_{i,j} \in Y, Y \in \{0, 1, \ldots, N - 1\}\), representing the semantic class. Here, \(h\) and \(w\) are the height and width of the input image \(x\), respectively, \(N\) is the number of classes, and we define class 0 to be the background. The setting for class-incremental learning (CIL) for semantic segmentation was first defined in [10], [11]. Training is conducted for CIL along \(T\) different training sessions. During each training session, we only have training data of newly available classes, while the training data of the previously learned classes are no longer accessible. Each session introduces novel categories to be learned.

Specifically, the \(t_{th}\) training session contains data \(T^t = (X^t, Y^t)\), where \(X^t\) contains the input images for the current task and \(Y^t\) is the corresponding ground truth. The current label set \(Y^t\) is the combination of the previous label set \(Y^{t-1}\) and a set of new classes \(C^t\) such that \(Y^t = Y^{t-1} \cup C^t\).
Only pixels of new classes are annotated in $X'$ and the remaining pixels are assigned as background. The loss function for the current training session is defined as follows:

$$L_{cc}(\theta^t) = \frac{1}{|T'|} \sum_{(x,y) \in T'} \ell_{cc}(\theta^t; x, y)$$  \hspace{1cm} (1)$$

where $\ell_{cc}$ is the standard cross-entropy loss used for supervised semantic segmentation and $|T'|$ is the total number of samples in current training session $T'$.

**B. Self-Training for Incremental Learning**

A naive approach to address the class-incremental learning (CIL) problem is to train a model $f_{0}$ on each set $X'$ sequentially by simply fine-tuning (FT) $f_{0}$ from the previous model $f_{0^{t-1}}$. This approach would suffer from catastrophic forgetting, as the parameters of the model are biased to the current categories because no samples from previous data $X^{t-1}$ are replayed. As discussed in the related work, various approaches to prevent forgetting could be considered, such as regularization [32]–[36] or rehearsal methods [37], [38]. Instead, we propose to use self-training. To the best of our knowledge, we are the first to investigate self-training for incremental learning. Our goal is to use unlabeled data to generate pseudo labels for replay. In this way, the models are able to “revisit” the previous knowledge and avoid catastrophic forgetting of previously learned categories.

We present an overview of our framework on incremental semantic segmentation with a self-training mechanism shown in Fig. 4. There are two training steps for each task. We first initialize a temporary model $f_{0}$ from the old model $f_{0^{t-1}}$ (trained in the training session $T^{t-1}$) and update parameters using $X'$ in the training session $T'$. This temporary model is trained to be optimal for the new classes $C'$; however, it forgets the previous classes $Y^{t-1}$. To overcome this problem, during the second step, we combine the knowledge from both the old and temporary models in order to predict all categories we have seen. We generate pseudo label $\hat{P}^{t-1}$ and $\hat{P}^{t}$ by feeding the unlabeled data $A$ to the previous model $f_{0^{t-1}}$ and the current temporary model $f_{0^{t}}$, respectively. The generated pseudo labels of the auxiliary data from both models have the potential to represent all categories the models have encountered.

We require a strategy to fuse the pseudo labels of both models for each image in the auxiliary dataset into a single pseudo-labeled image. We first propose a fusion based on the idea that the predictions from the previous model $\hat{P}^{t-1}$ should be trusted and we only change those background pixels that are considered foreground in the current temporary model $\hat{P}^{t}$. In Section III-C, we improve this fusion of pseudo labels by considering a conflict reduction mechanism.

When $\hat{P}^{t-1}$ and $\hat{P}^{t}$ both consider the pixel as background, we assign background label 0 to the corresponding pixel in the fused pseudo label $\hat{P}^{t}_{A}$ directly; when $\hat{P}^{t-1}$ considers the pixel as background (label is 0), $\hat{P}^{t}$ is equal to $\hat{P}^{t-1}$ since the pixel is likely to be the category learned in the current temporary model $f_{0^{t}}$. If $f_{0^{t-1}}$ considers the pixel as foreground, $\hat{P}^{t}_{A}$ is equal to $\hat{P}^{t-1}$, and we assume that the old model has higher priority than the current temporary model since the previous model is accumulating more and more knowledge. The final fusion of pseudo labels for these auxiliary data $\hat{P}^{t}_{A}$ can be presented as follows:

$$\hat{P}^{t}_{A} = \begin{cases} 0 & \text{if } \hat{P}^{t} = 0 \text{ and } \hat{P}^{t-1} = 0 \\ \hat{P}^{t} & \text{if } \hat{P}^{t} > 0 \text{ and } \hat{P}^{t-1} = 0 \\ \hat{P}^{t-1} & \text{if } \hat{P}^{t} = 0 \text{ and } \hat{P}^{t-1} > 0. \end{cases}$$  \hspace{1cm} (2)$$

Finally, we update the new model (initialized from the old model $f_{0^{t-1}}$) using the auxiliary data $A$ and pseudo label $\hat{P}^{t}_{A}$. We repeat the above procedure until all tasks are learned. For many practical applications, the assumption of an available auxiliary dataset is realistic. For instance, in the application of autonomous driving, there is abundant amount of unlabeled data available for training. In our experiments, we will show results where we use the COCO dataset as the auxiliary dataset for Pascal-VOC 2012 and the Places365 for ADE20K.
in Section IV. We also investigated how unrelated data affect the self-training process.

C. Conflict Reduction

In the above section, we introduce how self-training is adapted in the framework for incremental semantic segmentation to help mitigate catastrophic forgetting of old categories. The pseudo labels of the auxiliary data are generated from the old and temporary models. Then, the two pseudo labels are fused into the final pseudo label directly; however, there might be wrong fusions due to similar categories that are often misclassified (see Fig. 2). Therefore, we propose conflict reduction to further improve the accuracy of pseudo label fusion.

Assume that “bus” is the category added in the training session $T^{t-3}$ ($s \geq 1$) and “train” is learned in the current training session $T^t$ (as shown in Fig. 4). When an image from the auxiliary data containing “train” is fed into the model $f_{t-1}$, as “train” has never been learned in the previous training sessions, the model assigns the maximum probability to the most similar category label “bus” due to the usage of cross-entropy loss. Following (2), the fused pseudo labels are automatically assigned from $\hat{p}^{t-1}$ if the previous model regards the pixels as foreground with no need to check the pseudo label $\hat{p}^t$ obtained from the current temporary model. This results in misclassification and a drop in performance of the overall semantic segmentation system.

In this case, the current temporary model $f_t$ labels the train as “train” very confidently since it just learned to recognize trains from data $X^t$. Conflict frequently occurs when similar categories appear, such as “sheep” and “cow,” “sofa” and “chair,” and “bus” and “train.” Therefore, a conflict reduction module is proposed to obtain better fusion when both the old and temporary models consider the pixel as foreground ($\hat{p}^t > 0$ and $\hat{p}^{t-1} > 0$). Therefore, we update (2) as follows:

$$\hat{p}^t_A = \begin{cases} 0 & \text{if } \hat{p}^t = 0 \text{ and } \hat{p}^{t-1} = 0 \\ \hat{p}^t & \text{if } \hat{p}^t > 0 \text{ and } \hat{p}^{t-1} = 0 \\ \hat{p}^{t-1} & \text{if } \hat{p}^t = 0 \text{ and } \hat{p}^{t-1} > 0 \\ \hat{p}^t & \text{if } \hat{p}^t > 0 \text{ and max}(\hat{q}^t) > \text{max}(\hat{q}^{t-1}) \\ \hat{p}^{t-1} & \text{if } \hat{p}^t > 0 \text{ and max}(\hat{q}^t) < \text{max}(\hat{q}^{t-1}) \end{cases}$$

(3)

where $\hat{q}^{t-1}$ and $\hat{q}^t$ are output probabilities from the old and temporary models, respectively.

D. Maximizing Self-Entropy

Pseudo labeling is a simple yet effective technique commonly used in self-training learning. One major concern of training with pseudo label is the lack of guarantee for label correctness [14]. To address this problem, maximizing the self-entropy loss is explored in this work to relax the pseudo labels and redistribute a certain amount of confidence to other classes. We soften the pseudo label by maximizing the self-entropy loss $L_{se}(\theta^t)$ according to

$$L(\theta^t) = L_{se}(\theta^t) - \lambda \cdot L_{se}(\theta^t)$$

(4)

Algorithm 1: Self-Training for Semantic Segmentation CIL

\begin{algorithm}
\caption{Self-Training for Semantic Segmentation CIL}
\begin{algorithmic}[1]
\State \textbf{Input:} Sequence $T^1, \ldots, T^t, \ldots, T^T$, where $T^i = (X^i, Y^i)$. And auxiliary data $A$.
\State \textbf{Output:} A model $f_{\theta^t}$ after learning $T$ tasks incrementally.
\For{$t = 1, \ldots, T$}
\If{$t = 1$}
\State Train a standard semantic segmentation model $f_{\theta^t}$ using Eq. 1 or Eq. 4 if self-entropy maximization is applied.
\Else
\State Train a temporary model $f_{\theta^t}$ (initialized from the old model $f_{\theta^{t-1}}$) using current data $X^t$.
\EndIf
\State Given $f_{\theta^{t-1}}$ and $f_{\theta^t}$, we introduce auxiliary data $A$. Pseudo labels are predicted by both models represented as $\hat{p}^{t-1}$ and $\hat{p}^t$ respectively.
\State Final pseudo labels $\hat{P}_A^t$ are fused by $\hat{p}^{t-1}$ and $\hat{p}^t$ using the proposed conflict reduction mechanism (Eq. 3).
\State A new model $f_{\theta^t}$ (initialized from the old model $f_{\theta^{t-1}}$) is updated using auxiliary data $A$ and pseudo labels $\hat{P}_A^t$.
\EndFor
\end{algorithmic}
\end{algorithm}

Note that self-entropy loss is applied in different stages of training besides when learning the new model from pseudo labels. For incremental learning, the models are updated for different tasks, and the current new model will become the old model for the next task in the future. Therefore, reducing the overconfident predictions at all stages is crucial for generating more correct pseudo labels for incremental learning.

E. Algorithm

We provide a detailed algorithm of our incremental semantic segmentation procedure in Algorithm 1. For the first task (line 5), it is similar to standard incremental learning methods. For the remaining tasks, our method can be divided into two main parts: pseudo label fusion (lines 7–9) and model retraining (line 10).

IV. EXPERIMENTS

A. Experimental Setups

In this section, we provide details for the datasets, evaluation metrics, implementations, and compared methods. Code will be made available upon acceptance of this article.

1) Datasets: We evaluate all methods using Pascal-VOC 2012 and ADE20K. Pascal-VOC 2012 [1] has 10582 images for training, 1449 images for validation, and 1456 images for testing. Images contain 20 foreground object classes and one background class. ADE20K [2] is a large-scale dataset containing 20210 images in the training set, 2000 images in the validation set, and 3000 images in the testing set. It contains 150 classes of both stuff and objects. For auxiliary
datasets, we choose COCO 2017 train set [56] with 80 classes and 118k images for Pascal-VOC 2012 and Places365 [57] with 365 classes and 1.8M images for ADE20K.

2) Implementation Details: We follow the same implementation as proposed in [11]. For all methods, we use the same framework in Pytorch. The Deeplab-v3 architecture [24] with ResNet-101 [58] backbone is used for all methods. In-place activated batch normalization is used and the backbone network is initialized with an ImageNet pretrained model [59]. We train the network with SGD and an initial learning rate of $10^{-3}$ for the first task and $10^{-3}$ for the rest of the tasks as in [11]. We train the current model on Pascal-VOC 2012 for 30 epochs and on ADE20K for 60 epochs. We train the model with a batch size of 24. Also, we crop the images to 512 $\times$ 512 during both training and test phases. The self-training procedure is trained on unlabeled data for one pass, the tradeoff $\lambda$ between cross entropy and self-entropy is set to 1; 20% of training data are used as the validation set and the final results are on the standard test set.

3) Compared Methods: We consider the FT baseline and the joint training (joint) upper bound. In addition, we compare with several regularization-based methods adapted to incremental semantic segmentation, including ILT [10], LwF [32], LwF-MC [37], RW [38], EWC [33], and PI [34]. We also compare with the state-of-the-art method MiB [11]. In addition, MiB can be further trained with unlabeled data by generating pseudo labels with learned current model, and we denote it as MiB + Aux by leveraging unlabeled data using pseudo labels. All the results are reported in mean intersection-over-union (mIoU) in percentage. It is averaged for all classes of each setting and overlapped settings. MiB is superior to all previous methods but inferior to our method by a large margin. Our method surpasses MiB on overall mIoU by 8.0% for the disjoint setting and 6.7% for the overlapped setting. When MiB is further trained with unlabeled data (MiB + Aux) using generated pseudo labels, it improves by 2.2% for the disjoint setting and 11.3% for the overlapped setting on class 20 while keeping similar performance for the first 19 classes. Note that our method is very close to joint training performance on both settings (75.4%–77.4% and 74.5%–77.4%).

2) Single-Step Addition of Five Classes (15-5): Similar conclusions can be drawn in this scenario. Our method outperforms MiB by 6.6% in the disjoint setting and 2.1% in the overlapped setting. The performance gain due to additional pseudo labels for MiB+Aux is more obvious in this scenario (when compared to the 19-1 setting). However, there is still a big gap compared to our method. This shows that our proposed techniques are more efficient to leverage knowledge from unlabeled data. Our method achieves similar overall results (71.3% and 71.1%) in both settings, showing the robustness of our proposed method.

We report some qualitative results for different incremental methods (Ours, MiB, and FT) on Pascal-VOC 2012 with 15-5 scenario in Fig. 5. The results demonstrate the superiority of our approach. FT totally forgets previously learned classes (first row and third row) but correctly predicts new classes are learned one-by-one resulting in a total of six tasks. For all scenarios, we consider two different settings. Disjoint setting assumes that images of current task only contain the current or previous classes, while Overlapped setting assumes that at each training session, it contains all images with at least one pixel of novel classes. Previous classes are labeled as background for both settings.

| Method       | 19-1 Disjoint | 19-1 Overlapped | 15-5 Disjoint | 15-5 Overlapped | 15-1 Disjoint | 15-1 Overlapped |
|--------------|---------------|----------------|--------------|----------------|--------------|----------------|
| FT           | 58.3          | 6.2            | 6.8          | 12.9           | 7.1          | 1.1            |
| PI           | 5.4           | 14.1           | 7.5          | 14.0           | 7.8          | 1.3            |
| EWC          | 23.2          | 16.0           | 22.9         | 26.9           | 26.5         | 26.7           |
| RW           | 19.4          | 15.7           | 19.2         | 23.3           | 22.1         | 17.9           |
| LwF          | 53.0          | 9.1            | 50.8         | 51.2           | 8.5          | 49.1           |
| LwF-MC       | 63.0          | 13.2           | 60.5         | 64.4           | 13.3         | 61.9           |
| ILT          | 69.1          | 16.4           | 66.4         | 67.1           | 12.3         | 64.4           |
| MiB          | 69.6          | 25.6           | 67.4         | 70.2           | 22.1         | 67.8           |
| MiB + Aux    | 68.6          | 28.0           | 66.6         | 70.2           | 33.4         | 68.3           |
| Ours         | 76.6          | 36.0           | 75.4         | 76.1           | 43.4         | 74.5           |
| Joint        | 77.4          | 78.0           | 77.4         | 77.4           | 72.6         | 77.4           |

B. On Pascal-VOC 2012

We compare different methods in three different scenarios as in [11]; 19-1 means that we first learn a model with first 19 classes and then learn the remaining class as the second task. For 15-5 scenario, there are 15 classes for the first task and the remaining five classes as the second task. For 15-1, the first task is the same as in 15-5 scenario, but the remaining five
classes (second row), while our approach obtains sharper (e.g., person and bike), more coherent (e.g., potted plant), and finer border (e.g., cow) predictions compared to the state-of-the-art method MiB.

3) Multistep Addition of Five Classes (15-1): This is the most challenging scenario of the three. There are five more tasks after learning the first task, and therefore, it is more difficult to prevent forgetting. From Table I, we can observe that in general, all methods forget more in this scenario. MiB only achieves 46.2% and 35.1% for the first task in two settings after learning all tasks, while our method achieves 70.1% and 71.4% of mIoU. Meanwhile, our method outperforms MiB on new tasks by a large margin (21.4% and 26.5%) as well. Overall, we gain 23.3% for the disjoint setting and 33.9% for the overlapped setting. We also see a significant improvement for MiB + Aux compared to the original MiB, from 37.9% to 39.9% for disjoint and from 29.7% to 37.3% for the overlapped setting. Again, without any more expensive annotation processes, unlabeled data are beneficial not only for our proposed method but also very useful for the state-of-the-art method MiB to further boost performance.

C. Ablation Study

In this section, we perform an ablation study on several different aspects, including the proposed techniques, the relationship between performance and the amount

|     | 1-19 | 20 | all | 1-15 | 16-20 | all | 1-15 | 16-20 | all |
|-----|------|----|-----|------|-------|-----|------|-------|-----|
| FT  | 5.8  | 12.3 | 6.2 | 1.1  | 33.6 | 9.2 | 0.2  | 1.8  | 0.6 |
| MiB | 69.6 | 25.6 | 67.4 | 71.8 | 43.3 | 64.7 | 46.2 | 12.9 | 37.9 |
| ST* | 72.5 | 20.0 | 69.9 | 55.0 | 37.1 | 50.5 | 26.9 | 12.7 | 23.3 |
| ST  | 75.9 | 35.8 | 74.8 | 78.0 | 30.9 | 66.3 | 67.9 | 18.6 | 55.6 |
| ST+CR | 76.2 | 34.4 | 75.0 | 76.0 | 53.9 | 70.5 | 69.0 | 28.1 | 58.8 |
| ST+CR, MS | 76.6 | 36.0 | 75.4 | 76.9 | 54.3 | 71.3 | 70.1 | 34.3 | 61.2 |

Fig. 5. Visualization on Pascal-VOC 2012 with 15-5 scenario. From left to right: original image, ground truth, Ours, MiB, and FT. We can see that our method has higher quality predictions compared to state-of-the-art CIL method MiB.
of unlabeled data used, epochs for self-training, and hyperparameter $\lambda$.

1) Impact of Different Proposed Components: In Table II, we conduct an ablation study on the Pascal-VOC 2012 disjoint setting. In the first row, we show the performance of FT on three scenarios. Also, in the second row, MiB is the current state-of-the-art method. When we use self-training (ST) with unlabeled data (2), the performance is largely improved in all three scenarios, which shows the effectiveness of self-training on unlabeled data in incremental semantic segmentation. We also conducted experiments for self-training, assuming that the temporary model has higher priority than the old model for unlabeled data (denoted as ST*) as an ablation study. The results are much worse than ST, which shows that our original assumption results in much better performance. Conflict reduction (ST+CR) further improves the results on all of these three scenarios by 0.6%, 4.2%, and 3.2%. Maximizing self-entropy (ST + CR, MS) strategy further obtains 0.4% better on the 19-1 scenario and 0.8% better results on the 15-5 scenario. On more challenging 15-1 scenario, MS boosts the performance significantly by 2.4%. Note that the gain for the 19-1 scenario is small compared to the other two scenarios because there is only one category (“TV”) for the second task, and it does not contribute much to the overall performance even without learning it. Therefore, for incremental learning, the other two scenarios are more relevant.

2) Amount of Unlabeled Data: We also evaluate the relationship between mean IoU and the size of unlabeled data by randomly selecting a portion of unlabeled data (see Fig. 6). We experiment on disjoint setup for 19-1, 15-5, and 15-1 scenarios. Notably, for 19-1 scenario, our method beats MiB by using only 1% unlabeled data, and the mIoU continually increases when more unlabeled data are used. For 15-5 scenario, our method achieves similar results as MiB by only using 2% of unlabeled data. It keeps improving by increasing unlabeled data and peaks when 70% of unlabeled data are used. A similar conclusion can be observed for the 15-1 scenario, and our method outperforms MiB by a large margin with only 1% unlabeled data. When adding more unlabeled data, the curve goes up consistently until it reaches the best performance, and then, it drops a bit in the end.

3) Number of Self-Training Epochs: Without any specific mention, we only pass all unlabeled data once into the network for efficiency consideration throughout this article in our experiments. In this section, we consider different multiple passes when only 1% unlabeled data are available from the COCO dataset, as shown in Table III. As expected, compared to only passing it once, training for more epochs achieves significant gain for all three scenarios. Specifically, when we increase the self-training epochs from 1 to 20, it improves from 70.1% to 73.1 for 19-1 scenario, from 46.4 to 68.8 for 15-5 scenario, and from 50.5 to 58.1 for 15-1 scenario. It is the most effective to 15-5 scenario because the difficulty of this scenario is between the other two scenarios. It also shows that by training multiple runs using pseudo labels, the performance can be further boosted even in the low-data regime when we have little unlabeled data.

4) Impact of Tradeoff $\lambda$: This parameter controls the strength between the cross-entropy and self-entropy losses. In this section, we report results using various $\lambda$’s on the 15-5 scenario of the disjoint setup. As shown in Table IV, by changing $\lambda$ from 0.1 to 5, the overall performance goes up first and then goes down. When 0.5 and 1 are used, it obtains the best performance. Using $\lambda = 1$ has a similar performance as 0.5 and slightly better on the new task; without any specific mention, $\lambda = 1$ is used throughout this article.

5) Comparability of Two Output Probabilities: To generate pseudo labels for self-training, we fuse labels from the old model and the temporary model by comparing their output
In order to show whether the output probabilities from two different models are comparable, we conduct an ablation study by adding a bias on the probability of the old model before comparing these two. As shown in Table V, our method performs the best with bias \(= 0\), which is adopted throughout this article by default. When we increase or decrease the bias, the performance drops significantly. It shows that it is reasonable to compare the probabilities of different models.

### D. On Different Self-Training Datasets

In this section, we compare several datasets used as self-training datasets for Pascal-VOC to show how sensitive our method is with respect to the unlabeled data. We choose a general-purpose ImageNet validation set, two fine-grained CUB_200_2011 (Birds), and Flowers datasets together with the COCO dataset as different unlabeled data to show how our method performs on Pascal-VOC 15-5 scenario (see Table VI). It is surprising that using ImageNet (validation set) provides similar performance as using the COCO dataset (mIoU: 70.0 versus 71.3) for all classes. It suggests that datasets with diverse categories can be a good option. As expected, it fails on the CUB (mIoU: 10.7) and Flowers (mIoU: 5.1) datasets since fine-grained classes do not contain diverse objects.

We did a preliminary experiment on measuring the relatedness of datasets mathematically with Fréchet inception distance (FID) score [60]. FID is originally proposed to measure the similarity of two distributions, such as generated fake images and real images, which is popular in measuring the quality of generative models. We use FID here to measure the closeness between labeled datasets and unlabeled datasets. We compute FID between Pascal-VOC and other unlabeled datasets, and we obtain an FID score with COCO (25.8), ImageNet (43.8), CUB (153.5), and Flowers (222.3). Smaller FID scores indicate that the distribution of two datasets is closer, which is consistent with performance we obtained for self-training. Therefore, such a measure could be used to select good auxiliary data.

### E. On ADE20K

Following [11], we report the average mIoU of two different class orders on the ADE20K dataset. In this experiment, we only compare with activation-based regularization methods, LwF, LwF-MC, and ILT (much better than PI, EWC, and RW), and the state-of-the-art methods MiB and MiB with auxiliary data (MiB + Aux). Here, we consider three scenarios: 100-50 means 100 classes for the first task and 50 classes for the second task. 100-10 considers 100 classes as the first task and the rest classes are divided into five tasks. Finally, in the 50-50 scenario, 150 classes are equally distributed in three tasks with 50 classes each.

1) **Single-Step Addition of 50 Classes (100-50):** As shown in Table VII, FT forgets the first task totally because of the large number of classes. As seen from Joint, the overall mIoU is 38.9%, which is much less compared to Pascal-VOC 2012, indicating that this is a more challenging dataset. MiB achieves relatively robust results in this scenario, while our method outperforms it by 0.5% in average for 150 classes. Notably, our method prevents the forgetting of previous task effectively but obtains worse performance on the second task compared to MiB. Interestingly, we have seen that for Pascal-VOC dataset, MiB + Aux outperforms baseline MiB in most cases, while it fails to further improve the performance in this case. The reason could be that ADE-20K is more challenging and accuracy itself is much lower than on Pascal-VOC, which could introduce more noise during training. It further shows the importance of our proposed framework to leverage unlabeled data, which leads to superior performance.

We report some qualitative results for different incremental methods (Ours, MiB, and FT) on this scenario in Fig. 7. On this challenging scenario with more categories, our approach is capable of segmenting more objects correctly (e.g., the wall, apparel, floor, painting, box, and computer) than MiB.

2) **Multistep of Addition 50 Classes (100-10):** This is a more challenging scenario with six tasks in total. Most methods fail in this scenario, only achieving about 1.0% mIoU (FT, LwF, and ILT). ILT performs better than LwF-MC in most scenarios on Pascal-VOC but worse results on ADE20K. Our method still outperforms MiB overall by 2.2%. More specifically, the gain is significant for the first four tasks from 1.8% (first task) to 11.3% (third task). Also, the performance of the fifth task is comparable, while Ours obtains worse result for the last task. Again, using auxiliary data for MiB (MiB + Aux) leads to worse results compared to the original MiB, which shows that without specific design, the unlabeled data can be harmful.

3) **Three Steps of 50 Classes (50-50):** This is a more balanced scenario where each training session has the same number of classes. Similar to previous scenarios, we improve the overall mIoU of MiB from 27.0% to 29.0%. Specifically, the gain is 4.5% for the first task and 3.9% for the second task. One general observation for all scenarios is that there is still a large gap between incremental learning and Joint for...
Fig. 7. Visualization on ADE20K with 100-50 scenario. From left to right: original image, ground truth, Ours, MiB, and FT. It is clear that Ours obtains the most similar results to the ground truth.

| Method | 100-50  | 101-150 | all  | 100-10  | 101-110 | 110-120 | 120-130 | 130-140 | 140-150 | 1-50 | 51-100 | 101-150 | all  |
|--------|---------|---------|------|---------|---------|---------|---------|---------|---------|------|--------|---------|------|
| FT     | 0.0     | 24.9    | 8.3  | 0.0     | 0.0     | 0.0     | 0.0     | 16.6    | 1.1     | 0.0   | 0.0     | 22.0    | 7.3  |
| LwF    | 21.1    | 25.6    | 22.6 | 0.1     | 0.0     | 0.4     | 2.6     | 4.6     | 16.9    | 1.7   | 5.7     | 12.9    | 22.8 |
| LwF-MC | 34.2    | 20.5    | 26.3 | 18.7    | 2.5     | 8.7     | 4.1     | 6.5     | 5.1     | 14.3  | 27.8    | 7.0     | 10.4 |
| ILT    | 22.9    | 18.9    | 21.6 | 0.3     | 1.0     | 2.1     | 4.6     | 10.7    | 1.4     | 8.4   | 9.7     | 14.3    | 10.8 |
| MiB    | 37.9    | 27.9    | 34.6 | 31.8    | 10.4    | 14.8    | 12.8    | 13.6    | 18.7    | 25.9  | 35.5    | 22.2    | 23.6 |
| MiB + Aux | 29.0 | 28.0    | 28.7 | 24.0    | 8.3     | 7.4     | 6.2     | 4.0     | 11.9    | 18.9  | 20.2    | 21.5    | 23.5 |
| Ours   | 40.7    | 24.0    | 35.1 | 33.6    | 18.7    | 25.5    | 16.7    | 13.7    | 9.7     | 28.1  | 40.0    | 26.1    | 21.0 |
| 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 |

TABLE VII
MEAN IOU ON THE ADE20K DATASET FOR DIFFERENT INCREMENTAL CLASS LEARNING SCENARIOS

V. CONCLUSION

In this work, we improve incremental semantic segmentation with self-training. Unlabeled data are leveraged to combine the knowledge of the previous and current models. Importantly, conflict reduction provides more accurate pseudo labels for self-training. We have achieved the state-of-the-art performance on two large datasets Pascal-VOC 2012 and ADE20K. We show that our method can obtain superior results on Pascal-VOC 2012 using only 1% unlabeled data. Qualitative results show significantly that more accurate segmentation maps are generated compared to the other methods.

There are still several limitations of our method. As seen from the experimental results, our proposed method works relatively better on VOC than ADE, and one reason is the task difficulty of different datasets. Our method benefits more from a better segmentation model itself, which can provide better pseudo labels. We have also shown that our method can learn from general unlabeled datasets, such as ImageNet; however, large domain differences can still be a challenging problem for using unlabeled data.

As possible future directions, involving human monitors in the loop of using unlabeled data has the potential to reduce the errors in real applications. Moreover, it would be interesting to explore other ways of replay to avoid catastrophic forgetting, such as storing raw exemplars from previous data or generating...
object templates for replay. Besides, the development of the field of pseudo labeling is complementary to our method, which can further improve the performance of the incremental learning setting for semantic segmentation.

REFERENCES

[1] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. (2011). The Pascal Visual Object Classes Challenge 2012 (VOC2012) Results (2012). [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html

[2] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ADE20K dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 633–641.

[3] S. Maia, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image segmentation using deep learning: A survey,” 2020, arXiv:2001.05356.

[4] M. Delange et al., “A continual learning survey: Defying forgetting in classification tasks,” IEEE Trans. Pattern Anal. Mach. Intell., early access, Feb. 5, 2021, doi: 10.1109/TPAMI.2021.3057446.

[5] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, “Continual lifelong learning with neural networks: A review,” Neural Netw., vol. 113, pp. 54–71, May 2019.

[6] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in Psychology of Learning and Motivation, vol. 24. Amsterdam, The Netherlands: Elsevier, 1989, pp. 109–165.

[7] R. Kemker, M. McAlline, A. Abitino, T. Hayes, and C. Kanan, “Measuring catastrophic forgetting in neural networks,” in Proc. AAAI Conf. Artif. Intell., vol. 32, no. 1, 2018, pp. 1–15.

[8] K. Shmelkov, C. Schmid, and K. Alahari, “Incremental learning of object detectors without catastrophic forgetting,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3400–3409.

[9] O. Tasar, Y. Tarabalka, and P. Alliez, “Incremental learning for semantic segmentation of large-scale remote sensing data,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 12, no. 9, pp. 3552–3557, Sep. 2019.

[10] U. Michieli and P. Zanuttigh, “Incremental learning techniques for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW), Oct. 2019, pp. 3205–3212.

[11] F. Cermelli, M. Mancini, S. R. Bulo, E. Ricci, and B. Caputo, “Modeling the background for incremental learning in semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 9233–9242.

[12] K. Nigam and R. Ghan, “Analyzing the effectiveness and applicability of co-training,” in Proc. 9th Int. Conf. Inf. Knowl. Manage. (CIKM), 2000, pp. 86–93.

[13] Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang, “Unsupervised domain adaptation for semantic segmentation via class-balanced self-training,” in Proc. Eur. Conf. Comput. Vis., Sep. 2018, pp. 289–305.

[14] Y. Zou, Z. Yu, X. Liu, B. V. K. V. Kumar, and J. Wang, “Confidence regularized self-training,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 5982–5991.

[15] G. Pereyra, G. Tucker, J. Chowro, L. Kaiser, and G. Hinton, “Regularizing neural networks by penalizing confident output distributions,” in Proc. Int. Conf. Learn. Represent. Workshop, 2017, pp. 1–12.

[16] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 3431–3440.

[17] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in Proc. IEEE Int. Conf. Comput. Vis., May 2015, pp. 1520–1528.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481–2495, Dec. 2015.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2881–2890.

[20] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, “Adaptive pyramid context network for semantic segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7515–7528.

[21] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, “Attention to scale: Scale-aware semantic image segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3640–3649.
[46] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, “Incremental few-shot object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 13846–13855.

[47] F. Ozdemir, P. Fuerststahl, and O. Goksel, “Learn the new, keep the old: Extending pretrained models with new anatomy and images,” in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. Cham, Switzerland: Springer, 2018, pp. 361–369.

[48] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in Proc. Adv. Neural Inf. Process. Syst., 2005, pp. 529–536.

[49] D.-H. Lee et al., “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Proc. Workshop Challenges Represent. Learn., Int. Conf. Mach. Learn., vol. 3, no. 2, 2013, pp. 1–6.

[50] H. Wang, Y. Sun, and M. Liu, “Self-supervised drivable area and road anomaly segmentation using RGB-D data for robotic wheelchairs,” IEEE Robot. Autom. Lett., vol. 4, no. 4, pp. 4386–4393, Oct. 2019.

[51] I. Triguero, S. García, and F. Herrera, “Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study,” Knowl. Inf. Syst., vol. 42, no. 2, pp. 245–284, Feb. 2015.

[52] M. Chen, K. Q. Weinberger, and J. Blitzer, “Co-training for domain adaptation,” in Proc. Adv. Neural Inf. Process. Syst., 2011, pp. 2456–2464.

[53] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, “Label refinery: Improving ImageNet classification through label progression,” 2018, arXiv:1805.02641.

[54] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, “An ensemble of fine-tuned convolutional neural networks for medical image classification,” IEEE J. Biomed. Health Informat., vol. 21, no. 1, pp. 31–40, Jan. 2017.

[55] Z. Fan et al., “Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement,” Coatings, vol. 10, no. 2, p. 152, Feb. 2020.

[56] T.-Y. Lin et al., “Microsoft coco: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., Springer, 2014, pp. 740–755.

[57] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: A 10 million image database for scene recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 6, pp. 1452–1464, Jun. 2018.

[58] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[59] S. R. Bulo, L. Porzi, and P. Kontschieder, “In-place activated BatchNorm for memory-optimized training of DNNs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5639–5647.

[60] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local nash equilibrium,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6626–6637.