Evidential Deep Learning for Class-Incremental Semantic Segmentation

Karl Holmquist[1], Lena Klasén[1,2], and Michael Felsberg[1]

[1] Linköping University, Sweden
[2] Sweden Office of the National Police Commissioner, The Swedish Police Authority

Abstract. Class-Incremental Learning is a challenging problem in machine learning that aims to extend previously trained neural networks with new classes. This is especially useful if the system is able to classify new objects despite the original training data being unavailable. While the semantic segmentation problem has received less attention than classification, it poses distinct problems and challenges since previous and future target classes can be unlabeled in the images of a single increment. In this case, the background, past and future classes are correlated and there exist a *background-shift*.

In this paper, we address the problem of how to model unlabeled classes while avoiding spurious feature clustering of future uncorrelated classes. We propose to use Evidential Deep Learning to model the evidence of the classes as a Dirichlet distribution. Our method factorizes the problem into a separate foreground class probability, calculated by the expected value of the Dirichlet distribution, and an unknown class (background) probability corresponding to the uncertainty of the estimate. In our novel formulation, the background probability is implicitly modeled, avoiding the feature space clustering that comes from forcing the model to output a high background score for pixels that are not labeled as objects. Experiments on the incremental Pascal VOC, and ADE20k benchmarks show that our method is superior to state-of-the-art, especially when repeatedly learning new classes with increasing number of increments.

Keywords: Class-incremental learning · Continual-learning · Semantic Segmentation.

1 Introduction

Semantic segmentation is a challenging fundamental problem in computer vision with applications to many real-world tasks which require detailed knowledge regarding the surrounding environment. While the introduction of new architectures e.g., Convolutional Neural Networks (CNNs) [4,13] and transformers [19,36], as well as large-scale annotated datasets [9,37], has led to significant improvements in semantic segmentation, the models are typically constrained by pre-defined sets of classes. Thus, if current semantic segmentation methods are to be extended to new classes, retraining of the entire network and availability

of both the old and the new training data, fully annotated with all classes, is required.

Instead, Class-incremental Semantic Segmentation (CISS) [2,10,32] aims at expanding the set of known classes of a trained model by continual learning, addressing the problem of adding new classes to an existing network while avoiding *catastrophic forgetting* that causes the performance on the old classes to degrade.

The topic of continual learning has been studied in various fields, such as image classification [1,12,18], object detection [17], robotics [14], and long-term visual tracking [33]. More recently, continual learning has also been applied to semantic segmentation in the form of class-incremental semantic segmentation [3,7,20,23].

While standard semantic segmentation operates under a closed-world assumption where all possible classes are known during training. CSS is based on an open-world assumption [8], in which the background class is a mixture of different unlabeled classes, both previously learned and future classes, and a
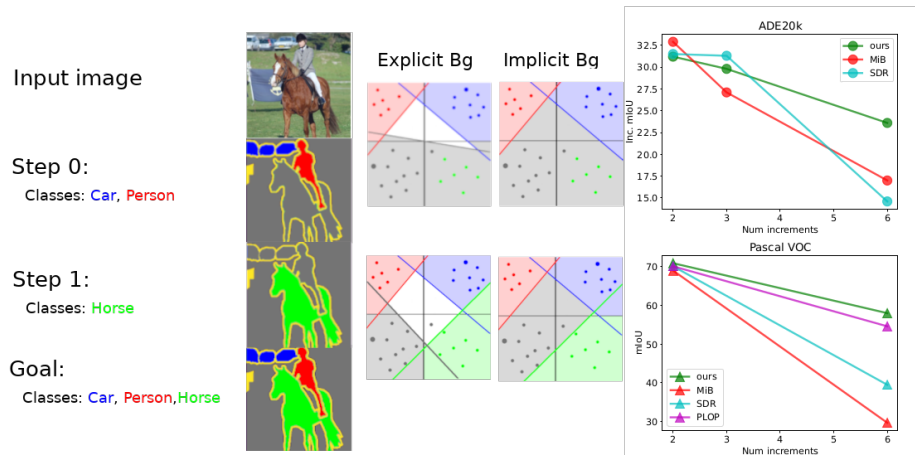

Fig. 1: The leftmost part of the figure illustrates the task, which is to incrementally learn all of the classes, *car*, *person*, *horse*, and the *background*. In the first step, the training data contains labels of the first two classes and in the second step only of the third class with the previous two marked as background. As illustrated in the middle using a linear classifier, an explicit background model requires a large change to separate the previous background class into the two subclasses, *horse* and *background.*. Contrary to our proposed implicit background model which classifies uncertain parts of the feature space as background, an explicit background model instead interpolate the class from the closest linear classifier. The right-most part of the image shows the performance of the current state-of-the-art models based on the number of increments in each task. Our method is clearly more robust to the increasing number of incremental steps in the learning.

generic background class. The open-world setting makes the common explicit background modeling approach less suitable since the weight layer of the background class requires large changes between the increments to handle the *background shift*, see Fig. 1.

Previous methods have addressed this problem using in-painting techniques [7] and by modeling the background as a mixture of a background class and multiple unlabeled classes [3]. However, both of these approaches are based on explicit background modeling, which clusters future classes together into the background class.

Contrary to previous works, our approach avoids this confusion of future classes and background by implicitly modeling the background as the absence of evidence for any of the known labeled classes, facilitating repeatedly learning new classes without the degradation of results seen in current state-of-the-art methods.

Our main contributions are:

– We introduce a novel approach to implicitly model the background based on Evidential Deep Learning for Class-incremental semantic segmentation.
– We perform multiple ablation studies to validate our approach, which gives evidence about its soundness.
– We perform exhaustive large-scale experiments and show that our method is superior to the state-of-the-art on class-incremental Pascal VOC and ADE20k, especially for a larger number of increments.

2   Related work

Continual Learning: The most prevailing type of approach to deep learning, especially for vision-based segmentation and classification, is done in a batched fashion, where all data and labels are provided before the learning starts. Traditionally, this has prohibited incrementally adding new knowledge to an already trained model since the previous knowledge tends to be easily forgotten and cause catastrophic forgetting.

Class-incremental learning was first introduced for classification where new classes are added to the set of known classes of an image classification model [15]. Since then, multiple approaches to this type of continual learning have been proposed. Most of these methods employs one or multiple of the following approaches: regularization-based approaches [1,12], topology-preserving approaches [27,26,31], or rehearsal-based approaches [21,24]. Regularization-based approaches can be separated into distillation-based methods and penalty-computing methods [23].

Distillation-based methods [11,15,34,35] use knowledge distillation to transfer knowledge from one or multiple *teachers* trained on previous tasks [6] to the new *student* network. Knowledge distillation is commonly used in most approaches above and can be applied both on the output level and the internal feature level [7] often in the form of a cross-entropy or ℓ_2 loss.

Penalty-computing methods aim at directing the parameter updates so that the new knowledge is interfering as little as possible with previous knowledge [1,12] and topology-preserving methods have a similar aim to feature-level distillation-based approaches in that they preserve the feature-space topology [27,26].

The other common category is formed by rehearsal-based approaches, which either maintain a small sample set of previous tasks [21,24] or use a Generative Adversarial Network (GAN) for generating samples of previous tasks [16,29,30].

Class-Incremental Semantic Segmentation: Recently, the class-incremental problem was formulated for semantic segmentation [3,23,22], which not only encounters the issue of catastrophic forgetting, but also *background shift* [7]. The background shift arises due to the inconsistent interpretation of background between separate increments, which contains also the respectively unlabeled classes.

One of the first methods to address this issue together with catastrophic forgetting was proposed by Cermelli et al. [3] who modeled the background as a mixture of the background class and unlabeled classes. However, this explicit background model constrains future classes to be similar in feature space. We argue that implicit modeling of the background class is the most appropriate formulation to address an open-world assumption on the class sets, enabling both previous and future classes to be part of the current background class.

Unlabeled previously seen classes was partly addressed by the pseudo-labeling in PLOP [7], who used an uncertainty measure based on median entropy to decide if the model was confident enough to use the predicted label or not. In addition to this, they also expanded the knowledge-distillation term used by [3] to a multi-scale local distillation.

Another aspect of importance to allow continual learning is the underlying learned feature space. SDR, based on representation learning [23] applied a clustering approach and a sparsity loss to improve the feature space and thus facilitate continuous learning.

In addition to the approaches above, RECALL [20] applies either a web crawler or a GAN to expand the training data with examples of previous classes and annotate these using a pseudo labeling approach based on the previous model. This significantly improved the performance when using multiple increments and a few classes per increment.

Our method is independent of the network architecture and maintains a simplistic distillation loss while it does not use any extra data as in [20]. However, expanding our method with these components would be straightforward.

Evidential Deep Learning (EDL) was proposed by Sensoy et.al. [25] as a more computationally efficient method compared to Bayesian inference and ensemble methods for estimating the epistemic uncertainty, i.e. the uncertainty of the predictions of a neural network. Instead of training multiple models and estimating the variance in their predictions, a single model is trained to output the parameters of a probability distribution, the normal distribution for regression problems, and the Dirichlet distribution for classification. The use of EDL has been shown to be useful for open-world action recognition settings [28].

EDL for classification problems is based on subjective logic, which assigns a belief, b_i, to each singleton (each class in our case) of a *frame of discernment*, i.e. the set of possible classes. These beliefs correspond to an observer's belief of each of the classes being true. However, the observer is normally not completely confident in their belief, which leads to an additional amount of uncertainty, u. The total belief and uncertainty mass is defined to always be one, i.e.

$$u + \sum_i b_i = 1. \tag{1}$$

To model these beliefs, they are formulated in terms of the concentration parameters of a Dirichlet distribution, which PDF lies on an Nd-simplex. The model is trained to predict scores, z_i, for each of the classes. These scores are rectified, normally by a ReLU or an exponential function, and offset to calculate the concentration parameters of a Dirichlet distribution,

$$\alpha_i = \text{Rectifier}(z_i) + 1 = e_i + 1. \tag{2}$$

To reconnect with the subjective logic heritage, the belief mass and the uncertainty mass are defined in terms of the evidence, e_i, as

$$b_i = \frac{e_i}{\sum_k (e_k + 1)}, \; u = \frac{K}{\sum_k (e_k + 1)}. \tag{3}$$

The class probabilities are obtained from the Dirichlet distribution as,

$$p_i^{\text{fg}} = \frac{\alpha_i}{\sum_k \alpha_k}. \tag{4}$$

In this paper, we adapt the Evidential Deep Learning framework to the continual semantic segmentation problem and formulate the training to account for the incremental nature, which to the best of our knowledge, has not been done before.

3   Method

The Semantic Segmentation task consists of predicting a pixel-wise semantic classification from a set of known classes, $\mathcal{C}$, i.e. given an RGB image $\mathcal{X} \in \mathrm{R}^{H \times W \times 3}$ the model should correctly predict the label image $\mathcal{Y} \in \mathcal{C}^{H \times W}$.

Class-Incremental Semantic Segmentation (CSS) aims at incrementally expanding the known set of classes at step t, $\mathcal{C}^t$, with new classes $\mathcal{C}^{t+1}$ so that the model learns the full class set $\mathcal{C}^{0:t} := \bigcup_{k=0}^{t} \mathcal{C}^k$. However, while **standard** Semantic Segmentation datasets contain all known classes, fully labeled, CSS only provides labels for the current class set, $\mathcal{C}^t$, and all previous and future class sets are labeled as background. This introduces a drift in the appearance of the background known as *background shift*. While recent methods have addressed the background shift [3,7], their choice to model the background as an

explicit class means that future classes are trained to output a high activation for the background class despite it being an incorrect prediction of the completely trained model.

Our proposed method uses implicit modeling of the background class, avoiding that the background is classified as foreground, while not constraining the future class labels.

3.1   Implicit background modeling

In this paper, we formulate the background class as the model dependent epistemic uncertainty estimate based on EDL, avoiding large updates of the background class and the assumption that no future classes are part of the background.

Formally, we learn a model, $\mathcal{M}$, which predicts the per-pixel class-evidence, $\mathcal{E}$, from an image, $\mathcal{X}$, i.e. $\mathcal{M} : \mathcal{X} \to \mathcal{E}$. The class evidence, e, is used to calculate the concentration parameters, α, of the underlying Dirichlet distribution, from which both the uncertainty and the predictive probability are calculated. We directly interpret the epistemic uncertainty as the background probability, i.e. $p_{\mathrm{bg}} = u$.

Semantic Segmentation using EDL: Given $\mathcal{X}$, we predict e_i for each class at each pixel. To interpret these scores as the concentration parameters of a Dirichlet distribution, they must be strictly larger than one. While the original formulation uses a ReLU function [25], we found that an exponential term [28] performs better. Based on the results in our ablation study (see Tab. 5a) we further found that combining an exponential function with a sigmoid, σ, to suppress the response from negative activation, performed best. Based on these findings, we

(a) RGB image   (b) $P_{\mathrm{Bkg}} = u$ (c) P_{Bicycle} (d) P_{Person} (e) Pred.

(f) GT labels   (g) MiB - P_{Bkg} (h) MiB - P_{Bicycle} (i) MiB - P_{Person} (j) MiB - Pred.
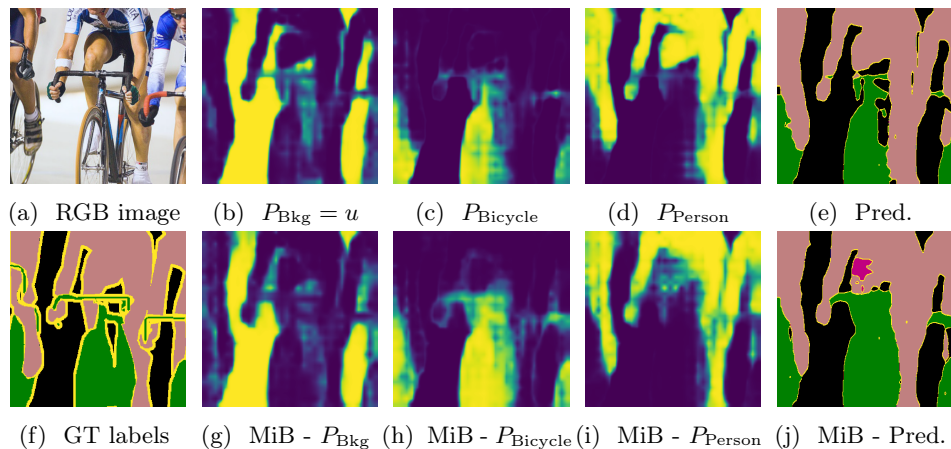
Fig. 2: Qualitative example comparing our proposed method to explicit background modeling, MiB [3].

calculate the concentration parameters of the Dirichlet distribution, α, according to

$$\alpha_i = \exp(e_i)\sigma(e_i) + 1. \tag{5}$$

From the Dirichlet parameters, the foreground probabilities and the epistemic uncertainties are calculated according to 3 and 4. The final predictive probability is factorized into the uncertainty u, corresponding to the probability of *not* being foreground (i.e. background probability), and the foreground probabilities, p_i^{fg}. To calculate the final foreground probabilities, the probability of *not* being background and the individual foreground probabilities are multiplied as follows,

$$p_i = \begin{cases} u, & i = 0(\text{bg}) \\ (1-u)p_i^{\text{fg}} & else \end{cases}. \tag{6}$$

3.2   Training

The training is separated into multiple sections for learning new classes, maintaining old class knowledge and balancing the increments. To furhter highlight the modeling part of our method and facilitate comparison with similar works, we keep the training as straightforward as possible.

Learning new classes: The training objective for learning new classes is the pixel-wise cross-entropy loss over the currently active classes, $\mathcal{C}^t$.

$$\mathcal{L}_{\text{new}} = -\sum_{i \in \mathcal{C}^t} y_i \log p_i, \tag{7}$$

where y is the one-hot encoded ground-truth label for a certain pixel. The final loss is averaged over all pixels.

Output-level Knowledge Distillation: The learned class knowledge is maintained by two output-level knowledge distillation losses [11] between the current probability, $p^{(t)}$, and the output from the teacher model trained on the previous increment, $p^{(t-1)}$. The first loss is the foreground multinomial cross-entropy loss that maintains the intraclass variations (8). The second loss is a binary cross-entropy loss, which maintains the uncertainty with respect to the previous classes (9). These two knowledge-distillation terms works as regularization terms to maintain the hidden knowledge of the previous model.

$$\mathcal{L}_{\text{KD-fg}} = -\sum_{i \in \mathcal{C}^{1:t-1}} p_i^{\text{fg},(t-1)} \log p_i^{\text{fg},(t)} \tag{8}$$

$$\mathcal{L}_{\text{KD-u}} = -u^{(t-1)} \log u^{(t)} + (1 - u^{(t-1)}) \log(1 - u^{(t)}) \tag{9}$$

Increment balancing: While the probabilities received by using softmax are independent of the number of classes, the formulation of EDL introduces a dependency between the number of classes and the final probabilities. To maintain the predictive foreground probability when adding new classes, the activation of each increment needs to be scaled dependently on the number of classes in the step compared to the total number of classes. Since the final number of classes is not necessarily known during the beginning of training, we formulate this compensation term for inference only.

The scaling is applied to the maximum activation in each increment that is larger than zero to not scale negative activations. By requiring the intraclass set probability and the probability after scaling to be the same (the derivation can be found in the supplementary material), we get the following scale factor,

$$o = \frac{(2K^t - 1)(K^T)^2}{(K^t)^2(2K^T - 1)}. \tag{10}$$

We apply this increment balancing term to all ADE20k [37] results because of the large number of classes and the difference between the number of base classes and incrementally added new classes.

Foreground-background balancing: Compared to Pascal VOC, the classes of ADE20k varies much more in size (average number of pixels per image). For the CISS setting this leads to large variations of the foreground-background ratio of the different increments, especially for certain class-orders. To adress this, we apply a class-balancing term between foreground and background regions by weighting the loss for the corresponding set of pixels according to the inverse ratio of pixels in the current batch. For training stability, we empirically found that clamping this weight factor to a maximum weight of 10 worked well, i.e. the weight is calculated as

$$w_i = \min \left(10, \begin{cases} \frac{N_{\text{pixels}}^{\text{tot}}}{N_{\text{pixels}}^{\text{bg}}} & i = 0 \\ \frac{N_{\text{pixels}}^{\text{tot}}}{N_{\text{pixels}}^{\text{fg}}} & \text{else} \end{cases} \right). \tag{11}$$

4   Experiments

We compare our method with recent state-of-the-art CSS methods on two datasets, the Pascal VOC 2012 [9] and the ADE20k [37] datasets. Multiple variants of the incremental learning task have been proposed in terms of the number of incremental steps, the number of classes in each step, and which classes that might be unlabeled in the background at each step. The *overlapped* scenario is the most realistic one, allowing both future and past classes to be labeled as background. This is what would be the usual case if the data collection is done incrementally since the data will contain classes that might be later annotated. The *disjoint* scenario is arguable less common practical use cases but is more challenging in terms of maintaining old class knowledge.

From a practical perspective, the overlapped scenario contains more data samples per increment since all images showing any of the current labels are used. In the disjoint scenario, only images that do not contain any of the future classes are used, which means that none of the samples will be reused, see Tab. 1 for a comparison.

It is worth noting that ADE20k is only evaluated on the pseudo-disjoint setting proposed by [3]. This setting is the same as a disjoint-setting but with the additional requirement that a minimum number of images of each class is included in each increment.

Setting	Bg can contain prev. classes	Bg can contain future classes	Reused images w/ different label
Overlapped	Yes	Yes	Yes
Disjoint	Yes	No	No

Table 1: Description of the two setting used on Pascal VOC

Formal problem definition : Let the total number of increments be T and a specific increment, t, while the current set of classes is notated as $\mathcal{C}^t$ and the current training set as $\mathcal{D}^t$. In this case, all sets of classes are disjoint, i.e.

$$\mathcal{C}^i \cup \mathcal{C}^j = \emptyset, i \neq j. \tag{12}$$

All labels in the dataset are part of the current class set, $\mathcal{Y}^t \in \mathcal{C}^t$, with previous and future classes labeled as *background*.

In addition to these class-incremental settings, the *joint* setting is the standard semantic segmentation setting in which all classes are learned in a single increment.

4.1 Evaluation

Our method is quantitatively evaluated using seven different setups [3] on two datasets, the Pascal VOC 2012 [9] and the ADE20k [37], listed in Tab. 2.

Pascal VOC is a widely used dataset for semantic segmentation, which contains 20 annotated foreground classes and one background class for unannotated pixels. The foreground objects are all *things*, i.e *person*, *car*, and *airplane*. We evaluate our method following four of the PASCAL setups in the evaluation protocol [3]. These consist of a shorter incremental setup with a single step, *15-5*, where the first 15 and the last 5 classes are learned in different steps, and the more complicated *15-1* setup where the last 5 classes are learned separately in different steps, resulting in a total of 6 steps. Both of these setups are evaluated in the *overlap* and *disjoint* settings.

ADE20k is, compared to the Pascal dataset, a more challenging dataset with 150 classes of both *thing* classes and *stuff* classes (i.e. *grass*, *sky*) but without an explicit background class. The challenging nature can be seen from the fact that the current state-of-the-art of semantic segmentation achieves a mean Intersection-over-Union (mIoU) score of 53.5 on the validation dataset [19] while it is 90.5 on Pascal [38]. While the Pascal dataset is reasonably class-balanced, ADE20k is a class-imbalanced dataset, since it contains both large *stuff* classes (i.e. *building*) and smaller *thing* classes (i.e. *fan*). This class imbalance has a major impact on how well the different increments can be learned. Because of this, the ADE20k evaluation protocol [3] evaluates each scenario on two different class orders, the original one in decreasing frequency order and a random order.

We follow the evaluation protocol proposed by [3] and evaluate on three distinct scenarios, *100-50* (2 steps), *100-10* (6 steps), and *50-50* (3 steps). Similar to the naming of the Pascal tasks, the first number is the number of classes in the base step and the second is the number of classes in each increment after that until all 150 classes have been learned.

While PLOP [7] evaluates on ADE20k, they do not evaluate their method on multiple splits and do not use the same disjoint setting as MiB [3]. Therefore, the numbers in [7] cannot be compared to the results in [3,23] and here.

Dataset	Pascal		ADE20k		
Task	15-5	15-1	100-50	50-50	100-10
Number of Increments	2	6	2	3	6
New classes per increment	5	1	50	50	10

Table 2: Table illustrating some of the properties of the different tasks.

Ablation study We perform an ablation study on the choice of the rectification function on the Pascal 15-5 overlap scenario, the choice of EDL for implicit background modeling, and the importance of class balancing on ADE20k 100-10 in Sec. 5.

4.2   Metrics

The evaluation metric for ADE20k and Pascal is the class-wise mean Intersection-over-Union (mIoU), which is also the metric that previous work reports. However, the standard mIoU metric does not fully illustrate the class-incremental performance of the model. To clearer show how the performance differs between different increments, we report three different mIoU metrics as previously proposed [3]:

 – Base: mIoU of the base classes trained in the first increment (Pascal: 0-15)
 – New: mIoU of the incrementally added classes (Pascal: 16-20).

– All: mIoU of all classes including background (Pascal: 0-20)

While the mIoU-metric for *all* is of interest for the final system performance, it can often be skewered and obscuring if the model sacrifice the performance of the new classes to maintain the performance of the base classes. In addition to the three metrics above, we also propose to use the incremental-wise averaged mIoU, incremental-mIoU, which puts equal importance on all incremental steps independently of the number of classes in each step. This metric differs from the *averaged* metric proposed in [7], which calculates the averaged mIoU of the performance after each incremental step.

$$\text{Inc-mIoU} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{\mathcal{C}^t} \sum_{k \in \mathcal{C}^t} IoU_k = \frac{1}{T} \sum_{t=1}^{T} \text{mIoU}(\mathcal{C}^t). \qquad (13)$$

4.3   Implementation details

We use the same network architecture as previous works ([3,23]), which is the DeepLab v3 [5] with a ResNet101 backbone. To facilitate the comparison, we are using the same pretrained weights, the same learning rates, and the same optimizer and learning rate scheduler as [3], that is, a polynomial learning rate scheduler with a decay rate of 0.9, a learning rate of $\lambda_0 = 0.01$, and a learning rate of the following steps of $\lambda_{t>0} = 0.001$. The optimizer is the Stochastic Gradient Descent (SGD) with a Nesterov momentum of 0.9.

The output stride during testing is kept the same as during training, e.g., 16, which means that the final class scores are upsampled from 32×32 to 512×512 using bilinear interpolation. We used a batch size of 20 and 8, respectively, and used 30 epochs for Pascal and 60 epochs for training each increment on ADE20k [37]. No early stopping was employed. As in previous works we use random crop and horizontal flips as data augmentation, cropping the images randomly to 512×512 during training and during validation and testing, using a center-crop of the same size. The relative weighting between the cross-entropy loss and the knowledge distillation loss is the same as in [3,23], $\lambda_{CE} = 1$ and $\lambda_{KD} = 10$.

5   Results

We compare our method with the state-of-the-art on Pascal VOC 2012 and ADE20k according to the experimental setup previously described.

Pascal VOC: Table 3 shows quantitative results on the 15-5 and the 15-1 setups. All of our results are reported as the mean and variance over three distinct random seeds (42, 1337, and 2001) while the compared methods have used a single seed for evaluation. In Table 3 we can see that our methods outperform all previous methods on Pascal *15-5*. The results from the more challenging *15-1* scenario (see Tab. 3) also show that our method outperforms most methods except for RECALL [20] which uses additional unlabeled data during training.

ADE20k: The ADE20k results in Table 4 show that our method performs on par with most approaches on the *100-50* and *50-50* settings while out-performing all methods on the most challenging incremental setting *100-10*.

Ablation study We evaluate three important components of our method.
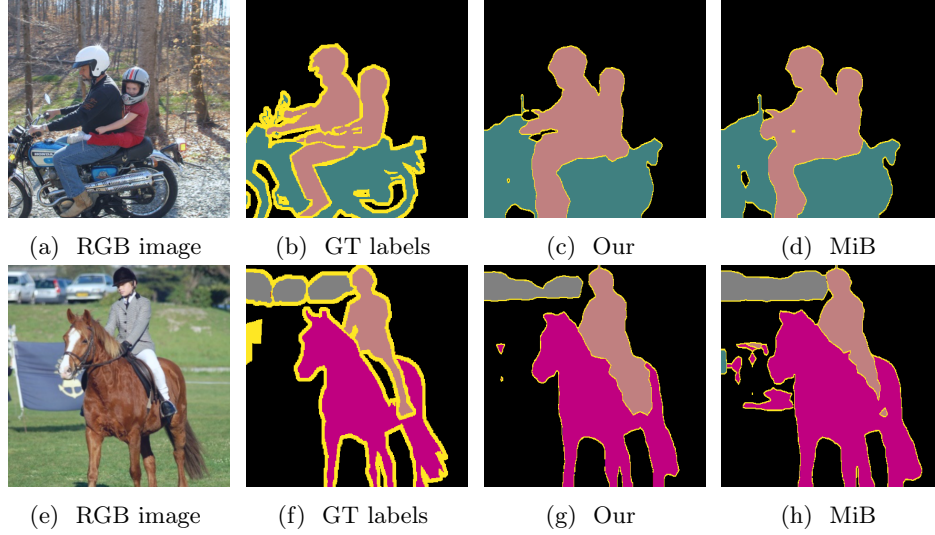

| (a) RGB image | (b) GT labels | (c) Our | (d) MiB |

| (e) RGB image | (f) GT labels | (g) Our | (h) MiB |

Fig. 3: Illustration of quantitative results of our method on Pascal *15-5* compared to MiB [3]

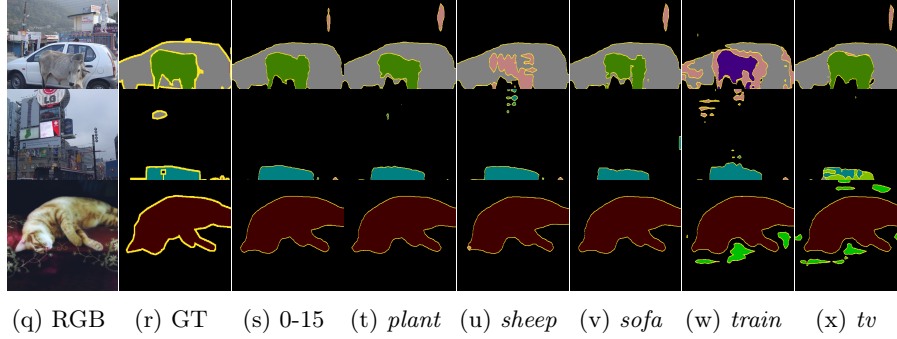(q) RGB   (r) GT   (s) 0-15   (t) *plant*   (u) *sheep*   (v) *sofa*   (w) *train*   (x) *tv*

Fig. 4: Illustration of how the class predictions of previous classes change while learning new classes. The figures show the predictions after each step on Pascal *15-1*. Note the class confusion that occurs right after learning the new class sheep in the top row, where the cow is miss-classified as such. Similar effects occur in the middle row where the bus and train classes are similar. Overall, we can see that even in the challenging scenario of Pascal *15-1*, our method maintains high accuracy for the learned classes.

	15-5								15-1							
	disjoint				overlapped				disjoint				overlapped			
	0-15	16-20	All	Inc.	0-15	16-20	All	Inc.	0-15	16-20	All	Inc.	0-15	16-20	All	Inc.
MiB [3]	71.4	43.3	64.7	57.4	75.1	49.4	69.0	62.3	45.7	12.9	37.9	18.4	34.8	13.5	29.7	17.1
MiB [23]	47.5	34.1	44.3	40.8	73.1	44.5	66.3	58.8	39.0	15.0	33.3	19.0	44.5	11.7	36.7	17.2
SDR [23]	73.5	**47.3**	67.2	**60.4**	75.4	*52.6*	69.9	*64.0*	59.2	12.9	48.1	20.6	44.7	*21.8*	39.2	25.6
SDR+MiB [23]	*73.6*	44.1	**67.3**	58.8	**76.3**	50.2	*70.1*	63.3	*59.4*	*14.3*	*48.7*	*21.8*	47.3	14.7	39.5	20.1
PLOP [7]	-	-	-	-	*75.7*	51.7	*70.1*	63.7	-	-	-	-	65.1	21.1	*54.6*	*28.4*
RECALL [20] (GAN)	67.8	49.8	63.5	58.8	68.1	50.9	64.0	59.5	67.5	44.9	62.1	48.7	69.0	49.2	64.3	52.5
RECALL [20] (Web)	70.5	52.9	66.3	61.7	69.1	54.3	65.6	61.7	67.4	47.8	62.7	51.1	69.1	50.9	64.8	53.9
Our	**73.8**	*44.8*	66.9	*59.3*	**76.3**	**53.6**	**70.9**	**64.9**	**63.8**	15.9	**52.4**	**23.9**	**68.5**	**24.4**	**58.0**	**31.8**
	±0.1	±0.0.8	±0.1	±0.3	±0.9	±7.4	±1.5	±2.8	±1.6	±3.8	±1.0	±1.6	±0.3	±2.6	±0.03	±1.6
Our (joint)	-	-	78.8	-	-	-	78.8	-	-	-	78.8	-	-	-	78.8	-

Table 3: Results on the Pascal VOC dataset, best in **bold** and runner-up in *italic*. The *joint* setting is used as an upper limit of what the model can achieve when learning all classes simultaneously in one step. RECALL [20] is part of the table but not part of the ranking since the method uses additional unlabeled data.

	100-50				50-50				100-10			
	Base	New	All	Inc.	Base	New	All	Inc.	Base	New	All	Inc.
MiB [3]	**37.9**	*27.9*	**34.6**	**32.9**	35.5	22.9	27.1	27.1	**31.8**	*14.1*	*25.9*	*17.0*
MiB [23]	*37.6*	24.7	33.3	31.2	39.1	22.6	28.1	28.1	21.0	5.3	15.8	7.9
SDR [23]	37.4	24.8	33.2	31.1	*40.9*	23.8	29.5	29.5	*28.9*	7.4	21.7	11.0
SDR+MiB [23]	37.5	25.5	*33.5*	*31.5*	**42.9**	*25.4*	**31.3**	**31.3**	*28.9*	11.7	23.2	14.6
Our	33.3	**29.1**	31.9	31.2	34.4	**27.4**	*29.8*	*29.8*	28.5	**22.6**	**26.5**	**23.6**
joint	-	-	41.1	-	-	-	41.1	-	-	-	41.1	-

Table 4: Results on the ADE20k dataset. The methods have been evaluated using two distinct class orders, the original class order, as well as the random order from [3]. The *Base* set correspond to classes 1-100 and 1-50 respectively, the *New* set correspond to the incrementally added classes 51-150 and 101-150 respectively. Class 0, which is seen as the background is absent in the ADE20k since all pixels are annotated.

First, we evaluate the choice of rectification function for the evidence score. Our choice (Tab. 5a) clearly outperforms ReLU and is superior to Exp.

Secondly, we compare our implicit background modeling approach compared to simply replacing the weight layer for the background class in MiB with a single bias value (see Tab. 5b). We also evaluate how freezing the bias term together with the previous weights after learning the first increment affects the performance. The results clearly show that even a simple implicit background model can be beneficial, however, our proposed EDL-based modeling is clearly superior.

Lastly, we evaluate the effect of the compensation term on ADE20k (see Tab. 5c) and show that it is highly beneficial, especially for the improving the Incremental mIoU.

Func.	All
ReLU	53.9
Exp	70.6
Ours	**70.9**

(a)

Method	mIoU All
MiB	68.8
+ Bias only	69.5
+ Frozen clas.	69.1
EDL Impl. bg	**70.9**

(b)

Balancing	All	Inc.
w/o	25.8	15.8
with	**26.5**	**23.6**

(c)

Table 5: Ablation study: a) Influence of choice of rectification function on Pascal 15-5 Overlap. b) Implicit background approach c) Increment balancing on ADE20k 100-10

6   Discussion

Quantitative results: From the results in Table 3 and Table 4 we see that our method based on implicit modeling of the background performs better than most methods using explicit background modeling and without additional data.The results for ADE20k show that we perform on par with the state-of-the-art on the two simpler scenarios and clearly surpass the results by SDR [23] and MiB [3] when using multiple increments.

Qualitative results: To illustrate the qualitative results of our method we pick image samples from the test set and visualize them. Figure 2 illustrates the class probabilities of our implicit background modeling compared to an explicit background modeling approach [3]. In Figure 4 we illustrate how the prediction on older classes is affected by learning new ones. It highlights the remaining problem that new classes, similar to previous ones, can cause confusion between the two classes. Some examples of this are *bus* and *train*, and *cow* and *sheep*, all of which can be similar in certain situations. Finally, we show some results on the Pascal *15-5* task, where we compare our model to MiB [3]. In the supplementary material, more qualitative examples can be found showing a comparison with additional methods, a comparison between *15-1* and *15-5* results on Pascal VOC, and a comparison between results on split A and B on ADE20k.

To summarize, our method shows that an implicit background model is not only feasible but also leads to improved performance on the Class-incremental Semantic Segmentation problem, in particular for many increments.

7   Conclusion

In this paper, we introduce a novel method for implicitly modeling the background, which consists of unlabeled classes. While the typical explicit background modeling is efficient for standard semantic segmentation, we argue that the class-incremental setting implies an open-world assumption. This assumption requires an implicit background model so as not to constrain the feature

space of the unlabeled classes unnecessarily. We have demonstrated the strengths of our method compared to the state-of-the-art on two datasets and shown that we outperform the state-of-the-art when learning an increasing number of increments.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 139–154 (2018)
2. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European conference on computer vision (ECCV). pp. 233–248 (2018)
3. Cermelli, F., Mancini, M., Bulo, S.R., Ricci, E., Caputo, B.: Modeling the background for incremental learning in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Choi, Y., El-Khamy, M., Lee, J.: Dual-teacher class-incremental learning with data-free generative replay. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3543–3552 (2021)
7. Douillard, A., Chen, Y., Dapogny, A., Cord, M.: Plop: Learning without forgetting for continual semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4040–4050 (2021)
8. Drummond, N., Shearer, R.: The open world assumption. In: eSI Workshop: The Closed World of Databases meets the Open World of the Semantic Web. vol. 15 (2006)
9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
10. He, C., Wang, R., Chen, X.: A tale of two cils: The connections between class incremental learning and class imbalanced learning, and beyond. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3559–3569 (2021)
11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
12. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
13. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012)
14. Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Dıaz-Rodrıguez, N.: Continual learning for robotics. arXiv preprint arXiv:1907.00182 pp. 1–34 (2019)
15. Li, Z., Hoiem, D.: Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence **40**(12), 2935–2947 (2017)
16. Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A.D., Jui, S., de Weijer, J.v.: Generative feature replay for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 226–227 (2020)

17. Liu, X., Yang, H., Ravichandran, A., Bhotika, R., Soatto, S.: Continual universal object detection. arXiv preprint arXiv:2002.05347 (2020)
18. Liu, Y., Su, Y., Liu, A.A., Schiele, B., Sun, Q.: Mnemonics training: Multi-class incremental learning without forgetting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12245–12254 (2020)
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
20. Maracani, A., Michieli, U., Toldo, M., Zanuttigh, P.: Recall: Replay-based continual learning in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7026–7035 (October 2021)
21. Mi, F., Kong, L., Lin, T., Yu, K., Faltings, B.: Generalized class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)
22. Michieli, U., Zanuttigh, P.: Incremental learning techniques for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
23. Michieli, U., Zanuttigh, P.: 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 (CVPR). pp. 1114–1124 (June 2021)
24. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
25. Şensoy, M., Kaplan, L., Kandemir, M.: Evidential deep learning to quantify classification uncertainty. Advances in Neural Information Processing Systems (2018)
26. Tao, X., Chang, X., Hong, X., Wei, X., Gong, Y.: Topology-preserving class-incremental learning. In: European Conference on Computer Vision. pp. 254–270. Springer (2020)
27. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12183–12192 (2020)
28. Wentao Bao, Q.Y., Kong, Y.: Evidential deep learning for open set action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
29. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Zhang, Z., Fu, Y.: Incremental classifier learning with generative adversarial networks. arXiv preprint arXiv:1802.00853 (2018)
30. Xiang, Y., Fu, Y., Ji, P., Huang, H.: Incremental learning using conditional adversarial networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6619–6628 (2019)
31. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
32. Zhang, B.F., Su, J.S., Xu, X.: A class-incremental learning method for multi-class support vector machines in text classification. In: 2006 International Conference on Machine Learning and Cybernetics. pp. 2581–2585. IEEE (2006)
33. Zhang, H., Zhu, M., Zhang, J., Zhuo, L.: Long-term visual object tracking via continual learning. IEEE Access **7**, 182548–182558 (2019). https://doi.org/10.1109/ACCESS.2019.2960321

34. Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., Kuo, C.C.J.: Class-incremental learning via deep model consolidation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (March 2020)
35. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
36. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890 (2021)
37. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)
38. Zoph, B., Ghiasi, G., Lin, T.Y., Cui, Y., Liu, H., Cubuk, E.D., Le, Q.: Rethinking pre-training and self-training. Advances in neural information processing systems **33**, 3833–3845 (2020)

Supplementary Material for

Evidential Deep Learning for Class-Incremental Semantic Segmentation

A   Derivation of Increment Balancing Term

Due to the relatively straightforward derivation and limited space, the derivation of the increment balancing term was omitted from the main paper. Instead, it is provided here for completeness.

The formulation of EDL introduces a dependency between the number of classes and the uncertainty, (3, 4) in the main paper. Compensating for the bias that occurs between increments in necessary since the number of classes varies over the increments. This is why a post-processing step to rebalance the logits of the different increments was proposed in (10) in the main paper. Under the assumption of perfect classification of the network, all evidence is centered on the correct class j, the complete class probability is written as,

$$p_i^{(t)} = p_i^{\text{fg},(t)}(1 - u^{(t)}) = \frac{e_i + 1}{e_i + K^{(t)}}\left(1 - \frac{K^{(t)}}{K^{(t)} + e_i}\right). \qquad (14)$$

Where K^t is the number of classes in increment t, i.e. $|\mathcal{C}^t|$. To maintain the foreground probability constant between the training of as single increment and during the inference oveer all learned classes, the class evidence are scaled differently based on the number of classes in the corresponding increment. That is, the unscaled single increment probability should be equal to the rescaled full probability when considering all classes, i.e.

$$\frac{p_i^{(t)}}{p_i^{(1:T)}} = 1 \iff \frac{\frac{e_i+1}{e_i+K^{(t)}}\left(1 - \frac{K^{(t)}}{K^{(t)}+e_i}\right)}{\frac{oe_i+1}{oe_i+K^{(1:T)}}\left(1 - \frac{K^{(1:T)}}{K^{(1:T)}+oe_i}\right)} = 1 \qquad (15)$$

Solving the above equation with respect to the scale factor, o, we arrive at the proposed compensation term,

$$o = \frac{(2K^t - 1)(K^{1:T})^2}{(K^t)^2(2K^{1:T} - 1)}. \qquad (16)$$

The compensation term is well defined since the number of classes is at least one.

B   Additional Qualitative results

In this section, additional qualitative results to Figures 2, 3, and 4 in the main paper are presented.Following is a short description of each method in the comparison:

Fine-tuning (FT). While fine-tuning has been proven insufficient for class-incremental learning [3], we include some qualitative results of using fine-tuning together with our implicit background formulation. This is accomplished by training our method with $\lambda_{KD} = 0$.

MiB [3] is trained in our framework using the code provided by the authors of MiB to run inference on Pascal VOC.

PLOP [7] is trained on the *overlap* scenario of Pascal VOC using the code provided by the authors and the default settings for their methods. Code can be found here: CVPR 2021 PLOP

B.1   Results on PASCAL

This section provides some general qualitative results on the Pascal VOC [9] *15-5 overlap* setting in fig. 5. They illustrating that our method manages finer details better and avoids spurious classifications in the background. The results from fine-tuning are surprisingly good when using our proposed implicit background modeling, showing a tendency of classes bleeding out but without introducing spurious classifications, see panel (c). This illustrates how the implicit background model facilitates maintaining previous knowledge of the model, even without any explicit constraints during learning.

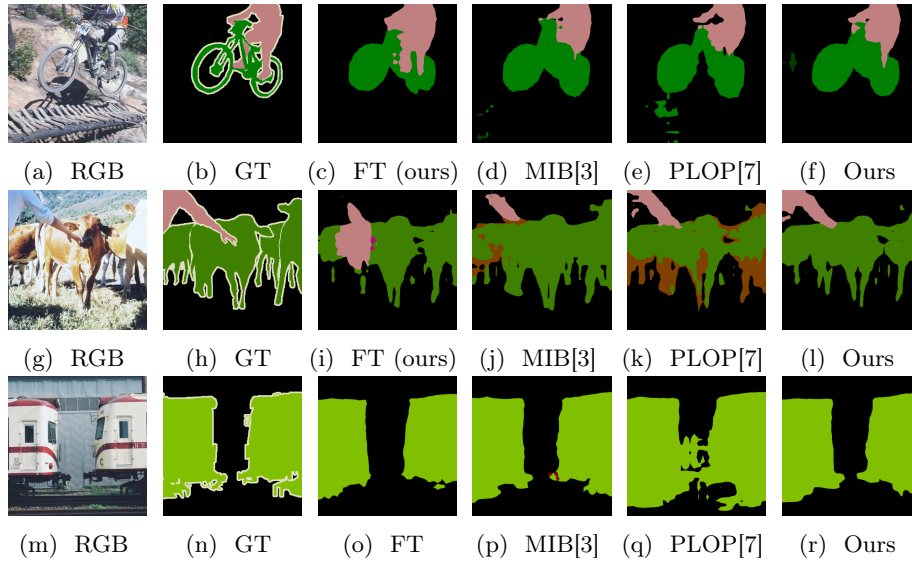|					
(a) RGB	(b) GT	(c) FT (ours)	(d) MIB[3]	(e) PLOP[7]	(f) Ours
(g) RGB	(h) GT	(i) FT (ours)	(j) MIB[3]	(k) PLOP[7]	(l) Ours
(m) RGB	(n) GT	(o) FT	(p) MIB[3]	(q) PLOP[7]	(r) Ours

Fig. 5: Qualitative example on Pascal VOC 15-5.

Additionally, fig. 6 shows how the predictions can differ when learning multiple increments (*15-1*) instead of a single one (*15-5*). These results show that the

out-of-focus background is complex for all methods to handle, resulting in large spurious classifications in the background. While our implicit background model handles the *15-5* scenario pretty well in comparison to the state-of-the-art, both our and other methods deliver drastically worse results on this failure case in the *15-1* scenario.
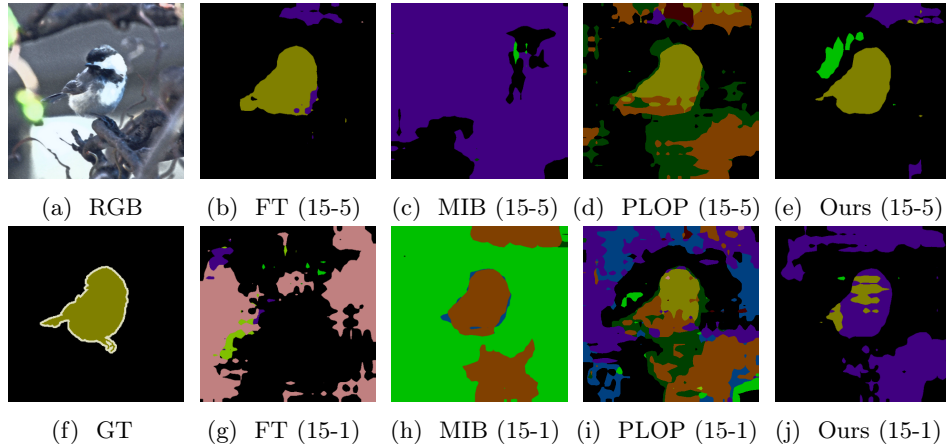

| (a) RGB | (b) FT (15-5) | (c) MIB (15-5) | (d) PLOP (15-5) | (e) Ours (15-5) |

| (f) GT | (g) FT (15-1) | (h) MIB (15-1) | (i) PLOP (15-1) | (j) Ours (15-1) |

Fig. 6: Qualitative example (failure case) on Pascal VOC, comparing prediction on the *15-1* and *15-5* tasks

B.2   Results on ADE20k

This section presents qualitative results on one scene from ADE20k [37] in order to illustrate the impact that different class orders have on the final predictions and how the foreground-background class-balancing can mitigate some of these problems.

ADE20k contains 150 different classes, ordered by frequency. The first set of classes, the base set, is composed of the first 50 or 100 classes depending on the task. Since these classes are the most frequent ones, the original order, the a-split, has a larger interclass imbalance between the base set and the following increments. The b-split, however, is based on a random reordering of the classes, done by Cermelli et.al. [3], leading to a larger intraclass frequency variation but lower inter-class imbalance.

From fig. 7 it can be seen that the original class order lead to a larger confusion between the building and house classes unless the background foreground class balancing term is utilized. However, this confusion does not seem to occur for the b-split.
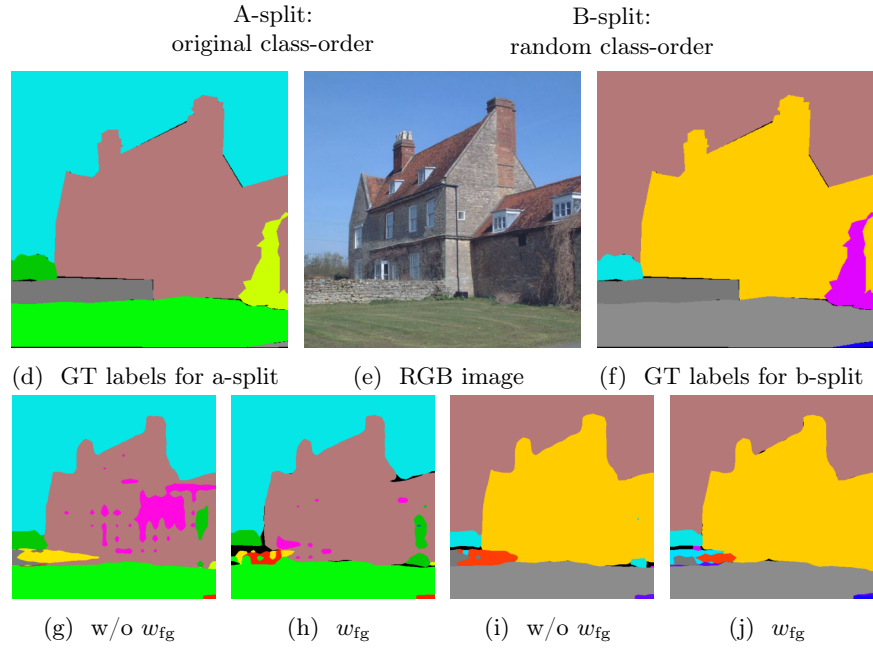
A-split:
original class-order

B-split:
random class-order

(d) GT labels for a-split   (e) RGB image   (f) GT labels for b-split

(g) w/o w_{fg} (h) w_{fg} (i) w/o w_{fg} (j) w_{fg}

Fig. 7: Figure illustrating the effect that the class order and the foreground-background-balancing weight, w_{fg}, have on ADE20k *100-10*. The a- and b-splits differ in the order in which classes are learned (a is the original order and b is an alternative order with a more balanced intra-increment class frequency).