Modeling the Background for Incremental and Weakly-Supervised Semantic Segmentation

# Modeling the Background for Incremental and Weakly-Supervised Semantic Segmentation

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Buló, Elisa Ricci and Barbara Caputo

**Abstract**—Deep neural networks have enabled major progresses in semantic segmentation. However, even the most advanced neural architectures suffer from important limitations. First, they are vulnerable to catastrophic forgetting, *i.e.* they perform poorly when they are required to incrementally update their model as new classes are available. Second, they rely on large amount of pixel-level annotations to produce accurate segmentation maps. To tackle these issues, we introduce a novel incremental class learning approach for semantic segmentation taking into account a peculiar aspect of this task: since each training step provides annotation only for a subset of all possible classes, pixels of the background class exhibit a semantic shift. Therefore, we revisit the traditional distillation paradigm by designing novel loss terms which explicitly account for the background shift. Additionally, we introduce a novel strategy to initialize classifier's parameters at each step in order to prevent biased predictions toward the background class. Finally, we demonstrate that our approach can be extended to point- and scribble-based weakly supervised segmentation, modeling the partial annotations to create priors for unlabeled pixels. We demonstrate the effectiveness of our approach with an extensive evaluation on the Pascal-VOC, ADE20K, and Cityscapes datasets, significantly outperforming state-of-the-art methods.

## 1 Introduction

The goal of semantic segmentation [1] is to correctly predict the semantic label associated to each pixel in an image. In the last years, thanks to the emergence of deep neural networks and to the availability of large-scale human-annotated datasets [2], [3], the state of the art in this task has improved significantly [1], [4], [5], [6], [7]. Current approaches are based on Fully Convolutional Networks (FCNs) [1] and mostly differ from the strategies used to combine multiscale representations [6], [7], to model spatial dependencies and contextual cues [4], [8], [9] or to integrate attention models [10].

Despite their effectiveness, semantic segmentation models need a large amount of images with paired pixel-level annotations during training, which are extremely costly to collect. This can be overcome by training semantic segmentation models with weaker forms of supervisions, such as image-level labels [11] and points [12]. Still, both fully-supervised and weakly-supervised learning (WSL) algorithms assume that the annotated data for all the semantic categories the model will be asked to recognize should be available beforehand. This assumption rarely holds in many practical applications; it would be desirable to dispose of semantic segmentation models able to continuously incorporate information about novel categories, while being able to retain knowledge about the previous classes. In this paper we study the problem of semantic segmentation in an incremental class learning (ICL) scenario [13], *i.e.* we aim to build a deep model able to incrementally learn new categories whilst preserving good performance on the old ones avoiding catastrophic forgetting [14].

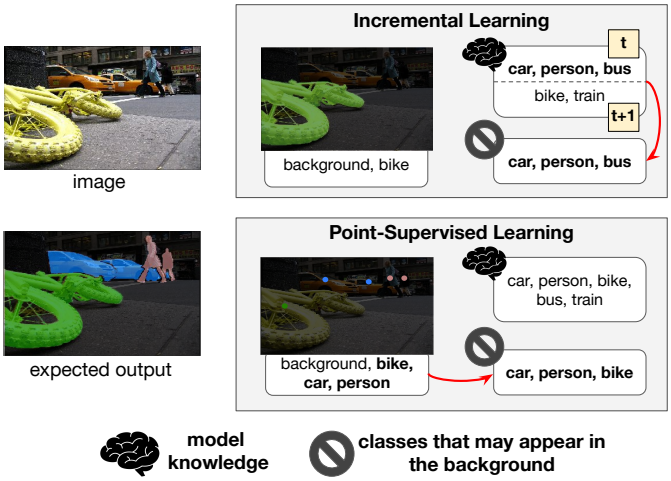



Fig. 1: The figure depicts the content of the background pixels in case of partial labels. In incremental learning (top), since we have labels only for pixels of novel classes in the current training step, the background may contain pixels of the old ones. In point-supervised learning, every class with at least one annotated point in the image is also present in the background. Image taken from the Pascal-VOC dataset [2].

Our approach is inspired by previous ICL methods on image classification [13], [15], [16], which address catastrophic forgetting through knowledge distillation [17]. However, here we show that a naive application of previous knowledge distillation strategies would not suffice in our setting. The reason is that none of these approaches take explicitly into account the evolving semantics of the background class among different training steps, a problem that we called *background shift*. Indeed,


- *F. Cermelli and B.Caputo are with DAUIN Department of Control and Computer Engineering of Politecnico di Torino, Turin, Italy. (Email: fabio.cermelli@polito.it, barbara.caputo@polito.it)*
- *M. Mancini is with Cluster of Excellence "Machine Learning", University of Tübingen, Germany. (Email: massimiliano.mancini@uni-tuebingen.de)*
- *S. Rota Buló is with Facebook (Email: rotabulo@fb.com), but the work has been done when he was with Mapillary Research, Graz, Austria.*
- *F.Cermelli and B. Caputo are with Italian Institute of Technology, Turin, Italy.*
- *E. Ricci is with Fondazione Bruno Kessler, Trento, Italy. (Email: elricci@fbk.eu)*
- *E. Ricci is with Department of Information Engineering and Computer Science, University of Trento, Trento, Italy*

since we have only partial annotations in each training step, unlabeled/background pixels might belong to some of the classes we have previously learned and even to classes we will learn in the future. For instance (Fig. 1, top), we might have learned the classes *car*, *person* and *bus* at a previous learning step $t$, and at step $t+1$ we learn new classes (e.g., *bike* and *train*). Since in the current training step we have annotations only for new classes, the background might contain pixels of old classes as well. Note that this problem is peculiar to semantic segmentation. To overcome this issue, we revisit the classical distillation-based framework in [15] by introducing two novel loss terms to properly account for the semantic distribution shift within the background class. Our approach is based on a simple principle: each background pixel in the ground-truth might contain either the background or one of the categories whose annotation is missing for the current image. This means that, in the incremental learning setting, we consider pixels labeled as background at a given learning step to contain either the background or any of the previously seen classes. A similar reasoning can be applied for the distillation loss, in a symmetric manner. We extensively evaluate our method on three datasets, Pascal-VOC [2], ADE20K [3] and Cityscapes [18] showing that our approach, coupled with a novel classifier initialization strategy, largely outperform traditional ICL methods.

Finally, we show how our ICL approach easily extends to other partially-annotated scenarios, such as weakly supervised semantic segmentation with point or scribble supervision. In this setting, we consider non-annotated pixels containing either the actual background or any of the weakly annotated classes in the current image. As an example, in Fig.1 (bottom), we have point-level annotations for *bike*, *car* and *person*. The definition of the setting entails that non-annotated pixels might contain either the background or one of the three classes above, but it does not contain other classes (e.g., *train* and *bus*) without any annotation in the current image. We encode this prior in the standard cross-entropy loss and we benchmark this approach in semantic segmentation in Pascal-VOC using point [12] and scribble [19] supervision, showing performance superior or comparable to the state of the art. We also evaluate our method on another scenario, scene parsing with point supervision, where the background is not present but unlabeled pixels might still contain any of the classes with at least one point in the current image. Experiments on ADE20k [20] demonstrate the effectiveness of our approach in this scenario.

To summarize, the main contributions of this paper are as follows:

- We identify the problem of semantic shift of the background class arising in incremental class learning for semantic segmentation.
- We revisit standard ICL approaches with a novel objective function that is applied both to a cross-entropy and a standard distillation loss. Coupled with a specific classifier initialization strategy, our approach greatly alleviates the catastrophic forgetting and the semantic shift of the background class, leading to the state of the art.
- We benchmark our approach over several previous ICL methods on three popular semantic segmentation datasets, considering different experimental settings. We hope that our results will serve as a reference for future works in incremental learning in semantic segmentation.
- We show how the same approach can be applied to the task of WSL using point or scribble supervision, achieving state-of-the-art results in three different experimental settings.

This paper extends our earlier work [21] in many aspects. In particular, we demonstrate that the key idea behind modeling the background, *i.e*. considering unlabeled/background pixels as belonging to any class in a specific set built through known priors (*i.e*. old classes), can be extended to any partially-annotated scenario, such as semantic segmentation with point or scribble supervision. Experiments demonstrate that our approach is very effective even in these new tasks, confirming that its underlying idea of modeling the semantic of the background class is general and it is applicable on multiple tasks consisting of noisy or partial annotations. Finally, we expanded our experimental evaluation on ICL considering other challenging scenarios in the Cityscapes dataset [18]. We also provide a more comprehensive review of related works, including the weakly-supervised learning literature.

The rest of this paper is organized as follows. We first introduce related work in Section 2 and then describe our ICL method and its extension to tackle WSL with weak supervision in Section 3. The results of our approach on ICL and WSL are provided in Section 6. We conclude the paper in Section 7.

The code is available at https://github.com/fcdl94/MiB.

## 2 Related Works

**Semantic Segmentation.** Deep learning has enabled great advancements in semantic segmentation [1], [4], [5], [6], [7]. State-of-the-art methods are based on Fully Convolutional Neural Networks [1], [22] and use different strategies to condition pixel-level annotations on their global context, *e.g*. using multiple scales [4], [5], [6], [7], [8], [9] and/or modeling spatial dependencies [8], [23]. The vast majority of semantic segmentation methods considers an offline setting, *i.e*. they assume that training data for all classes is available beforehand. To our knowledge, the problem of ICL in semantic segmentation has been addressed only in [24], [25], [26], [27]. Ozdemir *et al*. [24], [25] describe an ICL approach for medical imaging, extending a standard image-level classification method [15] to segmentation and devising a strategy to select relevant samples of old datasets for rehearsal. Tasar *et al*. [26] proposed a similar approach for segmenting remote sensing data. Differently, Michieli *et al*. [27] consider ICL for semantic segmentation in a particular setting where labels are provided for old classes while learning new ones. Moreover, they assume the novel classes to be never present as background pixels in previous learning steps. These assumptions strongly limit the applicability of their method.

Here we propose a more principled formulation of the ICL problem in semantic segmentation. In contrast with previous works, we do not limit our analysis to medical [24] or remote sensing data [26] and we do not impose any restrictions on how the label space should change across different learning steps [27]. Moreover, we are the first to provide a comprehensive experimental evaluation of state of the art ICL methods on commonly used semantic segmentation benchmarks and to explicitly introduce and tackle the semantic shift of the background class, a problem recognized but largely overseen by previous works [27]. Our strategy can be applied in different scenarios with partial annotations, such as weakly supervised learning.

**Incremental Learning.** The problem of catastrophic forgetting [14] has been extensively studied for image classification tasks

[28]. Previous works can be grouped in three categories [28]: replay-based [13], [16], [29], [30], [31], [32], regularization-based [15], [33], [34], [35], [36], and parameter isolation-based [37], [38], [39]. In replay-based methods, examples of previous tasks are either stored [13], [16], [30], [40] or generated [29], [31], [32] and then replayed while learning the new task. Parameter isolation-based methods [37], [38], [39] assign a subset of the parameters to each task to prevent forgetting. Regularization-based methods can be divided in prior-focused and data-focused. The former [33], [34], [35], [41] define knowledge as the parameters value, constraining the learning of new tasks by penalizing changes of important parameters for old ones. The latter [15], [36], [42] exploits distillation [17] and use the distance between the activations produced by the old network and the new one as a regularization term to prevent catastrophic forgetting.

Despite these progresses, very few works have gone beyond image-level classification. A first work in this direction is [43] which considers ICL in object detection proposing a distillation-based method adapted from [15] for tackling novel class recognition and bounding box proposals generation. In this work we also take a similar approach to [43] and we resort on distillation. However, here we specifically propose to address the problem of modeling the background shift which is peculiar of the semantic segmentation setting.

**Weakly Supervised Learning.** The significant burden of requiring annotations for each pixel of an image has lead to several research efforts toward building semantic segmentation models using cheaper (but weaker) annotations. Under this perspective, different types of annotation has been explored, such as image-level labels [11], [44], [45], [46], bounding boxes [47], [48], [49], scribbles [19], [50] and points [12], [20].

Image-level labels only provide information about which classes are contained in the image, without any hint on their locations. Most approaches in this direction [11], [44], [45], [46], [51], [52], [53], [54], [55] aim to generate pixel-wise pseudo-labels obtaining and refining an initial localization map, which is often a class activation maps (CAM) [56], [57] obtained from an image-level classifier. [11] introduced the idea to use CAMs as a seed for weak localization of the objects, expanding the object prediction based on the information provided by image-level labels and constraining the segmentation masks with CRF-based object boundaries. On the intuition that better localization cues may further improve performances, subsequent works proposed to refine the localization priors. In [45] the seeded region growing algorithm [58] is adapted to extend the prior, [52] modeled the pixel similarity from the initial CAMs and employed random walk to propagate the class labels, [44] extracted multiple class activation maps using different combinations of the image feature obtained with dropout [59], [46] exploited the cross-image semantic relation and [55] adopted consistency regularization to improve the localization seed.

A stronger form of weak annotations are bounding boxes [47], providing information about the classes in the image, their location and dimensions. In this scenario, various approaches explore the use of region proposal methods to refine the candidate object masks [47], [49]. To this extent, [47] uses multi-scale combinatorial grouping (MCG) [60], refining the masks in an iterative process involving the ground-truth bounding boxes and the network predictions. Similarly, in [49] the segmentation masks are refined using GrabCut [61], MCG and the network predictions.

Finally, cheaper than bounding boxes are scribbles [19] or points [12]. Scribble annotations provide a strong localization information and are very fast to collect, providing a class for each scribble. In this scenario, [19] first proposed to expand the scribble supervision by dividing pixels into super-pixels and exploiting pixel-similarity as additional source of supervision. Differently, [50], [62] integrates graphical models (*e.g.*, graph cut or dense CRFs) into regularization losses during training, forcing the model to produce consistent outputs on similar pixels. Recently, [63] proposed to use two additional sub-networks to fully exploit scribble-annotation: one sub-network refines the model's output with an iterative up-sampling while the other performs boundary prediction to obtain more precise segmentation results.

Point supervision is more challenging since it provides only one point for each instance in the image. To solve this problem, in [12] the authors propose to use three main components: (i) an image-level prior to predict which objects are present in the image, (ii) a partial cross-entropy on the labeled points, and (iii) an objectness prior, extracted from a shallow model, which helps in differentiating background and foreground pixels. In [20] the authors propose a method using point supervision for the task of scene parsing, where a model is asked to segment both objects and stuffs. The authors propose to use the partial cross entropy coupled with a distance metric regularization, forcing pixels of the same classes to produce similar feature vectors.

In this work we focus on weak supervision with points and scribbles, both in object segmentation and scene parsing settings. We show how our simple loss formulation considering the uncertainty on unlabeled pixels produces a boost on the performance of the standard partial cross-entropy adopted by multiple works, achieving state-of-the-art results in both scenarios.

## 3 Modeling the Uncertainty in Semantic Segmentation

### 3.1 Problem Definition

The goal of semantic segmentation is to produce a model capable of assigning a class for each pixel of a given input image. Let us denote as $\mathcal{X}$ the input space (*i.e.* the image space) and, without loss of generality, let us assume that each image $x \in \mathcal{X}$ is composed by a set of pixels $\mathcal{I}$ with constant cardinality $|\mathcal{I}| = N$. The output space is defined as $\mathcal{Y}^N$, with the latter denoting the product set of $N$-tuples with elements in a label space $\mathcal{Y}$. Given an image $x$ the goal of semantic segmentation is to assign each pixel $x_i$ of $x$ a label $y_i \in \mathcal{Y}$, representing its semantic class. The mapping is realized by learning a model $f_\theta$ with parameters $\theta$ from the image space $\mathcal{X}$ to a pixel-wise class probability vector, *i.e.* $f_\theta : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}|}$. To learn the mapping, it is provided a training set $\mathcal{T} \subset \mathcal{X} \times (\mathcal{Y} \cup \mathtt{u})^N$, where $\mathtt{u}$ indicates pixels that are not labeled, either because they contain objects which are not of interest or the labeling is partial. The output segmentation mask is obtained as $y^* = \{\arg\max_{c \in \mathcal{Y}} f_\theta(x)[i,c]\}_{i=1}^N$, where $f_\theta(x)[i,c]$ is the probability for class $c$ in pixel $i$. In the following, we will indicate the probability for class $c$ in pixel $i$ as $q_x(i,c) = f_\theta(x)[i,c]$.

### 3.2 Learning from the Unknown

Commonly, in the training procedure of semantic segmentation, unlabeled pixels are discarded from the loss computation since it is believed that they do not bring information or it is not known what loss function should be minimized on them. However, we

argue that, if it is possible to make assumptions on the classes they belong to, these pixels carry useful information that can be used in the training procedure. In particular, denoting with $\mathcal{U} \subseteq \mathcal{Y}$ the set of classes to whom unlabeled pixels might belong to, we can define a loss function on an image $x$, with label $y$ as:

$$\ell(x, y) = -\sum_{i \in \mathcal{I}} \log p_x(i, y_i)\,, \tag{1}$$

where $y_i$ is the ground truth label associated to pixel $i$ and $p_x$ is computed as follow:

$$p_x(i, c) = \begin{cases} q_x(i, c) & \text{if } c \neq \mathtt{u} \\ \sum_{k \in \mathcal{U}} q_x(i, k) & \text{if } c = \mathtt{u}\,. \end{cases} \tag{2}$$

The idea behind Eq. (1) and Eq. (2) is that unlabeled pixels should provide a positive feedback for all the semantic classes they might contain. Being simple and general, this loss allows to exploit the information provided by both the labeled pixels (it degenerates to the standard cross entropy when there are no unlabeled pixels) and the unlabeled ones, through the prior that we have on their semantic content.

In the following, we first show this idea can be effectively used to address the background shift problem of ICL in semantic segmentation. Next, we extend this idea to weakly-supervised learning, showing that modeling the unlabeled pixels is beneficial to improve the final performance.

## 4 Incremental Learning in Semantic Segmentation

In the ICL setting, training is realized over multiple phases, called *learning steps*, and each step introduces novel categories to be learnt. In other terms, during the $t_{\text{th}}$ learning step, the previous label set $\mathcal{Y}^{t-1}$ is expanded with a set of new classes $\mathcal{C}^t$, yielding a new label set $\mathcal{Y}^t = \mathcal{Y}^{t-1} \cup \mathcal{C}^t$. At learning step $t$ we are provided with a training set $\mathcal{T}$ which only contains labels for pixels of novel classes while all the other pixels of the image are unlabeled. However, ignoring these unlabeled pixels will prevent the model to learn the boundary of novel classes. For this reason, we decide to assign to the unlabeled pixels a special class, *i.e.* the background class $\mathtt{b}$. The background class is the only class which is shared by multiple learning steps and it is included in any label and class set, *i.e.* $\mathtt{b} \in \mathcal{C}^t$ for any step $t$. The learning is then performed using the current training set $\mathcal{T}^t \subset \mathcal{X} \times (\mathcal{C}^t)^N$ in conjunction to the previous model $f_{\theta^{t-1}} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}^{t-1}|}$ to obtain an updated model $f_{\theta^t} : \mathcal{X} \mapsto \mathbb{R}^{N \times |\mathcal{Y}^t|}$. As in standard ICL, in this paper we assume the sets of labels $\mathcal{C}^t$ that we obtain at the different learning steps to be disjoint, except for the special background class $\mathtt{b}$.

### 4.1 Modeling the Background

A naive approach to address the ICL problem consists in retraining the model $f_{\theta^t}$ on each set $\mathcal{T}^t$ sequentially. When the predictor $f_{\theta^t}$ is realized through a deep architecture, this corresponds to fine-tuning the network parameters on the training set $\mathcal{T}^t$ initialized with the parameters $\theta^{t-1}$ from the previous stage. This approach is simple, but it leads to catastrophic forgetting. Indeed, when training using $\mathcal{T}^t$ no samples from the previously seen object classes are provided. This biases the new predictor $f_{\theta^t}$ towards the novel set of categories in $\mathcal{C}^t$ to the detriment of the classes from the previous sets. In the context of ICL for image-level classification, a standard way to address this issue is coupling the supervised loss on $\mathcal{T}^t$ with a regularization term, either taking into account the importance of each parameter for previous tasks [29], [33], or by distilling the knowledge using the predictions of the old model $f_{\theta^{t-1}}$ [13], [15], [16]. We take inspiration from the latter solution to initialize the overall objective function of our problem. In particular, we minimize a loss function of the form:

$$\mathcal{L}(\theta^t) = \frac{1}{|\mathcal{T}^t|} \sum_{(x,y) \in \mathcal{T}^t} \left( \ell_{ce}^{\theta^t}(x, y) + \lambda \ell_{kd}^{\theta^t}(x) \right) \tag{3}$$

where $\ell_{ce}$ is a standard supervised loss (*e.g.* cross-entropy loss), $\ell_{kd}$ is the distillation loss and $\lambda > 0$ is a hyper-parameter balancing the importance of the two terms.

As stated at the beginning of the Sec. 4, differently from standard ICL settings considered for image classification problems, in semantic segmentation we have that two different label sets $\mathcal{C}^s$ and $\mathcal{C}^u$ share the common background class $\mathtt{b}$. However, the distribution of the background class changes across different incremental steps. In fact, background annotations given in $\mathcal{T}^t$ refer to classes not present in $\mathcal{C}^t$, that might belong to the set of seen classes $\mathcal{Y}^{t-1}$ and/or to still unseen classes *i.e.* $\mathcal{C}^u$ with $u > t$. In the following, we show how to account for the semantic shift of the the background class by revisiting standard choices for the general objective defined in Eq. (3) with our formulation in 3.2.

**Revisiting Cross-Entropy Loss.** In Eq. (3), a possible choice for $\ell_{ce}$ is the standard cross-entropy loss computed over all image pixels:

$$\ell_{ce}^{\theta^t}(x, y) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log q_x^t(i, y_i)\,, \tag{4}$$

where $q_x^t$ is the output of the model at the training step $t$, *i.e.* $q_x^t(i, c) = f_{\theta_t}(x)[i, c]$

The problem with Eq. (4) is that the training set $\mathcal{T}^t$ we use to update the model only contains information about novel classes in $\mathcal{C}^t$. However, the unlabeled pixels in $\mathcal{T}^t$, that are assigned to the background class, might include also pixels associated to the previously seen classes in $\mathcal{Y}^{t-1}$. We argue that, without explicitly taking into account this aspect, the catastrophic forgetting problem would be even more severe. In fact, we would drive our model to predict the background label $\mathtt{b}$ for pixels of old classes, further degrading the capability of the model to preserve semantic knowledge of past categories. To avoid this issue, we propose to replace the cross-entropy loss in Eq. (4) with the loss function in Eq. (1), by considering $\mathcal{U} = \mathcal{Y}^{t-1}$. Therefore, we define:

$$\ell_{ce}^{\theta^t}(x, y) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \log p_x^t(i, y_i)\,, \tag{5}$$

where:

$$p_x^t(i, c) = \begin{cases} q_x^t(i, c) & \text{if } c \neq \mathtt{b} \\ \sum_{k \in \mathcal{Y}^{t-1}} q_x^t(i, k) & \text{if } c = \mathtt{b}\,. \end{cases} \tag{6}$$

Our intuition is that by using Eq.(5) we can update the model to predict the new classes and, at the same time, account for the uncertainty over the actual content of the background class. In fact, in Eq. (5) the background class ground truth is not directly compared with its probabilities $q_x^t(i, \mathtt{b})$ obtained from the current model $f_{\theta^t}$, but with the probability of having *either an old class or the background*. A schematic representation of this procedure is depicted in Fig. 2 (blue block).

It is worth noting that the alternative of ignoring the unlabeled pixels within the cross-entropy loss is a worse solution than

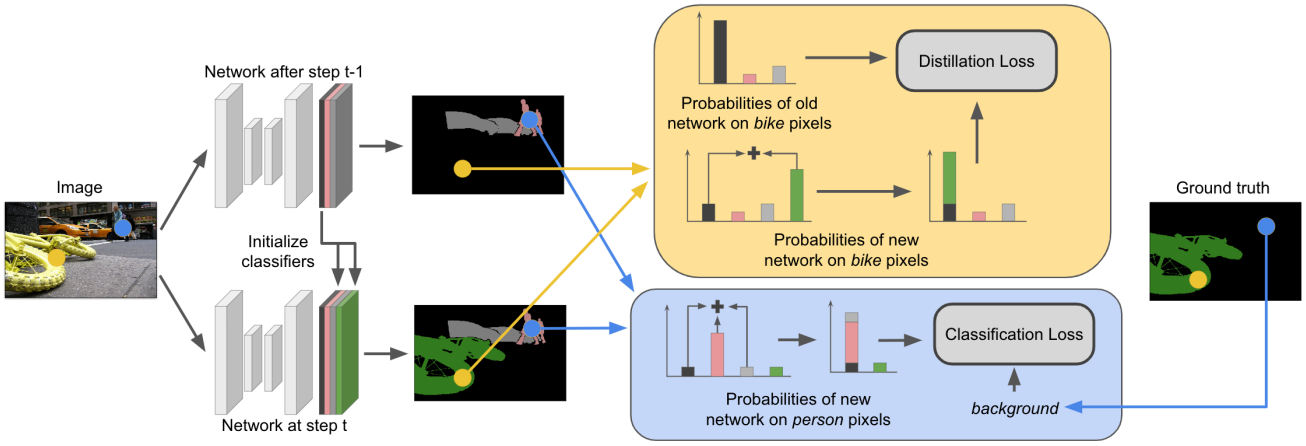



Fig. 2: Overview of our method. At learning step $t$ an image is processed by the old (top) and current (bottom) models, mapping the image to their respective output spaces. As in standard ICL methods, we apply a cross-entropy loss to learn new classes (blue block) and a distillation loss to preserve old knowledge (yellow block). In this framework, we model the semantic changes of the background across different learning steps by (i) initializing the new classifier using the weights of the old background one (left), (ii) comparing the pixel-level background ground truth in the cross-entropy with the probability of having either the background (black) or an old class (pink and grey bars) and (iii) relating the background probability given by the old model in the distillation loss with the probability of having either the background or a novel class (green bar). Image taken from the Pascal-VOC dataset [2].

considering them as background. In fact, this would not allow the model to correctly update its classifier to update its representation of the background and to learn the boundary of novel objects. Moreover, it would not allow to exploit the information that new images might contain about old classes.

**Revisiting Distillation Loss.** In the context of incremental learning, distillation loss [17] is a common strategy to transfer knowledge from the old model $f_{\theta^{t-1}}$ into the new one, preventing catastrophic forgetting. Formally, a standard choice for the distillation loss $\ell_{kd}$ is:

$$\ell_{kd}^{\theta^t}(x,y) = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{c \in \mathcal{Y}^{t-1}} q_x^{t-1}(i,c) \log \hat{p}_x^t(i,c)\,, \qquad (7)$$

where $\hat{p}_x^t(i,c)$ is defined as the probability of class $c$ for pixel $i$ given by $f_{\theta^t}$ but re-normalized across all the classes in $\mathcal{Y}^{t-1}$ *i.e.*:

$$\hat{p}_x^t(i,c) = \begin{cases} 0 & \text{if } c \in \mathcal{C}^t \setminus \{\mathtt{b}\} \\ q_x^t(i,c) / \sum_{k \in \mathcal{Y}^{t-1}} q_x^t(i,k) & \text{if } c \in \mathcal{Y}^{t-1}\,. \end{cases} \qquad (8)$$

The rationale behind $\ell_{kd}$ is that $f_{\theta^t}$ should produce activations close to the ones produced by $f_{\theta^{t-1}}$. This regularizes the training procedure such that the parameters $\theta^t$ remain anchored to the solution found for classifying pixels of previous classes, *i.e.* $\theta^{t-1}$.

The loss defined in Eq. (7) has been used either in its base form or variants in different contexts, from incremental task [15] and class learning [13], [16] in object classification to complex scenarios such as detection [43] and segmentation [27]. Despite its success, it has a fundamental drawback in semantic segmentation: it completely ignores that the representation of the background class evolves over time. While with Eq. (5) we tackled the first problem linked to the semantic shift of the background (*i.e.* $\mathtt{b} \in \mathcal{T}^t$ contains pixels of $\mathcal{Y}^{t-1}$), we use the distillation loss to tackle the second: annotations for background in $\mathcal{T}^s$ with $s < t$ might include pixels of classes in $\mathcal{C}^t$. From the latter considerations, the background probabilities assigned to a pixel by the old predictor $f_{\theta^{t-1}}$ and by the current model $f_{\theta^t}$ do not share the same semantic content. More importantly, $f_{\theta^{t-1}}$ might predict as background pixels of classes in $\mathcal{C}^t$ that we are currently trying to learn. Notice that this aspect is peculiar to the segmentation task and it is not considered in previous incremental learning models.

To address the semantic shift of the background class between the old and the current model, we explicitly revise the distillation loss in Eq. (7). In particular, we extend the reasoning behind Sec.3.2 to the distillation soft-targets and we design a novel distillation loss by rewriting the probability distribution $\hat{p}_x^t(i,c)$ in Eq. (8) as:

$$\hat{p}_x^t(i,c) = \begin{cases} q_x^t(i,c) & \text{if } c \neq \mathtt{b} \\ \sum_{k \in \mathcal{C}^t} q_x^t(i,k) & \text{if } c = \mathtt{b}\,. \end{cases} \qquad (9)$$

We note that Eq. (9) is identical to Eq. (2) by setting $\mathcal{U} = \mathcal{Y}^t$ and by substituting the background class $b$ to $\mathtt{u}$.

Similarly to Eq. (7), we still compare the probability of a pixel belonging to seen classes assigned by the old model, with its counterpart computed with the current parameters $\theta^t$. However, differently from classical distillation, in Eq. (9) the probabilities obtained with the current model are kept unaltered, *i.e.* normalized across the whole label space $\mathcal{Y}^t$ and not with respect to the subset $\mathcal{Y}^{t-1}$ (Eq. (8)). More importantly, the background class probability as given by $f_{\theta^{t-1}}$ is not directly compared with its counterpart in $f_{\theta^t}$, but with the probability of having *either a new class or the background*, as predicted by $f_{\theta^t}$ (see Fig. 2, yellow block).

We highlight that, with respect to Eq. (8) and other simple choices (*e.g.* ignoring unlabeled pixels from Eq. (8)) this solution has two advantages. First, we can use the full output space of the old model to distill knowledge in the current one, without ignoring any pixel or class. Second, we can propagate the uncertainty we have on the semantic content of the background in $f_{\theta^{t-1}}$ without penalizing the probabilities of new classes we are learning in the current step $t$.

**Classifiers' Parameters Initialization.** As discussed above, the background class $\mathtt{b}$ is a special class devoted to collect the

probability that a pixel belongs to an unknown object class. In practice, at each learning step $t$, the novel categories in $\mathcal{C}^t$ are unknowns for the old classifier $f_{\theta^{t-1}}$. As a consequence, unless the appearance of a class in $\mathcal{C}^t$ is very similar to one in $\mathcal{Y}^{t-1}$, it is reasonable to assume that $f_{\theta^{t-1}}$ will likely assign pixels of $\mathcal{C}^t$ to $\mathtt{b}$. Taking into account this initial bias on the predictions of $f_{\theta^t}$ on pixels of $\mathcal{C}^t$, it is detrimental to randomly initialize the classifiers for the novel classes. A random initialization would provoke a misalignment among the features extracted by the model (aligned with the background classifier) and the random parameters of the classifier itself. Notice that this could lead to possible training instabilities while learning novel classes since the network could initially assign high probabilities for pixels in $\mathcal{C}^t$ to $\mathtt{b}$.

To address this issue, we propose to initialize the classifier's parameters for the novel classes in such a way that given an image $x$ and a pixel $i$, the probability of the background $q_x^{t-1}(i,\mathtt{b})$ is uniformly spread among the classes in $\mathcal{C}^t$, *i.e.* $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathcal{C}^t| \;\; \forall c \in \mathcal{C}^t$, where $|\mathcal{C}^t|$ is the number of new classes (notice that $\mathtt{b} \in \mathcal{C}^t$). To this extent, let us consider a standard fully connected classifier and let us denote as $\{\omega_c^t, \beta_c^t\} \in \theta^t$ the classifier parameters for a class $c$ at learning step $t$, with $\omega$ and $\beta$ denoting its weights and bias respectively. We can initialize $\{\omega_c^t, \beta_c^t\}$ as follows:

$$\omega_c^t = \begin{cases} \omega_{\mathtt{b}}^{t-1} & \text{if } c \in \mathcal{C}^t \\ \omega_c^{t-1} & \text{otherwise} \end{cases} \tag{10}$$

$$\beta_c^t = \begin{cases} \beta_{\mathtt{b}}^{t-1} - \log(|\mathcal{C}^t|) & \text{if } c \in \mathcal{C}^t \\ \beta_c^{t-1} & \text{otherwise} \end{cases} \tag{11}$$

where $\{\omega_{\mathtt{b}}^{t-1}, \beta_{\mathtt{b}}^{t-1}\}$ are the weights and bias of the background classifier at the previous learning step. The fact that the initialization defined in Eq.(10) and (11) leads to $q_x^t(i,c) = q_x^{t-1}(i,\mathtt{b})/|\mathcal{C}^t| \;\; \forall c \in \mathcal{C}^t$ is easy to obtain from $q_x^t(i,c) \propto \exp(\omega_{\mathtt{b}}^t \cdot x + \beta_{\mathtt{b}}^t)$.

As we will show in the experimental analysis, this simple initialization procedure brings benefits in terms of both improving the learning stability of the model and the final results, since it eases the role of the supervision imposed by Eq.(5) while learning new classes and follows the same principles used to derive our distillation loss (Eq.(9)).

## 5 Semantic Segmentation using Weak Supervision

In the previous section, we revisited standard cross-entropy and distillation losses to take into account the prior we have on the content of the unlabeled/background pixels (for the cross-entropy loss) and the semantic of the predicted probabilities for the background class (for the distillation loss). The overall idea of the approach is that we can assume what is the set of semantic classes to which unlabeled/background pixels belong. This idea can be easily extended in other scenario where we can exploit partial annotations to impose priors on the content of unlabeled pixels. In the following, we will show how the same reasoning can be applied to tackle weakly supervised segmentation with point and scribble supervision.

### 5.1 Problem Formulation

In weakly supervised segmentation using points or scribbles the goal is to obtain a model capable of predicting, for each pixel of the image, its correct semantic class, as in standard semantic segmentation. However, differently from the standard segmentation task, we train our model using a training set in which we do not have full pixel-level annotation, but just points or scribbles. In particular, for each instance of a class presented in a training image, only one or few contiguous annotated pixels are provided. Formally, considering an image $x$ and its label $y$ belonging to the training set $\mathcal{T}$, the annotation is provided only for pixels $\mathcal{I}_S^x = \{i : \;\; \forall \; i \in \mathcal{I} \text{ s.t. } y_i \in \mathcal{Y}\}$, where $|\mathcal{I}_S^x| << |\mathcal{I}|$. All the other image pixels are unlabeled.

We address three weakly semantic segmentation setting: point-based [12] and scribble-based [19] object segmentation, and point-based scene parsing [20]. The goal of object segmentation is to predict object classes in a target image, where the objects are countable things, such as *cars*, *bikes*, and *dogs*. All the pixels that do not fall in these categories are labeled as background, which is considered a class in the output space $\mathcal{Y}$ of our model, similarly to Sec. 4. Formally, given a training set $\mathcal{T} \subset \mathcal{X} \times (\mathcal{Y} \cup \mathtt{u})^N$, the goal is to learn a model able to predict, for each pixel $i$, a label $y_i \in (\mathcal{Y})$. Following the protocols defined in [12] and [19], the point annotations are given only for the objects, while no points are provided for the background class. Differently, the scribble annotations also contain a scribble for the background class.

Scene parsing, instead, is a more complex task where the goal is to obtain a model able to predict both countable things and stuff classes (*i.e.* all the non-countable classes, such as *sky*, *road*, *ground*, etc.). In this setting, all the pixels in the image contain a semantic category and the background class is not included in the label space. Formally, the goal is to learn a model able to map each pixel $i$ to a label $y_i \in \mathcal{Y}$. The mapping is learned using a training set $\mathcal{T} \subset \mathcal{X} \times (\mathcal{Y} \cup \mathtt{u})^N$.

### 5.2 Modeling the Unlabeled

Being provided few labeled pixels, previous approaches [12], [20] proposed to apply a cross-entropy loss directly on the labeled points. In particular, they defined a partial cross-entropy (PCE) loss that considered only the pixels for which an annotation is given. Formally, given an image $x$ and the respective annotation $y$, the PCE loss has the form:

$$\ell_{PCE}(x,y) = -\frac{1}{|\mathcal{I}_S^x|} \sum_{i \in \mathcal{I}_S^x} \log q_x(i,y_i). \tag{12}$$

This loss is crucial for the network to discriminate the classes and to localize them in the image. However, while this solution is simple and easy to implement, it completely discards the information provided by the unlabeled pixels. In Section 3.2 we showed a simple principle to extract value from them and in this section we will revisit the principle to adapt it in this scenario.

We start from the assumption that, for each instance of a class in the image it has been provided at least one labeled pixel. This assumption implies that we know which are the classes present in the image and that all the pixels in the image belong to one of those classes. Denoting the set of classes appearing in the label $y$ of an image $x$ as $\mathcal{U}_x = \{c : \exists \, i \in I_S^x \text{ s.t. } c = y_i\}$ , we can use the loss in Eq. (1), with $\mathcal{U} = \mathcal{U}_x$ to consider the uncertainty we have on all the unlabeled pixels of the image. In particular, denoting as $\mathcal{I}_u^x = \mathcal{I} \setminus \mathcal{I}_S^x$ the set of unlabeled pixels, we propose to extend Eq. (1) as follows:

$$\ell_{UNL}(x,y) = \ell_{PCE}(x,y) - \frac{\gamma}{|\mathcal{I}_u^x|} \sum_{i \in \mathcal{I}_u^x} \log p_x(i,u)\,, \tag{13}$$

where $p_x$ is the probability distribution defined in Eq.(2) with $\mathcal{U} = \mathcal{U}_x$, and $\gamma$ is a hyper-parameter to weight the importance of unlabeled pixels since in this scenario unlabeled pixels are many more than the labeled ones.

Using the $\ell_{UNL}$ loss provides two important benefits to our training procedure when compared with $\ell_{PCE}$: (i) information from labeled pixels is propagated to unlabeled ones, providing an additional source of supervision; (ii) if the network predicts an unlabeled pixel as belonging to a class $c$ which is not in the current image (*i.e.* $c \notin \mathcal{U}_x$), receives a feedback on the error from the loss function.

To summarize, given a training set $\mathcal{T}$, we train the network to minimize the following objective function:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell_{UNL}(x, y). \tag{14}$$

We note that this loss function can be applied without any modification both with point and scribble supervision to the object segmentation and to the scene parsing tasks. The only difference relies on the classes contained in the image, since for object segmentation the background class $\mathtt{b}$ belongs to every image, *i.e.* $\mathtt{b} \in \mathcal{U}_x \quad \forall x \in \mathcal{T}$, while in scene parsing the background class is not considered. As far as we know, our method is the first applicable on point and scribble supervision both on object segmentation and scene parsing, achieving state of the art results without relying on any other prior learned on additional data (e.g. objectness prior [12]).

## 6 Experiments

### 6.1 Datasets

In this work, we use three datasets: Pascal-VOC 2012, ADE20K and Cityscapes. PASCAL-VOC 2012 [2] is a widely used benchmark that includes 20 foreground object classes. We use the extra annotation provided in [64], resulting in a dataset containing 10582 images in the training set and 1449 images in the validation. ADE20K [3] is a large-scale dataset that contains 150 classes. Differently from Pascal-VOC 2012, this dataset contains both stuff (*e.g.* *sky*, *building*, *wall*) and object classes. The dataset comprises more than 25K scene-centric images. Adopting the standard protocol [5] we use 20K images for training and we reported the results on the 2K images of the validation set. Cityscapes [18] is a dataset containing street-level images captured in central Europe that includes 19 classes, which are both objects or stuffs. The dataset provides high-resolution images with size $2048 \times 1024$, which are splitted in 2975 images for training, 500 for validation and 1525 for testing. However, since the test set ground truth are not available, we report results on the validation set as done by [65]. We exclude from the training protocol the coarse-annotations, and we use only the fine-grained annotations.

### 6.2 Incremental Learning in Semantic Segmentation

#### 6.2.1 ICL Baselines

We compare our method against standard ICL baselines, originally designed for classification tasks, on the considered segmentation task, thus segmentation is treated as a pixel-level classification problem. Specifically, we report the results of six different regularization-based methods, three prior-focused and three data-focused approaches.

In the first category, we chose Elastic Weight Consolidation (EWC) [33], Path Integral (PI) [35], and Riemannian Walks (RW) [34]. They employ different strategies to compute the importance of each parameter for old classes: EWC uses the empirical Fisher matrix, PI uses the learning trajectory, while RW combines EWC and PI in a unique model. We choose EWC since it is a standard baseline employed also in [43] and PI and RW since they are two simple applications of the same principle. Since these methods act at the parameter level, to adapt them to the segmentation task we keep the loss in the output space unaltered (*i.e.* standard cross-entropy across the whole segmentation mask), computing the parameters' importance by considering their effect on learning old classes.

For the data-focused methods, we chose Learning without forgetting (LwF) [15], LwF multi-class (LwF-MC) [13] and the segmentation method of [27] (ILT). We denote as LwF the original distillation based objective as implemented in Eq.(3) with basic cross-entropy and distillation losses, which is the same as [15] except that distillation and cross-entropy share the same label space and classifier. LwF-MC is the single-head version of [15] as adapted from [13]. It is based on multiple binary classifiers, with the target labels defined using the ground truth for novel classes (*i.e.* $\mathcal{C}^t$) and the probabilities given by the old model for the old ones (*i.e.* $\mathcal{Y}^{t-1}$). Since the background class is both in $\mathcal{C}^t$ and $\mathcal{Y}^{t-1}$ we implement LwF-MC by a weighted combination of two binary cross-entropy losses, on both the ground truth and the probabilities given by $f_{\theta^{t-1}}$. Finally, ILT [27] is the only method specifically proposed for ICL in segmentation. It uses a distillation loss in the output space, as in our adapted version of LwF [15] and/or another distillation loss in the features space, attached to the output of the network decoder. Here, we use the variant where both losses are employed. As done by [43], we do not compare with replay-based methods (*e.g.* [13]) since they violate the standard ICL assumption regarding the unavailability of old data.

In all tables we report other two baselines: simple fine-tuning (FT) on each $\mathcal{T}^t$ (*e.g.* Eq.(4)) and training on all classes offline (Joint). The latter can be regarded as an upper bound. In the tables we denote our method as MiB (**M**odel**i**ng the **B**ackground for incremental learning in semantic segmentation). All results are reported as mean Intersection-over-Union (mIoU) in percentage, averaged over all the classes of a learning step and all the steps.

#### 6.2.2 Implementation Details

For all methods we use the Deeplab-v3 architecture [8]. We use a ResNet-101 [66] backbone for Pascal-VOC 2012 and ADE20K, and following [65] a ResNeXt-101 [67] for Cityscapes. For both backbones we use an output stride of 16. Since memory requirements are an important issue in semantic segmentation, we use in-place activated batch normalization, as proposed in [65]. The backbone has been initialized using the ImageNet pretrained model [65]. We follow [8], training the network with SGD and the same learning rate policy, momentum and weight decay. For ADE20K and Pascal-VOC 2012 we use an initial learning rate of $10^{-2}$ for the first learning step and $10^{-3}$ for the followings, as in [43], while for Cityscapes we employed $2 \times 10^{-3}$ for the first step and $2 \times 10^{-4}$ in the following. We train the model with a batch-size of 24 for 30 epochs for Pascal-VOC 2012, 60 epochs for ADE20K and 360 epochs for Cityscapes in every learning step. We apply the same data augmentation of [8] and we crop the images to $512 \times 512$ during training. During test, we make a center crop $512 \times 512$ of for Pascal-VOC 2012 and ADE20K,

while we use the full-resolution images for Cityscapes. For setting the hyper-parameters of each method, we use the protocol of incremental learning defined in [28], using 20% of the training set as validation. The final results are reported on the standard validation set of the datasets.

#### 6.2.3 Pascal-VOC 2012

Following [27], [43], we define two experimental settings, depending on how we sample images to build the incremental datasets. Following [27], we define an experimental protocol called the *disjoint* setup: each learning step contains a unique set of images, whose pixels belong to classes seen either in the current or in the previous learning steps. Differently from [27], at each step we assume to have only labels for pixels of novel classes, while the old ones are labeled as background in the ground truth. The second setup, that we denote as *overlapped*, follows what done in [43] for detection: each training step contains all the images that have at least one pixel of a novel class, with only the latter annotated. It is important to note a difference with respect to the previous setup: images may now contain pixels of classes that we will learn in the future, but labeled as background. This is a more realistic setup since it does not make any assumption on the objects present in the images.

As done by previous works [27], [43], we perform three different experiments concerning the addition of one class (*19-1*), five classes all at once (*15-5*), and five classes sequentially (*15-1*), following the alphabetical order of the classes to split the content of each learning step.

**Addition of one class *(19-1)*.** In this experiment, we perform two learning steps: the first in which we observe the first 19 classes, and the second where we learn the *tv-monitor* class. Results are reported in Table 1. Without employing any regularization strategy, the performance on past classes drops significantly. FT, in fact, performs poorly, completely forgetting the first 19 classes. Unexpectedly, using PI as a regularization strategy does not provide benefits, while EWC and RW improve performance of nearly 15%. However, prior-focused strategies are not competitive with data-focused ones. In fact, LwF, LwF-MC, and ILT, outperform them by a large margin, confirming the effectiveness of this approach on preventing catastrophic forgetting. While ILT surpasses standard ICL baselines, our model is able to obtain a further boost. This improvement is remarkable for new classes, where we gain $11\%$ in mIoU, while do not experience forgetting on old classes. It is especially interesting to compare our method with the baseline LwF which uses the same principles of ours but without modeling the background. Compared to LwF we achieve an average improvement of about $15\%$, thus demonstrating the importance of modeling the background in ICL for semantic segmentation. These results are consistent in both the *disjoint* and *overlapped* scenarios.

**Single-step addition of five classes (*15-5*).** In this setting we add, after the first training set, the following classes: *plant, sheep, sofa, train, tv-monitor*. Results are reported in Table 1. Overall, the behavior on the first 15 classes is consistent with the 19-1 setting: FT and PI suffer a large performance drop, data-focused strategies (LwF, LwF-MC, ILT) outperform EWC and RW by far, while our method gets the best results, obtaining performances closer to the joint training upper bound. For what concerns the *disjoint* scenario, our method improves over the best baseline of $4.6\%$ on old classes, of $2\%$ on novel ones and of $4\%$ in all classes. These gaps increase in the *overlapped* setting where our method surpasses the baselines by nearly $10\%$ in all cases, clearly demonstrating its ability to take advantage of the information contained in the background class.

**Multi-step addition of five classes (*15-1*).** This setting is similar to the previous one except that the last 5 classes are learned sequentially, one by one. From Table 1 we can observe that performing multiple steps is challenging and existing methods work poorly for this setting, reaching performance inferior to 7% on both old and new classes. In particular, FT and prior-focused methods are unable to prevent forgetting, biasing their prediction completely towards new classes and demonstrating performances close to 0% on the first 15 classes. Even data-focused methods suffer a dramatic loss in performances in this setting, decreasing their score from the single to the multi-step scenarios of more than 50% on all classes. On the other side, our method is still able to achieve good performances. Compared to the other approaches, MiB outperforms all baselines by a large margin in both old ($46.2\%$ on the *disjoint* and $35.1\%$ on the *overlapped*), and new (nearly $13\%$ on both setups) classes. As the overall performance drop ($11\%$ on all classes) shows, the *overlapped* scenario is the most challenging one since it does not impose any constraint on which classes are present in the background.

**Ablation Study.** In Table 2 we report a detailed analysis of our contributions, considering the *overlapped* setup. We start from the baseline LwF [15] which employs standard cross-entropy and distillation losses. We first add to the baseline our modified cross-entropy (*CE*): this increases the ability to preserve old knowledge in all settings without harming (*15-1*) or even improving (*19-1*, *15-5*) performances on the new classes. Second, we add our distillation loss (*KD*) to the model. Our *KD* provides a boost on the performances for both old and new classes. The improvement on old classes is remarkable, especially in the 15-1 scenario (*i.e.* 22.8%). For the novel classes, the improvement is constant and is especially pronounced in the 15-5 scenario (7%). Notice that this aspect is peculiar of our *KD* since standard formulation work only on preserving old knowledge. This shows that the two losses provide mutual benefits. Finally, we add our classifiers' initialization strategy (*init*). This component provides an improvement in every setting, especially on novel classes: it doubles the performance on the *19-1* setting ($22.1\%$ vs $11.9\%$) and triplicates on the *15-1* ($4.5\%$ vs $13.5\%$). This confirms the importance of accounting for the background shift at the initialization stage to facilitate the learning of new classes.

#### 6.2.4 ADE20K

We create the incremental datasets $\mathcal{T}^t$ by splitting the whole dataset into disjoint image sets, without any constraint except ensuring a minimum number of images (*i.e.* 50) where classes on $\mathcal{C}^t$ have labeled pixels. Obviously, each $\mathcal{T}^t$ provides annotations only for classes in $\mathcal{C}^t$ while other classes (old or future) appear as background in the ground truth. In Table 3 we report the mean IoU obtained averaging the results on two different class orders: the order proposed by [3] and a random one. In this experiments, we compare our approach with data-focused methods only (*i.e.* LwF, LwF-MC, and ILT) due to their gap in performance with prior-focused ones.

**Single-step addition of 50 classes (*100-50*).** In the first experiment, we initially train the network on 100 classes and we add the remaining 50 all at once. From Table 3 we can observe

TABLE 1: Mean IoU (in %) on the Pascal-VOC 2012 dataset for different incremental class learning scenarios.

| | 19-1 | | | | | | 15-5 | | | | | | 15-1 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Disjoint | | | Overlapped | | | Disjoint | | | Overlapped | | | Disjoint | | | Overlapped | | |
| **Method** | *1-19* | *20* | *all* | *1-19* | *20* | *all* | *1-15* | *16-20* | *all* | *1-15* | *16-20* | *all* | *1-15* | *16-20* | *all* | *1-15* | *16-20* | *all* |
| FT | 5.8 | 12.3 | 6.2 | 6.8 | 12.9 | 7.1 | 1.1 | 33.6 | 9.2 | 2.1 | 33.1 | 9.8 | 0.2 | 1.8 | 0.6 | 0.2 | 1.8 | 0.6 |
| PI [35] | 5.4 | 14.1 | 5.9 | 7.5 | 14.0 | 7.8 | 1.3 | 34.1 | 9.5 | 1.6 | 33.3 | 9.5 | 0.0 | 1.8 | 0.4 | 0.0 | 1.8 | 0.5 |
| EWC [33] | 23.2 | 16.0 | 22.9 | 26.9 | 14.0 | 26.3 | 26.7 | 37.7 | 29.4 | 24.3 | 35.5 | 27.1 | 0.3 | 4.3 | 1.3 | 0.3 | 4.3 | 1.3 |
| RW [34] | 19.4 | 15.7 | 19.2 | 23.3 | 14.2 | 22.9 | 17.9 | 36.9 | 22.7 | 16.6 | 34.9 | 21.2 | 0.2 | 5.4 | 1.5 | 0.0 | 5.2 | 1.3 |
| LwF [15] | 53.0 | 9.1 | 50.8 | 51.2 | 8.5 | 49.1 | 58.4 | 37.4 | 53.1 | 58.9 | 36.6 | 53.3 | 0.8 | 3.6 | 1.5 | 1.0 | 3.9 | 1.8 |
| LwF-MC [13] | 63.0 | 13.2 | 60.5 | 64.4 | 13.3 | 61.9 | 67.2 | 41.2 | 60.7 | 58.1 | 35.0 | 52.3 | 4.5 | 7.0 | 5.2 | 6.4 | 8.4 | 6.9 |
| ILT [27] | 69.1 | 16.4 | 66.4 | 67.1 | 12.3 | 64.4 | 63.2 | 39.5 | 57.3 | 66.3 | 40.6 | 59.9 | 3.7 | 5.7 | 4.2 | 4.9 | 7.8 | 5.7 |
| MiB | **69.6** | **25.6** | **67.4** | **70.2** | **22.1** | **67.8** | **71.8** | **43.3** | **64.7** | **75.5** | **49.4** | **69.0** | **46.2** | **12.9** | **37.9** | **35.1** | **13.5** | **29.7** |
| Joint | 77.4 | 78.0 | 77.4 | 77.4 | 78.0 | 77.4 | 79.1 | 72.6 | 77.4 | 79.1 | 72.6 | 77.4 | 79.1 | 72.6 | 77.4 | 79.1 | 72.6 | 77.4 |

TABLE 2: Ablation study of the proposed method on the Pascal-VOC 2012 *overlapped* setup. *CE* and *KD* denote our cross-entropy and distillation losses, while *init* our initialization strategy.

| | 19-1 | | | 15-5 | | | 15-1 | | |
|---|---|---|---|---|---|---|---|---|---|
| | *1-19* | *20* | *all* | *1-15* | *16-20* | *all* | *1-15* | *16-20* | *all* |
| LwF [15] | 51.2 | 8.5 | 49.1 | 58.9 | 36.6 | 53.3 | 1.0 | 3.9 | 1.8 |
| + *CE* | 57.6 | 9.9 | 55.2 | 63.2 | 38.1 | 57.0 | 12.0 | 3.7 | 9.9 |
| + *KD* | 66.0 | 11.9 | 63.3 | 72.9 | 46.3 | 66.3 | 34.8 | 4.5 | 27.2 |
| + *init* | **70.2** | **22.1** | **67.8** | **75.5** | **49.4** | **69.0** | **35.1** | **13.5** | **29.7** |

that FT is clearly a bad strategy on large scale settings since it completely forgets old knowledge. Using a distillation strategy enables the network to reduce the catastrophic forgetting: LwF obtains 21.1% on past classes, ILT 22.9%, and LwF-MC 34.2%. Regarding new classes, LwF is the best strategy, exceeding LwF-MC by 18.9% and ILT by 6.6%. However, our method is far superior to all others, improving on the first classes and on the new ones. Moreover, we can observe that we are close to the joint training upper bound, especially considering new classes, where the gap with respect to it is only 0.3%. In Figure 3 we report some qualitative results which demonstrate the superiority of our method compared to the baselines.

**Multi-step addition of 50 classes (*100-10*).** We then evaluate the performance on multiple incremental steps: we start from 100 classes and we add the remaining classes 10 by 10, resulting in 5 incremental steps. In Table 3 we report the results on all sets of classes after the last learning step. In this setting the performance of FT, LwF and ILT are very poor because they strongly suffers catastrophic forgetting. LwF-MC demonstrates a better ability to preserve knowledge on old classes, at the cost of a performance drop on new classes. Again, our method achieves the best trade-off between learning new classes and preserving past knowledge, outperforming LwF-MC by 11.6% considering all classes.

**Three steps of 50 classes (*50-50*).** Finally, in Table 3 we analyze the performance on three sequential steps of 50 classes. Previous ICL methods achieve different trade-offs between learning new classes and not forgetting old ones. LwF and ILT obtain a good score on new classes, but they forget old knowledge. On the contrary, LwF-MC preserves knowledge on the first 50 classes without being able to learn new ones. Our method outperforms all the baselines by a large margin with a gap of 11.9% on the best performing baseline, achieving the highest mIoU on every step. Remarkably, the highest gap is on the intermediate step, where there are classes that we must both learn incrementally and preserve from forgetting on the subsequent learning step.

#### 6.2.5 Cityscapes

As done for the ADE20K dataset, we split the dataset into disjoint sets, one for each learning step $t$. Annotations are provided only for classes in $\mathcal{C}^t$ while other classes (old or future) appear as background in the ground truth. Also for Cityscapes, we compare our method with data-focused methods only (*i.e.* LwF, LwF-MC, and ILT). Table 4 reports the mean IoU obtained on three different settings: *vehicles*, *non-driving*, and *11-2*.

**Addition of vehicles classes (*vehicles*).** In the first setting, we initially train the network on the non-vehicles classes of Cityscapes (*i.e. road, sidewalk, building, wall, fence, pole, light, sign, vegetation, terrain, sky, person, rider*) and then we add in a single step all the vehicle classes (*i.e. car, truck, bus, train, motorcycle, bicycle*). From Table 4, we note that fine-tuning the network on the novel classes gives good results, but at cost of completely forgetting the old classes. Adding a distillation strategy to it improves the results, especially on old classes. In particular, LwF and ILT obtain respectively 69.0% and 68.3%, while LwF-MC achieves a lower mIoU 58.9% but it is the highest among the three on the novel classes, achieving 47.0%. Comparing MiB with the other methods, it is able to maintain a good performance on old classes, achieving the best result 69.4%, while it is also able to learn properly the novel classes, being inferior to FT only of 4.4%. Overall, the best method is MiB, exceeding other methods more than 7.2% mIoU on all classes and being inferior to the joint training upper-bound only by 4.6%.

**Addition of non-driving classes (*non-driving*).** In this experiment we use the same number of incremental classes as the previous but we propose a different grouping. The classes are semantically divided in two groups, depending if they are strictly related to driving or not. The first group, which we train first, is made by *driving* classes: *road, sidewalk, pole, light, sign, person, rider, car, truck, bus, train, motorcycle and bicycle*. The second group, which is learned incrementally, contains *non-driving* classes: *building, wall, fence, vegetation, terrain, sky*. As can be noted in Table 4, the results are coherent with the findings on the previous setting. FT performs well on novel classes, while it completely forgets about old ones. LwF, LwF-MC, and ILT achieve good performances on both old and novel classes. In particular, the best among the three is LwF, which obtains 63.9% on the *driving* classes, 63.1% on the *non-driving* classes and an overall mIoU of 63.6%. However, the best method is MiB, which exceeds LwF both on *driving* (2.5%) and *non-driving* (6.9%) classes. Overall, it achieves 67.6% mIoU, which is inferior to the upper-bound of 5.5%.

**Multi-step addition of 2 classes (*11-2*).** Finally, we analyze the

TABLE 3: Mean IoU (in %) on the ADE20K dataset for different incremental class learning scenarios.

| | 100-50 | | | 100-10 | | | | | | | 50-50 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | *1-100* | *101-150* | *all* | *1-100* | *100-110* | *110-120* | *120-130* | *130-140* | *140-150* | *all* | *1-50* | *51-100* | *101-150* | *all* |
| FT | 0.0 | 24.9 | 8.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 16.6 | 1.1 | 0.0 | 0.0 | 22.0 | 7.3 |
| LwF [15] | 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 | 13.9 |
| LwF-MC [13] | 34.2 | 10.5 | 26.3 | 18.7 | 2.5 | 8.7 | 4.1 | 6.5 | 5.1 | 14.3 | 27.8 | 7.0 | 10.4 | 15.1 |
| ILT [27] | 22.9 | 18.9 | 21.6 | 0.3 | 0.0 | 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** | **27.0** |
| 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 | 38.9 |

TABLE 4: Mean IoU (in %) on the Cityscapes dataset for different incremental class learning scenarios.

| | vehicles | | | non-driving | | | 11-2 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | *old* | *novel* | *all* | *old* | *novel* | *all* | *1-11* | *12-13* | *14-15* | *16-17* | *18-19* | *all* |
| FT | 0.0 | **71.0** | 22.4 | 0.0 | 69.0 | 21.8 | 0.0 | 0.0 | 0.0 | 0.0 | **57.3** | 6.0 |
| LwF [15] | 69.0 | 44.4 | 61.3 | 63.9 | 63.1 | 63.6 | 27.8 | 0.0 | 4.8 | 38.5 | 49.7 | 25.9 |
| LwF-MC [13] | 58.9 | 47.0 | 55.2 | 48.7 | 58.5 | 51.8 | 60.6 | 0.0 | 0.0 | 9.6 | 33.5 | 39.6 |
| ILT [27] | 68.3 | 37.4 | 58.5 | 64.9 | 54.8 | 61.7 | 28.9 | 0.0 | 6.8 | 27.3 | 33.2 | 23.8 |
| MiB | **69.4** | 66.6 | **68.5** | **66.4** | **70.0** | **67.6** | **70.2** | **33.7** | **53.7** | **49.0** | 53.9 | **60.7** |
| Joint | 72.8 | 73.8 | 73.1 | 72.5 | 74.3 | 73.1 | 73.6 | 68.3 | 79.0 | 72.4 | 69.9 | 73.1 |

TABLE 5: Results on point-based weakly supervised object segmentation on Pascal-VOC (mIoU in %).

| **Method** | **mIoU** | **P-Acc** |
|---|---|---|
| Img Lvl [12] | 33.2 | 76.0 |
| Img Lvl + PCE [12] | 34.7 | 58.9 |
| Img Lvl + PCE + Obj [12] | 42.1 | 81.5 |
| PCE + bkg | 38.8 | 81.9 |
| MiB (lr $10^{-5}$) | 45.3 | 82.3 |
| MiB (lr $10^{-4}$) | **46.7** | **83.6** |
| Full Supervision | 58.8 | 89.9 |

TABLE 6: Results on scribble-based weakly supervised object segmentation on Pascal-VOC (mIoU in %).

| **Method** | **wo/ CRF** | **w/ CRF** |
|---|---|---|
| PCE | 69.5 | 72.8 |
| MiB | **72.3** | **75.1** |
| Scribble-Sup [19] | - | 63.1 |
| NormalizedCut [50] | 72.8 | 74.5 |
| KernelCut [62] | 73.0 | 75.0 |
| BPG [63] | **73.2** | **76.0** |
| Full Supervision | 75.8 | 76.4 |

TABLE 7: Results on point-based weakly supervised scene parsing on ADE20K (mIoU in %).

| | Our protocol | | [20] protocol | |
|---|---|---|---|---|
| **Method** | **mIoU** | **P-Acc** | **mIoU** | **P-Acc** |
| PCE | 22.4 | 60.9 | 20.2 (17.7) | 58.3 (58.0) |
| PDML [20] | 21.1 | 56.6 | 19.3 (19.6) | 55.5 (61.0) |
| MiB | **22.9** | **62.2** | **21.0** | **59.5** |
| Full Supervision | 29.7 | 68.8 | 25.1 | 66.0 |

performance on a multi-step setting, where we add two classes in four different steps. We start from 11 classes (*road, sidewalk, building, wall, fence, pole, light, sign, vegetation, terrain, sky*) and then we add *person and rider*, then *car and truck*, then *bus and train*, and finally *motorcycle and bicycle*. In Table 4, we report the results for each group of classes, after all the classes have been learned. As before, fine-tuning the network provide good performance on the novel classes but it suffers catastrophic forgetting and obtains 0.0% mIoU on old classes. LwF-MC obtains a good results on the first set of classes (*1-11*) but it struggles to learn the novel ones, especially considering the intermediate classes. LwF and ILT demonstrate a similar behavior, forgetting old classes both on the first and intermediate steps. However, LwF achieves better results on the novel ones, exceeding ILT by 16.5%. Our method outperforms all the other baselines by more than 21% mIoU. In particular, it is the only method the only method able to maintain good performances on the intermediate steps.

### 6.3 Semantic Segmentation with Weak Supervision

#### 6.3.1 Point-based Object Segmentation on Pascal-VOC

Following the work of [12], we evaluate our method on object segmentation using the Pascal-VOC dataset and the point annotations the authors' provide. Differently from [12], we employ a Resnet-101 [66] as our backbone, with the modification of dilated convolutions, as in standard state-of-the-art architectures [8]. To recover the input resolution, we add after a bilinear interpolation layer on top of the Resnet-101, without additional trainable parameters. We initialize the backbone with an ImageNet pretrained model, as in [12], using the weights provided by [65]. However, differently from them, we did not initialize the classifier since we were not able to establish the correct mapping among the ImageNet indices published by [12] and the ImageNet classes. For fairness of comparison, we implemented [12] using our same backbone and training protocol. We follow [8] and we train the network using SGD with momentum 0.9, weight decay $10^{-4}$, and the same learning rate polynomial policy $base\_lr \times (1 - \frac{iteration}{max\_iterations})^{0.9}$. We use an initial learning rate of $10^{-5}$ for the methods in [12] and $10^{-4}$ for the fully-supervised baseline. We report the results for our method with both learning rates. For all the methods we train the network using a batch size of 24 for 30 epochs. We crop the images to $512 \times 512$ during training and we apply the same data augmentation of [8].

**Results.** In Table 5 we report the mIoU and the overall pixel accuracy (*P-Acc*). The first three rows of the table refer to the methods proposed in [12]. In particular, we refer to *Img Lvl* as the model trained only using Eq.2 of [12] which does not consider points location, but only image-level labels. This method achieves 33.2% mIoU, which is 4.4% better than the one reported by [12]. In the second row, we add the partial cross-entropy (PCE) loss, as

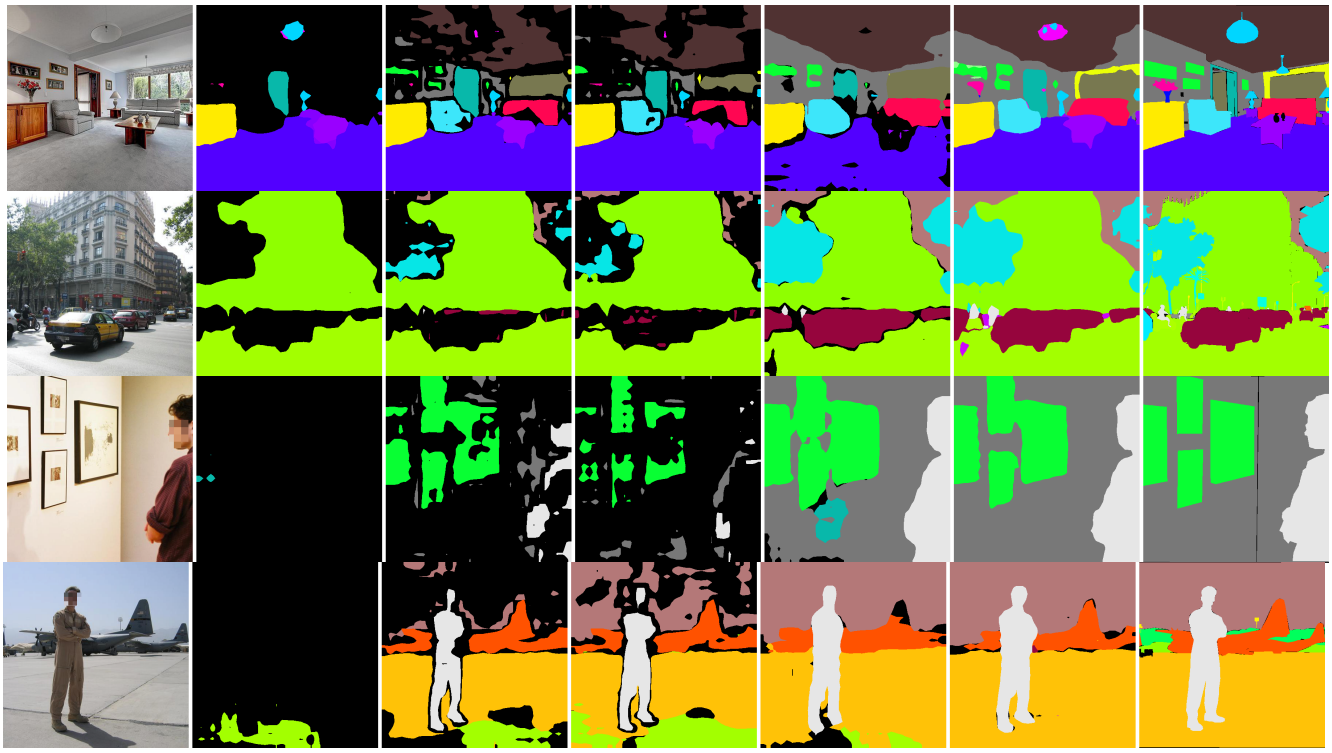


Fig. 3: Qualitative results on the *100-50* setting of the ADE20K dataset using different incremental methods. The image demonstrates the superiority of our approach on both new (*e.g. building*, *floor*, *table*) and old (*e.g. car*, *wall*, *person*) classes. From left to right: image, FT, LwF [15], ILT [27], LwF-MC [13], our method, and the ground-truth. Best viewed in color.

proposed by [12], and we refer to it as *Img Lvl + PCE*. For this method, we use all the points available in the annotation and we do not weight them ($\alpha_i = 1, \forall i \in \mathcal{I}_S$). Adding the PCE improves the mIoU of 1.5% but deteriorates the pixel accuracy by 17.1%. This is due to the bias of the model toward the semantic classes, which led the model to assign an object label even to background pixels, which are the majority. However, introducing the Objectness Prior (*Img Lvl + PCE + Obj*) [12] computed on an additional dataset (following [12]) improves the results, achieving 42.1% mIoU and 81.5% pixel accuracy.

Nonetheless, our method outperforms all the three variants of [12]. In particular, we report it twice to be fair in the comparison: *MiB (lr* $10^{-5}$*)* employs the same learning rate of [12], while *MiB (lr* $10^{-4}$*)* uses a learning rate $10^{-4}$ which we found better With both learning rates, MiB achieves better performance than [12], demonstrating that our method is better in modeling the unknown pixels. In particular, *MiB (lr* $10^{-4}$*)* achieves 46.7% mIoU and 83.6% pixel accuracy, being inferior to the fully supervised baselines of 12.1% 6.3% respectively. We would like to highlight that, differently from [12], MiB does not use any objectness prior.

Finally, to prove that the improvement of our method is given by the way we model unlabeled pixels and not by rescaling the contribution of the background, we introduce the baseline referred as *PCE + bkg*. In this method, we still use Eq. (14), but we consider as possible class for the unlabeled pixels only the background. As can be noted in Table 5, this method is not able to learn properly the classes, obtaining 38.8% mIoU, which is 7.9% less than *MiB (lr* $10^{-4}$*)*. In particular, considering all the unlabeled pixels as background biases PCE+bkg toward this class. Instead, MiB models the unlabeled pixels using the prior given by the point labels, *i.e.* $\mathcal{U} = \mathcal{U}_x$, pushing the network to predict them either as background or as any of the annotated classes.

#### 6.3.2 Scribble-based Object Segmentation on Pascal-VOC

To evaluate our method on scribble-supervised semantic segmentation we followed the experimental protocol defined in [62], [63], using the Pascal-VOC 2012 dataset and the scribble annotation released by [19]. We employ the Deeplab-v2 architecture [9] with the Resnet-101 backbone [66]. As in [62], [63], we use dilated convolutions obtaining an output resolution 8 times smaller than the input. Moreover, we follow the strategy used in [63] and we train the network on a single-scale resolution using a polynomial learning rate policy $base\_lr \times (1 - \frac{iteration}{max\_iterations})^{0.9}$ with a batch size of 10 images and with $base\_lr = 2.5 \times 10^{-4}$, momentum 0.9 and weight decay $5 \times 10^{-4}$. We train the network for 20K iterations using $321 \times 321$ cropped images, after applying horizontally flip (left-right) and randomly scaling the input images (from 0.5 to 2.0). In the testing stage, similarly to previous works [62], [63] we use multi-scale inputs (*i.e.* [0.5, 0.75, 1.0, 1.25, 1.5]) with max voting to get the final prediction.

**Results** Table 6.3.2 reports the mIoU with and without applying the dense CRF [68] post-processing using scribble-supervision. The top part reports the results of methods not explicitly designed for the scribble annotation (i.e. the PCE baseline and MiB), while the following reports the scribble-specific state-of-the-art approaches [19], [50], [50], [63], and the fully-supervised upper-bound. As for point supervision, the PCE baseline trains the network using the cross-entropy only on labeled pixels, as described in Eq.(12). We note that PCE is already a competitive baseline, obtaining 72.8% mIoU, *i.e.* 3.6% below the fully-supervised upper bound (76.4%), demonstrating that the model is able to extract meaningful information even from few pixels. However, introducing our loss as reported in Eq. (14), we are able to outperform the PCE baseline. In particular, MiB obtains 72.3% (+1.8% w.r.t. PCE) without CRF and 75.1% (+2.3) with CRF. This

remarks that unlabeled pixels bring crucial information to improve the results.

Comparing MiB with the state-of-the-art methods, we note that it achieves competitive performance. In particular, comparing with NormalizedCut [50] and KernelCut [62], we see that MiB obtains inferior performance without using the CRF, but it achieves superior performance while using it (+0.6% w.r.t. NormalizedCut and +0.1% w.r.t. KernelCut). We argue that KernelCut and NormalizedCut are superior to MiB without CRF since they already integrate the CRF in their training objective to better model the boundaries. However, the CRF post-processing is useful to correct boundary predictions and improves MiB performance while having less impact on NormalizedCut and KernelCut. Finally, BPG [63] achieves better results than MiB both without (+0.9%) and with (+0.9%) CRF post-processing. However, we remark that BPG introduces two sub-networks in the segmentation architecture to model boundaries, largely increasing the number of parameters and requiring additional supervision for boundary prediction. On the contrary, MiB is a general method that introduces only a loss function on unlabeled pixels, without requiring either to modify the network architecture or additional supervision.

#### 6.3.3 Scene Parsing on ADE20K

We evaluate our method also on the scene parsing task, as proposed by [20]. The task is based on the ADE20K dataset and on the point annotation used by [20], which have been released in the LID Challenge 2020 [1]. Since the code of [20] has not been released, we re-implemented it following the details and the algorithm provided in the paper. Moreover, we report the results using two different training protocols, since we noted that the protocol of [20] was sub-optimal. Both protocols employ a Resnet-101 [66] architecture with dilated convolutions, followed by a bilinear interpolation layer to recover the input resolution, as proposed by [20]. The first protocol we implemented is the same of [20]. The network is trained using SGD with momentum 0.9, weight decay $5 \times 10^{-4}$ and an initial learning rate of $2.5 \times 10^{-4}$ that is decayed following a polynomial schedule $base\_lr \times (1 - \frac{iteration}{max\_iterations})^{0.8}$. The dataset is iterated using a batch size of 16 and the images are randomly cropped with size $321 \times 321$. However, the number of epochs has not been specified in [20] and we train the network for 60 epochs. The second protocol follows the protocol and hyperparameters described in Sec. 6.3.1. We only change the base learning rate that we set to $10^{-3}$.

**Results.** The results are shown in Table 7 where we report the mean Intersection-over-Union (mIoU) and the overall pixel accuracy (P-Acc). In the bracket we reported the numbers as reported in [20]. Following [20], we implemented the partial cross entropy (PCE) baseline, which only applies the cross-entropy loss on the pixels to whom a label is provided, as described in Eq.12. As observed by [20], this is a strong baseline for point-supervised methods: it achieves 22.4% mIoU using our protocol and 20.2% mIoU on the one of [20]. However, we note that the result obtained by this baseline are better than the one found in [20] with a gap of 2.5% mIoU and 0.3% pixel accuracy. The PDML [20] baseline obtains results in line with [20]. However, comparing it with the PCE baseline, it perform worse, exhibiting a drop of performance of 1.3% and 0.9% mIoU in the two protocols. However, our method outperforms both baselines. It achieves 22.9% mIoU using our protocol, which is 0.5% more than PCE, and 21.0% using the [20] protocol, with a gap of 0.8% with respect to PCE.

## 7 Conclusions

In this work, we proposed a general loss function for semantic segmentation under partial or weak supervision. This formulation considers unlabeled pixels as ground-truth annotation for *any* possible class that pixel might contain. We considered two application scenarios for the method, incremental class learning and point-based weakly supervised semantic segmentation. In incremental class learning, we analyze the realistic scenario where the new training set does not provide annotations for old classes, leading to the semantic shift of the background class and exacerbating the catastrophic forgetting problem. We addressed this issue by revisiting standard distillation-based ICL algorithms with our general principle in both cross-entropy and distillation losses, where the uncertainty on the unlabeled/background pixels is on the presence of old classes for the former and of new classes for the latter. Additionally, we propose a classifiers' initialization strategy which allows our network to explicitly model the semantic shift of the background. Results show that our approach outperforms regularization-based ICL methods by a large margin, considering both small and large scale datasets.

In a second series of experiments, we apply our general formulation to semantic segmentation with point and scribble supervision, where the prior on unlabeled pixels is given by the set of classes present in the current image. We show how our model obtains competitive performance with respect previous approaches in both objects segmentation and scene parsing, without any additional prior on the objects or without making assumptions on the provided annotations.

Future works might consider the application of the approach under different levels of weak supervision (e.g. bounding boxes [49], polygons [18]) and on new tasks with partial knowledge on the unlabeled pixels, such as zero-shot learning [69].

## Acknowledgments

We acknowledge financial support from ERC grant 637076 - RoboExNovo obtained by Barbara Caputo. This work has been partially funded by the ERC (853489-DEXIM) and the DFG (2064/1–Project number 390727645). Computational resources were partially provided by HPC@POLITO[2].

## References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 1, 2

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 1, 2, 5, 7

[3] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017. 1, 2, 7, 8

[4] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018. 1, 2

[5] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017. 1, 2, 7

[6] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017. 1, 2

1. https://lidchallenge.github.io/challenge.html, see track 2.

2. http://www.hpc.polito.it

[7] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *ECCV*, 2018. 1, 2

[8] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017. 1, 2, 7, 10

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE T-PAMI*, vol. 40, no. 4, pp. 834–848, 2017. 1, 2, 11

[10] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *CVPR*, 2016. 1

[11] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *European conference on computer vision*. Springer, 2016, pp. 695–711. 1, 3

[12] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei, "What's the point: Semantic segmentation with point supervision," in *European conference on computer vision*. Springer, 2016, pp. 549–565. 1, 2, 3, 6, 7, 10, 11

[13] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *CVPR*, 2017. 1, 3, 4, 5, 7, 9, 10, 11

[14] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165. 1, 2

[15] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE T-PAMI*, vol. 40, no. 12, pp. 2935–2947, 2017. 1, 2, 3, 4, 5, 7, 8, 9, 10, 11

[16] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *ECCV*, 2018. 1, 3, 4, 5

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. 1, 3, 5

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223. 2, 7, 12

[19] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3159–3167. 2, 3, 6, 10, 11

[20] R. Qian, Y. Wei, H. Shi, J. Li, J. Liu, and T. Huang, "Weakly supervised scene parsing with point-based distance metric learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8843–8850. 2, 3, 6, 10, 12

[21] F. Cermelli, M. Mancini, S. R. Bulo, E. Ricci, and B. Caputo, "Modeling the background for incremental learning in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9233–9242. 2

[22] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE T-PAMI*, vol. 39, no. 12, pp. 2481–2495, 2017. 2

[23] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *ECCV*, 2016. 2

[24] F. Ozdemir, P. Fuernstahl, and O. Goksel, "Learn the new, keep the old: Extending pretrained models with new anatomy and images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2018, pp. 361–369. 2

[25] F. Ozdemir and O. Goksel, "Extending pretrained segmentation networks with additional anatomical structures," *International journal of computer assisted radiology and surgery*, pp. 1–9, 2019. 2

[26] O. Tasar, Y. Tarabalka, and P. Alliez, "Incremental learning for semantic segmentation of large-scale remote sensing data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 9, pp. 3524–3537, 2019. 2

[27] U. Michieli and P. Zanuttigh, "Incremental learning techniques for semantic segmentation," in *ICCV-WS*, 2019, pp. 0–0. 2, 5, 7, 8, 9, 10, 11

[28] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "Continual learning: A comparative study on how to defy forgetting in classification tasks," 2019. 3, 8

[29] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *NeurIPS*, 2017. 3, 4

[30] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *CVPR*, 2019. 3

[31] C. Wu, L. Herranz, X. Liu, J. van de Weijer, B. Raducanu *et al.*, "Memory replay gans: Learning to generate new categories without forgetting," in *NeurIPS*, 2018. 3

[32] O. Ostapenko, M. Puscas, T. Klein, P. Jahnichen, and M. Nabi, "Learning to remember: A synaptic plasticity driven framework for continual learning," in *CVPR*, 2019. 3

[33] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. 3, 4, 7, 9

[34] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *ECCV*, 2018. 3, 7, 9

[35] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *ICML*, 2017. 3, 7, 9

[36] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in *CVPR*, 2019. 3

[37] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *CVPR*, 2018. 3

[38] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *ECCV*, 2018. 3

[39] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016. 3

[40] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *CVPR*, 2019. 3

[41] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *ECCV*, 2018. 3

[42] E. Fini, S. Lathuilière, E. Sangineto, M. Nabi, and E. Ricci, "Online continual learning under extreme memory constraints," *ECCV*, 2020. 3

[43] K. Shmelkov, C. Schmid, and K. Alahari, "Incremental learning of object detectors without catastrophic forgetting," in *ICCV*, 2017. 3, 5, 7, 8

[44] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 5267–5276. 3

[45] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023. 3

[46] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *European conference on computer vision*. Springer, 2020. 3

[47] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1635–1643. 3

[48] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1742–1750. 3

[49] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 876–885. 3, 12

[50] M. Tang, A. Djelouah, F. Perazzi, Y. Boykov, and C. Schroers, "Normalized cut loss for weakly-supervised cnn segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1818–1827. 3, 10, 11, 12

[51] S. J. Oh, R. Benenson, A. Khoreva, Z. Akata, M. Fritz, and B. Schiele, "Exploiting saliency for object segmentation from image level labels," in *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2017, pp. 5038–5047. 3

[52] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4981–4990. 3

[53] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2209–2218. 3

[54] N. Araslanov and S. Roth, "Single-stage semantic segmentation from image labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4253–4262. 3

[55] Y.-T. Chang, Q. Wang, W.-C. Hung, R. Piramuthu, Y.-H. Tsai, and M.-H. Yang, "Weakly-supervised semantic segmentation via sub-category exploration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8991–9000. 3

[56] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE*

*conference on computer vision and pattern recognition*, 2016, pp. 2921–2929. 3

[57] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626. 3

[58] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 16, no. 6, pp. 641–647, 1994. 3

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014. 3

[60] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 328–335. 3

[61] C. Rother, V. Kolmogorov, and A. Blake, "" grabcut" interactive foreground extraction using iterated graph cuts," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 309–314, 2004. 3

[62] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, "On regularized losses for weakly-supervised cnn segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 507–522. 3, 10, 11, 12

[63] B. Wang, G. Qi, S. Tang, T. Zhang, Y. Wei, L. Li, and Y. Zhang, "Boundary perception guidance: a scribble-supervised semantic segmentation approach," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019. 3, 10, 11, 12

[64] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *International Conference on Computer Vision (ICCV)*, 2011. 7

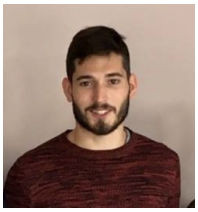
[65] S. Rota Bulò, L. Porzi, and P. Kontschieder, "In-place activated batchnorm for memory-optimized training of dnns," in *CVPR*, 2018. 7, 10

[66] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 7, 10, 11, 12

[67] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500. 7

[68] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in neural information processing systems*, vol. 24, pp. 109–117, 2011. 11

[69] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, "Semantic projection network for zero-and few-label semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8256–8265. 12



**Fabio Cermelli** is a Ph.D. student in Computer and Control Engineering at the Politecnico di Torino, funded by the Italian Insitute of Technology (IIT). He received his master thesis in Software Engineering (Computer Engineering) with honors at the Politecnico di Torino in 2018. He is member of the Visual Learning and Multimodal Applications Laboratory (VANDAL), supervised by Prof. Barbara Caputo. During his first year, he was a visiting Ph.D. student in the Technologies of Vision Laboratory at FBK.

**Massimiliano Mancini** is a post-doctoral researcher at the Cluster of Excellence in Machine Learning of the University of Tübingen, in the Explainable Machine Learning group, lead by Prof. Zeynep Akata. He completed his PhD in Engineering in Computer Science at the Sapienza University of Rome in 2020. During the Ph.D. he has been a member of the ELLIS PhD program, the Technologies of Vision lab at Fondazione Bruno Kessler, and the Visual Learning and Multimodal Applications Laboratory of the Italian Institute of Technology. His research interests include transfer learning across domains and learning from low supervision.

**Samuel Rota Buló** received the PhD in computer science at University of Venice in 2009. He worked there as PostDoc and held teaching positions until 2013. Then, he became a researcher for FBK in computer vision and machine learning. In 2017, he moved to Mapillary Research, where he worked as a senior researcher. He was awarded the prestigious Marr Prize in 2015. Since 2020, he is Research Scientist at Facebook. He serves on the editorial board for "Pattern Recognition" and "International Journal of Machine Learning and Cybernetics" and is regularly on the program committee of international conferences of his field. He participated to several EU projects (SIMBAD, VENTURI, REPLICATE).

**Elisa Ricci** is an associate professor at University of Trento and a researcher at Fondazione Bruno Kessler. She received her PhD from the University of Perugia in 2008. She has since been a post-doctoral researcher at Idiap Research Institute and an assistant professor at University of Perugia. She was also a visiting researcher at University of Bristol. Her research interests are mainly in the areas of computer vision and machine learning.

**Barbara Caputo** is Full Professor at the DAUIN Department of Control and Computer Engineering of Politecnico di Torino and Principal Investigator at the Italian Institute of Technology (IIT), where she leads the Visual Learning and Multimodal Applications Laboratory (VANDAL). Her main research interest is to develop algorithms for learning, recognition and categorization of visual and multimodal patterns for artificial autonomous systems. These features are crucial to enable robots to represent and understand their surroundings, to learn and reason about it, and ultimately to equip them with cognitive capabilities. Her research is sponsored by the Swiss National Science Foundation (SNSF), the Italian Ministry for Education, University and Research (MIUR), the European Commission (EC) and the European Research Council (ERC).