

# Weighting Pseudo-Labels via High-Activation Feature Index Similarity and Object Detection for Semi-Supervised Segmentation

Prantik Howlader<sup>1</sup>, Hieu Le<sup>2</sup>, and Dimitris Samaras<sup>1</sup>

<sup>1</sup> Stony Brook University, New York, USA

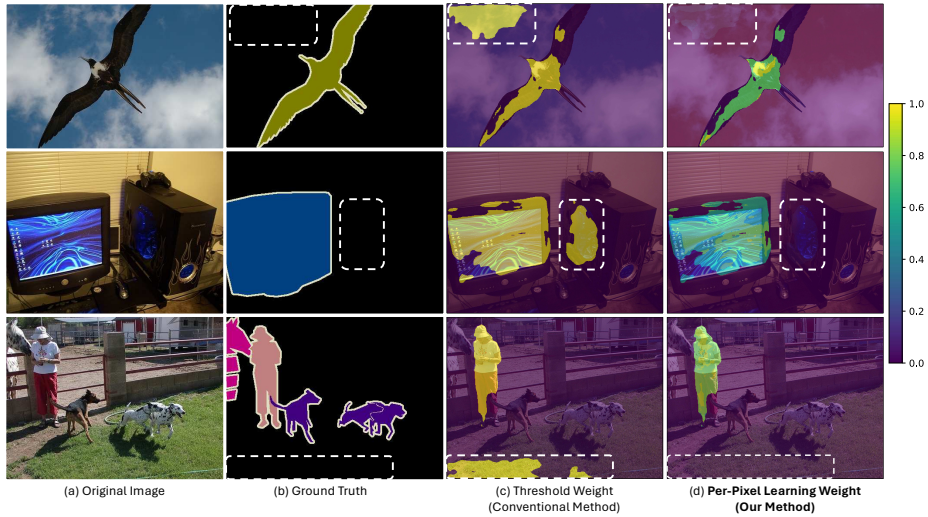
<sup>2</sup> EPFL, Lausanne, Switzerland

**Abstract.** Semi-supervised semantic segmentation methods leverage unlabeled data by pseudo-labeling them. Thus the success of these methods hinges on the reliability of the pseudo-labels. Existing methods mostly choose high-confidence pixels in an effort to avoid erroneous pseudo-labels. However, high confidence does not guarantee correct pseudo-labels especially in the initial training iterations. In this paper, we propose a novel approach to reliably learn from pseudo-labels. First, we unify the predictions from a trained object detector and a semantic segmentation model to identify reliable pseudo-label pixels. Second, we assign different learning weights to pseudo-labeled pixels to avoid noisy training signals. To determine these weights, we first use the reliable pseudo-label pixels identified from the first step and labeled pixels to construct a prototype for each class. Then, the per-pixel weight is the structural similarity between the pixel and the prototype measured via rank-statistics similarity. This metric is robust to noise, making it better suited for comparing features from unlabeled images, particularly in the initial training phases where wrong pseudo labels are prone to occur. We show that our method can be easily integrated into four semi-supervised semantic segmentation frameworks, and improves them in both Cityscapes and Pascal VOC datasets. Code is available at <https://github.com/cvlab-stonybrook/Weighting-Pseudo-Labels>.

## 1 Introduction

Semantic segmentation is essential for various applications, including autonomous driving [3, 26], and drone imagery [25, 48]. However, current models require large-scale pixel-level annotations for training, which are laborious and expensive to collect [52–55]. Semi-supervised segmentation [1, 23] alleviates this data dependency by learning from a limited set of labeled images and numerous unlabeled images.

An effective semi-supervised strategy is pseudo-labeling via a teacher-student framework [31, 45, 58, 69]. This strategy typically involves using a teacher model trained on limited labeled images to pseudo-label the unlabeled ones and then using these pseudo-labels as additional supervision to train the student model. The correctness of the pseudo labels is a major concern, as wrongly pseudo-labeled



**Fig. 1:** Per-pixel Learning Weight Visualization (heat-map). Our Per-pixel Learning Weight shows that the weight on unreliable high-confidence pseudo-labels (dotted white box) is reduced in contrast to conventional confidence thresholding ( $\geq 0.95$ ). Pseudo-labels are generated using AugSeg [65] after 50 epochs for  $\frac{1}{16}$  Pascal VOC Dataset.

pixels might never be corrected throughout the training process, leading to “confirmation bias” [33,47]. To remedy this, previous works [1,14,23,34,45,65,70] only pseudo-label pixels with high confidence scores. However, the confidence score is a sub-optimal proxy for pseudo-label correctness, especially in early training epochs. For example, in the initial training epochs on Pascal VOC dataset ( $\frac{1}{16}$  data partition) with a confidence threshold of 0.95,  $\sim 20\%$  of the pseudo-labels are still incorrect (see Section 5).

To address this issue, we introduce a novel approach for estimating the reliability of each pixel’s pseudo-label and then assign a learning weight to each pixel based on its reliability measure. The key observation is that pixels belonging to the same category frequently share a subset of  $k$  maximally activated representation components, *i.e.*, top- $k$  rank statistics [19,56]. For example, considering the last layer features of a network pre-trained on the Pascal VOC Dataset, we observe that pixel representations in the “dog” category typically exhibit the top-5 highest magnitudes in the 10<sup>th</sup>, 71<sup>st</sup>, 97<sup>th</sup>, 98<sup>th</sup>, and 111<sup>th</sup> dimensions of their 256-dim feature embeddings. Therefore, any pixel whose pixel representation exhibits the top-5 highest magnitudes in the same feature dimensions (*i.e.*, 10<sup>th</sup>, 71<sup>st</sup>, 97<sup>th</sup>, 98<sup>th</sup>, and 111<sup>th</sup> dimensions) has a high probability of belonging to the “dog” category. Notably, this pattern of index-based consistency is present even during the early stages of training the segmentation model with limited training data (see Section 5). On the other hand, the magnitude of each component in the embedding tends to vary significantly during the training, making value-based metrics such as entropy or confidence-score less reliable. As can be seen in Fig. 1, even pixels with very high confidence scores ( $\geq 0.95$ ) can still have incorrect

pseudo-labels (Fig. 1 (c)). However, these pixels are assigned very small weights by our method (Fig. 1 (d)) since their top-k maximally activated representation components are inconsistent with the majority of the same category pixels.

The question is how to identify these subsets of k maximally activated representation components for each class. To do so, we construct a class pixel-prototype by using labeled pixel representations and selecting a set of highly reliable pixel representations from unlabeled pixels. While previous methods simply define reliable pixels as the ones with high confidence scores from the segmentation model, we further improve this by training an additional object detection model and use the trained segmentation and the object detection model as an ensemble model to identify reliable pixels. We assert that if they predict the same label for a pixel, then we consider the pseudo label of the pixel highly reliable. The agreement between these two models indicate the pseudo-label’s correctness because they have distinct underlying inductive mechanisms: while the object detector assigns a single label to a group of pixels based on a holistic view of the image crop, the segmentation model assigns a label for each pixel based on the “*local*” patch and the surrounding context. In our experiments, this ensemble model identifies these reliable pixels more accurately, compared to a single segmentation model. Note that we only train this additional object detector from scratch on the limited labeled images and only for object-based categories. Reliable pixels for categories such as “sky” and “building” are obtained only from labeled pixels.

We incorporate our method into the four semi-supervised segmentation methodologies—UniMatch [57], AugSeg [65], AEL [23], and U2PL [50]—notably improving segmentation results for each across all data partitions in Pascal VOC [13] and Cityscapes [8] datasets. In summary, our contributions are:

1. We propose a novel method for weighing pseudo-labels to alleviate the potential noisy pseudo-label issue in semi-supervised segmentation via comparing top k maximally activated representation components.
2. We propose a novel method to identify reliable pixels by unifying the predictions from object detection and semi-supervised semantic segmentation models. The object detector is trained solely on the limited labeled data.
3. We show that our method can be easily integrated into other approaches by integrating it into four state-of-the-art approaches and getting consistent improvements across all settings.

## 2 Related Work

Semi-supervised learning (SSL) is a heavily studied problem. Recent works in SSL has been categorized into consistency regularization [2, 11, 12, 15, 38, 41, 58], entropy minimization [5, 16] and pseudo-labeling [27–31, 45]. Here, we focus on pseudo-labeling and consistency regularization.

**Pseudo labeling:** Pseudo-labeling [31, 43] and self-training [36, 59] aim to train a model on labeled images to generate pseudo-labels for unlabeled data. Because pseudo-labels are noisy, most approaches [40, 42, 45, 69] are focused on

refining pseudo-labels. For example, PseudoSeg [69] focuses on improving the quality of pseudo-labels using grad-cam [42] based attention. To select reliable pseudo-labels, FixMatch [45] uses confidence thresholding while UPS [40] uses uncertainty. ST++ [58], prioritizes unlabeled images that can provide more reliable pseudo-pixels. Recent approaches, AEL [23], propose adaptive frameworks to prefer under-performing categories. U2PL [50] considers extracting reliable pseudo-labels from unreliable pseudo-labels. CFCCG [32] proposed using cross fusion and contour guidance to improve the pseudo-labels. However, all these pseudo-labeling-based approaches rely on a segmentation network trained on limited labeled images to generate pseudo-labels. They further select reliable pseudo-labels based on high confidence (low entropy). Rizve *et al.* [40] observes that these high confidence predictions on unlabeled images can be incorrect due to poor network calibration [18]. A recent work [24] uses ground-truth bounding box annotations in the unlabeled images to improve semi-supervised segmentation. We on the contrary do not use any extra supervision in the unlabeled data. We train our object detector solely on the limited labeled data to generate object proposals in the unlabeled data.

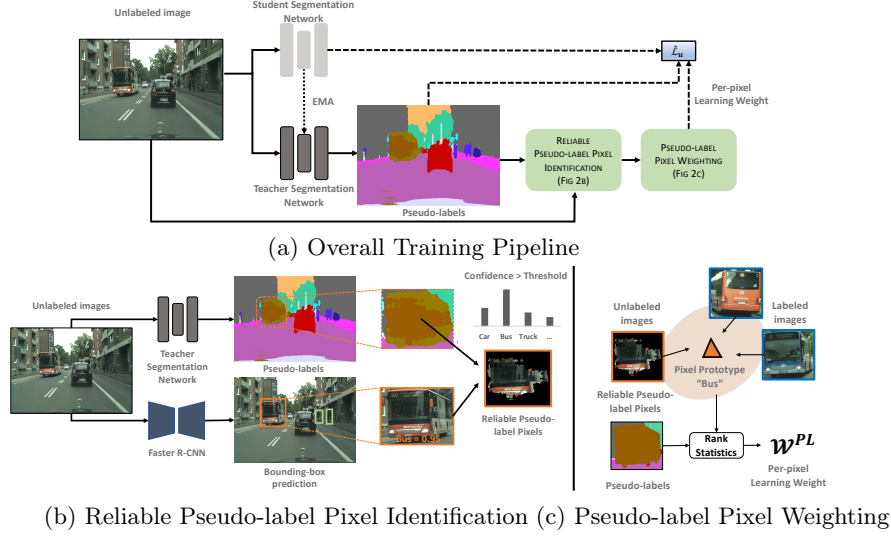
**Consistency Regularization:** Consistency Learning [45] enforces consistent predictions across different augmentations of unlabeled data. UniMatch [57] uses perturbation in the feature space to generate different augmentations of unlabeled data. ICT [49] uses Mixup [62] augmentation for consistency regularization. Recently, many methods [15, 58, 60] use Cutout [10], Cutmix [61], Classmix [37] as strong data augmentation. Consistency Regularization is used with pseudo-labeling techniques in Mixmatch [4] and TC-SSL [67]. In our work, we use a set of weak and strong data augmentations [1, 37, 57] to generate pseudo-labels.

### 3 Proposed Method

In semi-supervised semantic segmentation, we are given two distinct data sources: a set of labeled images denoted as  $\mathcal{D}^l = \{x_i^l, y_i^l\}_{i=1}^{N_l}$  and a collection of unlabeled images denoted as  $\mathcal{D}^u = \{x_i^u\}_{i=1}^{N_u}$ , where  $|\mathcal{D}_u| \gg |\mathcal{D}_l|$ . The central aim is to develop a semantic segmentation model that utilizes the knowledge from labeled and unlabeled data. This section first discusses the basic framework of semi-supervised semantic segmentation in Preliminary (Section 3.1). Next, we provide an overview of our proposed method (Section 3.2). Then we introduce the two steps that underpin our proposed novel method: (1) Reliable Pseudo-label Pixel Identification (Section 3.3), (2) Pseudo-label Pixel Weighting (Section 3.4).

#### 3.1 Preliminary

The conventional framework of semi-supervised semantic segmentation [1, 23, 65] is built on top of a student/teacher model. The teacher model shares the same architecture with the student model but uses a different set of parameters, which are updated by the exponential moving average (EMA) of the student model [23]. The teacher model generates a set of pseudo labels  $\hat{y}^u$  on the weakly



**Fig. 2: Overall Pipeline of our novel pseudo-labeling based semantic segmentation:** (a) End-to-end Teacher-Student Pipeline (b) We first identify pixels with reliable pseudo-labels using an object detector and segmentation model. The reliable pseudo-label pixels are defined as ones being labeled as the same class by both the detection and segmentation models with high confidence scores. (c) We constructed a pixel-representation prototype for each class using labeled images and identified reliable pseudo-label pixels. We then use rank statistics [19] to weight the pseudo-labels predicted by the teacher network.

augmented unlabeled data  $\mathcal{D}^u$ . Subsequently, the student model is trained on both weakly augmented labeled data  $\mathcal{D}^l$  with the ground truth and strongly augmented unlabeled data  $\mathcal{D}^u$  with the generated pseudo labels  $\hat{y}^u$ . The overall loss consists of the supervised loss  $\mathcal{L}_s$  and the unsupervised loss  $\mathcal{L}_u$ :

$$\mathcal{L}_s = \frac{1}{|\mathcal{D}^l|} \sum_{x^l \in \mathcal{D}^l} \frac{1}{WH} \sum_{i=1}^{WH} l_{ce}(y_i^l, p(x_i^{w,l})) \quad (1)$$

$$\mathcal{L}_u = \frac{1}{|\mathcal{D}^u|} \sum_{x^u \in \mathcal{D}^u} \frac{1}{WH} \sum_{i=1}^{WH} \mathbb{1}(\max(p(x_i^{w,u})) \geq \tau) l_i^u \quad (2)$$

$$l_i^u = l_{ce}(\hat{y}_i^u, p(x_i^{s,u})) \quad (3)$$

where  $x^{w,l}$  and  $x^{w,u}$  are weak augmentations of  $x^l$  and  $x^u$  respectively,  $x^{s,u}$  is strong augmentation of  $x^u$ .  $W$  and  $H$  correspond to the width and height of the input image,  $l_{ce}$  denotes the standard pixel-wise cross-entropy loss,  $p(\cdot)$  is the network prediction for labeled and unlabeled images,  $C$  is the number of classes.  $\tau$  is a predefined threshold to filter noisy labels. Further  $\hat{y}_i^u = \argmax(p(x_i^{w,u}))$  corresponds to the teachers prediction under weak augmentation view.

Consequently, the overall loss function can be defined as:

$$\mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_u \quad (4)$$

where  $\alpha$  controls the contribution of the unsupervised loss.

### 3.2 Overview

Fig. 2(a) presents a comprehensive view of our teacher-student framework method. There are two main ideas: we employ an additional object detector for better identifying reliable pseudo-label pixels, and we use rank statistics with class prototypes to assign a per-pixel learning weight for each pseudo-labeled pixel.

First, we use an ensemble model comprising of the teacher model and an additional Faster-RCNN [39] model to identify reliable pseudo-label pixels (Fig. 2(b)). We assert that if both models predict the same label for a pixel, then we consider the pseudo label highly certain. We use these reliable pseudo-label pixels and the pixels from the labeled images to construct class prototypes. The class prototypes are used to compute a per-pixel learning weight via rank statistics for training the network - Fig. 2(c).

### 3.3 Reliable Pseudo-label Pixel Identification

Semi-supervised segmentation aims to identify reliable pseudo-labels from the unlabeled images and use them for training. Existing methods [40,45,57,58,68,69] pseudo-label unlabeled images after filtering the pseudo-labels predicted by the teacher segmentation network based on confidence or entropy thresholds.

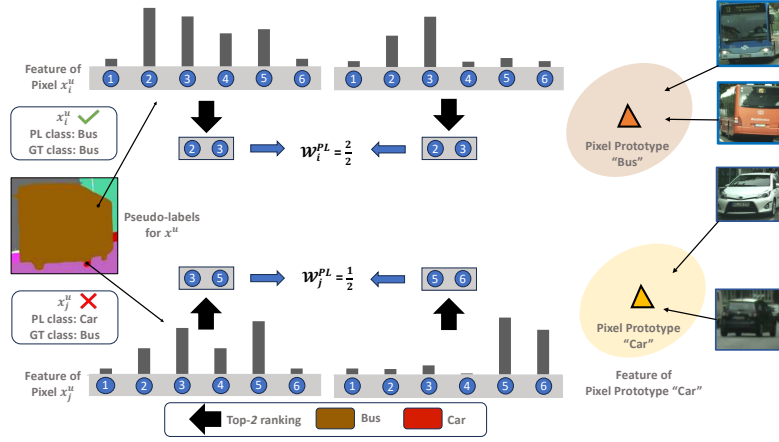
However, these reliable pseudo labels used during the training process are noisy, especially in the initial stages of training, due to the limited labeled data for training and poor model calibration [40]. To overcome this challenge, we propose a solution to reduce the reliance on segmentation model confidence by using an ensemble of teacher and an additional object detection model (Faster R-CNN [39]). Specifically, we first train a Faster R-CNN [39] only on the limited labeled data  $\mathcal{D}^l$ . We use the generated object proposals to complement the segmentation network in identifying a set of reliable pseudo-label pixels. We use two criteria to determine if a pixel is a reliable pseudo-label:

1. The segmentation and the detection model label the pixel as the same class.
2. The pixel confidence is higher than a predefined threshold.

The first criterion ensures that we exclusively consider pixels within the object proposal that share the same class as predicted by the segmentation network, i.e., only “*bus*” pixels in Fig. 2 (b). The second criterion considers pixels within the object proposal, on which the segmentation network is most confident, similar to previous work [1, 17, 57, 63]. Applying these two criteria, we identify a set of reliable pseudo-labeled pixels from the unlabeled images. The key point here is that if the two models agree over the label of a pixel, we consider the pseudo-label highly reliable because they have distinct underlying inductive mechanisms. While the object detector assigns a single label to a group of pixels based on the

holistic view of the image crop, the segmentation model assigns a label for each pixel based on the “*local*” patch and the surrounding context.

Note that we only re-purpose the segmentation labels in  $D_l$  to train the Faster R-CNN model. Given an image and its semantic segmentation mask, for each category, we define an object box for each set of connected pixels as the smallest bounding box containing them. Each box might contain more than one object, but it is not an issue for our semi-supervised segmentation method. We discard bounding boxes of background classes such as sky, vegetation, or buildings since their bounding boxes tend to cover almost the entire image. As we do not train the object detector on those background classes, reliable pixels for those classes only come from labeled images.



**Fig. 3: Demonstration of Pseudo-label Pixel Weighting via rank-statistics:** This diagram shows the top-2 ranking based pseudo-pixel weighing for two pixels  $x_i^u$  and  $x_j^u$  in unlabeled image  $x^u$ . PL class is Pseudo-label class, GT class is Ground Truth class. **Note, top-2 ranking is same between  $x_i^u$  and bus pixel prototype, while different between  $x_j^u$  and car pixel prototype.**

### 3.4 Pseudo-label Pixel Weighting

We assign an adaptive per-pixel learning weight to each pseudo-labeled pixel to avoid noisy training signals. To determine this weight, we first use the identified reliable pseudo-label (Section 3.3) and labeled pixels to construct a prototype for each class. Then, the per-pixel weight is the structural similarity between the pixel and the prototype measured via rank-statistics [19, 56]. This metric only considers the overlap between indices of the highest value elements (top- $K$ ) of the two representation vectors, *i.e.*, which feature components are activated the most. Specifically, given two features  $z_i$  and  $z_j$ , we rank the feature dimensions in the vector  $z_i$  and  $z_j$  by their magnitude. We consider two features belonging to

the same class if the set of their top- $k$  ranking of their feature dimensions match  $\{top_k(z_i) = top_k(z_j)\}$ . Rank statistics similarity is not based on raw feature magnitudes but rather the structural similarity between vector representations, making rank statistics similarity more robust to noise [9, 35, 44, 56] when comparing high-dimensional feature representations. Thus, it is better suited for comparing features in the initial training stages of semi-supervised segmentation when the model is poorly calibrated, and the features and predictions are noisy [40] (See Section 5 for analysis on rank statistics being more robust to noise). Here, we use rank statistics for features extracted from the second-to-last layer of the segmentation model [1]. It serves as a secondary source of confirmation besides the confidence score to verify the correctness of the pseudo-labels.

Specifically, we first construct a per-class feature prototype using pixel features in labeled images and reliable pseudo-label pixels in unlabeled images (Section 3.3). The prototype is the mean of the pixels’ latent embeddings, which are the outputs of the penultimate layer of the segmentation model at the pixel locations. It is computationally expensive to extract features from all labeled and unlabeled images in each iteration. So, we save their features in a feature memory bank [1], which we query to generate the per-class feature prototype (more details on memory bank in supplementary material for details).

For each unlabeled image  $x^u$ , we compute a per-pixel learning weight  $\mathcal{W}^{PL} \in \mathbb{R}^{W \times H}$ , where  $W$  and  $H$  are the width and height of the image. We use a soft extension of ranking statistics [19] proposed in [64] as our similarity metric. It measures the similarity between two features as the number of shared elements in their sets of top- $k$  ranking. The per-pixel learning weight  $\mathcal{W}^{PL}$  of the pseudo-label at the position  $i$  in an unlabeled image  $x^u$ , which has been assigned a pseudo-label of class  $c$  by the teacher model, is defined as  $\mathcal{W}_i^{PL} = \frac{s}{k}$ . Here,  $s$  represents the count of common elements within the sets of top- $k$  ranking of the pixel feature at position  $j$  and the feature prototype for class  $c$  (we use  $k = 5$  for all our experiments). We illustrate in Fig. 3 that per-pixel weighting based on our approach for top-2 ranking based feature similarity. We observe that pseudo-labels misclassified as “car” are provided lower weights than pseudo-labels correctly classified as “Bus”.

We modify the unsupervised loss of the conventional teacher-student framework (Equation 2) to incorporate the per-pixel learning weight  $\mathcal{W}^{PL}$ . Hence, the unsupervised loss for our approach is:

$$\mathcal{L}_u = \frac{1}{|\mathcal{D}_u|} \sum_{x^u \in \mathcal{D}_u} \frac{1}{WH} \sum_{i=1}^{WH} \mathbb{1}(\max(p(x_i^{w,u})) \geq \tau) l_i^u \quad (5)$$

$$l_i^u = \mathcal{W}_i^{PL} l_{ce}(\hat{y}_i^u, p(x_i^{s,u})) \quad (6)$$

Thus, our overall loss function consists of the supervised loss (Equation 1) and the unsupervised loss (Equation 5), is:

$$\mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_u \quad (7)$$

where  $\alpha$  controls the amount of contribution of the unsupervised loss.



## 4 Experiments

### 4.1 Setup

**Datasets:** The **PASCAL VOC 2012** dataset [13] is a widely recognized benchmark for semantic segmentation, with 20 object categories and an additional background class. It is partitioned into training, validation, and testing subsets, containing 1464, 1449, and 1556 images (*Classic*), respectively. Following [7, 65], we also include the additional augmented dataset [20] (*Blender*), which includes a collection of 10582 training images. We adopt the same partition protocols in [7, 65] to evaluate our method in both *Classic* and *Blender* sets. The **Cityscapes** dataset [8], tailored for urban scene analysis, comprises of 30 classes, though only 19 of these are employed for scene parsing assessments. This dataset is divided into 2975 training images, 500 validation images, and 1525 testing images.

**Implementation Details:** For fair comparison, we adopt DeepLabv3+ [6] as the decoder in all of our experimental setups, and compare with both ResNet-101 and ResNet-50 [21] as the backbone architecture. We incorporate our method into the framework of four semi-supervised segmentation methods: AugSeg [65], AEL [23], U2PL [50] and Unimatch [57]. Our method serves as a pseudo-label weighting strategy to alleviate the influence of confident, noisy pseudo-labels during training, without changing their original architecture and training procedures. Consistent with common practices [50], we train our models on the Cityscapes and Pascal datasets at resolutions of 801 and 513, respectively. It is important to note that in the interest of maintaining a fair comparison, our approach, labeled as “UniMatch+Ours” employs a training resolution of 321, aligning with the resolution used by UniMatch for training on the Classic set of the Pascal VOC dataset. Further when training a baseline integrated with our method, we use the same weak and strong augmentations as used by the corresponding baseline.

After the Faster R-CNN is trained on only the limited labeled dataset, the confidence threshold to select bounding boxes in unlabeled images is set at 0.95 for Cityscapes and 0.85 for the Pascal Dataset, respectively. In all our experiments we set  $K = 5$  for rank statistics.

### 4.2 Comparison with State-of-the-Art Methods

We conduct experiments on two popular benchmarks: PASCAL VOC 2012 and Cityscapes. We integrate our method to four semi-supervised methods: AugSeg [65], AEL [23], U2PL [50] and Unimatch [57]. Note, UniMatch [57] is a consistency regularization based method. The results demonstrate consistent improvement in performance over the corresponding baselines across all partitions. This notable improvement across datasets robustly validates the effectiveness of our proposed approach.

**PASCAL VOC 2012 Dataset.** Table 1 presents a comparative analysis with

other SOTA methods for both the *Classic* and *Blender* sets. Our method consistently enhances the performance of all baseline methods across all data partitions for both the *Classic* and *Blender* sets. In particular, the most significant improvements are observed in the partition with the least labeled data ( $\frac{1}{16}$ ), where our method boosts the performance of AugSeg [65], AEL [23], U2PL [50] and Unimatch [57] by 2.0%, 3.7%, 3.1% and 1.5% respectively on the *Classic* set and 1.9%, 3.3%, 2.9% and 2.1% respectively on the *Blender* set.

**Cityscapes Dataset.** Table 2 presents a comparative analysis with other SOTA methods. Our method consistently enhances the performance of all baseline methods across all data partitions. We observe that similar to results in PASCAL VOC 2012 dataset, our method brings the biggest improvement for the least labeled data partition ( $\frac{1}{16}$ ), improving performance of AugSeg [65], AEL [23], U2PL [50] and Unimatch [57] by 2.1%, 3.4%, 3.1% and 1.8% respectively for ResNet-101 based encoder.

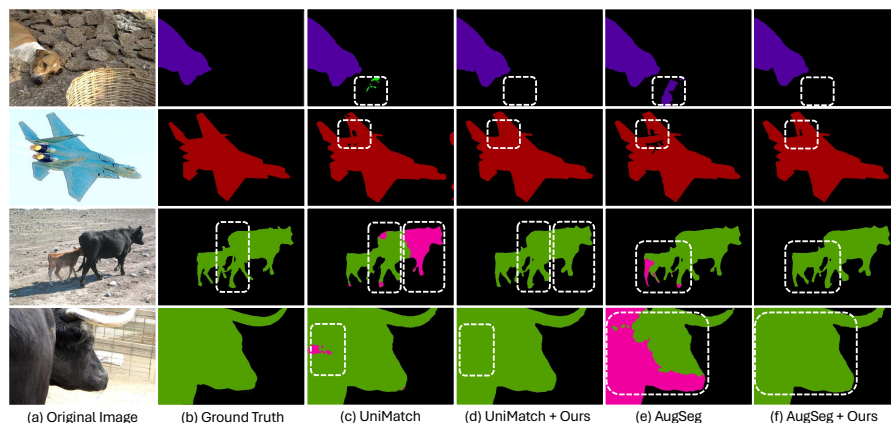
The consistent improvement in semi-supervised segmentation performance in both datasets and its ability to integrate in different methods substantiates the importance of our method for segmentation in the limited data domain.

**Table 1:** Quantitative results of different semi-supervised segmentation methods on Pascal VOC classic and blender set. We report Mean IoU under various partition protocols and show the improvements ( $\Delta$ ) over corresponding baseline.

Method	<i>Classic</i>					<i>Blender</i>		
	1/16	1/8	1/4	1/2	Full	1/16	1/8	1/4
<b>ResNet-50</b>								
<i>Supervised</i>	44.0	52.3	61.7	66.7	72.9	-	-	-
PC <sup>2</sup> Seg [66] [CVPR'21]	56.9	64.6	67.6	70.9	72.3	-	-	-
PseudoSeg [69] [ICLR'21]	56.9	64.6	67.6	70.9	72.2	-	-	-
ST++ [58] [CVPR'22]	-	-	-	-	-	72.6	74.4	75.4
AugSeg [65] [CVPR'23]	64.2	72.1	76.1	77.4	78.8	77.2	78.2	78.2
AugSeg + Ours/ $\Delta$	<b>66.4/2.2</b>	<b>73.9/1.8</b>	<b>77.6/1.5</b>	<b>78.3/0.9</b>	<b>79.3/0.5</b>	<b>79.5/2.3</b>	<b>79.1/0.9</b>	<b>79.8/1.6</b>
AEL [23] [NeurIPS'21]	62.9	64.1	70.3	72.7	74.0	74.1	76.1	77.9
AEL + Ours/ $\Delta$	<b>66.1/3.2</b>	<b>66.4/2.3</b>	<b>72.2/1.9</b>	<b>74.3/1.6</b>	<b>74.9/0.9</b>	<b>77.0/2.9</b>	<b>78.1/2.0</b>	<b>79.2/1.3</b>
U2PL [50] [CVPR'22]	63.3	65.5	71.6	73.8	75.1	74.7	77.4	77.5
U2PL + Ours/ $\Delta$	<b>66.0/2.7</b>	<b>67.6/2.1</b>	<b>73.2/1.6</b>	<b>75.5/1.7</b>	<b>75.9/0.8</b>	<b>77.7/3.0</b>	<b>79.8/2.4</b>	<b>79.4/1.9</b>
UniMatch [57] [CVPR'23]	71.9	72.5	76.0	77.4	78.7	78.1	79.0	79.1
UniMatch + Ours/ $\Delta$	<b>73.9/2.0</b>	<b>74.3/1.8</b>	<b>77.3/1.3</b>	<b>78.8/1.4</b>	<b>79.6/0.9</b>	<b>80.2/2.1</b>	<b>80.6/1.6</b>	<b>80.2/1.1</b>
<b>ResNet-101</b>								
<i>Supervised</i>	45.1	55.3	64.8	69.7	73.5	70.6	75.0	76.5
CPS [7] [CVPR'21]	64.1	67.4	71.7	75.9	-	72.2	75.8	77.6
PS-MT [34] [CVPR'22]	65.8	69.6	76.6	78.4	80.0	75.5	78.2	78.7
PCR [51] [NeurIPS'22]	70.1	74.7	77.2	78.5	80.7	78.6	80.7	80.8
DAW [46] [NeurIPS'23]	74.8	77.4	79.5	80.6	81.5	78.5	78.9	79.6
CFCG [32] [ICCV'23]	-	-	-	-	-	77.4	79.4	80.4
AugSeg [65] [CVPR'23]	71.1	75.5	78.8	80.3	81.4	79.3	81.5	80.5
AugSeg + Ours/ $\Delta$	<b>73.1/2.0</b>	<b>77.2/1.7</b>	<b>80.3/1.5</b>	<b>81.1/0.8</b>	<b>81.8/0.4</b>	<b>81.2/1.9</b>	<b>82.8/1.3</b>	<b>81.2/0.7</b>
AEL [23] [NeurIPS'21]	66.1	68.3	71.9	74.4	78.9	77.2	77.6	78.1
AEL+ Ours/ $\Delta$	<b>69.8/3.7</b>	<b>71.6/3.3</b>	<b>74.0/2.1</b>	<b>76.1/1.7</b>	<b>80.3/1.4</b>	<b>80.5/3.3</b>	<b>80.6/3.0</b>	<b>80.8/2.7</b>
U2PL [50] [CVPR'22]	68.0	69.2	73.7	76.2	79.5	77.2	79.0	79.3
U2PL + Ours/ $\Delta$	<b>71.1/3.1</b>	<b>72.0/2.8</b>	<b>75.6/1.9</b>	<b>78.0/1.8</b>	<b>81.0/1.5</b>	<b>80.1/2.9</b>	<b>81.5/2.5</b>	<b>81.6/2.3</b>
UniMatch [57] [CVPR'23]	75.2	77.2	78.8	79.9	81.2	80.9	81.9	80.4
UniMatch + Ours/ $\Delta$	<b>76.7/1.5</b>	<b>78.5/1.3</b>	<b>80.0/1.2</b>	<b>80.9/1.0</b>	<b>81.7/0.5</b>	<b>83.0/2.1</b>	<b>83.5/1.6</b>	<b>81.4/1.0</b>

**Table 2:** Quantitative results of different semi-supervised segmentation methods on the Cityscapes validation set. We report Mean IoU under various partition protocols and show the improvements ( $\Delta$ ) over the corresponding baseline.

Method	ResNet-50				ResNet-101			
	1/16	1/8	1/4	1/2	1/16	1/8	1/4	1/2
<i>Supervised</i>	63.34	68.73	74.14	76.62	66.3	72.8	75.0	78.0
CPS [7]	69.79	74.39	76.85	78.64	69.8	74.3	74.6	76.8
PS-MT [34]	-	75.76	76.92	77.64	-	76.9	77.6	79.1
PCR [51]	-	-	-	-	73.4	76.3	78.4	79.1
CFCG [32]	76.1	78.9	79.3	80.1	77.8	79.6	80.4	80.9
AugSeg [65]	73.7	76.4	78.7	79.3	75.2	77.8	79.6	80.4
AugSeg + Ours/ $\Delta$	<b>76.0/2.3</b>	<b>78.3/1.9</b>	<b>80.2/1.5</b>	<b>80.3/1.0</b>	<b>77.3/2.1</b>	<b>79.3/1.5</b>	<b>81.4/1.8</b>	<b>81.3/0.9</b>
AEL [23]	68.2	72.7	74.9	77.5	74.5	75.6	77.5	79.0
AEL + Ours/ $\Delta$	<b>71.6/3.4</b>	<b>75.4/2.7</b>	<b>77.0/2.1</b>	<b>79.9/2.4</b>	<b>77.9/3.4</b>	<b>78.8/3.2</b>	<b>79.6/2.1</b>	<b>80.1/1.1</b>
U2PL [50]	69.0	73.0	76.3	78.6	74.9	76.5	78.5	79.1
U2PL + Ours/ $\Delta$	<b>71.9/2.9</b>	<b>75.8/2.8</b>	<b>77.9/1.6</b>	<b>79.9/1.3</b>	<b>78.0/3.1</b>	<b>79.5/3.0</b>	<b>80.0/1.5</b>	<b>79.7/0.6</b>
UniMatch [57]	75.0	76.8	77.5	78.6	76.6	77.9	79.2	79.5
UniMatch + Ours/ $\Delta$	<b>77.1/2.1</b>	<b>78.5/1.7</b>	<b>78.7/1.2</b>	<b>79.3/0.7</b>	<b>78.4/1.8</b>	<b>79.6/1.7</b>	<b>80.5/1.3</b>	<b>80.7/1.2</b>



**Fig. 4: Qualitative Results on Pascal VOC** From left to right: original image, ground truth, UniMatch [57], UniMatch [57] + Ours, AugSeg [65], AugSeg [65] + Ours. The dotted white box shows the regions where our method improves the baseline.

**Qualitative Results** Figure 4 shows the results of different methods evaluated on the Pascal Validation set. Our method incorporated into both UniMatch [57] and AugSeg [65] shows clear improvements over their baselines. Both baselines get benefited from per-pixel pseudo label weighting, helping them achieve better segmentation performance. All methods are trained on  $\frac{1}{16}$  data partition.

## 5 Ablation Studies

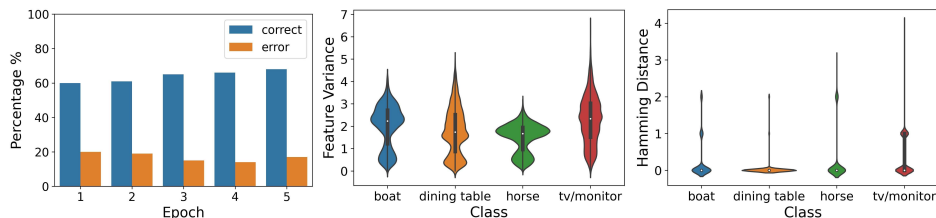
We conduct extensive experiments to study the impact of various components of our approach. Our experiments were conducted using the Pascal VOC dataset (*Classic*), focusing on one of its most challenging data partitions,  $\frac{1}{16}$ . For each

of these studies in this section, we have selected UniMatch [57] as the baseline.

### Analysis of pseudo-label noise and ranking based feature similarity

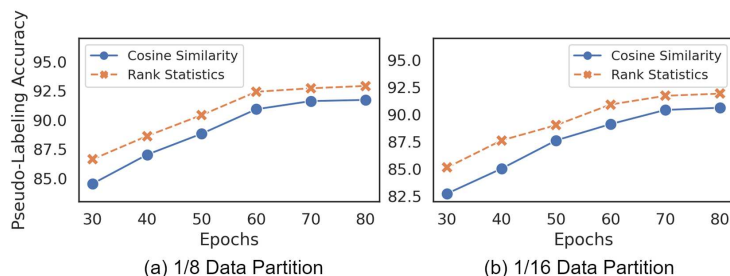
*The crux of our approach is we rely on the ranking of feature dimensions based on their magnitudes rather than their raw magnitudes in finding reliable pseudo-labels.* Here we analyse high confidence noisy pseudo-labels during training and the rationale behind using ranking of feature dimensions as similarity measure. Conventional pseudo-labeling approaches remove noisy pseudo-labels by confidence based thresholding. We observe in Fig. 5 (a) that even **high confidence ( $\geq 0.95$ ) pseudo-labels have significant proportions of incorrect pseudo-labels**. This observation validates that in the initial training epochs, the teacher network is miscalibrated leading to noisy pseudo-labels having high confidence [40].

Further, we analyse the features of the classes to understand why feature dimension based ranking is a good similarity metric. We first compare the variance of the features of correct pseudo-labels of 4 random classes in Fig. 5 (b). Further, we generated feature prototypes based on the most confident correct predictions (confidence threshold  $\geq 0.95$ ) of these classes in the labeled images. From the the class feature prototypes and the correct pseudo-labels we generate binary embeddings with the same dimension as the original features. With the indexes set to 1 based on the indexes of the original features with top-5 highest magnitudes. We calculate per-class hamming distance between the corresponding binary embeddings of the prototype and the correct pseudo-labels, as illustrated in Fig. 5 (c). Based on results in Fig. 5 (b) and (c), **we observe that feature values show more variation compared to indexes of the top-5 indexes based on magnitude**. This observation that rank ordering of the feature dimensions have less variation than their magnitudes aligns with previous works [9, 35, 44, 56].



**Fig. 5:** (a) The distribution of correct and incorrect pseudo-labels on Pascal VOC dataset above the confidence threshold of 0.95. (b) the variation of the features of correct pseudo-labels of 4 random classes, (c) the hamming distance between the binary embeddings of class prototypes of the most confident correct predictions in labeled images and the features of correct pseudo-labels.

**Comparison with cosine similarity** Our approach uses rank statistics to calculate the similarity between pixel features at a pseudo-pixel with a class



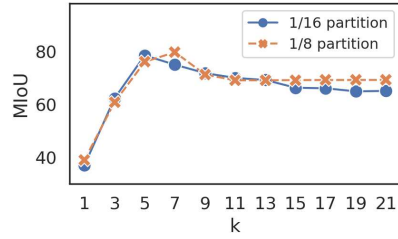
**Fig. 6:** Comparison of pseudo-labeling accuracy in the Pascal VOC unlabeled dataset, between two approaches of measuring similarity to assign the per-pixel Learning Weight  $\mathcal{W}^{PL}$  (Section 3.4).

prototype to assign per-pixel leaning weights (Section 3.4). A conventional approach is to use cosine-similarity. We compare the two approaches based on pseudo-label accuracy on both  $\frac{1}{16}$  and  $\frac{1}{8}$  partitions of the Pascal VOC dataset (Fig. 6). This confirms that rank statistics-based similarity performs better in comparing noisy features [56]. A plausible reason is, unlike cosine similarity, rank statistics matches features that share the same feature index ranking rather than their magnitude.

**The Effectiveness of Different Components of Our Approach** We ablate each component of our method step by step. Table 3 reports the studies. We use the basic teacher-student framework in Section 3.1 as our baseline, which achieves MIoU of 63.1, 67.4 and 70.18 under  $\frac{1}{16}$ ,  $\frac{1}{8}$  and  $\frac{1}{4}$  partition protocols respectively. As shown in the table, Pseudo-label Pixel Weighting (PPW) introduces a prototype-based pseudo-label per-pixel learning weight achieving an improvement of 3.9%, 2.7% and 2.5% under  $\frac{1}{16}$ ,  $\frac{1}{8}$  and  $\frac{1}{4}$  partition protocols respectively (class prototypes are only from labeled pixels). Reliable Pseudo-label Pixel Identification (RPPI) and and PPW together boost the performance by 7.4%, 5.9%, and 4.3%, demonstrating the effectiveness of our method (class prototypes from labeled pixels and reliable pseudo-labels).

**Impact of  $k$  in top- $k$  rank statistics** We analyze how our method performs with respect to  $k$ . The results on the 1/16 and 1/8 partition protocols of the Pascal VOC dataset are in Fig. 7. We observe that  $k = \{5, 7\}$  gave the best results, further low values of  $k$  lead to lower semi-supervised segmentation performance. A potential reason is that the number of dimensions to match is too few thus multiple prototypes can share the same top- $k$  dimensions, leading to pixels being matched with the wrong prototypes. A large value of  $k$ , however, makes it harder to match pixels with a prototype. This observation validates that in the context of ranking, that agreement among high-ranking coefficients is more important than the rest [9, 44]. For all our experiments we use  $k = 5$ .

**Per-class performance of our method** We compare the per-class performance of our method incorporated into UniMatch [57] with respect to the baseline under  $\frac{1}{16}$  Pascal VOC (*Classic*) partition. Table 4 shows the results. We



**Fig. 7:** Performance evolution with respect to  $k$ , for 1/16 and 1/8 partition protocols Pascal VOC Dataset

RPPI	PPW	1/16 (92)	1/8 (183)	1/4 (366)
		63.1	67.4	70.8
	✓	67.0	70.1	73.3
✓	✓	<b>70.5</b>	<b>73.3</b>	<b>75.1</b>

**Table 3:** Ablation study of different components: Reliable Pseudo-label Pixel Identification (RPPI) and Pseudo-label Pixel Weighting (PPW). RPPI and PPW both improve the performance

observe that our method improves the baseline across all classes. Using our rank statistics based pseudo-label weighting approach improves the performance of confusing classes like *Sheep* (**3.7%**) and *Sofa* (**3.4%**). These classes are often confused with Dog and Chair respectively.

**Table 4:** Per-class performance of our method incorporated into UniMatch [57], with respect to the baseline. Both methods are trained on  $\frac{1}{16}$  Pascal (*Classic*) set.

Method	mIoU	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	DiningTable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	TV/Monitor
UniMatch [57]	75.2	93.6	88.0	67.0	90.7	72.0	74.0	93.2	86.2	93.9	13.9	90.9	52.5	90.1	91.9	80.7	82.0	54.7	87.7	22.3	87.4	66.3
UniMatch + Ours	<b>76.7</b>	<b>94.0</b>	<b>90.9</b>	<b>67.8</b>	<b>91.1</b>	<b>73.7</b>	<b>74.8</b>	<b>93.3</b>	<b>86.3</b>	<b>94.2</b>	<b>15.2</b>	<b>94.2</b>	<b>54.2</b>	<b>91.1</b>	<b>92.1</b>	<b>81.5</b>	<b>82.2</b>	<b>60.9</b>	<b>91.4</b>	<b>25.7</b>	<b>88.1</b>	<b>68.3</b>

## 6 Conclusion

In this paper, we propose a novel approach that reduces the reliance on segmentation scores of the trained teacher model in pseudo-labeling unlabeled images. To do so, we propose a two-step approach. First, an ensemble of segmentation and detection models is used to identify reliable pseudo-labeled pixels. Second, a per-pixel weight is calculated to weigh the pseudo-labeled pixels. To determine this weight we first construct a prototype based on the labeled pixels and the reliable pseudo-labeled pixels identified from the first stage. Then the per-pixel weight is the similarity between the pixel and the prototype via rank-statistics. We show that our approach can be easily integrated into other approaches by integrating it into four approaches, which improves their performance in all data partitions if two recognized segmentation datasets, Cityscapes and Pascal VOC. **Acknowledgement.** This work is supported by the National Science Foundation (IIS-2123920, IIS-2212046).

# Weighting Pseudo-Labels via High-Activation Feature Index Similarity and Object Detection for Semi-Supervised Segmentation

## *Supplementary Material*

**Summary:** We provide additional analyses and results of our method, including:

- Obtaining Pseudo Object Detection Training Data from Segmentation Masks
- The Effect of the Object Detector in Improving Pseudo Label Accuracy
- Analysis of pseudo-labeling accuracy
- Analysis of training hyper-parameters
- Memory bank and the effect of its memory size
- Analysis of top-rank indices of class prototypes throughout training
- Comparison of pseudo-labeling accuracy with Euclidean distance
- Visualizing the top-ranked features
- Adapting our method to Transformer-based models
- Evaluation of performance on MS COCO
- Qualitative results

## 1 Obtaining Pseudo Object Detection Training Data from Segmentation Masks

We train the object detector from scratch using only the labeled segmentation data. Given an image and its semantic segmentation mask, we first separate the mask of each category into separate connected components. For each component, we extract the smallest bounding box containing it to use as pseudo-training data for the object detector. Apparently, each box might contain more than one object instance. Thus, our object detector is essentially trained to detect bounding boxes containing groups of pixels belonging to the same category rather than bounding boxes containing single object. This suffices for semantic segmentation tasks such as ours where instance differentiation is not necessary.

Further, we discard bounding boxes of background classes such as *sky*, *vegetation*, and *buildings* since their bounding boxes tend to cover almost the entire image. Specifically, for the Pascal VOC dataset, we generate bounding boxes for all classes except for “*background*”. For the Cityscapes dataset, we generate bounding boxes for the foreground classes: Person, Rider, Car, Bicycle, Motorcycle, Train, Truck, Bus, Traffic Light, and Traffic Sign.

## 2 The Effect of the Object Detector in Improving Pseudo Label Accuracy

The detection model is used in an ensemble with a segmentation model to identify reliable pseudo-label pixels. Here we analyze the performance of the detection model in limited labeled data scenarios and also how it improves the reliability of pseudo-labeled pixels when using together with the segmentation network. For our analysis, we use a Deeplabv3+ segmentation model with ResNet-101 backbone [6] and a Faster R-CNN object model. Both are trained on the  $\frac{1}{16}$  split of the Pascal VOC dataset (*Classic*) containing 92 labeled images.

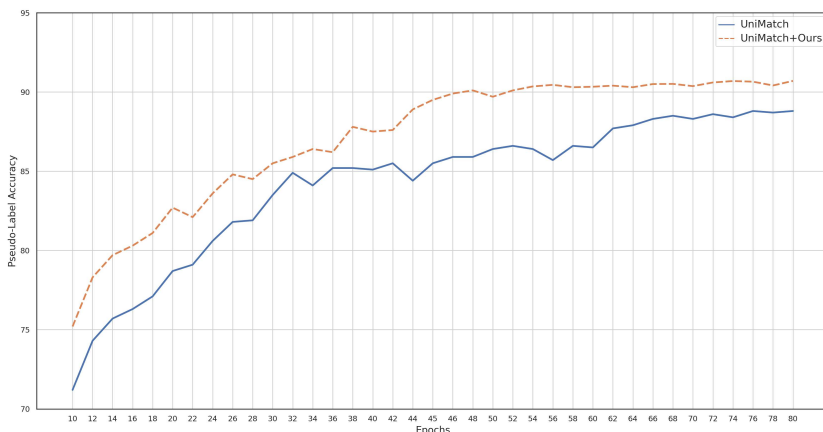
The performance of the Faster R-CNN is shown in the first row in Table 1, denoted as **Faster R-CNN**. We report the box-level accuracy for all detected bounding boxes with confidence scores  $\geq 0.85$ . A detected box is considered “correct” if its IoU  $\geq 0.8$  with a ground truth bounding box (discussed in Sec. 1) and “incorrect” otherwise. As can be seen, bounding boxes predicted by this Faster R-CNN model are fairly accurate: above 80% in all cases, even with only 92 training images.

More importantly, we show that this object detector improves the pseudo-label accuracy when used together with the segmentation network. We consider three group of pixels: 1) a baseline pseudo-labeling method that select pixels with high confidence scores from the segmentation model (denoted as “**Pixel**” in Table 1), 2) all pixels in the first group that are labeled as the same class by the object detector ( denoted as “**Pixel**  $\cap$  **BBOX**” in Table 1) and 3) all pixels in the first group that do not intersect with any detected bounding boxes of the same category (denoted as “**Pixel**  $\setminus$  **BBOX**” in Table 1). As can be seen, pixels that are labeled as the same classes by both models are more reliable than just using the segmentation model.

**Table 1:** Analysis of the performance of Faster R-CNN trained on limited labeled data and pseudo-label accuracy of pixels based on an ensemble of object detector and segmentation model on  $\frac{1}{16}$  Pascal (*Classic*) set. The first row reports the bounding box accuracy of Faster R-CNN. Pixel denotes the pseudo-labels by Deeplabv3+(ResNet-101) [6], BBox denotes bounding boxes generated by Faster R-CNN [39].

Category	Background	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	DiningTable	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	TV/Monitor
Faster R-CNN	-	83.7	89.6	80.4	83.9	87.5	82.3	90.1	92.6	80.1	92.3	80.4	94.7	93.9	80.7	93.6	81.3	82.5	80.8	83.1	84.7
Pixel	80.2	71.9	55.1	83.4	79.7	73.3	68.2	87.4	82.2	32.8	80.2	46.5	70.9	85.4	74.7	57.4	45.1	72.0	40.3	67.1	79.6
Pixel $\cap$ BBOX	-	<b>78.6</b>	<b>60.2</b>	<b>88.5</b>	<b>83.9</b>	<b>79.2</b>	<b>71.1</b>	<b>88.1</b>	<b>87.4</b>	<b>41.7</b>	<b>83.5</b>	<b>52.9</b>	<b>77.6</b>	<b>87.1</b>	<b>76.6</b>	<b>62.6</b>	<b>57.9</b>	<b>79.3</b>	<b>59.6</b>	<b>75.6</b>	<b>84.8</b>
Pixel $\setminus$ BBOX	-	54.1	43.3	61.2	60.1	71.4	37.2	75.6	60.1	6.8	73.1	13.3	53.0	74.5	56.0	36.2	12.7	56.8	11.1	21.4	70.5





**Fig. 1:** Pseudo-labeling accuracy in Pascal VOC unlabeled images

### 3 Analysis of Pseudo-label Accuracy

We analyze how our method, when integrated into UniMatch [57], can improve the pseudo-label accuracy of this method. We train a vanilla UniMatch model and a UniMatch model with our method with 1/16 data partition of the Pascal VOC dataset (*Classic*). As shown in Fig. 1, our method pseudo-label pixels with up to 90% accuracy and is consistently more accurate than UniMatch throughout the whole training process.

### 4 Analysis of Hyper-Parameters

In this section, we analyze the effects of different training hyper-parameters used by our method. We integrate our method into UniMatch [57] and train with 1/16 and 1/8 partition protocols of the Pascal VOC dataset (*Classic*)

#### 4.1 Analysis of the hyper-parameter $\alpha$ of semi-supervised segmentation loss

We analyze how our approach performs with different values of  $\alpha$ , which is used to balance the supervised loss and the unsupervised loss. The results of our approach are in Table 2.

#### 4.2 Analysis of Bounding Box Confidence Thresholds

We analyze how our method performs with different bounding box confidence thresholds. As shown in Table 3, the performance decreases with a very high confidence threshold. This is because the number of bounding boxes drastically reduces when increasing the threshold.

**Table 2:** Analysis of ( $\alpha$ ) of semi-supervised segmentation loss (1/16 and 1/8 partition protocols of Pascal VOC Dataset)

$\alpha$	1/16	1/8
0	45.1	55.3
0.2	75.1	78.0
0.4	<b>76.7</b>	<b>78.5</b>
0.6	76.3	78.1
0.8	75.5	77.8

**Table 3:** Analysis of BBox confidence threshold (1/16 and 1/8 partition protocols of Pascal VOC Dataset)

BBox Confidence	1/16	1/8
0.80	76.1	78.3
0.85	<b>76.7</b>	<b>78.5</b>
0.90	75.9	77.4
0.95	75.7	77.0

## 5 Memory Bank Implementation Details

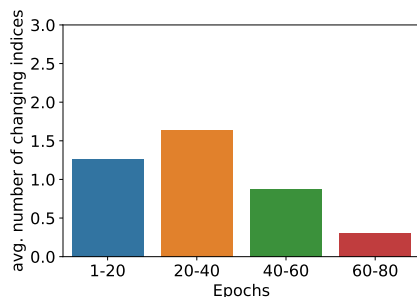
We use a memory bank to store features of labeled pixels and reliable pseudo-label pixels in each iteration, which is used to construct the per-class prototype. Since available memory is limited, only a random subset of features per class are included in the memory bank. Performing random sampling of the features to update the memory during training leads to a more diverse set of features per class. The memory follows First In First Out (FIFO) queue per class for computation and time efficiency [1]. This helps in maintaining recent high-quality feature vectors.

### 5.1 Effect of Memory Size

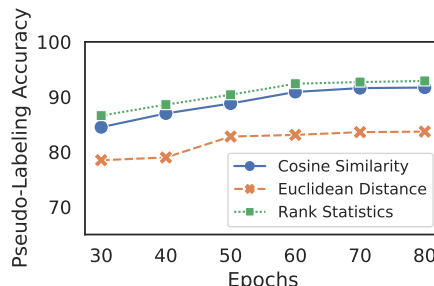
The effect of the memory bank size (per-class) is studied in Table 4. We observe that higher memory size leads to better performance, although from 256 the performance tends to stabilise. Since all elements from the memory bank are used during the prototype generation, the computation and memory complexity increases with a larger memory bank, we selected a size of 256 as a good tradeoff.

**Table 4:** Effect of our memory bank size (features per-class)  $\psi$ . (1/16 partition protocol of Pascal VOC Dataset)

$\psi$	32	64	128	256	512
mIoU	74.6	75.1	75.9	<b>76.7</b>	76.2



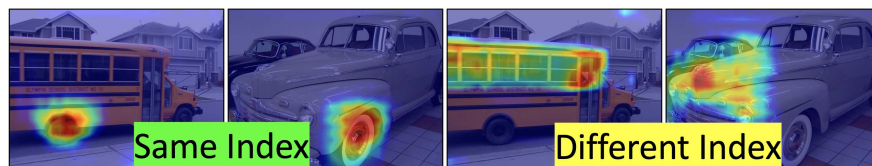
**Fig. 2:** Analysis of top-rank indices of class prototypes



**Fig. 3:** Pseudo-labeling accuracy with euclidean distance

## 6 Visualizing the top-ranked features

Top-ranked (TR) features often highlight discriminative object parts. In Fig. 4 we show gradcam visualization of a common TR feature between “car” and “bus” in the first two images and then show two different TR features, exclusive for each class, to illustrate the distinct regions each feature focuses on.



**Fig. 4:** Visualizing the top-ranked (TR) features

## 7 Adapting our method to Transformer-based models

We analyse how our method improves the transformer-based models. We experiment with SemiVL [22] on Cityscapes and show the results in the Table 5. We use the image embeddings prior to similarity map generation, to generate class prototypes and incorporate our per-pixel learning weight to the CLIP guidance loss.

## 8 Evaluation of performance on MS COCO

In Table 6, when using our method on UniMatch [57] (Xception-65) on MS COCO. We observe that performance improves across all data splits.

**Table 5:** Quantitative results of on the Cityscapes dataset

Method	Net	1/30	1/16	1/8	1/2
SemiVL [22]	Vit-B/16	76.2	77.9	79.4	80.6
SemiVL [22] + Ours	Vit-B/16	<b>77.5</b>	<b>79.1</b>	<b>80.4</b>	<b>81.4</b>

**Table 6:** Quantitative results of on the MS COCO dataset

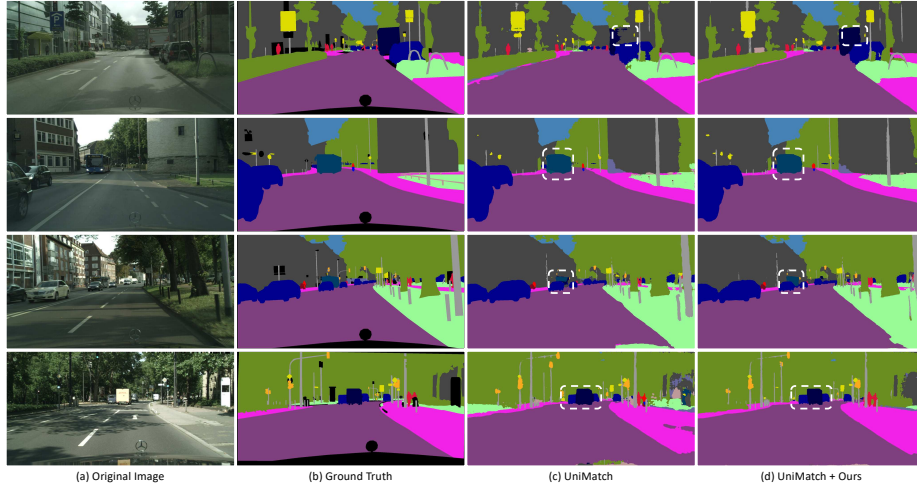
Method	1/512	1/256	1/128	1/64	1/32
UniMatch [57]	31.9	38.9	44.4	48.2	49.8
UniMatch [57] + Ours	<b>33.7</b>	<b>40.5</b>	<b>46.6</b>	<b>50.3</b>	<b>51.7</b>

## 9 Qualitative results

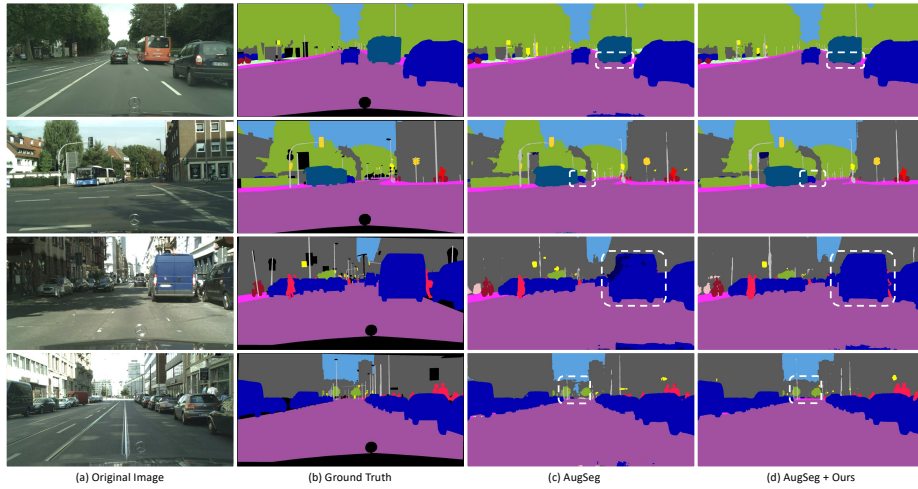
- **Detections:** In Fig. 5, we train a Faster R-CNN on  $\frac{1}{16}$  of Cityscapes labeled data and visualize the detection results (confidence  $\geq 0.9$ ) on Cityscapes unlabeled images. It can be observed that the detection boxes are relatively accurate when trained on limited labeled data.
- **Pascal VOC:** In Fig. 8, Fig. 9, Fig. 10 and Fig. 11 we compare our method integrated into UniMatch [57], AugSeg [65], U2PL [50] and AEL [23] respectively, with the corresponding baselines (UniMatch, AugSeg, U2PL and AEL). The visualization of the segmentation results indicate that our method improves the segmentation performance of all four baselines: UniMatch, AugSeg, U2PL and AEL.  
Note, all methods have been trained on  $\frac{1}{16}$  data partition of Pascal VOC dataset (*Classic*) and all visualizations are on Pascal VOC validation set.
- **Cityscapes:** In Fig. 6, Fig. 7, we compare our method integrated into UniMatch [57] and AugSeg [65] respectively, with the corresponding baselines (UniMatch and AugSeg). The visualization of the segmentation results indicate that our method improves the segmentation performance of both baselines: UniMatch and AugSeg.  
Note, all methods have been trained on  $\frac{1}{16}$  data partition of the Cityscapes dataset, and all visualizations are on the Cityscapes validation set.



**Fig. 5: Detection results on Cityscapes unlabeled images** Faster R-CNN model trained on 1/16 labeled data in Cityscapes dataset and confidence  $\geq 0.9$ . From left to right: Car, Person, Traffic Light, Bicycle.



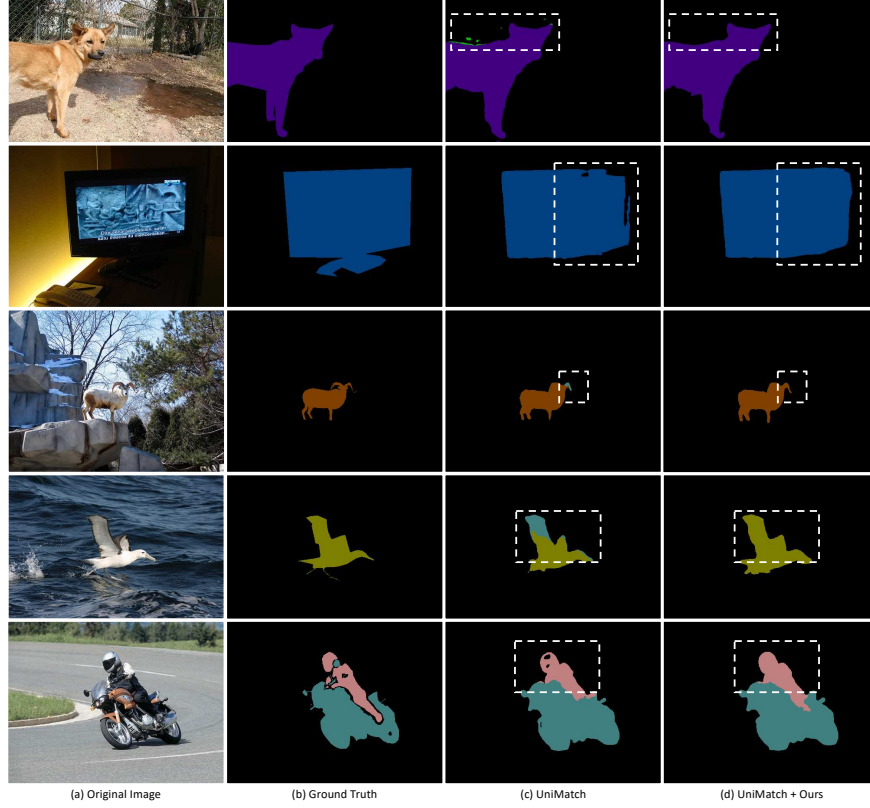
**Fig. 6: Qualitative Results on Cityscapes dataset:** (a) original image, (b) ground truth, (c) segmentations generated by UniMatch [57] compared to (d) which are segmentations generated when our method is integrated to UniMatch. The white boxes show the areas where our method improves the baseline [57].



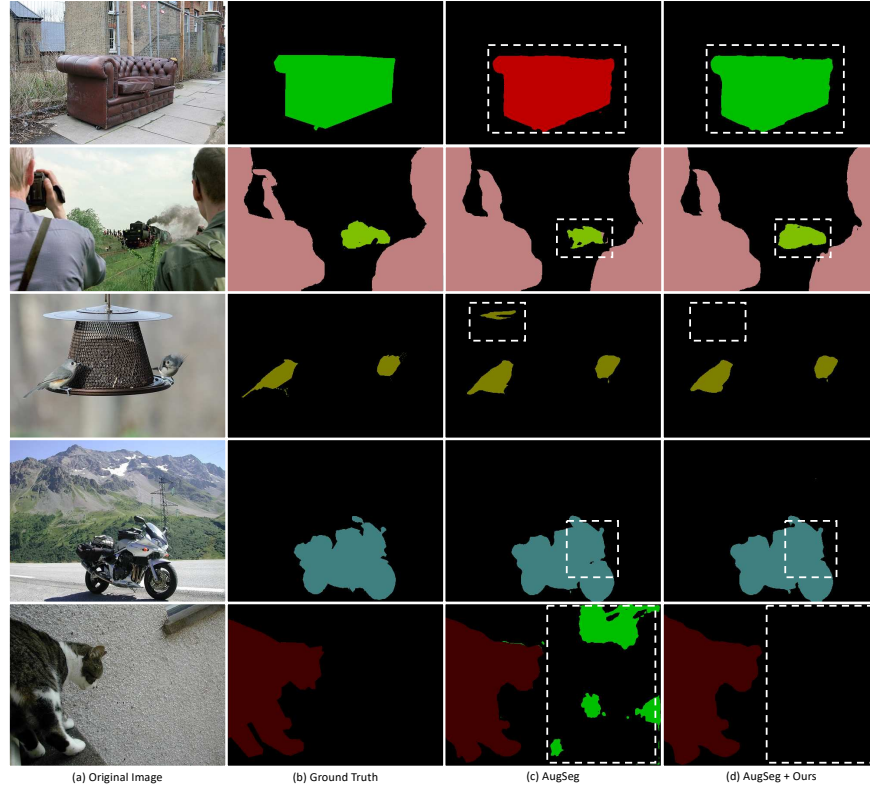
**Fig. 7: Qualitative Results on Cityscapes dataset:** (a) original image, (b) ground truth, (c) segmentations generated by AugSeg [65] compared to (d) which are segmentations generated when our method is integrated to AugSeg. The white boxes show the areas where our method improves the baseline [65].

## References

1. Alonso, I., Sabater, A., Ferstl, D., Montesano, L., Murillo, A.C.: Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In: *Int. Conf. Comput. Vis.* pp. 8219–8228 (2021)
2. Bachman, P., Alsharif, O., Precup, D.: Learning with pseudo-ensembles. *Advances in neural information processing systems* **27** (2014)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **39**(12), 2481–2495 (2017)
4. Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C.A.: Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems* **32** (2019)
5. Chen, H., Jin, Y., Jin, G., Zhu, C., Chen, E.: Semisupervised semantic segmentation by improving prediction confidence. *IEEE Transactions on Neural Networks and Learning Systems* (2021)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: *Eur. Conf. Comput. Vis.* pp. 801–818 (2018)
7. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 2613–2622 (2021)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 3213–3223 (2016)
9. Pinto da Costa, J., Soares, C.: A weighted rank measure of correlation. *Australian & New Zealand Journal of Statistics* **47**(4), 515–529 (2005)

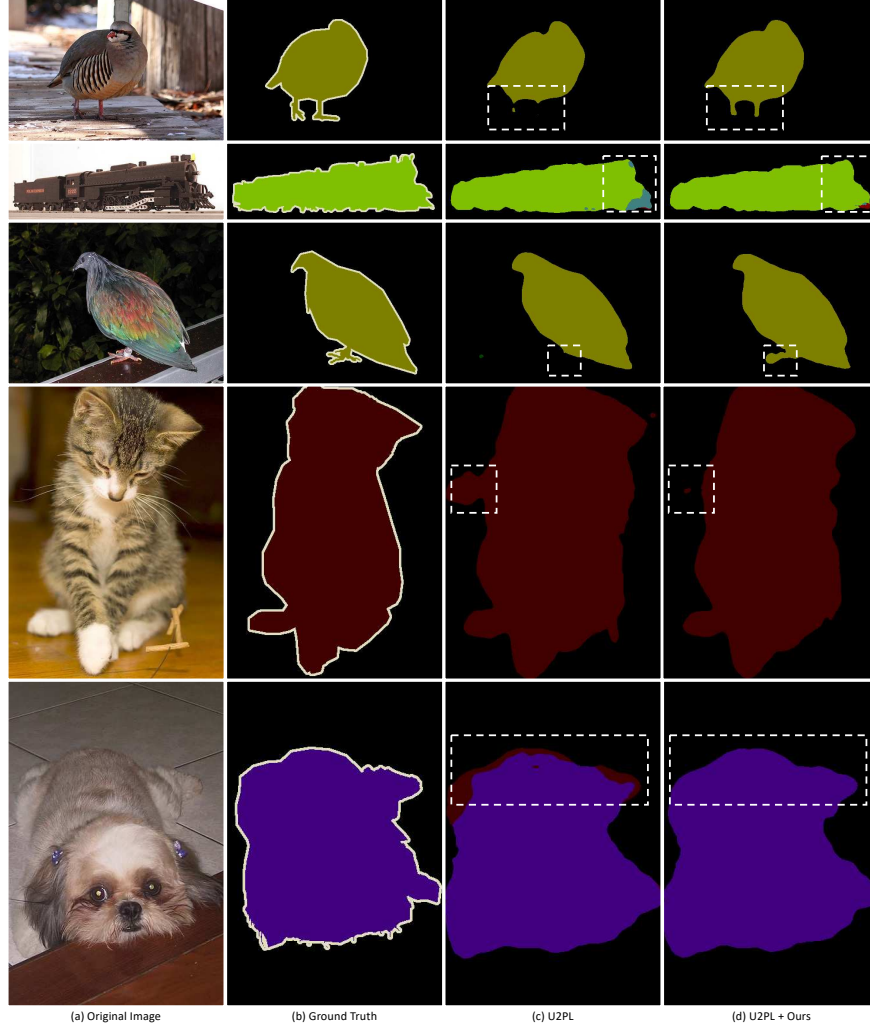


**Fig. 8: Qualitative Results on Pascal dataset:** (a) original image, (b) ground truth, (c) segmentations generated by UniMatch [57] compared to (d) which are segmentations generated when our method is integrated to UniMatch. The white boxes show the areas where our method improves the baseline [57].

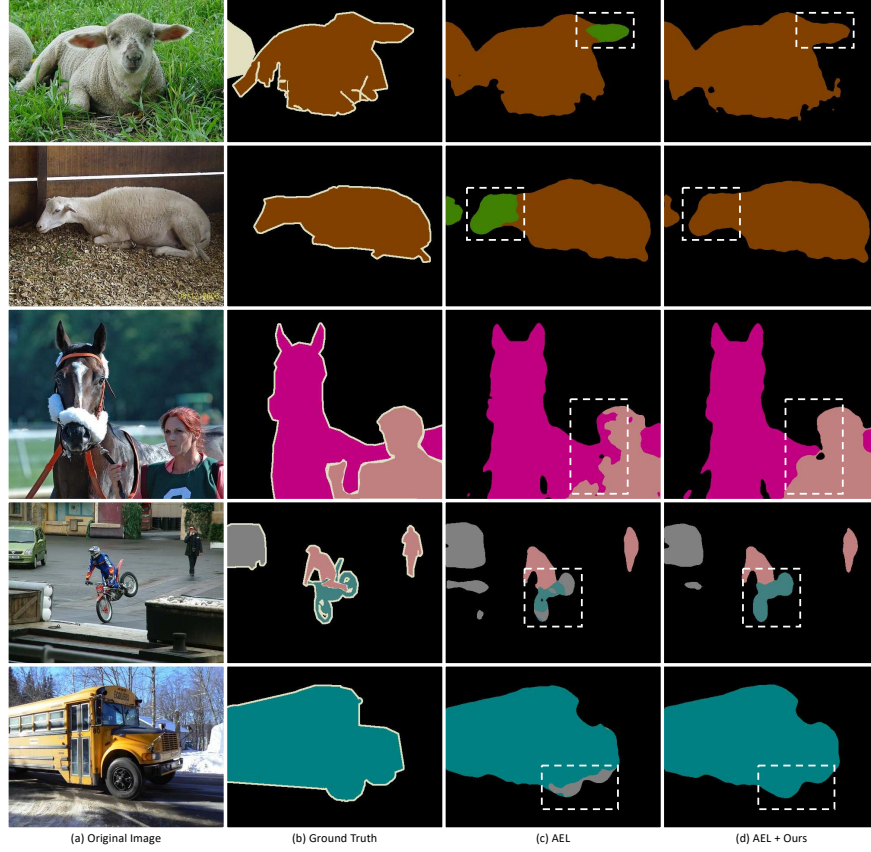


**Fig. 9: Qualitative Results on Pascal dataset:** (a) original image, (b) ground truth, (c) segmentations generated by AugSeg [65] compared to (d) which are segmentations generated when our method is integrated to AugSeg. The white boxes show the areas where our method improves the baseline [65].





**Fig. 10: Qualitative Results on Pascal dataset:** (a) original image, (b) ground truth, (c) segmentations generated by U2PL [50] compared to (d) which are segmentations generated when our method is integrated to U2PL. The white boxes show the areas where our method improves the baseline [50].



**Fig. 11: Qualitative Results on Pascal dataset:** (a) original image, (b) ground truth, (c) segmentations generated by AEL [23] compared to (d) which are segmentations generated when our method is integrated to AEL. The white boxes show the areas where our method improves the baseline [23].

10. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
11. Durasov, N., Dorndorf, N., Le, H., Fua, P.: Zigzag: Universal sampling-free uncertainty estimation through two-step inference. *Transactions on Machine Learning Research* (2024)
12. Durasov, N., Oner, D., Donier, J., Le, H., Fua, P.: Enabling uncertainty estimation in iterative neural networks. In: *Forty-first International Conference on Machine Learning* (2024)
13. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. *International journal of computer vision* **88**(2), 303–338 (2010)
14. Fan, J., Gao, B., Jin, H., Jiang, L.: Ucc: Uncertainty guided cross-head co-training for semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 9947–9956 (2022)
15. French, G., Laine, S., Aila, T., Mackiewicz, M., Finlayson, G.: Semi-supervised semantic segmentation needs strong, varied perturbations. arXiv preprint arXiv:1906.01916 (2019)
16. Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. *Advances in neural information processing systems* **17** (2004)
17. Guan, D., Huang, J., Xiao, A., Lu, S.: Unbiased subclass regularization for semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 9968–9978 (2022)
18. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: *International conference on machine learning*. pp. 1321–1330. PMLR (2017)
19. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. arXiv preprint arXiv:2002.05714 (2020)
20. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: *Int. Conf. Comput. Vis.* pp. 991–998. IEEE (2011)
21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 770–778 (2016)
22. Hoyer, L., Tan, D.J., Naeem, M.F., Van Gool, L., Tombari, F.: Semivl: Semi-supervised semantic segmentation with vision-language guidance. arXiv preprint arXiv:2311.16241 (2023)
23. Hu, H., Wei, F., Hu, H., Ye, Q., Cui, J., Wang, L.: Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems* **34**, 22106–22118 (2021)
24. Ibrahim, M.S., Vahdat, A., Ranjbar, M., Macready, W.G.: Semi-supervised semantic image segmentation with self-correcting networks. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 12715–12725 (2020)
25. Kattenborn, T., Eichel, J., Fassnacht, F.E.: Convolutional neural networks enable efficient, accurate and fine-grained segmentation of plant species and communities from high-resolution uav imagery. *Scientific reports* **9**(1), 1–9 (2019)
26. Kundu, A., Li, Y., Dellaert, F., Li, F., Rehg, J.M.: Joint semantic segmentation and 3d reconstruction from monocular video. In: *Eur. Conf. Comput. Vis.* pp. 703–718. Springer (2014)
27. Le, H., Goncalves, B., Samaras, D., Lynch, H.: Weakly labeling the antarctic: The penguin colony case. In: *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.* (June 2019)

28. Le, H., Samaras, D., Lynch, H.J.: A convolutional neural network architecture designed for the automated survey of seabird colonies. *Remote Sensing in Ecology and Conservation* **8**(2), 251–262 (2022)
29. Le, H., Vicente, T.F.Y., Nguyen, V., Hoai, M., Samaras, D.: A+D Net: Training a shadow detector with adversarial shadow attenuation. In: *European Conference on Computer Vision (ECCV)* (2018)
30. Le, H., Yu, C.P., Zelinsky, G., Samaras, D.: Co-localization with category-consistent features and geodesic distance propagation. In: *Int. Conf. Comput. Vis. Worksh.* (2017)
31. Lee, D.H., et al.: Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: *Workshop on challenges in representation learning, ICML*. vol. 3, p. 896 (2013)
32. Li, S., He, Y., Zhang, W., Zhang, W., Tan, X., Han, J., Ding, E., Wang, J.: Cfcg: Semi-supervised semantic segmentation via cross-fusion and contour guidance supervision. In: *Int. Conf. Comput. Vis.* pp. 16348–16358 (2023)
33. Liu, L., Tan, R.T.: Certainty driven consistency loss on multi-teacher networks for semi-supervised learning. *Pattern Recognition* **120**, 108140 (2021)
34. Liu, Y., Tian, Y., Chen, Y., Liu, F., Belagiannis, V., Carneiro, G.: Perturbed and strict mean teachers for semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 4258–4267 (2022)
35. Maturi, T.A., Abdelfattah, E.H.: A new weighted rank correlation. *Journal of mathematics and statistics*. **4**(4), 226–230 (2008)
36. McClosky, D., Charniak, E., Johnson, M.: Effective self-training for parsing. In: *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*. pp. 152–159 (2006)
37. Olsson, V., Tranheden, W., Pinto, J., Svensson, L.: Classmix: Segmentation-based data augmentation for semi-supervised learning. In: *Winter Conference on Applications of Computer Vision*. pp. 1369–1378 (2021)
38. Ouali, Y., Hudelot, C., Tami, M.: Semi-supervised semantic segmentation with cross-consistency training. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 12674–12684 (2020)
39. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* **28** (2015)
40. Rizve, M.N., Duarte, K., Rawat, Y.S., Shah, M.: In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329* (2021)
41. Sajjadi, M., Javanmardi, M., Tasdizen, T.: Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems* **29** (2016)
42. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. pp. 618–626 (2017)
43. Shi, W., Gong, Y., Ding, C., Tao, Z.M., Zheng, N.: Transductive semi-supervised deep learning using min-max features. In: *Eur. Conf. Comput. Vis.* pp. 299–315 (2018)
44. Shieh, G.S.: A weighted kendall’s tau statistic. *Statistics & probability letters* **39**(1), 17–24 (1998)

45. Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C.A., Cubuk, E.D., Kurakin, A., Li, C.L.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems* **33**, 596–608 (2020)
46. Sun, R., Mai, H., Zhang, T., Wu, F.: DAW: Exploring the better weighting function for semi-supervised semantic segmentation. In: *Thirty-seventh Conference on Neural Information Processing Systems (2023)*, <https://openreview.net/forum?id=KR1G7NJUCD>
47. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems* **30** (2017)
48. Umar, M., Babu Saheer, L., Zarrin, J.: Forest terrain identification using semantic segmentation on uav images (2021)
49. Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y., Lopez-Paz, D.: Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825* (2019)
50. Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X.: Semi-supervised semantic segmentation using unreliable pseudo-labels. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 4248–4257 (2022)
51. Xu, H.M., Liu, L., Bian, Q., Yang, Z.: Semi-supervised semantic segmentation with prototype-based consistency regularization. *arXiv preprint arXiv:2210.04388* (2022)
52. Xu, J., Le, H.: Generating representative samples for few-shot classification. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022)
53. Xu, J., Le, H., Huang, M., Athar, S., Samaras, D.: Variational feature disentangling for fine-grained few-shot classification. In: *Int. Conf. Comput. Vis.* (2021)
54. Xu, J., Le, H., Samaras, D.: Generating features with increased crop-related diversity for few-shot object detection. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2023)
55. Xu, J., Le, H.M., Nguyen, V., Ranjan, V., Samaras, D.: Zero-shot object counting. *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 15548–15557 (2023)
56. Yagnik, J., Strelow, D.W., Ross, D.A., Lin, R.S.: The power of comparative reasoning. *Int. Conf. Comput. Vis.* pp. 2431–2438 (2011)
57. Yang, L., Qi, L., Feng, L., Zhang, W., Shi, Y.: Revisiting weak-to-strong consistency in semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 7236–7246 (2023)
58. Yang, L., Zhuo, W., Qi, L., Shi, Y., Gao, Y.: St++: Make self-training work better for semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 4268–4277 (2022)
59. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: *33rd annual meeting of the association for computational linguistics.* pp. 189–196 (1995)
60. Yuan, J., Liu, Y., Shen, C., Wang, Z., Li, H.: A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In: *Int. Conf. Comput. Vis.* pp. 8229–8238 (2021)
61. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: *Int. Conf. Comput. Vis.* pp. 6023–6032 (2019)
62. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017)

63. Zhang, M., Shi, M., Li, L.: Mfnet: Multi-class few-shot segmentation network with pixel-wise metric learning. *IEEE Transactions on Circuits and Systems for Video Technology* (2022)
64. Zhao, B., Han, K.: Novel visual category discovery with dual ranking statistics and mutual knowledge distillation. *Advances in Neural Information Processing Systems* **34**, 22982–22994 (2021)
65. Zhao, Z., Yang, L., Long, S., Pi, J., Zhou, L., Wang, J.: Augmentation matters: A simple-yet-effective approach to semi-supervised semantic segmentation. In: *IEEE Conf. Comput. Vis. Pattern Recog.* pp. 11350–11359 (2023)
66. Zhong, Y., Yuan, B., Wu, H., Yuan, Z., Peng, J., Wang, Y.X.: Pixel contrastive-consistent semi-supervised semantic segmentation. In: *Int. Conf. Comput. Vis.* pp. 7273–7282 (2021)
67. Zhou, T., Wang, S., Bilmes, J.: Time-consistent self-supervision for semi-supervised learning. In: *International Conference on Machine Learning*. pp. 11523–11533. PMLR (2020)
68. Zhu, Y., Zhang, Z., Wu, C., Zhang, Z., He, T., Zhang, H., Manmatha, R., Li, M., Smola, A.: Improving semantic segmentation via self-training. *arXiv preprint arXiv:2004.14960* (2020)
69. Zou, Y., Zhang, Z., Zhang, H., Li, C.L., Bian, X., Huang, J.B., Pfister, T.: Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713* (2020)
70. Zuo, S., Yu, Y., Liang, C., Jiang, H., Er, S., Zhang, C., Zhao, T., Zha, H.: Self-training with differentiable teacher. *arXiv preprint arXiv:2109.07049* (2021)