This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Space Engage: Collaborative Space Supervision for Contrastive-based Semi-Supervised Semantic Segmentation

Changqi Wang[1]*, Haoyu Xie[1]*, Yuhui Yuan[2], Chong Fu[1,4]†, Xiangyu Yue[3]

[1] School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2] Microsoft Research Asia [3] The Chinese University of Hong Kong
[4] Key Laboratory of Intelligent Computing in Medical Image, Ministry of Education, NEU, China

2101668@stu.neu.edu.cn, xiehaoyu@stumail.neu.edu.cn, yuyua@microsoft.com
fuchong@mail.neu.edu.cn, xyyue@ie.cuhk.edu.hk

Abstract

Semi-Supervised Semantic Segmentation (S4) aims to train a segmentation model with limited labeled images and a substantial volume of unlabeled images. To improve the robustness of representations, powerful methods introduce a pixel-wise contrastive learning approach in latent space (i.e., representation space) that aggregates the representations to their prototypes in a fully supervised manner. However, previous contrastive-based S4 methods merely rely on the supervision from the model's output (logits) in logit space during unlabeled training. In contrast, we utilize the outputs in both logit space and representation space to obtain supervision in a collaborative way. The supervision from two spaces plays two roles: 1) reduces the risk of over-fitting to incorrect semantic information in logits with the help of representations; 2) enhances the knowledge exchange between the two spaces. Furthermore, unlike previous approaches, we use the similarity between representations and prototypes as a new indicator to tilt training those under-performing representations and achieve a more efficient contrastive learning process. Results on two public benchmarks demonstrate the competitive performance of our method compared with state-of-the-art methods.

1. Introduction

Semantic segmentation is a fundamental task in computer vision, aiming to classify each pixel in an image. Significant progress [22, 4] has been made in training on high-quality labeled images using segmentation models composed of an encoder and a segmentation head. However, annotating images is expensive and time-consuming. Semi-supervised Semantic Segmentation (S4) alleviates the thirst for annotation by leveraging unlabeled images to train segmentation models.

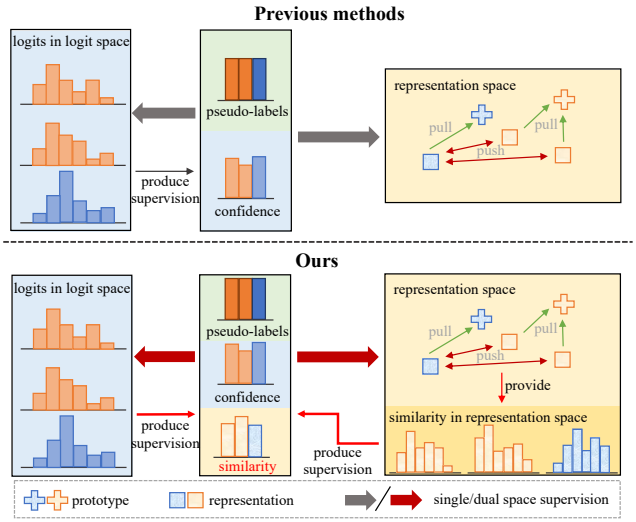


Figure 1. We enhance the knowledge exchange between the logit and representation spaces. Orange and blue represent different classes. Top: Existing contrastive-based S4 methods overlook the semantic information in representation space. Bottom: Our method uses dual-space collaborative supervision.

Most existing works learn from unlabeled images via self-training [44, 37, 15] or consistency regularization [40, 38, 13] strategies, both of which retrain the model with its predictions on unlabeled images. Recently, great success has been achieved by introducing pixel-wise contrastive learning to semantic segmentation, which endows the model with a stronger features-extracting ability by accessing a more discriminative representation space. Specially, these methods [45, 23] project each pixel to repre-

*equal contribution
†corresponding author

sentation space as a representation and regularize it in a fully supervised manner, *i.e.*, aggregating the representations with the same class and separating them with different classes. In semi-supervised settings, due to limited labels, most methods [1, 32, 46] obtain supervision from the model's output logits in logit space during the unlabeled training process. However, recent contrastive-based semantic segmentation methods [1, 32, 46, 45] mainly focus on the learning process in logit space while only taking that in representation space as an auxiliary task. The unidirectional supervision makes training dominated by the predicted logits, leading to the neglect of information in the representation space. We argue that this kind of single-space supervision may incorrectly provide semantic guidance to representation learning and fails to facilitate knowledge exchange between the two spaces (see Sec. 5.1).

In this work, we extend the single-space supervision to a dual-space supervision for contrastive-based S4 and propose Collaborative Space Supervision (CSS). Our key insight is to: **i)** utilize the semantic information in representations to obtain more reliable guidance during unlabeled training, and to enhance knowledge exchange between two spaces; **ii)** provide a more accurate reference for the model's performance on predicting each representation to tilt training those under-performing representations. To achieve objective **i)**, we obtain dense semantic predictions by retrieving the nearest class prototype for each representation in the representation space and then engage with predictions from the logits for collaborative supervision of the model. For objective **ii)**, we measure the similarity between the representations and prototypes and use the similarity after a normalization operation as the indicator for guiding the learning process in the representation space. Unlike previous works that utilize confidence as the indicator to involve representation learning, the similarity directly reflects the confusion level between representations and prototypes, resulting in more efficient representation learning.

To summarize, our main contributions are three-fold: **1)** The dual-space collaboration for contrastive-based S4, enhances the knowledge exchange between the logit and representation spaces. **2)** Utilizing similarity to provide a more accurate reference for the model's performance in representation learning. **3)** Extensive experiments on two S4 benchmarks demonstrate the effectiveness of our method.

2. Related Works

2.1. Semi-supervised Semantic Segmentation

The aim of S4 is to train a segmentation model with the semi-supervised setting (*i.e.*, a few labeled images and a large number of unlabeled images) to classify each pixel in an entire image. The critical issue of S4 is how to leverage unlabeled images to train the model. Some methods [25, 27, 30, 35] based on GANs [16], adversarial training [36], and consistency regularization paradigm [38, 13, 40, 7, 56]. Meanwhile, self-training [28, 44, 49, 61, 53] is also a striking paradigm, which always generates pseudo-labels from model and retrains the model with the combined supervision of human annotations and pseudo-labels. One essential issue of self-training is the accuracy of pseudo-labels. Some methods [29, 33, 14, 42, 52] try to polish pseudo-labels and provide reliable guidance. Some methods [21, 47, 24, 17] focus on the class-imbalance problems in the dataset and try to alleviate the negative effect from class-biased pseudo-labels generated by the model pre-trained on imbalanced labeled images. We build our framework based on the self-training and additionally explore semantic information among different images.

2.2. Pixel-wise Contrastive Learning

Pixel-wise contrastive learning explores semantic relations not only in the individual image but also among different images. Different from instance-wise contrastive learning [19, 6, 3], pixel-wise contrastive learning [50, 55, 5, 58] project each pixel to the representation in representation space with the cooperation of encoder and representation head. Representations are then aggregated in their prototypes and are separated from each other in different classes. In semi-supervised settings, most methods [1, 32, 46, 48] use pseudo-labels based on logits to provide semantic information contrastive learning process during training on unlabeled images. Meanwhile, the confidence of logit is used as an indicator to involve the contrastive learning process, *e.g.*, [32] uses the hard representations whose corresponding logit confidence is lower than a threshold to contrast for effective training. As opposed to the above methods, we use collaborative space supervision for contrastive learning on unlabeled images and use a new indicator to involve the contrastive learning progress.

2.3. Prototype-based Learning

Prototype-based learning has been widely studied in few-shot learning [41, 11, 31, 34] and unsupervised domain adaption [54, 26, 43, 39, 57]. Recently, it is restudied in semantic segmentation as known as a non-parametric prototype-based classifier [60]. Concretely, the classes in the dataset are presented by a set of non-learnable prototypes, and the dense semantic predictions are thus achieved by assigning the output features to its most similar prototype. Under semi-supervised settings, some methods [51] maintain the consistency between predictions from a linear predictor and a prototype-based predictor. The two predictors are followed by the encoder and project the features to logit space and representation space, respectively. In this work, we combine the semantic information in the logit and representation spaces to provide supervision in a collabora-

tive way during semi-supervised learning.

3. Methodology

In the S4 task, we have a small labeled set $\mathcal{D}_l = \{(\boldsymbol{x}_i^l, \boldsymbol{y}_i^l)\}_{i=1}^{N_l}$ and a large unlabeled set $\mathcal{D}_u = \{\boldsymbol{x}_i^u\}_{i=1}^{N_u}$, where $\boldsymbol{x}_i^l, \boldsymbol{x}_i^u \in \mathbb{R}^{H\times W\times 3}$, H, W denote the height, and the width, respectively. And ground truth $\boldsymbol{y}_i^l \in \{0,1\}^{H\times W\times|C|}$ with the set of class C. The goal is to boost model performance with $\mathcal{D}_u$. The base model consists of an encoder $f(\cdot)$ and a segmentation head $g(\cdot)$, which projects features to the logit space $\mathbb{R}^{H\times W\times|C|}$. We adopt Self-Training (ST) and pixel-wise contrastive learning to our framework, as described in Sec. 3.1. The supervision for $\mathcal{D}_u$ is produced by the collaboration between the logit and representation spaces, as described in Sec. 3.3.

3.1. ST and Pixel-wise Contrastive Learning

The main idea of self-training is to pre-train a model on labeled images and use it to produce pseudo-labels as supervision for unlabeled images. A typical framework is the teacher-student framework [44], which consists of a student model and a teacher model. Both the student model and the teacher model are constructed by an encoder and a segmentation head. Parameters of the student model are optimized via Stochastic Gradient Decent (SGD) while parameters of the teacher model are updated by the Exponential Moving Average (EMA) of student model parameters. We denote the encoder and the segmentation head in the student model by $f(\cdot)$ and $g(\cdot)$ while denoting those in the teacher model by $f'(\cdot)$ and $g'(\cdot)$. The pseudo-labels $\hat{\boldsymbol{y}}_i^{u,lgt}$ are produced based on the teacher model's output logits $\hat{\boldsymbol{p}}_i^u = g'(f'(\boldsymbol{x}_i^u))$ in logit space, formulated as:

$$\hat{\boldsymbol{y}}_i^{u,lgt} = \mathbf{1}_c(\arg\max_c\{\hat{\boldsymbol{p}}_{i,c}^u\}_{c\in C}), \tag{1}$$

where $\mathbf{1}_c(\cdot)$ denotes the one-hot encoding of class c.

In order to enhance the ability of the model itself to extract features, recent works [32, 46, 1] additionally employ pixel-wise contrastive learning and introduce a representation head to both teacher and student models. We denote the representation head in the student model as $h(\cdot)$ and that in the teacher model as $h'(\cdot)$. The pixel $\boldsymbol{x}_i$ of the class c are projected as representations $\boldsymbol{z}_{ci}$ in representation space by the cooperating of $f(\cdot)$ and $h(\cdot)$, *i.e.*, $\boldsymbol{z}_{ci} = h(f(\boldsymbol{x}_i))$. And the representation $\boldsymbol{z}_{ci}$ is then aggregated to its class centroid (prototype) while separated from representations in different classes $\boldsymbol{z}_{\tilde{c}i}$ (negatives). The semantic guidance for contrastive learning is from the combination of ground truth $\boldsymbol{y}_i^l$ and pseudo-labels $\hat{\boldsymbol{y}}_i^{u,lgt}$ in logit space. Moreover, in order to emphasize the reliable and crucial aspects during unlabeled and contrastive learning, a sampling strategy is adopted to select valid pixels $\boldsymbol{x}_i$ according to their corresponding confidence, *i.e.* the student model's output logits $\boldsymbol{p}_i$ after a Softmax operation.

Discussion. In recent works [46, 1, 32], the supervision of unlabeled images is derived solely from the logit space. This overlooks the potential benefits of the supervision from the representation space, leading to *two potential limitations*: **1)** the pseudo-labels $\hat{\boldsymbol{y}}_i^{u,lgt}$ obtained from the logit space may contain noise and miss the opportunity to be corrected by semantic information from the representation space; **2)** since the confidence from logit space is used as the indicator $\hat{j}_i$ for the sampling strategy, learning in the representation space may not be critical enough due to the different confusing parts between the two spaces.

To mitigate these limitations, we produce pseudo-labels from the representation space and combine them with pseudo-labels from the logit space to provide higher-quality supervision during unlabeled training. Meanwhile, we obtain a new indicator from the representation space for the more effective sampling strategy.

3.2. Supervision from Representation Space

In this section, we detail the approach to obtain the pseudo-labels from the representation space. Meanwhile, we simultaneously access a new indicator for the sampling strategy in representation spaces, which provides a critical reference in the contrastive learning process.

Specifically, we first build a set of prototypes for each class and obtain the pseudo-labels by retrieving the nearest prototype for each representation. We calculate the centroid of all representations in the current class c as the prototype $\boldsymbol{\rho}_c$, which is formulated as:

$$\boldsymbol{\rho}_c = \frac{1}{N_c}\sum_i^{N_c} \boldsymbol{z}_i', \tag{2}$$

where N_c is the total number of representations of current class c and $\boldsymbol{z}_i'$ is the representation projected by the cooperation of the $f'(\cdot)$ and $h'(\cdot)$. Meanwhile, to include more representation information, we update the prototype along the sequential iterations with EMA as follows:

$$\hat{\boldsymbol{\rho}}_c(t) = \alpha\hat{\boldsymbol{\rho}}_c(t-1) + (1-\alpha)\boldsymbol{\rho}_c(t), \tag{3}$$

where $\hat{\boldsymbol{\rho}}_c(t)$, $\hat{\boldsymbol{\rho}}_c(t-1)$ mean the current t^{th} prototype and last $(t-1)^{th}$ prototype in iterations , $\boldsymbol{\rho}_c(t)$ means the prototype calculated by Eq. 2 in current iteration, and α is a hyper-parameter that controls the updating speed. Thus, the pseudo-label from the representation space is achieved by:

$$\hat{\boldsymbol{y}}_i^{u,rep} = \mathbf{1}_{\hat{c}}(\hat{c}), with\ \hat{c} = \arg\max_c\{sim(\boldsymbol{z}_i', \hat{\boldsymbol{\rho}}_c(t))\}_{c\in C}, \tag{4}$$

where $sim(\cdot)$ is defined as the cosine similarity.

As for the indicator for the sampling strategy in the representation space, we use the Softmax function on the similarity among the representation and all prototypes, which is

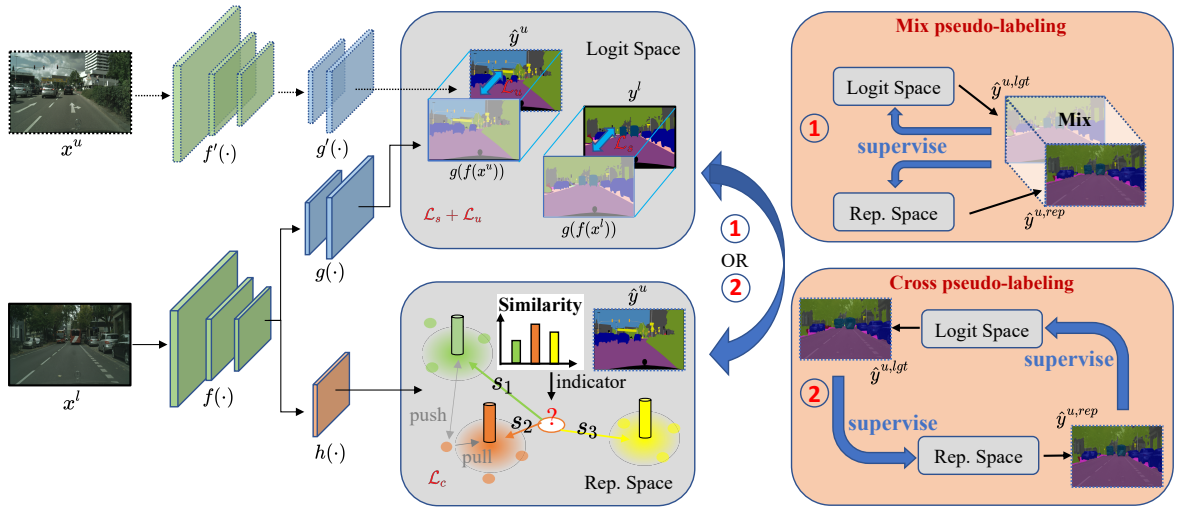


Figure 2. Overview of our framework. Our training pipeline consists of learning in two spaces: logit space and representation space. The pseudo-labels $\hat{\boldsymbol{y}}_i^u$ during unlabeled training are produced by the collaboration of two spaces with mix pseudo-labeling strategy (1) or cross pseudo-labeling strategy (2). The indicator for representation learning is produced by similarity (s_1, s_2, and s_3).

followed as:

$$\hat{j}_i^{u,rep} = \frac{e^{sim(\boldsymbol{z_{ci}}, \hat{\boldsymbol{\rho}}_c(t))/\tau}}{e^{sim(\boldsymbol{z_{ci}}, \hat{\boldsymbol{\rho}}_c(t))/\tau} + \sum_{\tilde{c}\in\tilde{C}} e^{sim(\boldsymbol{z_{ci}}, \hat{\boldsymbol{\rho}}_{\tilde{c}}(t))/\tau}}, \quad (5)$$

where $\hat{\boldsymbol{\rho}}_{\tilde{c}}(t)$ means the prototype with different classes from $\boldsymbol{z}_{ci}$ and τ is a hyper-parameter. Different from using confidence from logit space as the indicator to involve representation learning [32, 46], the Softmax similarity directly helps the model to discover the confusion between representations and their prototypes and focus on them during the subsequent training.

3.3. Collaboration Between Two Spaces

With the pseudo-labels in two spaces, we propose two pseudo-labeling strategies to strengthen the collaboration between two spaces and obtain more reliable pseudo-labels.

- **Mix pseudo-labeling.** To mitigate the negative effects of inherent noise from both two spaces during pseudo-labeling, we adopt the mix pseudo-labeling strategy that only considers the mutually agreeable pseudo-labels between the two spaces. Specifically, we define the set of final pseudo-labels as $\hat{Y}^u = \hat{Y}^{u,lgt} \cap \hat{Y}^{u,rep}$, where $\hat{\boldsymbol{y}}_i^{u,lgt} \in \hat{Y}^{u,lgt}$ and $\hat{\boldsymbol{y}}_i^{u,rep} \in \hat{Y}^{u,rep}$.

- **Cross pseudo-labeling.** Inspired by recent researches [7, 38] that maintain consistency among the predictions of the same image across different models or decoders in different views, we propose a cross pseudo-labeling strategy that leverages pseudo-labels from one space to supervise the other. Specifically, we use pseudo-labels $\hat{\boldsymbol{y}}_i^{u,rep}$ to supervise the logit space, and vice versa.

The strengths of using pseudo-labels from two spaces in the collaborative way are twofold: **1)** obtaining more reliable supervision during unlabeled training, and **2)** enabling the strengths of learning in different spaces to complement each other. Since the learning in different feature spaces concentrates on different parts of features, *i.e.*, the logit space mainly focuses on the most discriminative part of features while the representation space treats all parts of features equally, the performance of pseudo-labels from two spaces varies across different classes or regions of images. Therefore, our collaborative pseudo-labeling strategies exchange knowledge between two spaces and provide higher-quality supervision during unlabeled training. The experimental proof is in Sec. 5.1.

As for indicators, we use confidence as the indicator $\hat{j}_i^{u,lgt}$ for learning logit space and Softmax similarity as the indicator $\hat{j}_i^{u,rep}$ for learning representation space. We argue that the confusing parts of learning in both two spaces are varied due to the different parts of features being concentrated in each space. Therefore, this strategy allows the learning in different spaces to focus on their own confusing parts, which can be more effective than using a single indicator when mining confusing parts for both spaces in the training process. The experimental proof is in Sec. 5.2.

3.4. Training Objective

With the indicators $\hat{j}_i^{u,lgt}$ and $\hat{j}_i^{u,rep}$, we adopt some threshold sampling strategies. In logit space, we set a threshold δ_u during unlabeled learning and logits $\hat{\boldsymbol{p}}_i^u$ whose indicator $\hat{j}_i^{u,lgt}$ is higher than δ_u will be regarded as the valid logits in logit space. In representation space, our sample strategy can be divided into three parts: **1)** *Valid Sam-*

pling Strategy. Similar to the sampling strategy in logit space, a threshold δ_w is used to sample representations whose indicator $\hat{j}_i^{u,rep}$ is higher than δ_w. **2)** *Hard Sampling Strategy*. We adopt the hard sampling strategy for tilting to train those confusing representations. Specifically, we set a threshold δ_s to sample representations whose indicator $\hat{j}_i^{u,rep}$ is lower than δ_s. **3)** *Negative Sampling Strategy*. We sample negatives according to the similarity between the prototype of current class c and other prototypes. Concretely, the negatives are more likely to be sampled if its prototype is more similar to the prototype of the current class.

Cooperated with the ground truth $\boldsymbol{y}_i^l$, pseudo-labels $\boldsymbol{y}_i^u$ produced from two spaces in a collaborative way, and different sampling strategies in two spaces, the total learning object is composed with a supervised loss $\mathcal{L}_s$, an unsupervised loss $\mathcal{L}_u$, and a contrastive loss $\mathcal{L}_c$ as follows:

$$\mathcal{L} = \mathcal{L}_s + \mathcal{L}_u + \lambda_c \mathcal{L}_c, \tag{6}$$

where λ_c is used to tune the contribution between logit space and representation space. Specifically, $\mathcal{L}_s$ and $\mathcal{L}_u$ are constructed by the Cross-Entropy (CE) ℓ_{ce} and can be formulated as:

$$\mathcal{L}_s = \frac{1}{|\mathcal{B}_l|} \sum_{\boldsymbol{x}_i^l \in \mathcal{B}_l} \ell_{ce}(\boldsymbol{p}_i^l, \boldsymbol{y}_i^l), \tag{7}$$

$$\mathcal{L}_u = \frac{1}{|\hat{\mathcal{B}}_u|} \sum_{\boldsymbol{x}_i^u \in \hat{\mathcal{B}}_u} \ell_{ce}(\boldsymbol{p}_i^u, \hat{\boldsymbol{y}}_i^u), \tag{8}$$

where $\mathcal{B}_l$ denotes the labeled images in a mini-batch and $\hat{\mathcal{B}}_u$ is the subsets that sampled from unlabeled images in a mini-batch according to the sampling strategy. Meanwhile, the contrastive loss $\mathcal{L}_c$ is formulated as:

$$\mathcal{L}_c = -\frac{1}{|C| \times |\hat{\mathcal{Z}}_c|} \sum_{c \in C} \sum_{\boldsymbol{z}_{ci} \in \hat{\mathcal{Z}}_c} log[\frac{e^{s(\boldsymbol{z}_{ci}, \hat{\boldsymbol{\rho}}_c)/\tau}}{e^{s(\boldsymbol{z}_{ci}, \hat{\boldsymbol{\rho}}_c(t)))/\tau} + \sum_{\tilde{c} \in \tilde{C}} \sum_{\boldsymbol{z}_{\tilde{c}i} \in \hat{\mathcal{Z}}_{\tilde{c}}} e^{s(\boldsymbol{z}_{ci}, \boldsymbol{z}_{\tilde{c}i})/\tau}}], \tag{9}$$

where $\hat{\mathcal{Z}}_c$ is the subset sampled from the set of the representations which belong to class c according to the sampling strategy, $\hat{\mathcal{Z}}_{\tilde{c}}$ is the subset sampled from the set of the representations which bot belong to class c, $\tilde{C}$ denotes the subset of C with class c removed, and the supervision information comes from the final pseudo-labels $\hat{\boldsymbol{y}}_i^u$ after the pseudo-labeling strategies.

The whole framework is shown in Fig. 2. All the pseudo-code of producing pseudo-labels and their indicators from the logit and representation spaces is shown in Algorithm 1, and the pseudo-code of the mix pseudo-labeling strategy is shown in Algorithm 2. The pseudo-code is PyTorch-like.

Algorithm 1 Pseudo-code of producing pseudo-labels and indicators

Network: Teacher's encoder f', segmentation head g', representation head h'.
Input: Unlabeled mini-batch B_u consists of X^u, the set of prototypes $\{\hat{\boldsymbol{\rho}}_c(t)\}_{c \in C}$.

for $X_u \in B_u$ **do**
  # Predict logits from unlabeled images.
  $\hat{P}^{u,lgt} \leftarrow g'(f'(X^u))$
  # Produce labels and indicators based on logits with Eq. 1.
  $\hat{Y}^{u,lgt}, \hat{J}^{u,lgt} \leftarrow lgt\_pseudo(\hat{P}^{u,lgt})$
  # Predict representations from unlabeled images.
  $\hat{R}^{u,rep} \leftarrow h'(f'(X^u))$
  # Produce labels and indicators based on representations with Eq. 4 and Eq. 5.
  $\hat{Y}^{u,rep}, \hat{J}^{u,rep} \leftarrow rep\_pseudo(\hat{R}^{u,rep}, \{\hat{\boldsymbol{\rho}}_c(t)\}_{c \in C})$
end for

Algorithm 2 Pseudo-code of mix pseudo-labeling strategy

Input: Pseudo-labels from logit space $\hat{Y}^{u,lgt}$ and from representation space $\hat{Y}^{u,rep}$. Indicators $\hat{J}^{u,lgt}$ and $\hat{J}^{u,rep}$.
Notation: Threshold δ_u for sampling strategy in $\mathcal{L}_u$, weak threshold δ_w.

Sampling valid pseudo-labels according to the sampling strategy in logit space.
$\hat{Y}_{val}^{u,lgt} \leftarrow sample\_lgt(\hat{Y}^{u,lgt}, \hat{J}^{u,lgt}, \delta_u)$
Sampling valid pseudo-labels according to the *valid sampling strategy* in logit space.
$\hat{Y}_{val}^{u,rep} \leftarrow sample\_rep(\hat{Y}^{u,rep}, \hat{J}^{u,rep}, \delta_w)$
Produce the mask for mix pseudo-labels.
$mask_y \leftarrow \hat{Y}^{u,rep}.eq(\hat{Y}^{u,lgt})$
Produce mix pseudo-labels according to the mask.
$\hat{Y}^u \leftarrow mask\_pseudo(\hat{Y}^{u,lgt}, mask_y)$

3.5. Discussions

We discuss the relations between our framework and other most-related S4 frameworks.

Compare with PCR [51]. The key insight of PCR and our method lies in the promotion of consistency between the output in the logit and representation spaces. However, PCR exclusively relies on pseudo-labels derived from logit space as the guidance in the logit and representation spaces during the unsupervised training process, thereby neglecting the exploitation of the extensive semantic information inherent in representations and prototypes. In contrast, our method adopts a collaborative way by combining pseudo-labels obtained from both the logit and representation spaces, which improves the quality of pseudo-labels and enhances the knowledge exchange between the two spaces. In addition, since these two approaches are orthogonal, one can employ

our pseudo-labeling and indicator strategies within the PCR framework, or alternatively incorporate multiple prototypes and consistency loss into our approach. Such combinations may yield further improvements in performance, however, also increase the computational complexity.

Compare with CPS [7]. CPS contains two segmentation models initialized differently. It leverages pseudo-labels generated by one model to supervise the other, thereby maintaining the consistency between the output of the two models in the logit space. In a similar vein to CPS, our cross pseudo-labeling strategy in Sec. 3.3 is also motivated by the objective of maintaining consistency during unsupervised training. However, our method is distinct from CPS in that we focus on preserving consistency between the logit space and the representation space. This distinction confers a distinct advantage in terms of memory efficiency, as we solely introduce a MLP as the representation head, rather than introducing an additional segmentation model as in CPS.

Compare with other contrastive based S4 works [32, 1, 46, 23]. We adhere to prior works to build our prototype, describe in Eq. 2. Different from them, we update our prototypes iteratively, similar to [60]. Our sampling strategies are based on the threshold, similar to [32, 23, 45]. However, it is worth noting that our indicators which serve as the basis for comparison with the threshold, are derived from both the logit and representation spaces. This stands in contrast to prior approaches where indicators solely originate from the logit space. The efficacy of this modification is substantiated and extensively discussed in Sec. 5.2.

4. Experiments

4.1. Setup

Datasets. We conduct experiments on PASCAL VOC 2012 dataset [12], Cityscapes dataset [9], ADE20K dataset [59], and COCO-Stuff 10K dataset [2] to validate the effectiveness of our proposed method. The original PASCAL VOC 2012 dataset contains 1,464 labeled images in `train` set and 1,449 images for validation in `val` set. Following [7, 37], we additionally introduce 9,118 images from SBD [18] as training images. Since the labels in SBD are coarsely annotated, following [46], we use both *classic* VOC `train` set (1,464 candidate labeled images) and *blender* VOC `train` set (10,582 candidate labeled images). Cityscapes dataset is a dataset for urban scene understanding, which contains 2,975 images in `train` set and 500 images in `val` set. ADE20K has 20,210 and 2,000 images in `train` and `val` set, with 150 classes in total. COCO-Stuff 10K has 9,000 images in `train` set and 1,000 images in `val` set, with 181 classes. The approach of preprocessing labels is followed by MMSegmentation [8].

Network structure. We use Deeplabv3+ [4] with ResNet-101 [20] pre-trained on ImageNet [10] as our network structure. The segmentation and representation head are composed of `Conv-BN-ReLU-Conv`.

Implementation details. For training on PASCAL VOC 2012 dataset and COCO-Stuff 10K dataset, we set the learning rate as 0.0064, weight decay as 0.0005, crop size as 512 $\times$ 512, batch size as 16, and a total of 40,000 iterations. For training on the Cityscapes dataset, we set the learning rate as 0.0038, weight decay as 0.0005, crop size as 768 $\times$ 768, batch size as 8, and a total of 80,000 iterations. For training on ADE20K dataset, we set the learning rate as 0.0064, weight decay as 0.0005, crop size as 512 $\times$ 512, batch size as 16, and a total of 80,000 iterations. We use the poly scheduling to decay the learning rate during the training process: $lr = lr_{base} \times (1 - \frac{epoch}{total\_epoch})^{0.9}$. We use the mean of Intersection over Union (mIoU) as the metric in evaluation. We use the sliding window strategy to evaluate the performance of our method on the Cityscapes dataset, following [7]. In addition, when adopting our cross-labeling strategy, due to the requirement of a set of high-quality prototypes when classifying each representation, we first solely use the supervision from logit space for 20 epochs to initialize the prototypes.

4.2. Comparison with Existing Methods

In this subsection, we first reproduce three baselines: MT [44], CutMix [15], and a same contrastive-based framework with us but with only logit space pseudo-labels and indicators (Baseline) on *classic* VOC `train` set. Meanwhile, we make the comparison of our method with mix (CSS (mix)) and cross (CSS (crs.)) pseudo-labeling strategy on *blender* VOC `train` set and Cityscapes `train` set with following recent SOTA S4 methods: CCT [38], CPS [7], U^{2}PL [46], ST++ [52], PRCL [48], PCR [51], and PSMT [33]. Since the data split will dramatically affect the performance in S4, *i.e.*, choosing labeled images plays an important role in the final results, we conduct experiments with three different data splits and report the mean value and standard deviation (blue color numbers). Since the mix pseudo-labeling strategy has better performance, we only use CSS (mix) when compared with SOTAs. Meanwhile, we use ResNet-101 with deep stem blocks as our network structure when compared with SOTAs. Since there is no uniform data split, we use the data splits in U^{2}PL [46]. We use the OHEM loss when training Cityscapes.

Results on PASCAL VOC 2012. Tab. 1 shows the comparison with our baselines on *Classic* PASCAL VOC 2012 set. Our method consistently outperforms baselines with an acceptable standard deviation on all label rates. Tab. 2 shows the comparison with the SOTAs on PASCAL VOC 2012.

Results on Cityscapes. Tab. 3 shows the performance of our method on Cityscapes.

Results on ADE20K and COCO-Stuff 10K. Tab. 4 shows the results of our method on ADE20K and COCO-Stuff.

Table 1. Results on *classic* VOC `train` set with four different label rate. Labeled data splits are from the original VOC `train` set. All approaches are reproduced with three runs and three different data splits.

Pascal VOC 2012 (*Classic*)				
Method	92	183	366	732
Sup.	$51.57_{\pm 3.58}$	$54.69_{\pm 2.44}$	$64.86_{\pm 1.04}$	$70.77_{\pm 0.76}$
MT	$58.92_{\pm 2.99}$	$61.63_{\pm 1.76}$	$66.79_{\pm 0.53}$	$71.58_{\pm 0.51}$
CutMix	$65.82_{\pm 3.60}$	$67.91_{\pm 1.47}$	$72.53_{\pm 0.50}$	$74.08_{\pm 0.49}$
Baseline	$66.91_{\pm 4.21}$	$70.32_{\pm 1.86}$	$73.97_{\pm 1.04}$	$76.49_{\pm 0.58}$
CSS(crs.)	$\underline{67.03}_{\pm 4.58}$	$\underline{71.41}_{\pm 1.67}$	$\underline{74.47}_{\pm 1.08}$	$\underline{77.08}_{\pm 0.37}$
CSS(mix)	$\mathbf{68.09}_{\pm 4.89}$	$\mathbf{71.93}_{\pm 1.88}$	$\mathbf{74.91}_{\pm 1.12}$	$\mathbf{77.57}_{\pm 0.73}$

Table 2. Results on *blender* VOC `train` set. All the results from the recent papers [46, 52, 13, 29, 33, 51]. Labeled data is from the augmented VOC `train` set and the data splits are from [46, 33].

Pascal VOC 2012 (*Blender*)				
Method	662	1323	2646	5291
CCT [38]	71.86	73.68	76.51	77.40
CPS [7]	74.48	76.44	77.68	78.64
U^2PL [46]	77.21	79.01	79.30	80.50
ST++ [52]	74.70	77.90	77.90	-
PRCL [48]	76.96	78.16	79.02	79.59
PCR [51]	78.60	**80.71**	80.78	80.91
PSMT [33]	75.50	78.20	78.72	79.76
CSS (mix)	**78.73**	$\underline{79.54}$	**80.82**	**81.06**

Table 3. Results on Cityscapes. The model is trained on the Cityscapes `train` set, which consists of 2,975 samples in total, and tested on Cityscapes `val` set. And all the results from the recent papers [46, 51, 33].

Cityscapes				
Method	186	372	744	1488
CCT [38]	69.32	74.12	75.99	78.10
CPS [7]	69.78	74.31	74.58	76.82
U^2PL [46]	70.30	74.37	76.47	79.05
PCR [51]	73.41	76.31	**78.40**	79.11
PSMT [33]	-	76.89	77.60	79.09
CSS (mix)	**74.02**	**76.93**	$\underline{77.94}$	**79.62**

5. Ablative Study

The main contribution of our work lies in **1)** collaborative pseudo-labeling strategies and **2)** a new indicator for representation learning. To further prove the effectiveness of our proposed method, we conduct ablative studies on these two points. We choose Deeplabv3+ with ResNet-101 pre-trained on ImageNet as our backbone and leverage 92 labeled images and 183 labeled images in PASCAL VOC 2012. The other settings are the same as those in Sec. 4.

5.1. Effectiveness of Collaborative pseudo-labeling

Quality of pseudo-labels. To illustrate the superiority of using pseudo-labels from two spaces in a collaborative way as supervision, we conduct experiments to show the quality of pseudo-labels obtained **1)** from logit space (lgt.), **2)** from representation space (rep.), and **3)** from the mix pseudo-labeling strategy (mix). The pseudo-labels are sampled with corresponding sampling strategies in Sec. 3.4. Tab. 5 illustrates the IoU of pseudo-labels for each class on PASCAL VOC 2012 with 92 labeled images. The results clearly indicate that employing pseudo-labels from the representation space enhances the accuracy of the final pseudo-labels in most classes. This improvement is particularly evident in classes that are originally under-performing, such as the IoU improvement of 11.42% for the `chair` class and 11.58% for the `sofa` class.

(a) Image (b) GT (c) lgt. mask (d) lgt. label (e) rep. mask (f) rep. label

Figure 3. Differences between pseudo-labels in different spaces.

Meanwhile, we also visualize the pseudo-labels obtained from logit space (lgt.) and representation space (rep.) in Fig. 3. Fig. 3 (c) and (e) show the masks for pseudo-labels produced by the sampling strategies. In particular, the white color represents the valid pixels used during unlabeled learning, while the black color indicates the discarded pixels. Fig. 3 (d) and (f) are the pseudo-labels we obtained from two spaces. The figure clearly illustrates the differences between pseudo-labels produced by different spaces. For example, the parts of the instance edge in pseudo-labels are usually discarded since they are challenging for learning in logit space. However, pseudo-labels from representation space will easily tackle this problem (first row). The pseudo-labels from representation space are more inaccurate in some complex scenes, which be resolved by combining pseudo-labels from logit space (second row).

We mainly attribute the differences in pseudo-label to the differing concentrations of learning in the two spaces. Specifically, learning in the logit space primarily emphasizes the most discriminative part of features, whereas that in the representation space treats each part of features equally. As a result, learning in the logit space may overlook minor feature differences, leading to sub-optimal performance in predicting instance edges and distinguishing between similar classes (*e.g.*, `chair` and `sofa`). Conversely, learning in the representation space produces balanced performance across all image parts and classes. However, this can lead to erroneous predictions for classes with high intra-class variance (*e.g.*, `background`). By leveraging pseudo-labels from the two spaces, we capitalize on the strengths of the learning in each space and enhance the knowledge exchange between the two spaces.

Table 4. Results on ADE20K and COCO-Stuff 10K with four different label rates and single data splits.

	ADE20K				COCO-Stuff 10K			
Method	1/16	1/8	1/4	1/2	1/16	1/8	1/4	1/2
Cutmix	28.91	31.89	34.62	39.15	23.09	26.14	29.88	30.91
Baseline	31.11	33.24	36.46	39.91	26.57	27.91	28.97	31.08
CSS (crs.)	32.01	33.92	**37.84**	39.97	27.06	28.85	**30.59**	31.73
CSS (mix)	**32.55**	**34.21**	37.01	**40.85**	**27.46**	**29.51**	29.98	**31.91**

Table 5. The quality of pseudo-labels from different pseudo-labeling strategies. The pseudo-labels are sampled by sampling strategies.

source	back	aero.	bicy.	bird	boat	bott.	bus	car	cat	chair	cow
lgt.	96.78	95.14	77.47	93.38	81.37	87.54	96.76	95.48	94.47	4.09	92.15
rep.	90.71	96.50	61.42	75.75	53.30	54.65	84.46	80.55	91.11	28.03	88.16
mix	96.66$_{\downarrow 0.12}$	98.12$_{\uparrow 2.98}$	82.76$_{\uparrow 5.29}$	94.47$_{\uparrow 1.09}$	85.12$_{\uparrow 3.75}$	89.94$_{\uparrow 2.40}$	97.03$_{\uparrow 0.27}$	95.98$_{\uparrow 0.50}$	94.65$_{\uparrow 0.18}$	19.01$_{\uparrow 14.92}$	94.81$_{\uparrow 2.66}$
source	tabel	dog	horse	motor	pers.	plant	sheep	sofa	train	tv	mIoU
lgt.	58.34	94.12	93.01	91.37	93.68	50.33	91.10	18.16	86.65	64.89	78.87
rep.	52.23	86.97	73.22	88.70	91.75	56.13	66.78	35.25	88.51	62.61	71.57
mix	64.35$_{\uparrow 6.01}$	94.33$_{\uparrow 0.21}$	93.65$_{\uparrow 0.64}$	91.50$_{\uparrow 0.13}$	93.01$_{\downarrow 0.67}$	55.64$_{\uparrow 5.31}$	91.56$_{\uparrow 0.46}$	29.74$_{\uparrow 11.58}$	93.54$_{\uparrow 6.89}$	69.48$_{\uparrow 4.59}$	82.17$_{\uparrow 3.33}$

Table 6. Results on pseudo-labels from different sources on two different label rates.

source	92 labels	183 labels
logit space	67.11	70.32
representation space	64.20	67.52
mix pseudo-labeling	68.41$_{\uparrow 1.30}$	72.74$_{\uparrow 2.42}$
cross pseudo-labeling	67.85$_{\uparrow 0.74}$	71.98$_{\uparrow 1.66}$

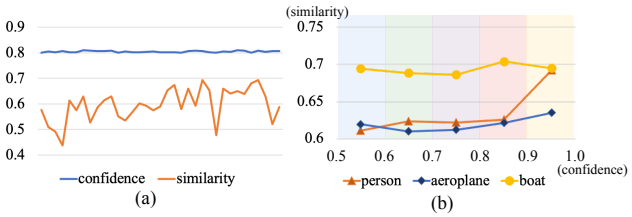


Figure 4. Relations between similarity and confidence.

Results of different strategies To investigate the involvement of different pseudo-labeling strategies, we conduct the experiments as follows: **1)** Using pseudo-labels from logit space. **2)** Using pseudo-labels from representation space. **3)** Using mix pseudo-labeling strategy. **4)** Using cross pseudo-labeling strategy. Tab. 6 shows the effectiveness of our proposed strategy in two different label rates. The results show that the performances of experiments with collaborative pseudo-labeling strategies are better than the ones whose pseudo-labels come from a single space with two different label rates, which proves the effectiveness of our proposed collaboration between the two spaces. It is worth noting that even though the quality of pseudo-labels from representation space is lower than that from logit space, the performance of the model is also boosted by using the cross pseudo-labeling strategy to maintain consistency between the predictions in two spaces. In addition, our method with the mix pseudo-labeling strategy outperforms that with the cross pseudo-labeling strategy.

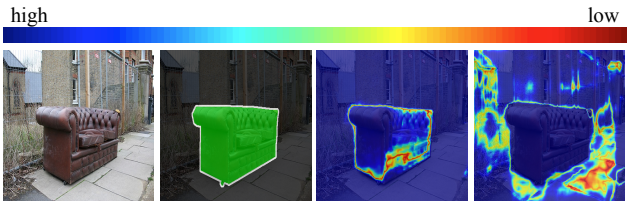
5.2. Effectiveness of the Indicator

Limitation of merely using confidence. To explain the limitation of merely using confidence for learning in the logit and representation spaces, we conduct experiments to show the relations between confidence and similarity. The similarity is the cosine similarity between representations and prototypes, which directly shows the confusion level between the representation and the prototype of its class. Fig. 4 (a) shows the comparison between the confidence of each prediction and the corresponding similarity. We use the class `person` for demonstrating in our experiments. It shows clearly that even though fixing the confidence into a small range (from 0.8 to 0.81 in our settings), the similarity varies. Meanwhile, in Fig. 4 (b), different color bars stand for different intervals of confidence, and the lines denote the mean similarity between the prototype and each representation whose corresponding confidence is in the current interval. Fig. 4 (b) illustrates that the mean similarity of the class fluctuates when its interval of confidence rises. Both two figures imply that confidence is not able to represent the confusion level between representations and prototypes since there are no direct and close relations between confidence and similarity.

Fig. 5 visualizes the similarity and confidence of an image in both logit (lgt.) and representation (rep.) spaces, indicating the varying levels of confusion in the same region when learning in different spaces.

We also attribute it to the different concentrations of learning two spaces, *i.e.*, the confusing region in one space can be more readily addressed in the other space.

Thus, it is inappropriate to choose confidence as the indicator to involve representation learning, *e.g.*, sampling more hard samples with threshold and indicator. In contrast, our indicator directly employs similarity between the represen-



(a) Image (b) GT (c) lgt. space (c) rep. space

Figure 5. Visualization of the confusing part in different spaces.

tation and prototype of its class, which directly reflects the confusion level in representation learning. It is more accurate to use similarity as the indicator to sample hard and critical samples in representation learning.

Results of different indicators. Tab. 7 shows the impact of using different indicators. We conduct experiments on two different label rates (92 and 183) and three different indicators for pseudo-labels: only confidence (conf.), only similarity (smlr.), and confidence for learning logit space while similarity for learning representation space (mix). We use two different approaches to obtain pseudo-labels: from logit space only (seg label) and from mix pseudo-labeling strategy (mix label). It is clear that using both confidence and similarity to involve the learning in their own spaces obtains the best performance.

Table 7. Results on indicators from different spaces on two different label rates.

	92 labels		183 labels	
source	seg label	mix label	seg label	mix label
conf.	67.11	68.33	70.32	71.92
smlr.	66.16	66.89	68.90	69.25
mix	67.80$_{\uparrow 0.69}$	68.41$_{\uparrow 1.30}$	71.50$_{\uparrow 1.18}$	72.74$_{\uparrow 2.42}$

5.3. Ablation study of Components

In this section, we conduct experiments to introduce our components in CSS step by step, with results shown in Tab. 8. Our baseline is the conventional contrastive-based S4, achieving mIoU of 67.11% on 92 labels and 70.32% on 183 labels. Mix and cross means the pseudo-labels are from the mix pseudo-labeling strategy and cross pseudo-labeling strategy while the indicator is still the confidence in two spaces. Ind means we use the different indicators in different spaces while the pseudo-labels are from logit space. The last two rows represent our proposed two pseudo-labeling strategies with indicators from two spaces.

5.4. Qualitative Results

Fig. 6 shows the qualitative results of different methods on PASCAL VOC 2012 with 92 labeled images. Baseline means the conventional contrastive-based method. Compared with the original self-training methods (CutMix), thanks to introducing pixel-wise contrastive learning, the baseline, and our method perform better in some ambiguous regions. Furthermore, benefiting from the supervision of two spaces and different indicators in different spaces, our method outperforms the baseline.

Table 8. Ablation study on different components of our CSS.

component	92 labels	183 labels
baseline	67.11	70.32
mix	68.33$_{\uparrow 1.22}$	71.92$_{\uparrow 1.60}$
cross	67.21$_{\uparrow 0.10}$	70.87$_{\uparrow 0.55}$
ind	67.80$_{\uparrow 0.69}$	71.50$_{\uparrow 1.18}$
mix + ind	68.41$_{\uparrow 1.30}$	72.74$_{\uparrow 2.42}$
cross + ind	67.85$_{\uparrow 0.74}$	71.98$_{\uparrow 1.66}$

(a) Image (b) GT (c) CutMix (d) Baseline (e) CSS (cross) (f) CSS (mix)

Figure 6. Visualization on PASCAL VOC 2012 with 92 labeled images. Yellow boxes highlight the main differences.

6. Conclusion

In this paper, we propose two collaborative pseudo-labeling strategies to take full use of the semantic information in the representation space and enhance the knowledge exchange between the logit and representation spaces. Moreover, we employ a new indicator for the learning process in the representation space. Extensive experiments demonstrate that our pseudo-labeling strategies obtain more reliable supervision during unlabeled training and our indicator helps the model to concentrate on more critical parts during representation learning.

Future work: In this paper, we employ pseudo-labeling strategies to utilize the semantic information in both logit and representation spaces. In the future, we will investigate more powerful strategies to enhance the knowledge exchange between two spaces.

References

[1] Iñigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C. Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8219–8228, October 2021.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[5] Mu Chen, Zhedong Zheng, Yi Yang, and Tat-Seng Chua. Pipa: Pixel-and patch-wise self-supervised learning for domain adaptative semantic segmentation. *arXiv preprint arXiv:2211.07609*, 2022.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[8] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3, 2018.

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[13] Jiashuo Fan, Bin Gao, Huan Jin, and Lihui Jiang. Ucc: Uncertainty guided cross-head co-training for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9947–9956, June 2022.

[14] Zhengyang Feng, Qianyu Zhou, Qiqi Gu, Xin Tan, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Dmt: Dynamic mutual training for semi-supervised learning. *Pattern Recognition*, page 108777, 2022.

[15] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, high-dimensional perturbations. In *In 31th British Machine Vision Conference*, 2020.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[17] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Unbiased subclass regularization for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9968–9978, 2022.

[18] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 international conference on computer vision*, pages 991–998. IEEE, 2011.

[19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[21] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6930–6940, October 2021.

[22] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[23] Hanzhe Hu, Jinshi Cui, and Liwei Wang. Region-aware contrastive learning for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16291–16301, 2021.

[24] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems*, 34:22106–22118, 2021.

[25] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

[26] Zhengkai Jiang, Yuxi Li, Ceyuan Yang, Peng Gao, Yabiao Wang, Ying Tai, and Chengjie Wang. Prototypical contrast adaptation for domain adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIV*, pages 36–54. Springer, 2022.

[27] Tarun Kalluri, Girish Varma, Manmohan Chandraker, and CV Jawahar. Universal semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5259–5270, 2019.

[28] Rihuan Ke, Angelica I Aviles-Rivero, Saurabh Pandey, Saikumar Reddy, and Carola-Bibiane Schönlieb. A three-stage self-training framework for semi-supervised semantic segmentation. *IEEE Transactions on Image Processing*, 31:1805–1815, 2022.

[29] Donghyeon Kwon and Suha Kwak. Semi-supervised semantic segmentation with error localization network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9957–9967, 2022.

[30] Daiqing Li, Junlin Yang, Karsten Kreis, Antonio Torralba, and Sanja Fidler. Semantic segmentation with generative models: Semi-supervised learning and strong out-of-domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8300–8311, 2021.

[31] Jinlu Liu, Liang Song, and Yongqiang Qin. Prototype rectification for few-shot learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 741–756. Springer, 2020.

[32] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew Davison. Bootstrapping semantic segmentation with regional contrast. In *International Conference on Learning Representations*, 2022.

[33] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and Gustavo Carneiro. Perturbed and strict mean teachers for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4258–4267, 2022.

[34] Binjie Mao, Xinbang Zhang, Lingfeng Wang, Qian Zhang, Shiming Xiang, and Chunhong Pan. Learning from the target: Dual prototype network for few shot semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1953–1961, 2022.

[35] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high-and low-level consistency. *IEEE transactions on pattern analysis and machine intelligence*, 43(4):1369–1379, 2019.

[36] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[37] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1369–1378, 2021.

[38] Yassine Ouali, Celine Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[39] Yingwei Pan, Ting Yao, Yehao Li, Yu Wang, Chong-Wah Ngo, and Tao Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2239–2247, 2019.

[40] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. Deep co-training for semi-supervised image segmentation. *Pattern Recognition*, 107:107269, 2020.

[41] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[42] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[43] Korawat Tanwisuth, Xinjie Fan, Huangjie Zheng, Shujian Zhang, Hao Zhang, Bo Chen, and Mingyuan Zhou. A prototype-oriented framework for unsupervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:17194–17208, 2021.

[44] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[45] Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Ender Konukoglu, and Luc Van Gool. Exploring cross-image pixel contrast for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7303–7313, 2021.

[46] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[47] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10857–10866, June 2021.

[48] Haoyu Xie, Changqi Wang, Mingkai Zheng, Minjing Dong, Shan You, and Chang Xu. Boosting semi-supervised semantic segmentation with probabilistic representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2938–2946, 2023.

[49] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[50] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16684–16693, 2021.

[51] Hai-Ming Xu, Lingqiao Liu, Qiuchen Bian, and Zhen Yang. Semi-supervised semantic segmentation with prototype-based consistency regularization. *Advances in Neural Information Processing Systems*, 2022.

[52] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4268–4277, 2022.

[53] Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8229–8238, 2021.

[54] Xiangyu Yue, Zangwei Zheng, Shanghang Zhang, Yang Gao, Trevor Darrell, Kurt Keutzer, and Alberto Sangiovanni Vincentelli. Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13834–13844, 2021.

[55] Xiangyun Zhao, Raviteja Vemulapalli, Philip Andrew Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label efficient semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10623–10633, October 2021.

[56] Xu Zheng, Yunhao Luo, Hao Wang, Chong Fu, and Lin Wang. Transformer-cnn cohort: Semi-supervised semantic segmentation by the best of both students, 2022.

[57] Xu Zheng, Jinjing Zhu, Yexin Liu, Zidong Cao, Chong Fu, and Lin Wang. Both style and distortion matter: Dual-path unsupervised domain adaptation for panoramic semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1285–1295, June 2023.

[58] Yuanyi Zhong, Bodi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7273–7282, 2021.

[59] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[60] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022.

[61] Yanning Zhou, Hang Xu, Wei Zhang, Bin Gao, and Pheng-Ann Heng. C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7036–7045, 2021.