# Semi-supervised Semantic Segmentation via Prototypical Contrastive Learning

Zenggui Chen
Peking University
China
zgc731956@gmail.com

Zhouhui Lian*
Wangxuan Institute of Computer Technology, Peking University
China
lianzhouhui@pku.edu.cn

## ABSTRACT

The key idea of semi-supervised semantic segmentation is to leverage both labeled and unlabeled data. To achieve the goal, most existing methods resort to pseudo-labels for training. However, the dispersed feature distribution and biased category centroids could inevitably lead to the calculation deviation of feature distances and noisy pseudo labels. In this paper, we propose to denoise pseudo labels with representative prototypes. Specifically, to mitigate the effects of outliers, we first employ automatic clustering to model multiple prototypes with which the distribution of outliers can be better characterized. Then, a compact structure and clear decision boundary can be obtained by using contrastive learning. It is worth noting that our prototype-wise pseudo segmentation strategy can also be applied in most existing semantic segmentation networks. Experimental results show that our method outperforms other state-of-the-art approaches on both Cityscapes and Pascal VOC semantic segmentation datasets under various data partition protocols.

## CCS CONCEPTS

• **Computing methodologies** → **Image segmentation**.

## KEYWORDS

pseudo labels; prototype; prototypical contrastive learning

**ACM Reference Format:**
Zenggui Chen and Zhouhui Lian. 2022. Semi-supervised Semantic Segmentation via Prototypical Contrastive Learning. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisbon, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3548353

## 1 INTRODUCTION

Semantic segmentation is a fundamental task in computer vision. It has been widely used in many applications such as scene understanding, autonomous driving, etc. However, supervised semantic segmentation requires pixel-level fine annotated data, which is labor-intensive and time-consuming to acquire. Therefore, semi-supervised semantic segmentation has attracted intensive attention in the last few years, the key idea of which is to leverage both labeled data and unlabeled data. Thus, how to boost the performance of semantic segmentation by exploiting unlabeled data becomes the key issue.

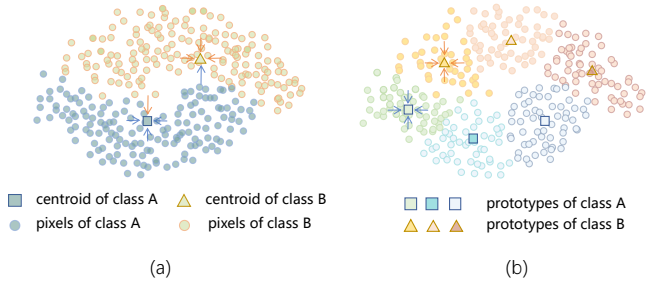



**Figure 1: Illustration of our motivation. (a) shows that a single category centroid is biased in representing the distribution, causing incorrect classification of outliers. (b) represents each class as a set of prototypes, with which the distribution of outliers can be better characterized.**

Typical solutions are to use consistency regularization [13, 21, 30] or self-training [47, 48]. Specifically, for self-training based methods, a model is trained with labeled data and is used to generate pseudo labels of the unlabeled data with which the segmentation model is retrained. While, consistency regularization based methods achieve semi-supervised semantic segmentation by making the network outputs invariant to disturbance [13, 22], such as different network parameters [7, 20] or different data augmentations [49]. Actually, both the consistency regularization and self-training methods resort to pseudo labeling [24], and thus the performance of these methods mainly depends on the prediction accuracy of pseudo labels. However, without sufficient supervision, a semantic segmentation network is typically confused in some pixels and produces noisy pseudo labels, leading to performance decay. Thereby, pseudo labels denoising plays an important role in semi-supervised semantic segmentation methods.

There exists many prior methods[28, 32] that improve pseudo labeling accuracy accounting confidence and uncertainty [32] as criterion, and drop out low-confidence pixels. However, those deserted pixels may be still under-performing and dropped in the next iteration. Furthermore, due to the existence of long-tailed distribution, many of the dropped pixels belong to under-performing tailed

*Corresponding author


*MM '22, October 10–14, 2022, Lisbon, Portugal.*

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548353

categories, leading to a biased learning. We can see that the existence of noisy pseudo labels boils down to the dispersed features and biased class centroids in the unlabeled data domain. As shown in Figure 1(a), due to the mixed and dispersed distribution of two category spaces, a single centroid is insufficient in representing the category, vulnerable to intra-class variation. This could inevitably lead to the calculation deviation of feature distances. Formally, the classification can be considered as retrieving the nearest category centroid. Therefore, with only a single centroid for each category, an outlier may be incorrectly classified because of the minimum feature distance to the other category centroids. In addition, the decision boundary lies in interlaced regions rather than low density ones.

In this paper, we propose to denoise pseudo labels with representative prototypes via pixel-level prototypical contrastive learning [25]. More specifically, to better characterize the distribution of outliers especailly those lie near the border, we first employ automatic clustering to model multiple prototypes in each category space. We implement clustering using a transformer decoder [39] with a series of query embeddings to produce corresponding mask embeddings, each of which refers to a prototype. Then, a pre-defined prototype-anchored assignment strategy is applied. As shown in Figure 1(b), prototypes, i.e., sub-centroids of classes, can better correspond to the distribution of local regions. With additional conformable prototypes, outliers are clustered to be discriminative from other categories. Furthermore, we combine pseudo supervision and contrastive learning [42, 43] in a prototype-wise manner, encouraging the network to learn a separable structure. For the sake of convenience, we denote out approach as PPS (Prototypical Pseudo Segmentation).

With the help of the above-mentioned techniques, our method obtains superior performance compared to existing approaches. It is worth noting that our prototype-wise pseudo segmentation strategy can also be applied in most existing semantic segmentation networks. Major contributions of this paper can be summarized as follows:

- We propose to implement clustering and model multiple prototypes in each category space and apply pseudo supervision in a prototype-wise manner. By doing so, unbiased class sub-centroids can be obtained and the distribution of outliers can be better characterized.
- To mitigate the issue of dispersed features and enforce the decision boundary lying in the low density regions, we take advantage of pixel-level prototypical contrastive learning by using the prediction as pseudo guidance for sampling of positive and negative pairs.
- Extensive experiments have been conducted on Cityscapes and Pascal VOC semantic segmentation datasets to demonstrate the superiority of our method compared to the state of the art under various data partition protocols.

## 2 RELATED WORK

### 2.1 Semantic Segmentation

Semantic Segmentation, a pixel-level classification task, plays a fundamental role in computer vision. Most methods follow the paradigm of fully convolutional network(FCN) [27]. The subsequent work like U-net [33], aggregate information between different layers through connections from encoder to decoder. ASPP [4, 5] uses atrous convolutions with different atrous rates to capture long-range context avoiding losing too much spatial information during downsampling.

Lately, due to the capacity of global receptive field, transformers [36, 39] benefit the semantic segmentation. SETR [44] and Segmenter [35] are two classical models based on vision transformer [10]. Moreover, MaX-DeepLab [41] and MaskFormer [8] propose to predict class-labeled masks rather than perform per-pixel classification, which build a unified model for semantic segmentation and panoptic segmentation. ProtoSeg[46] analyses semantic segmentation in a prototype view.

In this paper, we employ DeepLabv3+ [5] as our segmentation module. And our formation of prototypes and clustering is similar to MaskFormer [8], but we use different inference strategy that constitutes a distinct concept of mask embedding.

### 2.2 Semi-Supervised Semantic Segmentation

Semi-supervised semantic segmentation has attracted intensive attention for less demand of pixel-level finely annotations, the goal of which is to leverage unlabeled data to boost the performance. Consistency regularization and self-training are two typical paradigms when designing the framework.

**Consistency regularization.** Consistency regularization[7, 20, 40] based methods aim to obtain consistent representations in spite of the perturbation applied to the input. Diverse network parameters initialization or various data augmentation is commonly used for input perturbation.

CutMix [13] make use of mask-based augmentation as the input perturbation. GCT[20] imposes the consistency constraint upon outputs from differently initialled segmentation models. CPS [7] employs pseudo segmentation labels from two parallel network with different initialization as the supervision from each other. CCT [30] introduce feature perturbation to the out of encoder and enforce the consistency of different outputs from multiple decoders.

**Self-training.** Self-training [3, 29, 34, 47, 48] leverage unlabeled data by utilizing a segmentation network pretrained on labeled data to produce pseudo segmentation maps with which the network is retrained. s4GAN[29] introduces discriminator under self-training paradigm for low-level and high-level learning. AEL [17] extends the CutMix [13] to mitigate long-tailed distribution problem in semi-supervised semantic segmentation.

Pseudo-labeling is commonly used in whether consistency regularization or self-training paradigm. However, Without sufficient supervision, the network produce inexact predictions. Noisy pseudo labels may lead to poor calibration of network. Consequently, many methods seek to how to decide the pseudo segmentation maps.

UPS [32] improves pseudo labeling accuracy accounting confidence and uncertainty as criterion. PseudoSeg [49] focus on designing well-calibrated structured pseudo labels for unlabeled data. ECS [28] introduce a correction network to locate correct predictions and amend incorrect ones.

In this paper, we refer to CPS [7] to design our consistency regularization scheme with the modified prototype-wise pseudo

supervision. Furthermore, we propose a prototype based strategy to denoise pseudo labels.

## 2.3 Contrastive Learning

Contrastive learning [15, 16, 42, 43] has made significant progress in unsupervised or self-supervised learning. It aims to push negative samples apart and pull positive samples closer, that can reinforce the learning of representation. Both the sampling of positive pairs and negative pairs as well as the number of negative samples exert an influence on the performance. SimCLR [6] enforces consistency under different views and puts it over with big batch size. MoCo [15] maintains and updates a memory bank using queue. SwAV [2] takes advantage of clustering in contrastive learning. PCL [25] introduce prototypes to help find the maximum-likelihood estimation of the network parameters.

Except instance-wise contrastive learning, there are many semantic segmentation models [1, 11, 26, 45] referred to pixel-wise contrastive learning. ReCo [26] uses hard negative pixels to perform contrastive learning. PC2Seg [45] leverages both pixel-consistency in the label space and pixel-contrastive in the feature space.

Similar to [25], we find that representing classes with multiple prototypes is robust to intra-class variation. Besides, we apply contrastive learning in pixel-level and sample positive and negative pairs in prototype-wise.

# 3 METHOD

In this section, we present the details of our proposed method. We first describe how semantic segmentation can be formulated as a semi-supervised problem using the consistency regularization scheme with pseudo labels in Sec 3.2. Then, we introduce our transformer based formulation of prototypes modelling and clustering, and further implement contrastive learning in a prototype-wise manner (Sec 3.3). Afterwards, we elaborate a simple inference strategy to assign prototypes to semantic categories (Sec 3.4). Moreover, prototype activation regularization is proposed to eliminate trivial solutions (Sec 3.5).

## 3.1 Preliminary

The optimization of classifier can be considered as the procedure of approaching actual category centroids. Formally, a parametric softmax can be viewed as the calculation of the distance from category centroids:

$$p_i(c) = \frac{\exp\left(\boldsymbol{w}_c^\top \boldsymbol{\rho_i}\right)}{\sum_{c'=1}^{C} \exp\left(\boldsymbol{w}_{c'}^\top \boldsymbol{\rho_i}\right)}, \tag{1}$$

where $p_i(c)$ represents the probability of the $i$-th pixel belonging to the $c$-th class, $\boldsymbol{\rho_i}$ is the embedding of pixel $i$, $\boldsymbol{W} = \{(\boldsymbol{w}_c)\}_{c'=1}^{C} \in \mathbb{R}^{C\times D}$, $\boldsymbol{w}_c \in \mathbb{R}^D$ denotes the parameters of classifier and can be also regarded as the coordinates of centroids in the category space. Therefore, $\boldsymbol{w}_c^\top \boldsymbol{\rho_i}$ is equal to measuring the cosine distance between the pixel $i$ and the centroid $c$.

## 3.2 Overview

Fig. 2 illustrates an overview of the proposed method. Given a labeled dataset $\mathcal{D}_l = \{(x_l, y_l)\}$, where $x_l$ and $y_l$ denote an image and its corresponding annotation, respectively, and a unlabeled dataset $\mathcal{D}_u = \{x_u\}$, the goal of semi-supervised semantic segmentation is to train a network which leverages both labeled and unlabeled data. To handle this task, our approach resorts to consistency regularization. Specifically, there exist two parallel segmentation networks with the same structure but different initial parameters, i.e., $\theta_1$ and $\theta_2$, respectively. The segmentation procedure can be formulated as:

$$P_i = \xi(h(X;\theta_i)), i \in \{1,2\}, \tag{2}$$

where $X$ and $P_i$ are input images and the confidence map predicted by the $i$-th branch, respectively, $h(\cdot)$ represents the segmentation network producing the prototypical classification $\mathcal{P}_m = h(X)$ which denotes the probability of pixels belonging to each prototype, and $\xi(\cdot)$ represents the prototype-anchored assignment.

For labeled data, we apply the standard pixel-wise cross-entropy loss:

$$\mathcal{L}_s = \frac{1}{HW}\sum_{i=1}^{HW} \ell_{ce}\left(\boldsymbol{y}_l^{(i,k)}, \boldsymbol{p}_l^{(i,k)}\right), \tag{3}$$

where $p_l^{(i,k)}$ and $y_l^{(i,k)}$ denote the probability prediction and the corresponding ground truth of the $i$-th pixel belonging to the $k$-th category, respectively, $k \in \{1, 2, \cdots, C\}$ with C denoting the number of categories, $\ell_{ce}$ denotes the cross-entropy loss, and H and W denote the height and weight of images. Besides, we exploit the information of unlabeled data by utilizing pseudo labels based self-supervision in a prototype-wise form which differs from previous methods. Our contrastive learning is implemented between prototypes to learn a more compact feature space and clear decision boundary. Along with the prototype activation regularization term $\mathcal{L}_r$, our full loss is defined as:

$$\mathcal{L}_t = \mathcal{L}_s + \lambda_1 \mathcal{L}_{contra} + \lambda_2 \mathcal{L}_{sp} + \lambda_3 \mathcal{L}_r, \tag{4}$$

where $\lambda_1$, $\lambda_2$, and $\lambda_3$ denote the weighting coefficients, $\mathcal{L}_{contra}$ and $\mathcal{L}_{sp}$ represent the contrastive learning loss and the self-supervision loss, respectively.

## 3.3 Prototype Modelling and Contrastive Learning

Consistency regularization is implemented with pseudo labels to leverage unlabeled data to enhance the segmentation performance. Previous methods formulate pseudo-labeling segmentation using the cross-entropy loss as Eq. 3 based on the class-wise pseudo label prediction $\hat{y}^{(i,k)}$:

$$\mathcal{L}_u = \frac{1}{HW}\sum_{i=1}^{HW} \ell_{ce}\left(\hat{\boldsymbol{y}}_u^{(i,k)}, \boldsymbol{p}_u^{(i,k)}\right). \tag{5}$$

However, due to the lack of efficient supervision for unlabeled data, those previous methods suffer from mixed and dispersed feature distribution among different categories. A single category centroid is biased in characterizing a feature distribution causing the incorrect classification of outliers as shown in Fig. 1(a). Obviously, pseudo labels $\hat{y}^{(i,k)}$ are so noisy that typically lead to poor network calibration. Although there already exist many works [28, 32] that attempt to denoise pseudo labels by setting confidence or uncertainty thresholds, the problem remains because most under-performing pixels

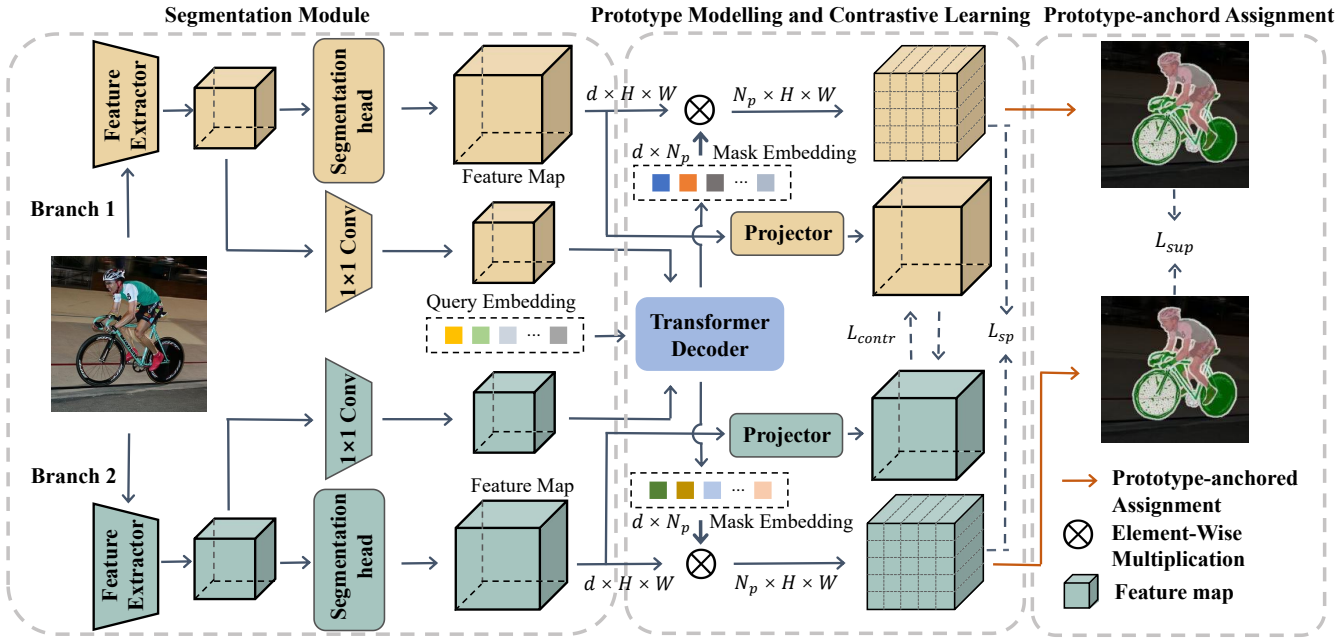



**Figure 2: An overview of our method. The overall framework consists of two parallel networks with different weights initialization. The whole process can be considered as three modules: segmentation module, prototypes modelling and contrastive learning, prototype-anchored assignment. Concretely, the segmentation module aims to produce per-pixel embedding with the dimension d, $\mathcal{F} \in \mathbb{R}^{d \times H \times W}$. Then, the transformer decoder and $N_p$ learnable query embeddings which are shared between two parallel networks are employed in modelling prototypes, i.e., mask embeddings $\mathcal{M} \in \mathbb{R}^{d \times N_P}$. By element-wise multiplication, the probability map of prototypical classification $\mathcal{P}_m \in \mathbb{R}^{N_p \times H \times W}$ is produced and applied to guide the positive and negative sampling for contrastive learning. Finally, we infer through a pre-defined prototype-anchored assignment operation $\mathcal{P}_c = \xi(\mathcal{P}_m)$.**

are excluded from training. Besides, those pixels are prone to being classified into under-performing categories in datasets with long-tailed distribution.

Motivated by above-mentioned analyses, we propose to mine more semantic information and achieve pixel correlation in each category space via unsupervised clustering. Our key idea is to model multiple prototypes in each category space, with which the distribution of outliers can be better characterized. Many existing approaches implement clustering to form centroids via some non-parametric strategies (e.g., k-means). Here, inspired by MaskFormer [8], we employ a transformer decoder to model prototypes and further cluster pixels based on the feature distance. Specifically, a feature map extracted by the backbone is fed into the multi-layer transformer decoder alongside $N_p$ learnable query embeddings to produce corresponding mask embeddings $\mathcal{M} \in \mathbb{R}^{d \times N_P}$, where $d$ is the dimension of pixel embedding and $N_p$ is the number of prototypes. Each mask embedding is theoretically equivalent to the cluster centroid, i.e., the prototype. We obtain the per-pixel prototype prediction via a dot product between pixel embedding and mask embedding, denoted as $p_i^m = \mathcal{F}[:, i]^{\mathrm{T}} \cdot \mathcal{M}, i \in \mathbb{R}^{HW}$. It is equal to measure the cosine similarity of pixels and prototypes. Here, $p_i^m$ with the dimension $N_p$ represents the probability of the $i$-th pixel belonging to the $k^{'}$-th($k^{'} \in \{1, 2, \cdots, N_p\}$) prototype.

Accordingly, we implement clustering to divide the category space into several semantic parts. Following CPS [7], we implement self-supervision to the prototypes probability map $\mathcal{P}_m$. Specifically, $\hat{y}^{(i,k^{'})} = \mathrm{argmax}_{k^{'} \in \{1,2,\cdots,N_p\}} p_i, p_i \in \mathcal{P}_m$ from branch 1 is taken as pseudo labels for training branch 2, and vice versa. It can be formulated as:

$$\mathcal{L}_{sp} = \frac{1}{HW} \sum_{i=1}^{HW} \ell_{ce}\left(\hat{\boldsymbol{y}}_1^{(i,k^{'})}, \boldsymbol{p}_2^{(i,k^{'})}\right) + \ell_{ce}\left(\hat{\boldsymbol{y}}_2^{(i,k^{'})}, \boldsymbol{p}_1^{(i,k^{'})}\right), \quad (6)$$

where $\mathcal{L}_{sp}$ is formed in a prototype-wise manner and differs from the counterpart of CPS which is category-wise.

After obtaining prototypes, we implement contrastive learning among them. Specifically, we first project the original feature map $\mathcal{F}$ into a low dimensional feature map $\mathcal{F}_p$ via a non-linear projector $\Phi$. Then, we depend on the prototype prediction $\mathcal{P}_m$ to guide the sampling of positive and negative pairs as shown in Fig. 3.

**Positive sampling.** With pseudo labels $\hat{y}^{(i,k^{'})}$, we take those pixels belonging to the same prototypes from both branch 1 and branch 2 as positive samples: $\mathcal{V} = \left\{\boldsymbol{v}_i^k \mid \hat{y}^{(i,k)} = k\right\}_{k=1}^{N_p}$. Unlike previous works that put pixels of the same category together as the positive samples, we aim to encourage the network to learn a compact structure in each cluster space.

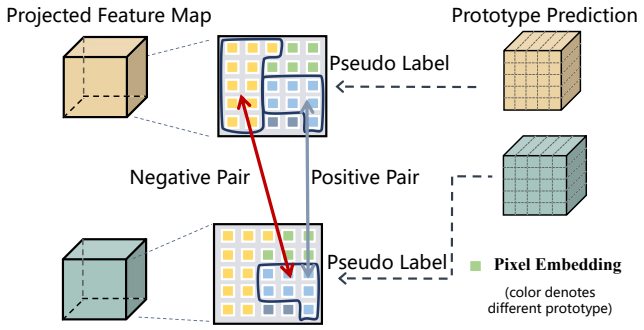



**Figure 3: Prototypical contrastive learning. We employ the prototype prediction as pseudo labels to guide the sampling of positive and negative pairs. Pixel embeddings with the same color belong to the same prototype.**

**Negative sampling.** There are two strategies for negative sampling, the first one is to sample pixels from different categories as negative pairs. For the second one, apart from those pixels sampled via the above strategy, we also take the part belonging to the same category but different prototypes as negative pairs with the insight to enforce the decision boundary lying in a low density region. In our method, we choose the second strategy, and the negative pair can be formulated as: $\boldsymbol{O} = \left\{ \boldsymbol{o}_i^k \mid \hat{y}^{(i,k)} \neq k \right\}_{k=1}^{N_p}$.

Afterwards, pixel-level contrastive learning is applied based on the positive and negative pairs sampled above, which can be defined as:

$$\mathcal{L}_{\text{contra}} = \sum_{k=1}^{N_p} \sum_{i=1}^{n} -\log \frac{\exp\left(\boldsymbol{v}_i^k \cdot \boldsymbol{v}_{i'}^k / \tau\right)}{\exp\left(\boldsymbol{v}_i^k \cdot \boldsymbol{v}_{i'}^k / \tau\right) + \sum_{j=0}^{r} \exp\left(\boldsymbol{v}_i^k \cdot \boldsymbol{o}_j^k / \tau\right)}, \tag{7}$$

where $i \neq i^{'}$, $n$ and $r$ denote the numbers of positive and negative samples, respectively, and $\tau$ is a temperature hyper-parameter.

## 3.4 Prototype-Category Assignment

Given the prototypical prediction map $\boldsymbol{\mathcal{P}}_m$, the prediction of categories can be obtained by the summation of corresponding prototypes:

$$p_c^{(i,k)} = f\Big(\sum_{n^{'} \in N} p_m^{(i,k^{'})}(n^{'})\Big) \tag{8}$$

$$f(x) = \text{softmax}(\psi(\text{sigmoid}(x))), \tag{9}$$

where $N$ represents the corresponding prototypes assigned to category $k$ (we assign the same amount of prototypes for each category, so $N_p = n \times C$), and $\psi(t) = \frac{t}{1-t}$. Eq. 8 indicates that we align prototypes to categories in a fixed many-to-one form. Given the final prediction, the supervised loss in Eq. 3 is applied for labeled data.

It is worth noting that our prototype-wise prediction $\boldsymbol{\mathcal{P}}_m$ is different from MaskFormer's mask prediction which is category-level or instance-level on account of the disparate inference implementation. Our method also differs from the superpixel-based approaches [19, 31] which focus on pixels correlation in local regions.

## 3.5 Prototype Activation Regularization

Due to the under-constrained prototype modelling together with the prototype-anchored assignment strategy, the proposed model is prone to giving trivial solutions. In our method, we apply prototype activation regularization to mitigate the problem. Specifically, in each iteration, we randomly sample one of prototypes with the same category label to produce the final inference result, which can be formulated as:

$$p_c^{(i,k)}[j] = p_m^{(i,k^{'})}[nj + \text{random(n)}], j \in [1, c], \tag{10}$$

where $n$ represents the number of prototypes per category. We penalize those inactive prototypes using the cross-entropy loss with both labels and pseudo labels:

$$\mathcal{L}_r = \frac{1}{HW} \sum_{i=1}^{HW} \ell_{ce}\left(\boldsymbol{y}^{(i,k)}, \boldsymbol{p}_c^{(i,k)}\right). \tag{11}$$

**Data Augmentation.** Besides the commonly used data augmentation strategies such as random flipping and scaling, we also apply the CutMix [13] augmentation in our approach following CPS [7]. Unless otherwise specified, results reported in this paper are obtained under the settings with CutMix.

# 4 EXPERIMENTS

## 4.1 Setup

**Datasets.** PASCAL VOC 2012 [12] is a widely-used semantic segmentation dataset including annotations of 20 object classes and 1 background class. There are 1,464, 1,449, and 1,456 images for training, validation, and testing, respectively. And the full augmented set [14] includes 10,582 images and we use it as training set for the common practice. Cityscapes [9] consists of 5,000 images with finely annotations of 19 semantic classes. There are 2,975, 500, and 1,525 images for training, validation, and testing, respectively.

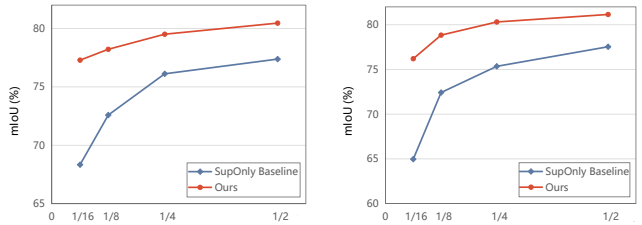
Following the common practice, 1/2, 1/4, 1/8, and 1/16 training images are randomly sampled as the labeled data and the remaining images are regarded as unlabeled data. We set the crop size as $512 \times 512$ in PASCAL VOC 2012 and the crop size as $800 \times 800$ in Cityscapes.

**Evaluation.** For all partition protocols, we adopt mean Intersection-over-Union (mIoU) as the evaluation metric and report results on the PASCAL VOC 2012 val set and Cityscapes val set.

**Implementation details.** We use ResNet-101 pre-trained on ImageNet [23], and DeepLab v3+ [5] as our backbone and segmentation module separately. We initialize the weights of two parallel networks differently except the backbone. Following CPS[7], we use the stochastic gradient descent (SGD) optimizer with the weight decay 0.0005 and the momentum 0.9. We set the initial learning rate as 0.01 for PASCAL VOC 2012 and 0.02 for Cityscapes, respectively, and employ a poly learning rate policy where the initial learning rate is multiplied by $\left(1 - \frac{iter}{max\_iter}\right)^{0.9}$. In addition, we adopt Sync-BN [18] for stable training.

## 4.2 Quantitative Results

In Tab. 1 and Tab. 2, we report our results on Cityscapes val set and PASCAL VOC 2012 val set under different partition protocols




(a) PASCAL VOC 2012 (b) Cityscapes

**Figure 4: Comparison with the supervised baseline on (a) PASCAL VOC 2012 and (b) Cityscapes under 1/16, 1/8, 1/4, 1/2 partition protocols.**

and show improvements over the baseline. Besides, we make comparison with several recently-proposed semi-supervised semantic segmentation methods, including Mean Teacher (MT) [37], Cross-Consistency Training (CCT) [30], Guided Collaborative Training (GCT) [20], CutMix [13], Cross Pseudo Supervision (CPS) [7], and Adaptive Equalization Learning (AEL) [17]. For fair comparison, the supervised baseline and all these methods are implemented using ResNet-101 and DeepLabv3+. The results of all other methods are from AEL [17] except CPS [7].

**Comparison with the supervised baseline.** in Fig. 4, we compare our method with the supervised only baseline. On the PASCAL VOC 2012 dataset, our method outperforms the baseline by +8.95%, +5.63%, +3.39%, and +3.08% under 1/16, 1/8, 1/4, and 1/2 partition protocols, respectively. On the Cityscapes dataset, our method achieves the improvements by +11.23%, +6.42%, +4.96%, and +3.61%, respectively. There is greater superiority over the baseline with fewer training data available, which verifies the effectiveness of our semi-supervised semantic segmentation paradigm.

**Comparison with state-of-the-art methods.** Compared to recent semi-supervised semantic segmentation approaches, our method achieves state-of-the-art performance both on Cityscapes and PASCAL VOC 2012 under various partition protocols as shown in Tab. 1 and Tab. 2. We also report the gain of performance over the semi-supervised baseline in brackets, i.e., CPS[7] which employs cross pseudo supervision as the consistency regularization strategy. Our method consistently promotes the baseline, achieving the improvements of +2.81%, +1.78%, +1.83%, +1.82% on Cityscapes and +1.48%, +1.22%, +1.1%, +0.94% on PASCAL VOC 2012 under various partition. It turns out that the network can benefit from our engineered prototypes modelling and contrastive learning.

Besides, we also make comparison on PASCAL VOC 2012 with only 732, 366, 183, 92 images available, respectively, in Tab. 3. Our method consistently outperforms the baseline by +1.75%, +1.74%, +1.9%, +1.89% under various partition protocols. Our method still performs better with few supervision.

## 4.3 Qualitative Results

Fig. 5 visualizes the activation of prototypes, demonstrating that prototypes capture some certain discriminative patterns. In Fig. 5(a), some prototypes focus on tiny parts like the feet and belly of the bird. Fig. 5(b) shows that prototypes can capture local and global information separately. As shown in Fig. 5(c), the activation maps

| Method | 1/16(186) | 1/8(372) | 1/4(744) | 1/2(1488) |
|---|---|---|---|---|
| MT [37] | 68.05 | 73.56 | 76.66 | 78.39 |
| CCT [30] | 69.32 | 74.12 | 75.99 | 78.10 |
| CutMix [13] | 72.13 | 75.83 | 77.24 | 78.95 |
| GCT [20] | 66.75 | 72.66 | 76.11 | 78.34 |
| CPS [7] | 74.72 | 77.62 | 79.21 | 80.21 |
| AEL [17] | 75.83 | 77.90 | 79.01 | 80.28 |
| PPS(ours) | **76.20**(+1.48) | **78.84**(+1.22) | **80.31**(+1.1) | **81.15**(+0.94) |

**Table 1: Comparison with state-of-the-art methods on Cityscapes val set under different partition protocols. All the methods are based on DeepLabv3+ with ResNet-101.**

| Method | 1/16(662) | 1/8(1323) | 1/4(2646) | 1/2(5291) |
|---|---|---|---|---|
| MT [37] | 71.29 | 73.33 | 76.61 | 78.08 |
| CCT [30] | 71.86 | 73.78 | 76.51 | 77.40 |
| CutMix [13] | 73.56 | 73.96 | 77.58 | 78.12 |
| GCT [20] | 70.90 | 73.29 | 76.66 | 77.98 |
| CPS [7] | 74.48 | 76.44 | 77.68 | 78.64 |
| AEL [17] | 77.20 | 77.57 | 78.06 | 80.29 |
| PPS(ours) | **77.29** (+2.81) | **78.22**(+1.78) | **79.51**(+1.83) | **80.46**(+1.82) |

**Table 2: Comparison with state-of-the-art methods on PASCAL VOC 2012 val set under different partition protocols. All the methods are based on DeepLabv3+ with ResNet-101.**

| Method | #labeled images | | | |
|---|---|---|---|---|
| | 732 | 366 | 183 | 92 |
| CCT [30] | 62.10 | 58.80 | 47.60 | 33.10 |
| MT [37] | 69.16 | 63.01 | 55.81 | 48.70 |
| GCT [20] | 70.67 | 64.71 | 54.98 | 46.04 |
| CutMix [13] | 69.84 | 68.36 | 63.20 | 55.58 |
| PseudoSeg [49] | 72.41 | 69.14 | 65.50 | 57.60 |
| CPS [7] | 75.88 | 71.71 | 67.42 | 64.07 |
| PPS(ours) | **77.63**(+1.75) | **73.45**(+1.74) | **69.32**(+1.9) | **65.96**(+1.89) |

**Table 3: Comparison for few-supervision scenarios. We report the results of all other methods from CPS [7]**

of some prototypes are complementary. Specially, some prototypes concentrate on contours or indistinguishable regions, which typically contain pixels that are hard to be classified. Those pixels are prone to be away from centroids in the feature space, such as the mis-classified outliers in Fig. 1(a). With extra prototypes built in the category space, outliers can be clustered and assigned to a correct semantic label.

Furthermore, we compare the qualitative results of our method with the supervised baseline and AEL [17]. As we can see from Fig. 6, our method achieves a more precise classification. We also visualize the pixel representation using T-SNE [38] as shown in Fig. 7. For better visualization, we only present five categories. Fig. 7(a) shows the distributions of pixel representation for the semi-supervised

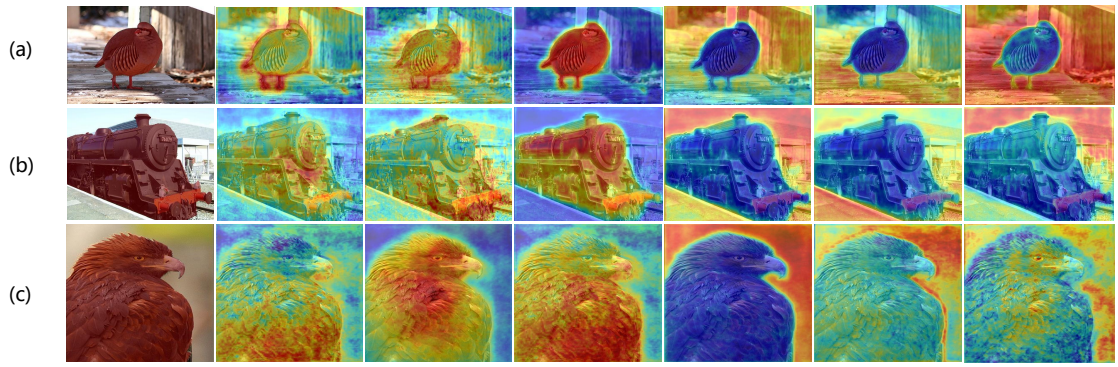



**Figure 5: Activation map of prototypes. We only visualize three prototypes per category here. The second to fourth columns refer to prototypes assigned to objects and the fifth to seventh columns refer to prototypes belonging to background category.**


**Figure 6: Qualitative results on PASCAL VOC 2012 and comparison with the supervised only baseline and AEL [17]. Orange bounding boxes show some regions that our method can make a better prediction.**

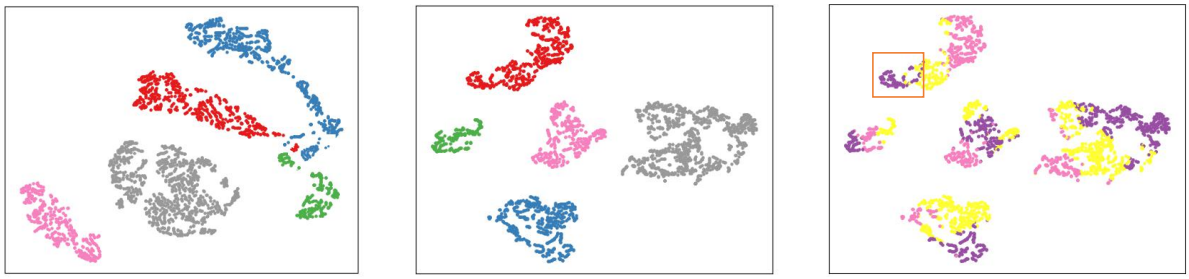
baseline, where pixel features of the categories in red, blue, and green are interlaced and dispersive. It can be observed from Fig. 7(b) that, through the modelling of prototypes and the implementation of prototypical contrastive learning, the distributions of pixel representation in the category space are compact and the decision boundaries lie in low-density regions. Fig. 7(c) demonstrates that multiple prototypes can capture and characterize the distribution of outliers and are robust to intra-class variation.

## 4.4 Ablation Study

To validate the effectiveness of each module of our method, we conduct ablation studies for these modules, including the multi-prototype modelling (MPM), prototypical contrastive learning (PCL), and prototype activation regularization (PAR). Experiments are conducted on PASCAL VOC 2012 under the 1/16 partition protocol if not specified. We report the results on val set as shown in Tab. 4.

**Effectiveness of multi-prototype modelling.** Benefiting from the utilization of multi-prototype modelling, the network achieves a performance lift of +1.2% (see Tab. 4). Besides, the ablation study conducted for the number of prototypes per category in Tab. 5 demonstrates that the gain of performance is brought by our multi-prototype strategy but not the introduction of the transformer decoder. It turns out that the multiple sub-center enhances the representative ability and makes our network perform better in pseudo segmentation.

**Effectiveness of prototypical contrastive learning.** As shown in Tab. 4, our prototypical contrastive learning (PCL) based on pseudo prediction further boosts the performance by +0.45% and +0.82% under the settings with or without regularization, respectively. This is mainly due to the fact that PCL enforces the network separating different categories from each other and the decision boundary



(a) Baseline (b) Ours (category space) (c) Ours (prototype space)

**Figure 7: T-SNE [38] visualization of pixel representation with five classes in view. (a) and (b) present the distribution of baseline and our method in category space. (c) shows the result in prototype space under three prototypes per category setting.**

| MPM | PCL | PAR | mIoU |
|---|---|---|---|
| | | | 74.48 |
| ✓ | | | 75.68 |
| ✓ | ✓ | | 76.13 |
| ✓ | | ✓ | 76.47 |
| ✓ | ✓ | ✓ | **77.29** |

**Table 4: Ablation study on the effectiveness of different modules in our method.**

lying in low-density regions. Moreover, it reinforces the learning of discriminative patterns.

**Effectiveness of prototype activation regularization.** Further improvements can be observed with the prototype activation regularization by +1.16% (see Tab. 4). This module penalizes those inactive prototypes and prevents the network from degradation.

## 4.5 Parameter Study

We evaluate the performance of our methods with different numbers of prototypes per category and report results in Tab. 5 and Tab. 6. When $n = 1$, there is only a single centroid in each category space and the network degrades to the baseline. As we can see, the increase of the prototype number can bring improvements due to the stronger representation capacity. However, too many prototypes are redundant and may boot the pressure for the network to distinguish among different prototypes. In addition, it will lead to sparse distribution in the feature space. Therefore, the best hyper-parameter is $n = 5$ for PASCAL VOC 2012 and $n = 7$ for Cityscapes. Unless otherwise specified, our method is configured with the above-mentioned optimal hyper-parameters by default to conduct other experiments.

In addition, Tab. 7 investigates the influences of the weights of different losses, including the contrastive learning loss, the self-supervision loss, and the prototype activation regularization term. We can see that the self-supervision loss is less influential for the segmentation performance. Experimentally, We choose the settings of $\lambda_1 = 0.5$, $\lambda_2 = 1.5$, $\lambda_3 = 0.01$ as our default hyper-parameters.

## 5 CONCLUSION

In this paper, we investigated the cause of noisy pseudo labels in semi-supervised semantic segmentation and proposed to build multiple prototypes in the category space to mitigate effects of outliers.

| num_prototypes/category | 1 | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|
| mIoU | 74.29 | 75.94 | **77.29** | 77.03 | 76.74 |

**Table 5: Parameter study 1 conducted on PASCAL VOC 2012 under 1/16 partition: the number of prototypes per category.**

| num_prototypes/category | 1 | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|
| mIoU | 74.68 | 75.12 | 75.93 | **76.20** | 76.04 |

**Table 6: Parameter study 2 conducted on Cityscapes under 1/16 partition: the number of prototypes per category.**

| $\lambda_1$ | mIoU | $\lambda_2$ | mIoU | $\lambda_3$ | mIoU |
|---|---|---|---|---|---|
| 0.2 | 76.47 | 0.5 | 75.94 | 0.005 | 76.67 |
| 0.5 | **76.95** | 1 | 76.87 | 0.01 | **76.94** |
| 0.8 | 76.21 | 1.5 | **77.03** | 0.05 | 76.02 |
| 1.5 | 74.57 | 2 | 76.53 | 0.1 | 75.78 |
| 2 | 74.65 | 3 | 76.35 | 0.2 | 74.52 |

**Table 7: Ablation study on hyper-parametesr, $\lambda_1$: the weight of $\mathcal{L}_{contra}$, $\lambda_2$: the weight of $\mathcal{L}_{sp}$, $\lambda_3$: the weight of $\mathcal{L}_r$**

Besides, we implemented contrastive learning in a prototype-wise manner to learn a compact distribution of feature space and the distinct boundary between prototypes. We also employed prototype activation regularization to keep the proposed model from trivial solutions. Experimental results showed that our method achieved state-of-the-art performance on several widely-used datasets. Furthermore, the proposed prototypes modelling is a general strategy for semi-supervised semantic segmentation frameworks. In the future, we are planing to apply our method to handle other tasks, such as supervised or weakly supervised semantic segmentation.

## ACKNOWLEDGMENTS

This work was supported by Beijing Nova Program of Science and Technology (Grant No.: Z191100001119077), Project 2020BD020 supported by PKU-Baidu Fund, and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

# REFERENCES


[1] Inigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana C Murillo. 2021. 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*. 8219–8228.

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

[3] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. 2020. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *European Conference on Computer Vision*. Springer, 695–714.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* 40, 4 (2017), 834–848.

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

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

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

[8] Bowen Cheng, Alex Schwing, and Alexander Kirillov. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 17864–17875.

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

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[11] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. 2022. Weakly supervised semantic segmentation by pixel-to-prototype contrast. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4320–4329.

[12] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision* 111, 1 (2015), 98–136.

[13] Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. 2019. Semi-supervised semantic segmentation needs strong, varied perturbations. *arXiv preprint arXiv:1906.01916* (2019).

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

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

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670* (2018).

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

[18] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

[19] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. 2018. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 352–368.

[20] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. 2020. Guided collaborative training for pixel-wise semi-supervised learning. In *European conference on computer vision*. Springer, 429–445.

[21] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson WH Lau. 2019. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6728–6736.

[22] Jongmok Kim, Jooyoung Jang, and Hyunwoo Park. 2020. Structured consistency loss for semi-supervised semantic segmentation. *arXiv preprint arXiv:2001.04647* (2020).

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[24] Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3. 896.

[25] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. 2020. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966* (2020).

[26] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J Davison. 2021. Bootstrapping semantic segmentation with regional contrast. *arXiv preprint arXiv:2104.04465* (2021).

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[28] Robert Mendel, Luis Antonio de Souza, David Rauber, João Paulo Papa, and Christoph Palm. 2020. Semi-supervised segmentation based on error-correcting supervision. In *European Conference on Computer Vision*. Springer, 141–157.

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

[30] Yassine Ouali, Céline Hudelot, and Myriam Tami. 2020. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12674–12684.

[31] Xiaofeng Ren and Jitendra Malik. 2003. Learning a classification model for segmentation. In *Computer Vision, IEEE International Conference on*, Vol. 2. IEEE Computer Society, 10–10.

[32] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. 2021. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329* (2021).

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[34] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems* 33 (2020), 596–608.

[35] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. 2021. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7262–7272.

[36] Rui Sun, Yihao Li, Tianzhu Zhang, Zhendong Mao, Feng Wu, and Yongdong Zhang. 2021. Lesion-aware transformers for diabetic retinopathy grading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10938–10947.

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

[38] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[40] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. 2019. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825* (2019).

[41] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2021. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5463–5474.

[42] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3733–3742.

[43] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6210–6219.

[44] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6881–6890.

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

[46] Tianfei Zhou, Wenguan Wang, Ender Konukoglu, and Luc Van Gool. 2022. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2582–2593.

[47] Yi Zhu, Zhongyue Zhang, Chongruo Wu, Zhi Zhang, Tong He, Hang Zhang, R Manmatha, Mu Li, and Alexander Smola. 2020. Improving semantic segmentation via self-training. *arXiv preprint arXiv:2004.14960* (2020).
[48] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in neural information processing systems* 33 (2020), 3833–3845.
[49] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. 2020. Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713* (2020).