# Hunting Sparsity: Density-Guided Contrastive Learning for Semi-Supervised Semantic Segmentation

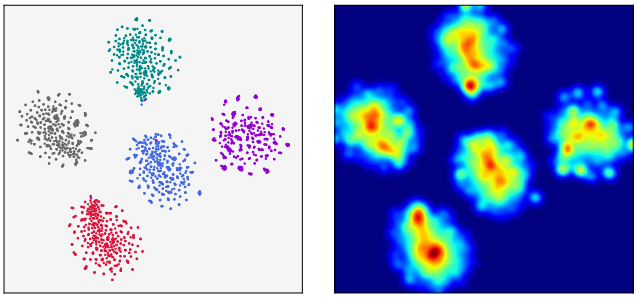
Xiaoyang Wang[1,2,3] Bingfeng Zhang[4] Limin Yu[1] Jimin Xiao[1*]
[1]XJTLU [2]University of Liverpool [3]Metavisioncn [4]China University of Petroleum (East China)
wangxy@liverpool.ac.uk, bingfeng.zhang@upc.edu.cn, {limin.yu, jimin.xiao}@xjtlu.edu.cn

## Abstract

*Recent semi-supervised semantic segmentation methods combine pseudo labeling and consistency regularization to enhance model generalization from perturbation-invariant training. In this work, we argue that adequate supervision can be extracted directly from the geometry of feature space. Inspired by density-based unsupervised clustering, we propose to leverage feature density to locate sparse regions within feature clusters defined by label and pseudo labels. The hypothesis is that lower-density features tend to be under-trained compared with those densely gathered. Therefore, we propose to apply regularization on the structure of the cluster by tackling the sparsity to increase intra-class compactness in feature space. With this goal, we present a Density-Guided Contrastive Learning (DGCL) strategy to push anchor features in sparse regions toward cluster centers approximated by high-density positive keys. The heart of our method is to estimate feature density which is defined as neighbor compactness. We design a multi-scale density estimation module to obtain the density from multiple nearest-neighbor graphs for robust density modeling. Moreover, a unified training framework is proposed to combine label-guided self-training and density-guided geometry regularization to form complementary supervision on unlabeled data. Experimental results on PASCAL VOC and Cityscapes under various semi-supervised settings demonstrate that our proposed method achieves state-of-the-art performances. The project is available at* https://github.com/Gavinwxy/DGCL.

## 1. Introduction

Semantic segmentation, as an essential computer vision task, has seen significant advances along with the rise of deep learning [4, 30, 46]. Nevertheless, training segmentation models requires massive pixel-level annotations which can be time-consuming and laborious to obtain. Therefore, semi-supervised learning is introduced in semantic segmentation and is drawing growing interest. It aims to design a label-efficient training scheme with limited annotated data to allow better model generalization by leveraging additional unlabeled images.



(a) Feature clusters (b) Local density estimation

Figure 1. Illustration of feature density within clusters. (a) Pixel-level features of 5 classes extracted from PASCAL VOC 2012 [12] with prediction confidence over 0.95. (b) Feature density estimated by the averaged distance of 16 nearest neighbor features.

The key in semi-supervised semantic segmentation lies in mining extra supervision from unlabeled samples. Recent studies focus on learning consistency-based regularization [13, 29, 32, 51] and designing self-training pipelines [15, 19, 43, 44]. Inspired by the advances in representation learning [17, 24], another line of works [28, 40, 47–49] introduce contrastive learning on pixel-level features to enhance inter-class separability. Though previous works have shown effectiveness, their label-guided learning scheme solely relies on classifier knowledge, while the structure information of feature space is under-explored. In this work, we argue that effective supervision can be extracted from the geometry of feature clusters to complement label supervision.

Feature density, measured by local compactness, has shown its potential to reveal feature patterns in unsupervised clustering algorithms such as DBSCAN [11]. The density-peak assumption [33] states that cluster centers are more likely located in dense regions. Inversely, features in

*Corresponding author.

sparse areas tend to be less representative within the cluster, so they require extra attention. The sparsity exists even within features that are confidently predicted by the classifier. As shown in Fig. 1, pixel-level features are extracted on labeled and unlabeled images from 5 classes in PASCAL VOC 2012 dataset, all with high prediction confidence. Evident density variation still exists within each cluster, indicating varying learning difficulty among features, which the classifier fails to capture.

In this work, we propose a learning strategy named Density-Guided Contrastive Learning (DGCL) to mine effective supervision from cluster structure of unlabeled data. Specifically, we initialize categorical clusters based on labeled features and enrich them with unlabeled features which are confidently predicted. Then, sparsity hunting is conducted in each in-class feature cluster to locate low-density features as anchors. Meanwhile, features in dense regions are selected to approximate the cluster centers and serve as the positive keys. Then, feature contrast is applied to push the anchors toward their positive keys, explicitly shrinking sparse regions to enforce more compact clusters.

The core of our method is feature density estimation. We measure local density by the average distance between the target feature and its nearest neighbors. For robust estimation, categorical memory banks are proposed to break the limitation on mini-batch, so in-class density can be estimated in a feature-to-bank style where class distribution can be approximated globally. When building the nearest neighbor graph, densities estimated by fewer neighbors tend to focus on the local region, which prevents capturing true cluster centers. On the other hand, graphs with too many neighbors cause over-smoothed estimation, which harms accurate sparsity mining. Therefore, we propose multi-scale nearest neighbor graphs to determine the final density by combining estimations from graphs of different sizes.

We evaluate the proposed method on PASCAL VOC 2012 [12] and Cityscapes [8] under various semi-supervised settings, where our approach achieves state-of-the-art performances. Our contributions are summarized as follows:

- We propose a density-guided contrastive learning strategy to tackle semi-supervised semantic segmentation by mining effective supervision from the geometry in feature space.
- We propose a multi-scale density estimation module combined with dynamic memory banks to capture feature density robustly.
- We propose a unified learning framework consisting of label-guided self-training and density-guided feature learning, in which two schemes complement each other. Experiments show that our method achieves state-of-the-art performances.

## 2. Related Works

**Semi-Supervised Learning.** Semi-supervised learning (SSL) is a well-studied topic and recent researches can be summarized in two branches which are entropy minimization [14, 41, 50] and consistency regularization [2, 21, 26, 35, 37, 38, 45]. Entropy minimization aims to assign pseudo labels on unlabeled data with knowledge from provided labels and then re-train the model on the combined dataset. Consistency regularization focuses on designing perturbations on input data and enforcing the model to have similar predictions on the same data under different augmentations. MixMatch [2] performs label-guessing on unlabeled data as the average predictions on its multiple augmented versions. FixMatch [37] proposes to generate pseudo labels from confident predictions on weakly augmented images and use them to supervise the prediction of its strongly augmented version. Moreover, FlexMatch [45] notices that training difficulty varies among classes, so it proposes class-dependent confidence thresholds for pseudo-label filtering.

**Semi-Supervised Semantic Segmentation.** The success of SSL in image classification inspires research on semantic segmentation. Early works [20, 31] apply generative models combined with adversarial training to generate high-quality pseudo labels. Recent works pay attention to consistency regularization to design sophisticated data perturbation strategies. French et al. [13] introduces image-level strong augmentations including CutOut and CutMix. Feature perturbations [29, 32] are proposed to explore consistency in feature level. Then, CPS [7] applies two models with the same architecture but different initialization to create pseudo labels for each other to conduct cross-pseudo supervision. PseudoSeg [51] adopts grad-CAM based on image-level labels to enhance the quality of pseudo labels on unlabeled data. Motivated by contrastive learning, a series of work [1, 25, 28, 40, 47–49] focus on enforcing relations among representations in feature space. Alonso et al. [1] introduces a memory bank to store high-quality class features and perform positive-only contrastive learning. PC$^2$Seg [48] proposes to align features of the same pixel under different augmentations while such alignment is also introduced on the same pixel under different context [25]. To take full advantage of pseudo labels, U$^2$PL [40] transforms unreliable pseudo labels into effective supervision for negative contrast. The methods above explore supervision on unlabeled data solely relying on the categorical information provided by the model classifier. Unlike previous works, our approach probes the geometry of feature clusters and extracts effective supervision from carefully selected anchors and keys.

**Contrastive Learning.** Image-level contrastive learning has shown promising results in self-supervised learning. Typical methods MoCo [17] and SimCLR [6] propose

pipelines to learn augmentation-invariant representations which show superior performance over their supervised pre-training counterparts. SupCon [24] points out that contrastive learning can also benefit supervised learning with label-guided feature contrast. In our work, we adopt the spirit of contrastive learning to guide the learning of pixel-level features. Apart from label information, feature density is transformed as training signals to boost the model performance at few cost.

# 3. Methodology

## 3.1. Overview

Given a small labeled dataset $D_l = \{(x_i^l, y_i^l)\}$, and massive unlabeled images $D_u = \{x_i^u\}$, semi-supervised semantic segmentation aims to mine effective supervision from unlabeled images with the help of limited annotations, to achieve comparable segmentation performance to its fully supervised counterpart.

Fig. 2 gives an overview of the proposed method. The framework is built upon a teacher-student network described in Section 3.2. It consists of two main training strategies: a) Label-guided training, where model is supervised by annotations and pseudo labels to learn pixel-level category assignment, which is described in Section 3.3. b) Density-guided contrastive learning to regularize the structures of clusters in feature space, which is described in Section 3.4. The core of our method is the density estimation process, which is shown in grey region in Fig. 2 (b) and described in Section 3.4.1 in details. In each mini-batch, we evaluate density to locate features in sparse regions as anchors. Meanwhile, representative features in the memory bank are extracted to serve as positive keys. Contrastive loss is minimized by pushing low-density anchors towards the approximated cluster center, dynamically shrinking the cluster volume and increasing in-cluster compactness.

## 3.2. Teacher-Student Network

We build our framework on a teacher-student network. The student network $f_\theta$, parameterized by $\theta$, is optimized by gradient descent while the teacher model $f_{\theta'}$ is updated as the exponential moving average (EMA) of the student as:

$$\theta' = \alpha\theta' + (1-\alpha)\theta, \tag{1}$$

where $\alpha \in [0, 1]$ controls the pace of update. For unlabeled images, pseudo labels are generated from teacher model predictions as robust supervision to guide the training. The pseudo labels provide class information to learn pixel classification, and also they help to extract feature maps in each mini-batch to serve for in-class feature contrast. The student model is optimized by a unified loss function:

$$\mathcal{L} = \mathcal{L}_{sup} + \lambda_{unsup}\mathcal{L}_{unsup} + \lambda_{contra}\mathcal{L}_{contra}, \tag{2}$$

where $\mathcal{L}_{sup}$ is the supervised loss and $\mathcal{L}_{unsup}$ is the self-training loss on unlabeled data. $\mathcal{L}_{contra}$ is the density-guided contrastive loss. Losses are described in detail in the following sections.

## 3.3. Learning Label-Guided Pixel Classification

**Learning with Ground Truth.** Given a batch of $N_l$ labeled images $\{(x_i^l, y_i^l)\}_{i=1}^{N_l} \in D_l$, cross-entropy loss is applied to supervise pixel-level classification:

$$\mathcal{L}_{sup} = -\frac{1}{N_l}\frac{1}{HW}\sum_{i=1}^{N_l}\sum_{j=1}^{HW}\ell_{ce}(p_{ij}^l, y_{ij}^l), \tag{3}$$

where $p_{ij}^l = f_\theta(x_{ij}^l)$ is the probability prediction by the student network on $j$-th pixel in $i$-th labeled image and $y_{ij}^l$ the corresponding pixel annotations. $W$ and $H$ represent the image width and height.

**Learning with Pseudo Labels.** For unlabeled images $\{x_i^u\}_{i=1}^{N_u}$, we adopt predictions from teacher model $\hat{p}^u = f_{\theta'}(x^u)$ to generate online pseudo labels as:

$$y_{ij}^u = \arg\max_c \hat{p}_{ij}^u. \tag{4}$$

A progressive label filtering strategy is applied to pseudo labels to guarantee robust training. Entropy is adopted to measure prediction certainty. In early training, we only select most certain teacher predictions as pseudo labels to avoid potential label noise. As training proceeds, more predictions can be involved to expand effective supervision. Entropy is calculated for each pixel prediction as follows:

$$\mathcal{H}(\hat{p}_{ij}) = -\sum_{c=1}^{C}\hat{p}_{ij}^c \log \hat{p}_{ij}^c. \tag{5}$$

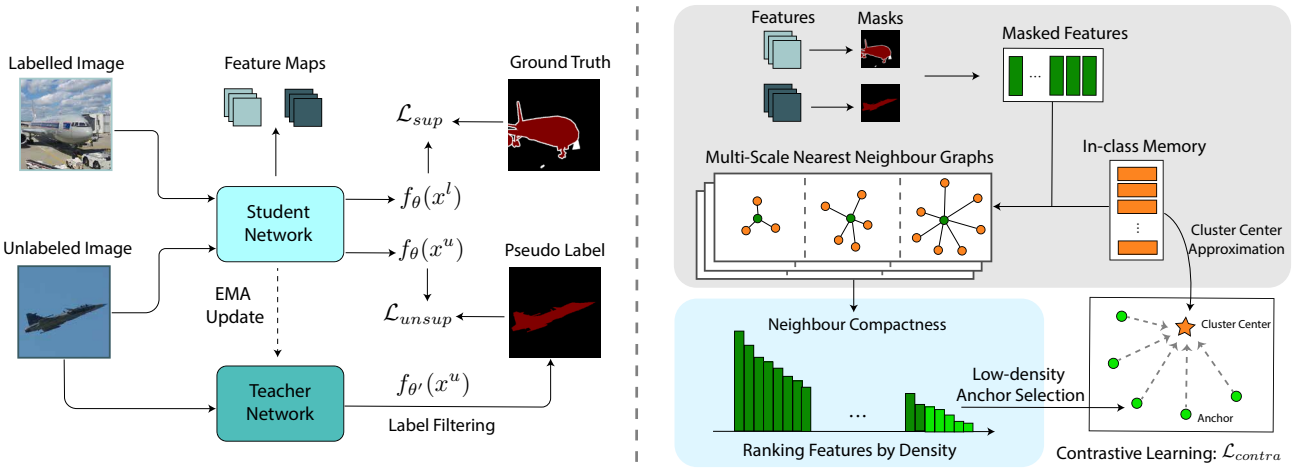
For iteration $t$, predictions with top $\beta_t$ percentile entropy values are filtered out in each mini-batch. $\beta_t$ are decreased linearly with $t$ from $\beta_0$ to 0 as $\beta_t = \beta_0(1 - t/T)$ where $T$ is total training iterations. The threshold value $\eta_{\beta_t}$ is extracted from entropy values of current prediction batch $\{\mathcal{H}(\hat{p}^u)\}$ as the $\eta_{\beta_t}$-th percentile value. Then, the student model can be supervised by the filtered pseudo labels with loss:

$$\mathcal{L}_{unsup} = -\frac{1}{N_u}\frac{1}{HW}\sum_{i=1}^{N_u}\sum_{j=1}^{HW}\ell_{ce}(p_{ij}^u, y_{ij}^u)\cdot\mathbb{1}(\mathcal{H}(\hat{p}_{ij}^u) < \eta_{\beta_t}), \tag{6}$$

where $p^u = f_\theta(x^u)$ are student predictions on unlabeled images.

## 3.4. Learning Density-Guided Feature Contrast

The student segmentation model contains a feature encoder $h$ which encodes pixels into features $v_{ij} = h(x_{ij})$, before being classified by classifier. Following [6], a projection head $g$ is attached to map extracted features into a




(a) Learning Label-Guided Pixel Classification

(b) Learning Density-Guided Feature Contrast

Figure 2. Overview of the proposed unified learning framework (best viewed in color). (a) shows the label-guided training on the teacher-student network. The upper branch depicts the supervised training process on student model $f_\theta$ to minimize $\mathcal{L}_{sup}$, while in the lower branch, teacher model $f_{\theta'}$ is updated as the exponential moving average (EMA) of the student to provide robust pseudo labels on the unlabeled images. The filtered pseudo labels are used to supervise the student model to minimize $\mathcal{L}_{unsup}$. (b) presents our density-guided contrastive learning strategy. In mini-batch, feature maps (green blocks) of *plane* are extracted by true and pseudo masks. Then, for each feature vector, we build per-feature nearest neighbor graphs where all the neighbors are from the memory containing *plane* features (orange blocks). The density is estimated by averaged neighbor similarity. Then *plane* features in mini-batch are ranked by their density values, and those with the lowest density are selected as anchors. The contrastive loss $\mathcal{L}_{contra}$ is minimized by pushing those anchors towards the cluster center approximated by high-density features in the memory bank.

projection space $z_{ij} = g(v_{ij})$ where contrastive learning is applied. We estimate feature density on $\mathcal{V} = \{v\}$ for sample selection while the contrastive loss is performed on their corresponding projections $\mathcal{Z} = \{z\}$.

#### 3.4.1 Neighbor Compactness as Feature Density

Inspired by density-peak assumption [33], our contrastive learning strategy aims to increase in-class cluster compactness by pushing features in sparse regions towards the class center approximated by densely gathered features. We propose to use feature density as an indicator to decide which feature should be selected as anchors.

**Modelling In-class Feature Distribution.** Before we can measure in-class feature density, we must acquire an approximation of the class feature distribution. In-image or in-batch categorical features can only provide a limited view of the class. Therefore, we propose dynamic memory banks to collect categorical features across the whole dataset for more comprehensive modeling of class features. In each mini-batch, class features are extracted by student encoder $h$ based on annotations and filtered pseudo labels. For class $c$, feature memory $\mathcal{P}^c$ is updated in a First-In, First-Out (FIFO) style while preserving a fixed size. To avoid features in large objects dominating the bank, we set a threshold for a single update, and sub-sample image features to absorb more images to increase the diversity of the bank.

**Building Nearest Neighbor Graphs.** As shown in Fig. 2 (b), features of class $c$ are extracted from mini-batch as $\mathcal{V}^c = \{v_1, v_2, \ldots, v_m\}$, with the help of ground truth and pseudo labels. In grey region of Fig. 2 (b), we measure the density of each $v$ in a feature-to-bank style, by building a k-nearest neighbor graph for $v$ in $\mathcal{P}^c \cup \{v\}$ as $\mathcal{N}_k(v) = \{v'_1, v'_2, \cdots, v'_k\}$. Such a setting aims to avoid in-image feature connections to guarantee an estimation in a global view. Then, following [27], the density $d(v)$ is calculated as the averaged cosine similarity between $v$ and its $k$ neighbors as:

$$d(v) = \frac{1}{|\mathcal{N}_k(v)|} \sum_{v' \in \mathcal{N}_k(v)} \frac{v^T v'}{\|v\| \cdot \|v'\|}, \tag{7}$$

where $\|\cdot\|$ denotes the L2-norm of a vector. The calculated densities $\{d(v)\}$ are collected for later sample selections. Meanwhile, batch features with their density values $\{(v, d(v))\}$ are used to update the corresponding memory.

**Multi-Scale Density Estimation.** Density estimated by a small number of neighbors tends to focus on the local region, while graphs with many neighbors lead to a more smoothed estimation which provides a larger view.

To obtain robust density estimation, we propose multi-scale nearest neighbor graphs for target feature vector $v$ as $\{\mathcal{N}_{k_1}, \mathcal{N}_{k_2}, ..., \mathcal{N}_{k_n}\}$, where $k_1 < k_2 < ... < k_n$. We calculate density values $d_1, d_2, ..., d_n$ with different graphs by Eq. 7. The final estimation is calculated as the averaged density values:

$$d(v) = \frac{1}{n} \sum_{i=1}^{n} d_i(v). \tag{8}$$

#### 3.4.2 Density-Guided Sampling on Anchors and Keys

With density estimation from Section 3.4.1, we perform density-guided anchor and key selection for categorical features in each mini-batch.

**Low-Density Anchor Sampling.** Anchor selection is critical in our proposed method since it largely determines the quality of supervision. Our sampling strategy aims to locate the anchor features that can provide the most effective supervision. We propose to use density to measure the "hardness" of training for each feature. Features with low-density values indicate that they are less representative in the class, in other words, under-trained for the current model state. Thus, tackling features in the sparse region will primarily benefit the learning process.

For all features of one specific category in a mini-batch, we first estimate their density referring to the corresponding memory bank. Then they are ranked by their density values. For class $c$, features with the lowest $N_q$ densities are selected as anchors and stored in $\mathcal{Q}^c$:

$$\mathcal{Q}^c = \{\, q^c \mid d(q^c) \leqslant a \,\}, \tag{9}$$

where $a$ indicates the $N_q$-th least density values for in-class features in current mini-batch.

**High-Density Positive Key Sampling.** We assume that most representative in-class features lie in dense regions. Thus, we approximate class centers with only high-density features to guide the anchors. In training, a total $N_r^+$ high-density positive keys are sampled from the mini-batch and the memory bank. In-batch features can provide fresh and in-object contrast, while the global memory bank shows a more comprehensive and diversified categorical pattern. Thus, two sets of features are assumed to complement each other to provide a robust center estimation. In each mini-batch, positive keys of class $c$ with largest $\frac{1}{2}N_r^+$ densities are selected and stored in $\mathcal{R}_{local}^{c,+}$ as:

$$\mathcal{R}_{local}^{c,+} = \{\, r_{local}^c \mid d(r_{local}^c) \geqslant b_{local}^c \,\}, \tag{10}$$

where $b_{local}^c$ represents the $\frac{1}{2}N_r^+$-th largest in-class density values for current batch. Similarly, in the memory bank of class $c$, features are ranked by their density values and another $\frac{1}{2}N_r^+$ keys are selected as:

$$\mathcal{R}_{global}^{c,+} = \{\, r_{global}^c \mid d(r_{global}^c) \geqslant b_{global}^c \,\}, \tag{11}$$

with $b_{global}^c$ indicating $\frac{1}{2}N_r^+$-th largest in-bank density. Then, with unified positive keys $\mathcal{R}^{c,+} = \mathcal{R}_{local}^{c,+} \cup \mathcal{R}_{global}^{c,+}$, categorical cluster center is approximated as:

$$r_{center}^{c,+} = \frac{1}{|\mathcal{R}^{c,+}|} \sum_{r^+ \in \mathcal{R}^{c,+}} r^+. \tag{12}$$

**Random Negative Key Sampling.** Unlike other works that carefully select negative keys, our method sets a relatively loose standard for negative key sampling. We randomly sample $N_r^-$ out-of-class features in current batch for each anchor of class $c$ as $\mathcal{R}^{c,-} = \{r^-\}$ to form negative contrast pairs.

#### 3.4.3 Optimization on Feature Contrast

A pixel-level contrastive loss function is performed to encourage a low-density anchors $a$ to be similar to the cluster center $r_{center}^+$, which is approximated by high-density positive keys $r^+$ locally and globally. Note that all features need to be projected by the projection head $g$ before being optimized by contrastive loss. Our loss design follows SupCon [24]. The sample selection and optimization are conducted across all classes in a mini-batch as shown in Eq. 13, where $\tau$ is the temperature parameter.

$$\mathcal{L}_{contra} = -\sum_{c \in \mathcal{C}} \sum_{q \in \mathcal{Q}^c} \log \frac{\exp(g(q) \cdot g(r_{center}^{c,+})/\tau)}{\sum_{r^{c,-} \in \mathcal{R}^{c,-}} \exp(g(q) \cdot g(r^{c,-})/\tau)}. \tag{13}$$

## 4. Experiments

### 4.1. Experimental Setup

**Datasets.** PASCAL VOC 2012 [12] is a standard semantic segmentation benchmark containing 20 foreground classes and one background class. It was initially built with 1464 finely annotated training images and 1449 validation images. Later the SBD dataset [16] extended the training set with extra coarse annotations to 10582 labeled images. Following previous works [29, 40], we conduct experiments on the original set and blended set separately. Cityscapes [8] is designed for urban scene understanding. It defines 19 semantic categories, and the training and validation sets contain 2975 and 500 finely annotated images separately.

To generate semi-supervised data splits for both datasets, we follow the partition protocols in [29] to randomly sample 1/16, 1/8, 1/4 and 1/2 from the whole training set as the labeled set while the rest serves as the unlabeled set.

Table 1. Comparison with state-of-the-art methods on PASCAL VOC 2012 *val* set with mIoU results (%) ↑. Methods are trained under *classic* setting. Labeled images are from the high-quality training set comprising 1464 samples. The fractions and the following integers denote the proportions and numbers of labeled images, respectively.

| Method | 1/16 (92) | 1/8 (183) | 1/4 (366) | 1/2 (732) | Full (1464) |
|---|---|---|---|---|---|
| Supervised | 45.77 | 54.92 | 65.88 | 71.69 | 72.50 |
| MT [38] | 51.72 | 58.93 | 63.86 | 69.51 | 70.96 |
| CutMix [13] | 52.16 | 63.47 | 69.46 | 73.73 | 76.54 |
| CPS [7] | 64.07 | 67.42 | 71.71 | 75.88 | - |
| U$^2$PL [40] | 67.98 | 69.15 | 73.66 | 76.16 | 79.49 |
| ST++ [43] | 65.20 | 71.00 | 74.60 | 77.30 | 79.10 |
| PS-MT [29] | 65.80 | 69.58 | 76.57 | 78.42 | 80.01 |
| PCR [42] | 70.06 | 74.71 | 77.16 | 78.49 | 80.65 |
| GTA-Seg [22] | 70.02 | 73.16 | 75.57 | 78.37 | 80.47 |
| **Ours** | **70.47** | **77.14** | **78.73** | **79.23** | **81.55** |

Table 2. Comparison with state-of-the-art methods on PASCAL VOC 2012 *val* set with mIoU results (%) ↑. Methods are trained under *blended* setting. Labeled images are randomly sampled from the extended training set, which consists of 10582 samples.

| Method | 1/16 (662) | 1/8 (1323) | 1/4 (2646) | 1/2 (5291) |
|---|---|---|---|---|
| Supervised | 67.87 | 71.55 | 75.80 | 77.13 |
| MT [38] | 70.51 | 71.53 | 73.02 | 76.58 |
| CutMix [13] | 71.66 | 75.51 | 77.33 | 78.21 |
| CCT [32] | 71.86 | 73.68 | 76.51 | 77.40 |
| GCT [23] | 70.90 | 73.29 | 76.66 | 77.98 |
| CPS [7] | 74.48 | 76.44 | 77.68 | 78.64 |
| U$^2$PL$^*$ [40] | 74.43 | 77.60 | 78.70 | 79.94 |
| PS-MT [29] | 75.50 | 78.20 | 78.72 | 79.76 |
| **Ours** | **76.61** | **78.37** | **79.31** | **80.96** |

$*$ denotes that the results are reproduced with CPS [7] splits.

Table 3. Comparison with state-of-the-art methods on Cityscapes *val* set with mIoU results (%) ↑. Labeled images are selected from Cityscapes *train* set, which contains 2975 samples.

| Method | 1/16 (186) | 1/8 (372) | 1/4 (744) | 1/2 (1488) |
|---|---|---|---|---|
| Supervised | 65.74 | 72.53 | 74.43 | 77.83 |
| MT [38] | 69.03 | 72.06 | 74.20 | 78.15 |
| CutMix [13] | 67.06 | 71.83 | 76.36 | 78.25 |
| CCT [32] | 69.32 | 74.12 | 75.99 | 78.10 |
| GCT [23] | 66.75 | 72.66 | 76.11 | 78.34 |
| CPS [7] | 69.78 | 74.31 | 74.58 | 76.81 |
| U$^2$PL [40] | 70.30 | 74.37 | 76.47 | 79.05 |
| PS-MT [29] | - | 76.89 | 77.60 | 79.09 |
| PCR [42] | **73.41** | 76.31 | 78.40 | 79.11 |
| GTA-Seg [22] | 69.38 | 72.02 | 76.08 | - |
| **Ours** | 73.18 | **77.29** | **78.48** | **80.71** |

**Network Architecture.** We adopt DeepLabv3+ [5] as the segmentation head with ResNet-101 [18] pre-trained on ImageNet [10] as the feature encoder. The projection head is designed as a multi-layer perceptron with a structure as Linear (256) → BatchNorm → ReLU → Linear (256). The projection head receives pixel-level features with 512 channels from the student encoder and projects each feature vector into a 256-dimensional space.

**Evaluation Protocols.** We adopt mean of intersection of union (mIoU) as our evaluation metric. For PASCAL, we perform single-scale evaluation on center-cropped images. For Cityscapes, sliding evaluation is performed on the original image to obtain the final result. All the evaluation is based on the results from the teacher network.

**Implementation Details.** We apply SGD optimizer with weight decay in training. For PASCAL, the initial learning rate is 0.001 with weight decay of 0.0001. The model is trained for 50k iterations with a batch size of 16. For Cityscapes, we set the initial learning rate as 0.01 with a weight decay of 0.0001. We train the model for 200k iterations under a batch size of 8. We use polynomial policy to decay the learning rate in training as: $lr = lr_{init} \cdot (1 - \frac{iter}{total\_iters})^{0.9}$. OHEM loss [36] is adopted for Cityscapes.

Data augmentations are introduced for both labeled and unlabeled images, including random resize, random crop and random horizontal flip. Images are cropped as $513 \times 513$ for PASCAL and $769 \times 769$ for Cityscapes in training. Strong data augmentation CutMix is only applied for unlabeled images for consistency training.

The initial entropy percentile threshold $\beta_0$ in Section 3.3 is set as 20%. The loss coefficients $\lambda_{unsup}$ and $\lambda_{contra}$ in Eq. 2 are both set as 1. The multi-scale estimation in Section 3.4.1 is based on graphs with neighbors of $k_1 = 8$, $k_2 = 16$ and $k_3 = 32$. We set $N_q$ the number of anchors in Section 3.4.2 as 256 per class in each mini-batch. For each anchor, positive keys $N_r^+$ and negative keys $N_r^-$ are both set as 512. Note that half positive keys (256) are from in-batch features while the other half are from the memory bank. The temperature coefficient $\tau$ in Eq. 13 is set as 0.5.

Table 4. Quantitative comparison of clustering on PASCAL VOC *val* images trained on *classic* 1/8 set. Clusters are extracted by model predictions. Baseline is trained with self-training.

| | Silhouette ↑ | Calinski-Harbasz ↑ | Davies-Boulding ↓ |
|---|---|---|---|
| Baseline | 0.46 | 3421.94 | 1.53 |
| DGCL | **0.70** | **7937.87** | **1.13** |

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

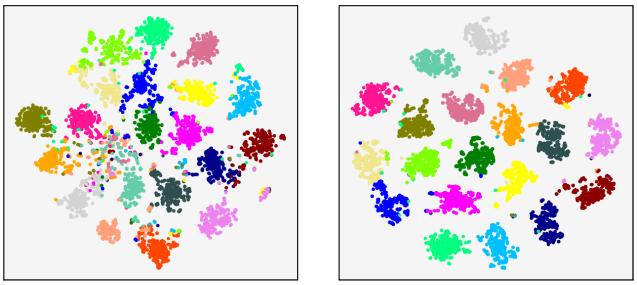
In this section, we compare our framework extensively with recent semi-supervised semantic segmentation methods across various datasets and data protocols. For a fair comparison, all the methods are based on DeepLabV3+ with ResNet-101 backbone. In PASCAL VOC, we evaluate our method in two settings: 1). The *classic* set where labeled images are only selected from the high-quality set with 1464 samples. 2). The *blended* set where annotations are randomly drawn from the extended set with 10582 labeled images. All data splits strictly follow the settings adopted in CPS [7], ST++ [43] and PS-MT [29] for both PASCAL VOC and Cityscapes datasets.

**Results on PASCAL VOC 2012.** Table 1 reports comparison results on PASCAL VOC 2012 *val* set on *classic* setting. The plain supervised baseline shows unsatisfactory results, especially on low-data regime. Our method brings significant performance boost over supervised baseline by +24.70%, +22.22%, +12.85% under 1/16, 1/8 and 1/4 label partitions respectively. Compared with previous methods, our approach also yields superior performance. With only 183 and 366 labeled images, our approach outperforms the previous best by +2.43% and +1.57%, respectively.

Table 2 presents the results on the *blended* setting. Our method achieves consistent performance gains over other baselines. Specifically, our method improves the supervised baseline by +8.74% and +6.82% on the 1/16 and 1/8 splits, respectively. Compared to the previous best-performed PS-MT [29], our method achieves significant improvements on 1/16 and 1/2 by +1.11% and +1.20%, respectively.

**Results on Cityscapes.** Table 3 shows the comparison results on the Cityscapes *val* set. Our method brings consistent performance gains on supervised baselines, e.g., +4.76%, 4.05% and +2.88% on 1/8, 1/4 and 1/2 data partitions. Our approach outperforms previous state-of-the-art methods by a notable margin. Specifically, our method achieves performance gains over the previous best +1.60% on 1/2 partition protocols.

The results on both PASCAL VOC 2012 and Cityscapes demonstrate the superiority of our approach, especially on the low-data regime. The presented performance indicates the effectiveness of our density-guided contrastive learning strategy in leveraging unlabeled data.



(a) Baseline method (b) DGCL

Figure 3. t-SNE [39] visualization of feature spaces on PASCAL VOC *val* images trained on *classic* 1/8 set. Clusters are extracted by model predictions without accessing true labels. We randomly select 10000 data points to for each class to perform t-SNE with perplexity parameter of 32. We sample 500 points per class for the plot. Baseline is trained with self-training. Our DGCL yields overall better clustering results.

Table 5. Ablation study on main components of the proposed framework. $\mathcal{L}_{unsup}$: Unsupervised training with pseudo labels. $\mathcal{L}_{contra}$: Plain contrastive learning where anchors are sampled randomly and cluster centers are obtained as average pooling of class features. Density: Density-guided anchor and key sampling where density is estimated in batch and positive anchors are sampled in batch. Memory: Density is estimated with class memory, and positive anchors are extended by memory features.

| | $\mathcal{L}_{unsup}$ | $\mathcal{L}_{contra}$ | Density | Memory | 183 | 1323 |
|---|---|---|---|---|---|---|
| I | | | | | 54.92 | 71.55 |
| II | ✓ | | | | 69.22 | 75.15 |
| III | ✓ | ✓ | | | 71.82 | 76.73 |
| IV | ✓ | ✓ | ✓ | | 75.83 | 77.97 |
| V | ✓ | ✓ | ✓ | ✓ | **77.14** | **78.37** |

Table 6. Ablation study on different sampling strategies in contrastive learning. Random: Random sampling. Average: Average pooling on in-class features. Low/High Conf: Select samples with the least/highest prediction confidence. Low/High Density: Select samples with the least/highest density values. For all the cases, negative keys are randomly sampled in batch features.

| Anchors | Positive Keys | 183 | 1323 |
|---|---|---|---|
| Random | Average | 71.82 | 76.73 |
| Low Conf | High Conf | 71.42 | 77.11 |
| Low Denisty | High Density | **77.14** | **78.37** |

## 4.3. Ablation Studies

To investigate contribution of each component in our approach, we conduct ablation study on 1/8 (183) *classic* and 1/8 (1323) *blended* data splits in PASCAL VOC 2012.

**Clustering Performance Comparison.** One important claim is that our DGCL learns better representations in fea-

Table 7. Ablation study on neighbor numbers. Multi-Scale: Multi-scale density estimation on [8, 16, 32] neighbors with Eq. 8.

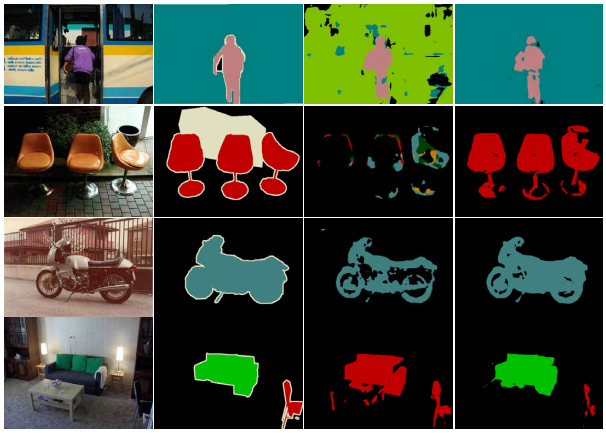
| Neighbors | 8 | 16 | 32 | 64 | Multi-Scale |
|---|---|---|---|---|---|
| 183 | 76.57 | 76.71 | 76.89 | 74.59 | **77.14** |
| 1323 | 77.74 | 78.16 | 78.12 | 77.66 | **78.37** |

ture space. We conduct experiments to verify it visually and quantitatively. We extract clusters from PASCAL *val* set with model trained on *classic* 1/8 set. Feature categories are assigned by model predictions. Fig. 3 compares feature spaces learned by self-training baseline and DGCL. It is evident that our method generates more separable clusters. In Table 4, we evaluate the clustering performance from different aspects by metrics of Silhouette [34], Calinski-Harbasz [3] and Davies-Boulding [9] scores, which show that DGCL significantly improve over baseline to generate more compact clusters with higher inter-cluster separability.

**Density-Guided Contrastive Learning.** We ablate the main framework in Table 5 to manifest the effectiveness of DGCL. We use models from supervised only (Experiment I) and self-training (Experiment II) as two baselines. Then plain contrastive learning with random sampling strategy is introduced in Experiment III and only brings limited gain on two data splits. Experiment VI demonstrates that introducing density information to guide the training (in-batch density estimation and positive key sampling) significantly boosts the performance by +4.01% and +1.24% on 183 and 1323 partitions respectively. In Experiment V, the performance is further improved by +1.31% and +0.40% on two settings when density is estimated in memory and positive keys are extended by memory features, which indicates the importance of leveraging memory bank in our framework.

**Different Sampling Strategies.** We validate the effectiveness of density-guided sampling strategy by comparing it with other strategies as shown in Table 6. As baseline, random anchor sampling select anchors without considering their hardness. Also, the cluster center is simply estimated as the average pooling of all in-class features, inevitably involving less representative samples, leading to inefficient training with lowest performances. Prediction confidence is widely used in sample selection and prediction filtering. We also implement confidence-guided sampling strategies where we use prediction confidence from classifier as selection criterion. We notice that when we adopt hard anchors denoted by low confidence and push them toward class center estimated by reliable features with high confidence, we cannot achieve comparable results as our density-guided sampling. It shows density information is more effective in guiding the contrastive learning.

**Multi-Scale Nearest Neighbor Graphs.** We also investigate the impact of size of nearest neighbor graphs on density estimation. Table 7 compares graphs with different number of neighbors. We found that the optimal choice is around 16 neighbors. The performance slightly degenerates when the graph size is further expanded to contain 64 neighbors. Then, our multi-scale estimation upon 8, 16 and 32 neighbors achieves best result among all settings.



(a) (b) (c) (d)

Figure 4. Qualitative results on PASCAL VOC 2012 *val* set based on 1/8 *classic* set. (a) Input images. (b) Ground truth. (c) Results from baseline with $\mathcal{L}_{sup}$ and $\mathcal{L}_{unsup}$. (d) Results by ours.

### 4.4. Qualitative Results

Fig. 4 presents the qualitative results on PASCAL VOC 2012 datasets under 1/8 *classic* set. With minimal annotations, self-training baseline performs poorly and even lose track of categories as shown in the second and last row in Fig. 4, where the model misclassifies pixels of *sofa* and *chair*. The model with our DGCL training strategy can capture semantics and object structures more accurately.

## 5. Conclusion

In this work, we propose a novel Density-Guided Contrastive Learning (DGCL) strategy for semi-supervised semantic segmentation, which aims to mine effective supervision from the geometry of clusters and enhance the training by tackling sparse regions inside. The core of our method is the robust estimation of feature density, which is achieved by multi-scale estimation within in-class feature distribution approximated by the memory bank. Extensive experiments under various settings have shown that the proposed method outperforms existing state-of-the-art methods, and the ablation study has revealed the effectiveness of our DGCL.

**Acknowledgements.** This work was supported by National Key R&D Program of China (No.2022YFE0200300) and National Natural Science Foundation of China under 61972323.

## References


[1] Iñigo Alonso, Alberto Sabater, David Ferstl, Luis Montesano, and Ana Cristina Murillo. Semi-supervised semantic segmentation with pixel-level contrastive learning from a class-wise memory bank. In *ICCV*, 2021. 2

[2] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 2

[3] Tadeusz Caliński and Joachim Harabasz. A dendrite method for cluster analysis. *Commun. Stat. - Theory Methods*, 3(1):1–27, 1974. 8

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2018. 1

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 6

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3

[7] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, 2021. 2, 6, 7

[8] 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 *CVPR*, 2016. 2, 5

[9] David L. Davies and Donald W. Bouldin. A cluster separation measure. *PAMI*, 1(2):224–227, 1979. 8

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[11] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996. 1

[12] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2009. 1, 2, 5

[13] Geoffrey French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham D. Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020. 1, 2, 6

[14] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2004. 2

[15] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Unbiased subclass regularization for semi-supervised semantic segmentation. In *CVPR*, 2022. 1

[16] Bharath Hariharan, Pablo Arbeláez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[18] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[19] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *ICCV*, 2021. 1

[20] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018. 2

[21] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. In *NeurIPS*, 2019. 2

[22] Ying Jin, Jiaqi Wang, and Dahua Lin. Semi-supervised semantic segmentation via gentle teaching assistant. In *NeurIPS*, 2022. 6

[23] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson W. H. Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *ECCV*, 2020. 6

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020. 1, 3, 5

[25] Xin Lai, Zhuotao Tian, Li Jiang, Shu Liu, Hengshuang Zhao, Liwei Wang, and Jiaya Jia. Semi-supervised semantic segmentation with directional context-aware consistency. In *CVPR*, 2021. 2

[26] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017. 2

[27] Suichan Li, B. Liu, Dongdong Chen, Q. Chu, Lu Yuan, and Nenghai Yu. Density-aware graph for deep semi-supervised visual recognition. In *CVPR*, 2020. 4

[28] Shikun Liu, Shuaifeng Zhi, Edward Johns, and Andrew J Davison. Bootstrapping semantic segmentation with regional contrast. In *ICLR*, 2022. 1, 2

[29] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and G. Carneiro. Perturbed and strict mean teachers for semi-supervised semantic segmentation. In *CVPR*, 2022. 1, 2, 5, 6, 7

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[31] Sudhanshu Mittal, Maxim Tatarchenko, and Thomas Brox. Semi-supervised semantic segmentation with high- and low-level consistency. *PAMI*, 43(4):1369–1379, 2021. 2

[32] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *CVPR*, 2020. 1, 2, 6

[33] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492 – 1496, 2014. 1, 4

[34] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *JCAM*, 20:53–65, 1987. 8

[35] Mehdi S. M. Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and

perturbations for deep semi-supervised learning. In *NeurIPS*, 2016. 2

[36] Abhinav Shrivastava, Abhinav Kumar Gupta, and Ross B. Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016. 6

[37] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 2

[38] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, 2017. 2, 6

[39] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. *JMLR*, 9(86):2579–2605, 2008. 7

[40] 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 *CVPR*, 2022. 1, 2, 5, 6

[41] Qizhe Xie, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 2

[42] Hai-Ming Xu, Lingqiao Liu, Qiuchen Bian, and Zhengeng Yang. Semi-supervised semantic segmentation with prototype-based consistency regularization. In *NeurIPS*, 2022. 6

[43] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *CVPR*, 2022. 1, 6, 7

[44] Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In *ICCV*, 2021. 1

[45] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *NeurIPS*, 2021. 2

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[47] Xiangyu Zhao, Raviteja Vemulapalli, P. A. Mansfield, Boqing Gong, Bradley Green, Lior Shapira, and Ying Wu. Contrastive learning for label efficient semantic segmentation. In *ICCV*, 2021. 1, 2

[48] Yuanyi Zhong, Bodi Yuan, Hong Wu, Zhiqiang Yuan, Jian Peng, and Yu-Xiong Wang. Pixel contrastive-consistent semi-supervised semantic segmentation. In *ICCV*, 2021. 1, 2

[49] Yanning Zhou, Hang Xu, Wei Zhang, Bin-Bin Gao, and Pheng-Ann Heng. C3-semiseg: Contrastive semi-supervised segmentation via cross-set learning and dynamic class-balancing. In *ICCV*, 2021. 1, 2

[50] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. In *NeurIPS*, 2020. 2

[51] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister. Pseudoseg: Designing pseudo labels for semantic segmentation. In *ICLR*, 2021. 1, 2