Diving Segmentation Model into Pixels

Anonymous ICCV submission

Paper ID ****

Abstract

*More meaningful pixel features will benefit the semantic segmentation under various settings. Existing efforts to mine better pixel-level features attempt to explicitly model the categorical distribution, which fails to achieve optimal due to the large pixel feature variance. In this work, We raise the concept of **pixel learning** to concentrate on the tailored learning process of pixels, handle pixel-level variance, and enhance the segmentation model's per-pixel recognition capability. Under the context of the pixel learning scheme, each image is viewed as a distribution of pixels, and pixel learning aims to pursue consistent pixel representation inside an image, continuously align pixels from different images (distributions), and eventually achieve consistent pixel representation for each category. We proposed a pure pixel-level learning framework, namely PiXL, which consists of a PixPar module to partition pixels into subdomains, a prototype generation and selection module to prepare targets for subsequent alignment, and a pixel alignment module to guarantee pixel feature consistency intra- and inter-images. Extensive evaluations of multiple learning paradigms, including unsupervised domain adaptation, semi-, and fully-supervised segmentation, show PiXL outperforms the state-of-art performances, especially when annotated images are scarce. The code is available upon acceptance.*

1. Introduction

Semantic segmentation is challenging because of requiring the categorical pixel-level annotations precisely and consistently under different background knowledge contexts, *e.g.*, organs or lesions [11, 19] in medical image analysis, city scenes in autonomous driving [28, 13]. In theory, pixels belonging to the same category, though from different images, should follow an implicit **global distribution**, while the pixels inside the images are subsets of this global distribution, referred to **local distribution**. Existing research mainly attempted to directly model or represent the implicit global distribution based on one or multiple prototypes or speculated prior distribution. Zhang *et al*. [43], Liu *et al*. [21] and Yang *et al*. [40] adopted the former methods to utilize those prototypes to stand for the global distributions. Xie *et al*. [35] employed the Gaussian distribution to model the global distribution. However, the complexity of global distribution exceeds the representation ability of prototypes or primary prior distribution. The notable pixel-level variance caused by the factors summarized in Tab 1 also aggravates the problem. Although multiple

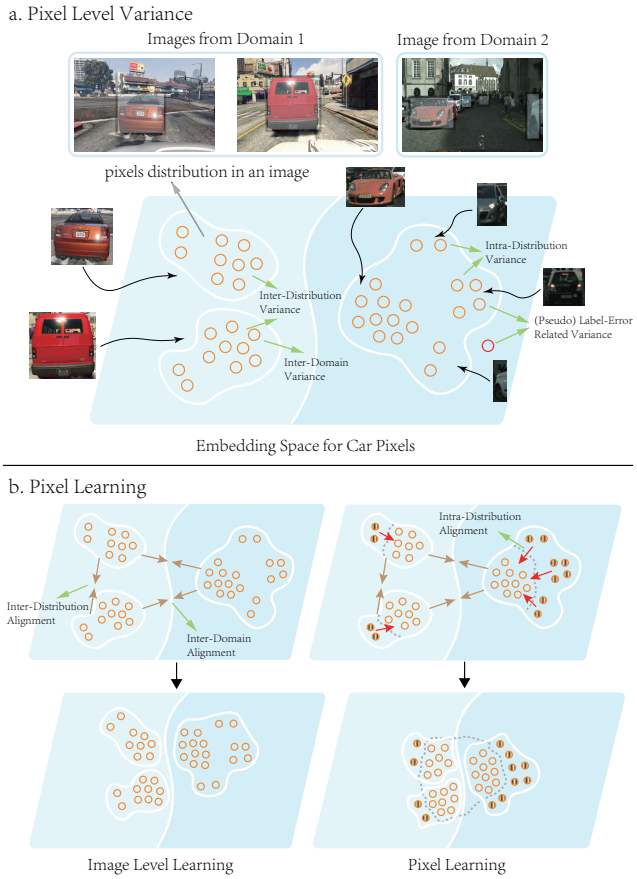


Figure 1: **a.** Causes of pixel-level variance. **b.** Highlights of pixel learning.

prototypes or complex prior distribution may alleviate these problems, the generation strategy of a large number of prototypes and complex prior distribution requiring tailored design based on a specific scenario is not affordable. Contrary to directly modeling or representing the global distribution, we advocate gradually approximating the global distribution by diving into the pixel level and addressing the pixel-level variance in a divide-and-conquer manner. In this work, we propose the concept of **pixel learning** where each image is viewed as a distribution of pixels. Therefore, pixels from the same distribution (image), different distributions, and different domains contain variance. Moreover, the unreliable pseudo labels in unsupervised domain adaptation or semi-supervised settings also exacerbate the variance as summarized in Tab 1. Pixel learning concentrates on addressing intra- and inter-distribution (image) variance to eventually achieve a consistent global distribution for each category. Under the context of pixel learning, We introduce a novel segmentation framework PiXL designed with a pure pixel-based learning strategy. Specifically, at each learning step, pixels from several images form the local distribution. Only a portion of pixels in the local distribution follows the global distribution, while the others are not, denoted as joint pixels and drift pixels, respectively. We proposed the Pixel-Level Sub-Domain Partition (PixPar) module to distinguish the pixels based on entropy. To align pixels in both intra- and inter-distributions, the pixel learning conducts asymmetric alignment between joint pixels and drift pixels utilizing our Drift Pixel Asymmetric Contrast (DPA) module in Fig 2. The alignment target is prototypes generated from joint pixels which stand for joint points of local and global distribution. Compared with previous prototype-based methods, our prototypes embed rich local distribution information besides global distribution information, denoted as local prototypes. We further designed an Adaptive Prototype Selection strategy (APS) to select the most representative local prototypes from multi-resolution images. Extensive experiments, including unsupervised domain adaptation (UDA), semi-, and fully-supervised segmentation, are conducted to evaluate PiXL. In the UDA setting, PiXL outperforms the state-of-the-art methods on GTA5 and SYNTHIA $\rightarrow$ Cityscapes respectively. In the semi-supervised setting on the Cityscapes dataset, PiXL produces robust and competitive performance, especially in 3.3% images labeled setting, PiXL outperforms the SOTA by a large margin. The performance in a fully supervised setting on cityscapes is also competitive.

Our contributions are as follows:

- We propose a novel learning scheme named pixel learning to dive semantic segmentation into the pixels to enhance the segmentation models.
- We propose a novel learning framework PiXL under the context of pixel learning. The pixel learning scheme in PiXL emphasizes pixel-level variance and continually handles that by focusing on intra- and inter-distribution alignment.
- Extensive experiments validate the performance of PiXL, showing promising results on label-scarce settings.

Table 1: Causes of pixel-level variance: **a.** Intra-Distribution Variance. **b.** Inter-Distribution Variance. **c.** Inter-Domain Variance. **d.** Label Error Related Variance.

Causes of Pixel-Level Variance				
Task Setting	Intra.Dist.Var.	Inter.Dist.Var.	Inter.Dom.Var.	Label.Err.Var.
Fully Supervised Segmentation	✓	✓		
Semi-Supervised Segmentation	✓	✓		✓
Unsupervised Domain Adaptation	✓	✓	✓	✓

2. Related Work

2.1. Pixel-level learning

For image segmentation, image-level learning methods only roughly recognize each pixel, while pixel learning is more precise. In unsupervised or self-supervised learning, Xie *et al*. [38] adopted pixel-level contrastive learning as a pre-training task. In semi-supervised learning, Alonso *et al*. [2] aligned per-pixel feature to high-quality pixels in the memory bank. In weakly-supervised learning, Anh *et al*. [1] and Du *et al*. [8] mining the pixel-level feature under supervision from CAMs[45]. In fully supervised learning, Wang *et al*. [32] explored forming contrastive pairs with the cross-image pixels. In unsupervised domain adaptation, Vayyat *et al*. [29] conducts contrastive learning on pixels from multi-resolution images in CLUDA. Xie *et al*. [35] unified different forms of pixel contrastive learning and added Gaussian distribution prior in SePiCo. However, few of these previous methods thoroughly analyzed pixel-level variance and tailored learning strategies in a per-pixel manner. Our PiXL dives into pixel-level variance and designs a per-pixel learning strategy to cope with that.

2.2. Refined Variance Handling

Although inter-domain variance is mostly considered in settings like UDA, intra-domain variance, and even intra-distribution variance matter but is rarely explored. Pan *et al*. [25] proposed IntraDA to explicitly conduct intra-domain adaptation. Cai *et al*. [3] extended IntraDA to an iterative adaptation manner. Yan *et al*. [39] proposed PixIntraDA to conduct intra-domain alignment at the pixel level. Whereas, the lack of comprehensive investigation of variance at pixel level hampers handling of the issue. We summarize the causes of pixel-level variance in Tab 1. Besides, we approach the pixel-level variance based on the pixel learning scheme in a divide-and-conquer manner to align drift pixels in local distribution to joint pixels following global distribution.

2.3. Contrastive learning

Contrastive learning is adopted either as a pretext task for pre-training[10, 34, 44] or a plug-and-play module in a model[35, 29]. Based on the InfoNCE loss[23] according to equation (1), features move closer to positive samples while farther from negative ones. In early works[42, 5], contrastive learning is applied to image-level learning scenarios. Recently, some works attempted to apply contrastive learning at region[36] or pixel[35, 32] level. Xie *et al*. [36] introduced patch-level contrastive learning by adding patch-level InfoNCE loss. Wang *et al*. [33] applied contrastive learning at pixel level with carefully selected contrastive pairs. Vayyat *et al*. [29] introduced an explicit pixel-to-pixel contrastive manner in UDA. Nevertheless, these previous works lack further consideration in contrastive mechanism design and proper positive sample selection which may cause adverse effects and misleading signals. PiXL developed an asymmetric contrast mechanism, inspired by [41], collaborated with PixPar and APS to guarantee the reliability of positive pixels and correct approximation of global distribution,

$$\mathcal{L}_{InfoNCE}\left(\boldsymbol{f}_a, \mathcal{P}, \mathcal{N}\right) = \frac{1}{|\mathcal{P}|}\sum_{\boldsymbol{f}_p \in \mathcal{P}}\left[-\log\left(\frac{e^{\left(\boldsymbol{f}_a \cdot \boldsymbol{f}_p/\tau\right)}}{e^{\left(\boldsymbol{f}_a \cdot \boldsymbol{f}_p/\tau\right)} + \sum_{\boldsymbol{f}_n \in \mathcal{N}} e^{\left(\boldsymbol{f}_a \cdot \boldsymbol{f}_n/\tau\right)}}\right)\right] \tag{1}$$

3. Method

3.1. Overview

Our PiXL framework follows the pixel learning scheme, where each image $X \in \mathbb{R}^{H\times W\times 3}$ is viewed as a distribution of pixels with its label Y according to equation (2).

$$(X, Y) = \{(x_j, y_j)\}, j \in 1, \cdots, N, \tag{2}$$

where x_j stands for pixel j from image X, and y_j is its category. N denotes the number of pixels in image X. For the sake of narrative, **image** is also referred to as **distribution** in this paper.

Then, the PiXL Framework is composed of four modules, i.e., i. Multiple Resolution Feature Extraction, ii. Pixel Level Sub-Domain Partition, iii. Drift Pixel Alignment, iv. Adaptive Prototype Selection. Specifically, at each training step in PiXL, two distributions are sampled to conduct intra- and inter-distribution alignment. We follow [35, 29, 32] to acquire the feature representation of pixels in an extracted feature map. In the multiple-resolution feature extraction module, an image X is fed into model E to acquire its feature map $\boldsymbol{F}$. Hence, the correspondence between pixel feature representation and pixel label can be given by equation (3).

$$(\boldsymbol{F}, Y) = \{(\boldsymbol{f}_j, y_j)\}, j \in 1, ..., N \tag{3}$$

Then the downstream learning will concentrate on alignment at the pixel level to handle pixel-level variance and approximate the global distribution. According to our key hypothesis, each image contains a few joint pixels following the global distribution and the other drift pixels, denoted as $\hat{\boldsymbol{f}}$ and $\tilde{\boldsymbol{f}}$, respectively. These two parts are partitioned from the distribution X in the pixel-level sub-domain partition module as follows:

$$\begin{aligned}\hat{\boldsymbol{F}} &= \{\hat{\boldsymbol{f}}_1, \cdots, \hat{\boldsymbol{f}}_{N_j}\},\\ \tilde{\boldsymbol{F}} &= \{\tilde{\boldsymbol{f}}_1, \cdots, \tilde{\boldsymbol{f}}_{N_d}\}.\end{aligned} \tag{4}$$

where N_j and N_d represent the number of joint pixels and drift pixels respectively. The pixel learning scheme emphasizes the handling of pixel-level variance. Hence, in our PiXL framework joint pixels from two sampled distributions are viewed as joint points between the implicit global distribution and the local distributions. Then, the drift pixels are pushed to local prototypes generated from those joint pixels, to align with the global distribution in the drift pixels alignment module. Moreover, the adaptive prototype selection module is proposed to guarantee the quality of prototypes from multi-resolution features.

3.2. Multiple Resolution Feature Extraction

To ensure that the model can extract richer semantic features and maintain robustness to the resolution of input images, a multiple-resolution input strategy is developed following [14]. We follow [14] to cut a high-resolution part $X^H \in$ from image X while resizing X to the same size of X^H to form low-resolution image X^L according to equation (5). The X^H provides abundant details while the X^L offers sufficient context semantic information.

$$\begin{aligned}&(X, Y) \xrightarrow{crop} (X^H, Y^H),\\ &(X, Y) \xrightarrow{resize} (X^L, Y^L),\\ &\text{where } X \in \mathbb{R}^{H\times W\times 3}, X^L, X^H \in \mathbb{R}^{h\times w\times 3},\\ &h = H \times 0.5, w = W \times 0.5\end{aligned} \tag{5}$$

Then the X^H and X^L are fed into feature extractor E to get pixel feature representation $\boldsymbol{F}^H$ and $\boldsymbol{F}^L$ given by:

$$\begin{aligned}(\boldsymbol{F}^H, Y^H) &= \{(\boldsymbol{f}_j^H, y_j)\}, j = 1, ..., N_H,\\ (\boldsymbol{F}^L, Y^L) &= \{(\boldsymbol{f}_j^L, y_j)\}, j = 1, ..., N_L.\end{aligned} \tag{6}$$

where N_L and N_H represent the number of low-resolution pixels and high-resolution pixels respectively. $\boldsymbol{f}_j^H$ and $\boldsymbol{f}_j^L$ stand for pixel feature extracted from X^H and X^L.

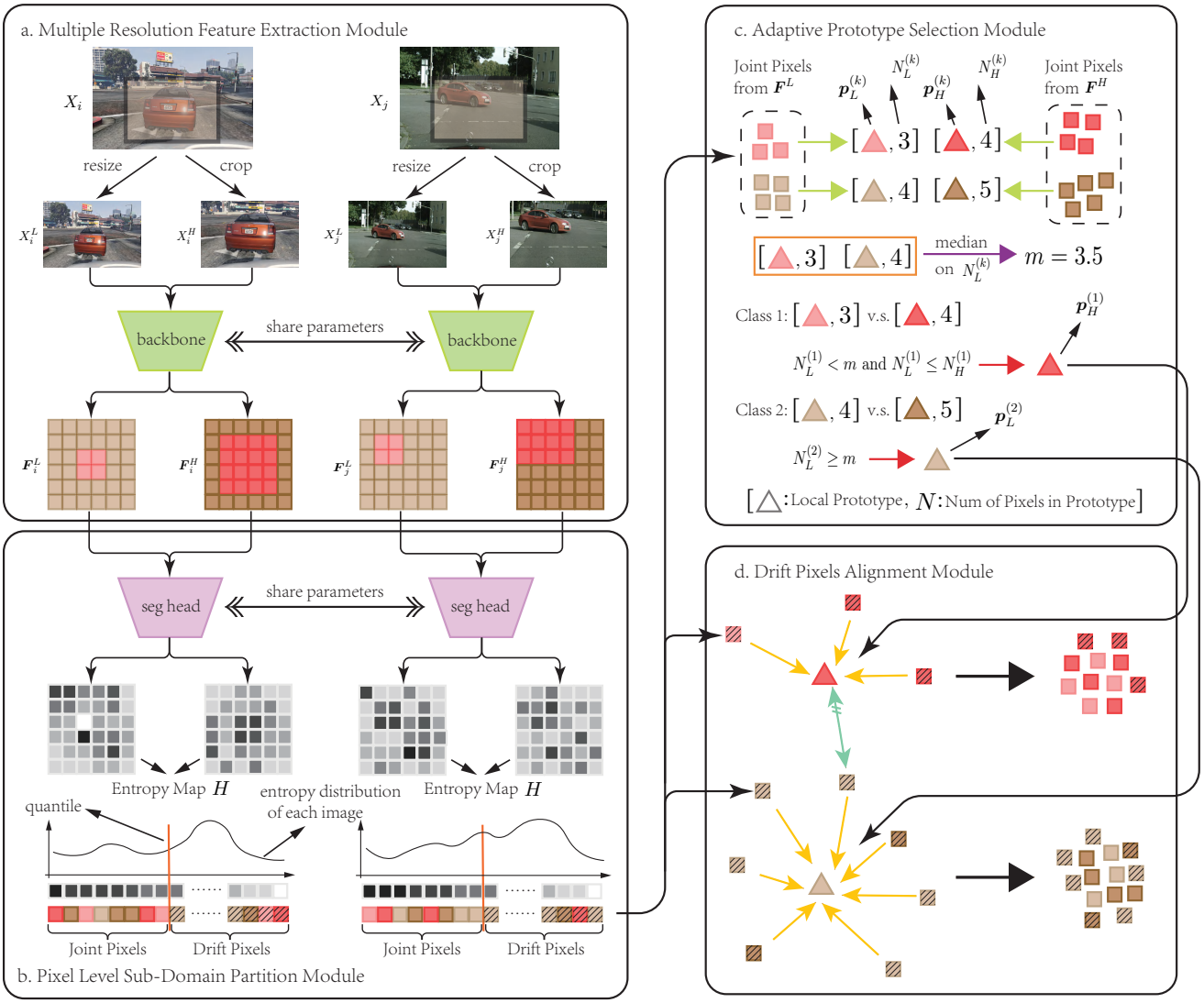


Figure 2: An overview of PiXL framework. At each training step, two images are sampled and fed into the subsequent modules. **Multiple Resolution Feature Extraction Module** extracts features from low- and high-resolution images. **Pixel-Level Sub-Domain Partition Module** arranges pixels from each image into joint pixels and drift pixels based on entropy. **Adaptive Prototype Selection Module** selects prototypes from different resolutions to balance context and details. **Drift Pixels Alignment Module** aligns drift pixels to local prototypes.

3.3. Pixel Level Sub-Domain Partition

Pixel entropy. In PiXL, we employ entropy as the measurement criteria to split the pixels into joint pixels and drift pixels. For a pixel feature representation $\boldsymbol{f}$ with its predicted probability distribution, its entropy h is given by:

$$h = -\sum_{k=1}^{C} p_k \log p_k. \tag{7}$$

where C is the number of classes and p_k is the probability of the pixel belonging to class k.

Entropy-based partition. The pixel-level sub-domain partition is based on the ranking of pixel entropy. We first concatenate both high- and low-resolution pixels to conduct ranking and partition following equation (8) and (9).

$$\begin{aligned} \boldsymbol{F} &= \{\boldsymbol{f}_1^L \cdots \boldsymbol{f}_{N_L}^L, \boldsymbol{f}_1^H \cdots \boldsymbol{f}_{N_H}^H\}, \\ H &= \{h_1^L \cdots h_{N_L}^L, h_1^H \cdots h_{N_H}^H\}. \end{aligned} \tag{8}$$

Then, these pixels are divided into two parts based on quantile $1-\eta$, which corresponds to entropy $h_{1-\eta}$. The former ones are joint pixels while the latter ones are drift pixels. It should be noted that our pixel ranking and partitioning

are performed on each distribution independently to guarantee a fair and targeted partition that is conducive to intra-distribution pixel alignment.

$$\hat{\boldsymbol{F}} = \{\hat{\boldsymbol{f}}_j | h_j < h_{1-\eta}\},$$
$$\tilde{\boldsymbol{F}} = \{\tilde{\boldsymbol{f}}_j | h_j \geq h_{1-\eta}\}. \tag{9}$$

3.4. Drift Pixels Alignment

Local prototypes. To conduct intra- and inter-distribution alignment on pixels from two sampled distributions, local prototypes for each category are calculated on joint pixels. Given joint pixels from distribution i and distribution j, local prototype for category k is given by:

$$\boldsymbol{p}^{(k)} = \frac{1}{|\hat{\boldsymbol{F}}_i^{(k)} \cup \hat{\boldsymbol{F}}_j^{(k)}|} \sum \hat{\boldsymbol{f}}_t,$$
$$\text{where } \hat{\boldsymbol{f}}_t \in \hat{\boldsymbol{F}}_i^{(k)} \cup \hat{\boldsymbol{F}}_j^{(k)}. \tag{10}$$

The $\hat{\boldsymbol{F}}_i^{(k)}$ and $\hat{\boldsymbol{F}}_j^{(k)}$ refer to feature of joint pixels belongs to category k from distribution i and j respectively.

Drift pixels asymmetric contrast. The intra- and inter-distribution pixels alignment is conducted in a contrastive manner. As we speculate the drift pixels are drifted from the global distribution, we design an asymmetric alignment strategy to not only push these pixels to local prototypes $\boldsymbol{p}$ but also suppress the misleading signals from drift pixels. Specifically, given a drift pixel $\tilde{\boldsymbol{f}}_t$ affiliated to category k as the anchor. The positive sample is the local prototypes $\boldsymbol{p}^{(k)}$ while other local prototypes $\{\boldsymbol{p}^{(j)} | j = 1, \cdots C, j \neq k\}$ are negative samples. Thus, for the single drift pixel $\tilde{\boldsymbol{f}}_t$, the asymmetric contrast loss is given as following:

$$\mathcal{L}_t^{DPA} = \mathcal{L}_{InfoNCE}\left(\tilde{\boldsymbol{f}}_t, \mathcal{P}_t, \mathcal{N}_t\right),$$
$$\textbf{where } \mathcal{P}_t = \{\boldsymbol{p}^{(k)}\},$$
$$\mathcal{N}_t = \{\boldsymbol{p}^{(j)} | j = 1, \cdots, C, j \neq k\}. \tag{11}$$

Additionally, in order to realize the asymmetric movement for drift pixels, we stop the gradients accumulation on $\boldsymbol{p}$ to suppress the signal from $\tilde{\boldsymbol{f}}_t$. Then, the asymmetric alignment loss to address all drift pixels from two distributions is given by equation (12).

$$\mathcal{L}_{DPA}(\tilde{\boldsymbol{F}}_i, \tilde{\boldsymbol{F}}_j) =$$
$$\frac{1}{|\tilde{\boldsymbol{F}}_i \cup \tilde{\boldsymbol{F}}_j|} \sum_{\tilde{\boldsymbol{f}}_t} \mathcal{L}_t^{DPA}. \tag{12}$$

3.5. Adaptive local prototype selection.

Considering the semantic variations in the features extracted from images of different resolutions, the local prototypes calculated from them should share complementary information. For common classes, like *sky*, *road* and *building*, context information gets compromised in features extracted from high-resolution image X^H. On the contrary, the pixel features extracted from low-resolution image X^L lack sufficient details in feature representation. Thus, we propose an adaptive local prototype selection module in PiXL to choose the most meaningful pixels per class to form local prototypes. As shown in Fig 2, local prototypes are calculated for each category according to equation (10) on joint pixels from $\boldsymbol{F}^L$ and $\boldsymbol{F}^H$ respectively, which are pixel features from two distributions. Then we quantify the number of pixels for each prototype and perform adaptive selection based on these statistical results.

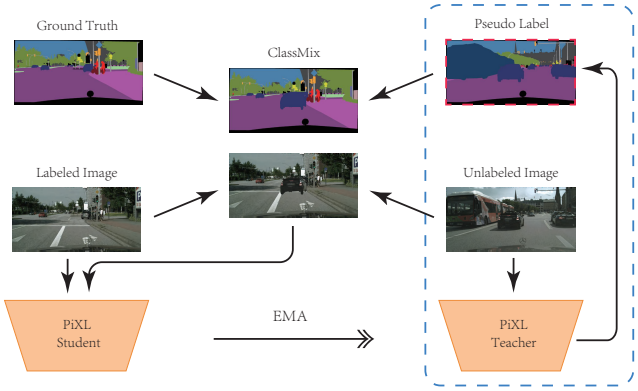


Figure 3: Pseudo Label Generation: For unlabeled images in SSL or UDA settings, our PiXL generates pseudo labels using an EMA-updated teacher model. ClassMix augmentation is also adapted to generate new distributions.

$$\{(\boldsymbol{p}_L^{(k)}, N_L^{(k)})\}, k \in \{1, \cdots, C\},$$
$$\{(\boldsymbol{p}_H^{(k)}, N_H^{(k)})\}, k \in \{1, \cdots, C\}. \tag{13}$$

In particular, we sort these prototypes $\boldsymbol{p}_L^{(k)}$ according to their corresponding $N_L^{(k)}$ and compute the median of $N_L^{(k)}$, denoted as m. For each low-resolution prototype that contains fewer pixels than m, we substitute the corresponding high-resolution prototype $\boldsymbol{p}_H^{(k)}$ for it when $N_H^{(k)} \geq N_L^{(k)}$ is satisfied. Otherwise, the $\boldsymbol{p}_L^{(k)}$ will be retained. The whole process can be formulated as equation (14).

$$\boldsymbol{p}^{(k)} = \begin{cases} \boldsymbol{p}_L^{(k)}, N_L^{(k)} \geq m, \\ \boldsymbol{p}_L^{(k)}, N_L^{(k)} < m \text{ and } N_H^{(k)} < N_L^{(k)}, \\ \boldsymbol{p}_H^{(k)}, \text{ otherwise.} \end{cases} \tag{14}$$

Based on this selection mechanism, we maintain the context feature of major categories by selecting prototypes from low-resolution images while emphasizing details of minor categories by utilizing the corresponding prototypes from high-resolution images.

Table 2: Comparison with previous methods in UDA setting on GTA5 → Cityscapes and SYNTHIA → Cityscapes

	Road	S.Walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.Bike	Bike	mIoU
GTA5 → Cityscapes																				
AdaptSeg[28]	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4
CBST[46]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
DACS[27]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
CorDA[31]	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47.0	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
BAPA[20]	94.4	61.0	88.0	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40.0	90.2	60.4	59.0	0.0	45.1	54.2	57.4
ProDA[43]	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
DAFormer[13]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
SePiCo[35]	96.9	76.7	89.7	55.5	49.5	53.2	60.0	64.5	90.2	50.3	90.8	74.5	44.2	93.3	77.0	79.5	63.6	61.0	65.3	70.3
HRDA[14]	96.4	74.4	91	**61.6**	51.5	57.1	63.9	69.3	91.3	48.4	94.2	79.0	52.9	93.9	**84.1**	85.7	75.9	63.9	67.5	73.8
CLUDA[29]	**97.1**	**78**	91	60.3	**55.3**	56.3	64.3	**71.5**	91.2	51.1	94.7	78.4	52.9	94.5	82.8	86.5	73	64.2	**69.7**	74.4
PiXL	97.0	77.6	**91.1**	59.9	54.1	**57.2**	**64.8**	69.1	**91.5**	**51.8**	**94.8**	**80.5**	**57.3**	**94.6**	83.8	**88.7**	**78.0**	**65.6**	67.8	**75.0**
SYNTHIA → Cityscapes																				
AdaptSeg[28]	79.2	37.2	78.8	-	-	-	9.9	10.5	78.2	-	80.5	53.5	19.6	67.0	-	29.5	-	21.6	31.3	37.2
CBST[46]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	-	78.3	60.6	28.3	81.6	-	23.5	-	18.8	39.8	42.6
DACS[27]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	-	90.8	67.5	38.3	82.9	-	38.9	-	28.5	47.6	48.3
CorDA[31]	**93.3**	**61.6**	85.3	19.6	5.1	37.8	36.6	42.8	84.9	-	90.4	69.7	41.8	85.6	-	38.4	-	32.6	53.9	55.0
BAPA[20]	91.7	53.8	83.9	22.4	0.8	34.9	30.5	42.8	86.8	-	88.2	66.0	34.1	86.6	-	51.3	-	29.4	50.5	53.3
ProDA[43]	87.8	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	–	84.4	74.2	24.3	88.2	–	51.1	–	40.5	45.6	55.5
DAFormer[13]	84.0	40.7	88.4	41.5	6.5	50.0	55.0	54.6	86.0	-	89.8	73.2	48.2	87.2	-	53.2	-	53.9	61.7	60.9
SePiCo[35]	87.0	52.6	88.5	40.6	**10.6**	49.8	57.0	55.4	86.8	-	86.2	75.4	52.7	**92.4**	-	**78.9**	-	53.0	62.6	64.3
HRDA[14]	85.2	47.7	88.8	49.5	4.8	57.2	65.7	60.9	85.3	-	92.9	79.4	52.8	89.0	-	64.7	-	63.9	64.9	65.8
CLUDA[29]	87.7	46.9	**90.2**	49	7.9	**59.5**	**66.9**	58.5	88.3	-	**94.6**	80.1	**57.1**	89.8	-	68.2	-	65.5	**65.8**	66.8
PiXL	89.9	53.8	90.1	**52.8**	7.1	58.8	49.9	**63.5**	**88.4**	-	**94.6**	**80.5**	55.9	90.8	-	68.1	-	**67.0**	62.9	**67.1**

Table 3: Comparison with previous methods in the semi-supervised setting with different proportions of the labeled image on Cityscapes.

	1/30(100)	1/8(372)	1/4(744)
ClassMix[22]	54.07	61.35	63.63
SemiSegContrast[2]	64.90	70.10	71.70
CCT[24]	-	74.12	75.99
GCT[17]	-	72.66	76.11
MT[26]	-	72.03	74.47
AEL[15]	-	**77.90**	79.01
U2PL[33]	-	76.48	78.51
CPS[6]	-	77.62	**79.21**
ReCo[18]	60.28	66.44	67.53
SegSDE[12]	62.09	68.01	69.38
PiXL	**71.73**	76.37	78.91

Table 4: Comparison with previous methods in the fully supervised setting on Cityscapes.

	Cityscapes
HRNetV2[30]	81.10
HRViT-b1[9]	81.63
HRViT-b2[9]	82.81
SegFormer[37]	84.00
Mask2Former[7]	84.30
SeMask[16]	**84.98**
PiXL	81.43

3.6. Loss

In PiXL, we employ contrastive learning loss to handle pixel-level variance while the cross-entropy(CE) loss is also adopted to achieve a meaningful implicit global semantic distribution. Given pixel features $\boldsymbol{F}$, its CE loss is calculated following equation (15).

$$\mathcal{L}_{CE}(\boldsymbol{F}) = -\frac{1}{|\boldsymbol{F}|}\sum_{t=1,\cdots,|\boldsymbol{F}|}\sum_{k=1,\cdots,C}\mathbb{I}_{[y_t=k]}\log p_{t_k} \tag{15}$$

where y_t is label of feature $\boldsymbol{f}_t$, p_{t_k} denotes the probability of $\boldsymbol{f}_t$ belongs to class k.

Thus, the CE loss in a training step is formulated as follows:

$$\begin{aligned}\mathcal{L}^*_{CE} = \mathcal{L}_{CE}(\boldsymbol{F}^L_i) + \mathcal{L}_{CE}(\boldsymbol{F}^L_j) + \\ \lambda_H\mathcal{L}_{CE}(\boldsymbol{F}^H_i) + \lambda_H\mathcal{L}_{CE}(\boldsymbol{F}^H_j).\end{aligned} \tag{16}$$

Following [13, 14, 29], we adopt the Thing-Class ImageNet Feature Distance, denoted as $\mathcal{L}_{FD}$, to utilize the recognition ability of ImageNet pre-trained model. Therefore, the total loss employed in our PiXL framework at each training step is a combination of equation (12), (16) and $\mathcal{L}_{FD}$ as follows:

$$\mathcal{L}_{PiXL} = \mathcal{L}^*_{CE} + \lambda_{FD}\mathcal{L}_{FD} + \mathcal{L}_{DPA}. \tag{17}$$

4. Experiments

4.1. Setup

4.1.1 Dataset

GTA-V: A large-scale synthetic dataset contains 24,966 annotated images with resolution 1914×1052. We resize the image to 1280×720. **SYNTHIA:** A collection of generated urban images including 9,400 images whose resolution is 1280 × 760. **Cityscapes:** A real-world street scene dataset contains 2,975 training images and 500 validation images with resolution 2048×1024.

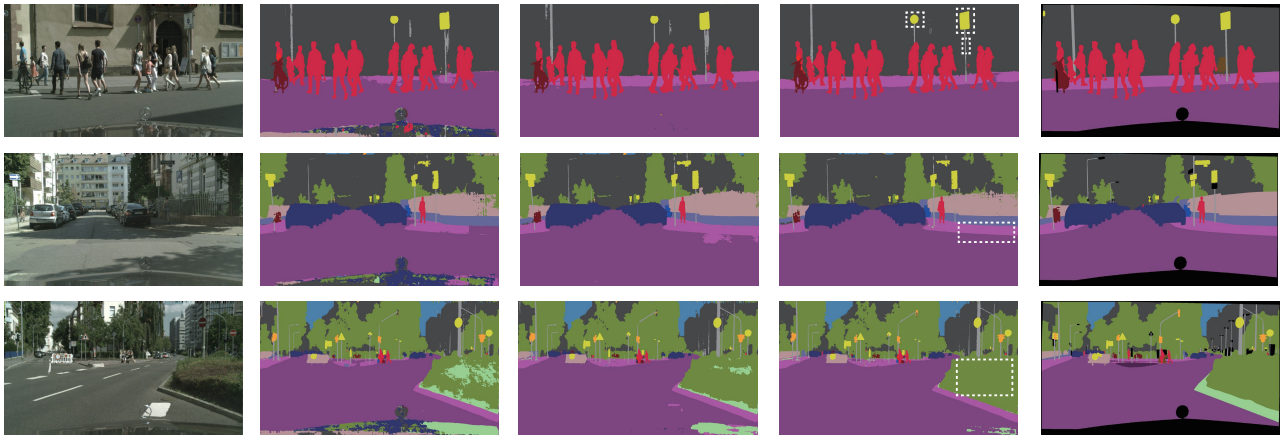


Figure 4: Qualitative comparison with baseline and SOTA methods in UDA setting on GTA5 → Cityscapes. PiXL performs better on boundary pixels and internal pixels.

4.1.2 Task setting

Fully supervised learning: We use all the training images from Cityscapes to train our PiXL and evaluate that on the corresponding validation part. **Semi-Supervised learning:** We randomly select 1/4, 1/8, and 1/30 labeled images from the Cityscapes training set while utilizing the left images as unlabeled images to train PiXL. Performance is reported on the validation part. **Unsupervised Domain Adaptation:** We evaluate our PiXL on GTA5→Cityscapes and SYNTHIA →Cityscapes.

4.1.3 Implementation details

Network architecture: We adopt the HRDA model[14] as our baseline which consists of a MiT-B5 encoder [37] and a feature fusion decoder from DAFormer [13]. To further validate the effectiveness and plug-and-play property of PiXL. We also implement PiXL based on DeepLabV3+[4] model, referring to supplementary for more details. We implement the PiXL based on the mmsegmentation framework[1].

Training: We follow [29] *et al*. to set the training parameters, i.e. a batch size of 2, λ_H is 0.1, λ_{FD} is 0.005, the optimizer is AdamW with a learning rate of 6×10^{-5} for the encoder and 6×10^{-4} for the decoder with linear warmup policy, DACS data augmentation, rare class sampling strategy on labeled training data, the self-training method for unlabeled training data. In UDA and SSL settings, we employ the Exponential Moving Average (EMA) updated teacher model to generate the pseudo labels for unlabeled images as shown in Fig 3. Moreover, we extend the training epoch to 60,000 and set the $\eta = 0.2$ and decrease in a linear manner until $\eta = 0.001$. The model is trained on a single Tesla V100 with 32 GB Memory.

[1]https://github.com/open-mmlab/mmsegmentation

4.1.4 Evaluation

The evaluation metric adopts in our experiment is mIoU which averages the intersection over Union in each category.

4.2. Results

4.2.1 Unsupervised Domain Adaptation

In the UDA setting, our PiXL outperforms the baseline model HRDA and achieves commensurate performance with the SOTA. In GTA → Cityscapes and SYNTHIA → Cityscapes, our model PiXL surpassed the baseline by a margin of 1.2% and 1.3% respectively in Tab 2. Compared with the SOTA model CLUDA, our method outperforms them by a margin of 0.6% and 0.3%. Specifically, in GTA → Cityscapes, PiXL obviously improves the performance on rare classes, like *pole*, *motorbike etc*. Moreover, the performance improvement is also consistent in SYNTHIA → Cityscapes. In the common classes pixels occupy the majority, PiXL also achieves comparable results or surpasses the previous works, like *vegetation*, *road*, *sidewalk etc*. Fig 4 provides more details. Experiments in the UDA setting validate the ability of PiXL in mining more meaningful pixel features not only in major classes but also in minor classes to adapt the model across domains at the pixel level.

4.2.2 Semi-Supervised Semantic Segmentation

In the semi-supervised segmentation setting, our PiXL achieves competitive performance with the SOTA models

in Tab 3. When the number of labeled images decreases sharply, PiXL maintains its excellent performance. Especially in 3.3% images labeled setting, PiXL outperforms SOTA models by a large margin in Fig 6. The robustness of limited annotations validates PiXL's ability to thrive in label-efficient scenarios, which further highlights the strengths of pixel learning.

4.2.3 Fully Supervised Semantic Segmentation

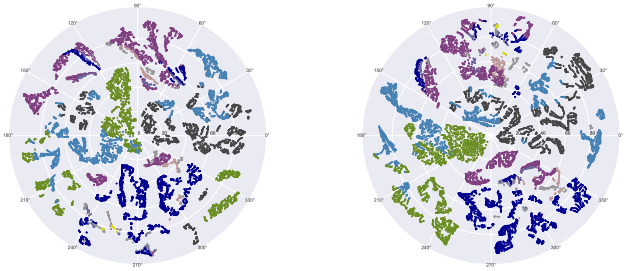
In the fully supervised setting as shown in Tab 4, PiXL achieves comparable performance with HRNetV2 and HRViT-b1 which both concentrated on mining the multi-resolution feature representation. The SOTA methods, like SeMask and Mask2Former, surpassed our model. However, compared with those universal segmentation frameworks, PiXL concentrates on generalizing to label-efficient scenarios while tailored optimization in the fully supervised setting is not included in this work.

4.2.4 Ablation Study

We conduct an ablation study to evaluate each module in PiXL in Tab 5. Compared to the baseline, our drift pixels alignment module improves the performance by 0.9%. This proves our contrast-based pixel learning scheme performs better in addressing pixel-level variance and approximating the implicit global distribution. The addition of an adaptive prototype selection strategy further improves the performance of PiXL to 75.0%, which outperforms the baseline by a margin of 1.2%. It proves the APS module strengthens the generation of local distribution representations from multi-resolution pixel features. The ablation on different quantiles in Tab 6 proves that PiXL is relatively robust to the selection of quantiles. Meanwhile, given the varying degree of variance at the pixel level across different datasets and task settings, it is reasonable that a delicately selected quantile may yield superior performance.

4.2.5 Visualization

We adopt t-SNE to compare the pixel feature presentation of PiXL and the SOTA method CLUDA. We also compute the Calinski-Harabasz Index (CH) and Davies-Bouldin Index (DB). A larger CH and a smaller DB indicate better inter-class separability and intra-class compactness. The visualization in Fig 5 demonstrates that PiXL produces more compact and discriminative pixel features, especially the compactness in *vegetation* (green) and *car* (dark blue). The CH and DB on PiXL are 1240.2, 6.9 while 693.8, 7.3 on CLUDA, which further validates that pixel learning guarantees more meaningful pixel feature representation. In comparison, misleading signals from drift pixels in CLUDA reduce the quality of feature representation.



(a) CLUDA(SOTA): CH:693.8, DB:7.3 (b) PiXL(Ours): CH:1240.2, DB:6.9

Figure 5: Visualization of pixel feature representation on CLUDA and PiXL. A larger CH and a smaller DB indicate better inter-class separability and intra-class compactness.

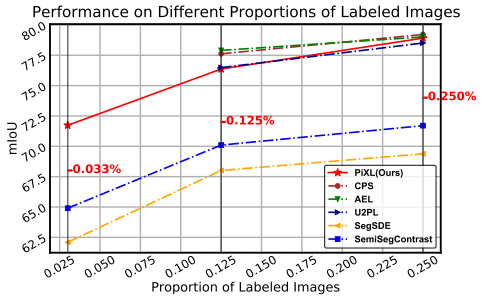


Figure 6: Performance on different proportions of labeled images:

Table 5: Component Ablation of PiXL.

	APS	DPA	mIoU	↑
baseline			73.8	
PiXL		✓	74.7	**+0.9**
PiXL	✓	✓	**75.0**	**+1.2**

Table 6: Quantile Ablation of PiXL.

η	**0.1**	**0.2**	**0.4**	**0.6**	**0.8**
mIoU	74.8	75.0	75.0	**75.2**	74.1

5. Conclusion

In this paper, we raise the concept of pixel learning to dive semantic segmentation into pixel level and explicitly address the pixel-level feature variance. We propose the PiXL framework which gradually approximates the global distribution by continually addressing intra- and inter-distribution (image) alignment in a divide-and-conquer manner. PiXL exhibits competitive performance in various settings and has revealed the feasibility of pixel learning in segmentation tasks, deserving further study.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4981–4990, 2018.

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

[3] Yuxiang Cai, Yingchun Yang, Yongheng Shang, Zhenqian Chen, Zhengwei Shen, and Jianwei Yin. Iterdanet: Iterative intra-domain adaptation for semantic segmentation of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022.

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

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

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

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022.

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

[9] Jiaqi Gu, Hyoukjun Kwon, Dilin Wang, Wei Ye, Meng Li, Yu-Hsin Chen, Liangzhen Lai, Vikas Chandra, and David Z Pan. Multi-scale high-resolution vision transformer for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12094–12103, 2022.

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

[11] Kelei He, Chen Gan, Zhuoyuan Li, Islem Rekik, Zihao Yin, Wen Ji, Yang Gao, Qian Wang, Junfeng Zhang, and Dinggang Shen. Transformers in medical image analysis: A review. *Intelligent Medicine*, 2022.

[12] Lukas Hoyer, Dengxin Dai, Yuhua Chen, Adrian Koring, Suman Saha, and Luc Van Gool. Three ways to improve semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11130–11140, 2021.

[13] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9924–9935, 2022.

[14] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pages 372–391. Springer, 2022.

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

[16] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021.

[17] Zhanghan Ke, Di Qiu, Kaican Li, Qiong Yan, and Rynson WH Lau. Guided collaborative training for pixel-wise semi-supervised learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*, pages 429–445. Springer, 2020.

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

[19] Xiaoming Liu, Quan Yuan, Yaozong Gao, Kelei He, Shuo Wang, Xiao Tang, Jinshan Tang, and Dinggang Shen. Weakly supervised segmentation of covid19 infection with scribble annotation on ct images. *Pattern recognition*, 122:108341, 2022.

[20] Yahao Liu, Jinhong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8801–8811, 2021.

[21] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 142–158. Springer, 2020.

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

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[24] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency

training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[25] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

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

[27] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021.

[28] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018.

[29] Midhun Vayyat, Jaswin Kasi, Anuraag Bhattacharya, Shuaib Ahmed, and Rahul Tallamraju. Cluda: Contrastive learning in unsupervised domain adaptation for semantic segmentation. *arXiv preprint arXiv:2208.14227*, 2022.

[30] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[31] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8525, 2021.

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

[33] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4248–4257, 2022.

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

[35] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[36] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021.

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

[39] Zizheng Yan, Xianggang Yu, Yipeng Qin, Yushuang Wu, Xiaoguang Han, and Shuguang Cui. Pixel-level intra-domain adaptation for semantic segmentation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 404–413, 2021.

[40] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 763–778. Springer, 2020.

[41] Qiying Yu, Jieming Lou, Xianyuan Zhan, Qizhang Li, Wangmeng Zuo, Yang Liu, and Jingjing Liu. Adversarial contrastive learning via asymmetric infonce. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 53–69. Springer, 2022.

[42] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel Ni, and Harry Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2022.

[43] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12414–12424, 2021.

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

[45] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[46] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the Eu-*

ropean conference on computer vision (ECCV), pages 289–305, 2018.