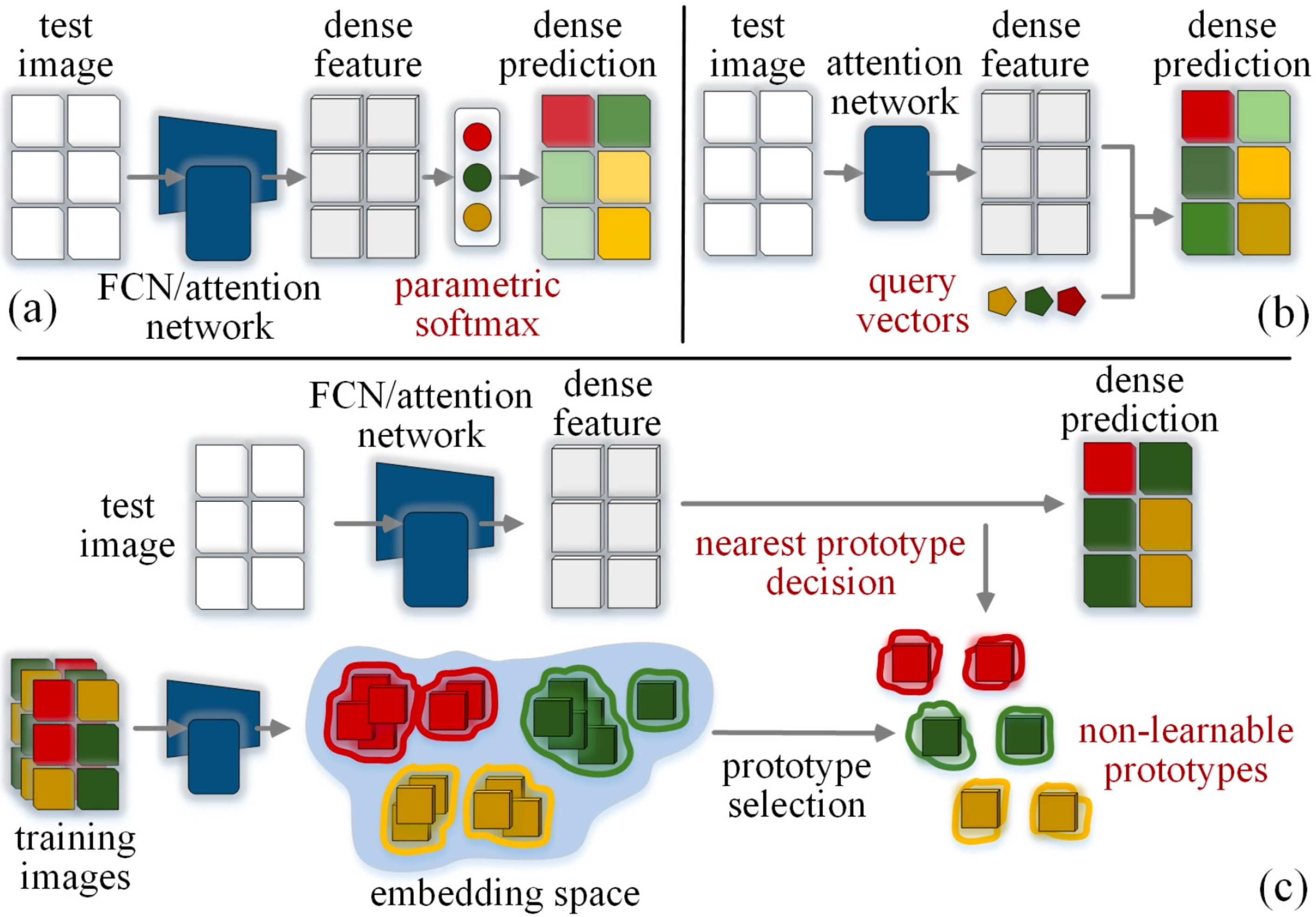
Rethinking Semantic Segmentation: A Prototype View

Tianfei Zhou[1], Wenguan Wang[2,1]*, Ender Konukoglu[1], Luc Van Gool[1]
[1] Computer Vision Lab, ETH Zurich [2] ReLER, AAII, University of Technology Sydney

CVPR 2022, Oral

Parametric & Non-Parametric

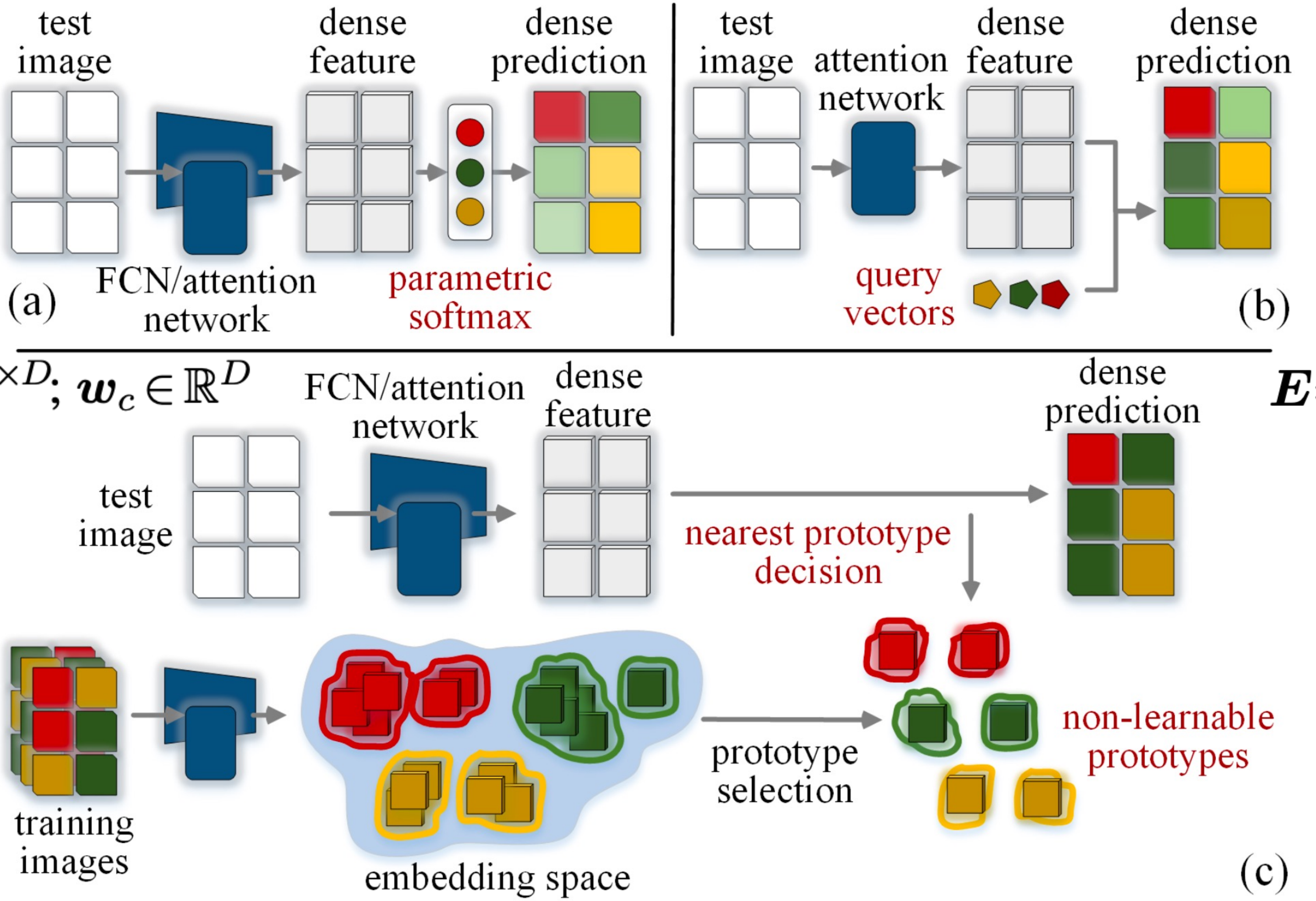


Introduction

Question:

1. What are the relation and difference between FCN based and attention based mask decoding strategies?
2. If the learnable query vectors indeed implicitly capture some intrinsic properties of data, is there any better way to achieve this?

Parametric & Non-Parametric



$\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C] \in \mathbb{R}^{C \times D}; \ \boldsymbol{w}_c \in \mathbb{R}^D$

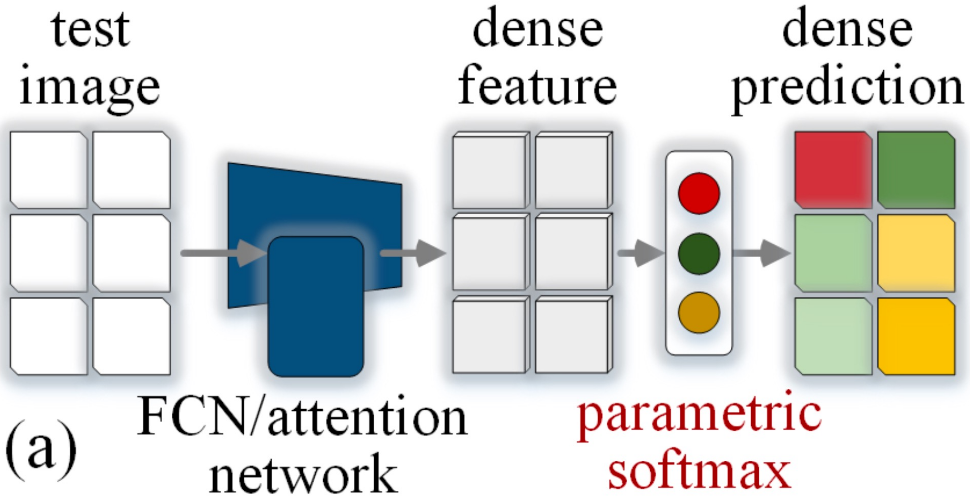
$\boldsymbol{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_C] \in \mathbb{R}^{C \times D}$

Introduction

Question:

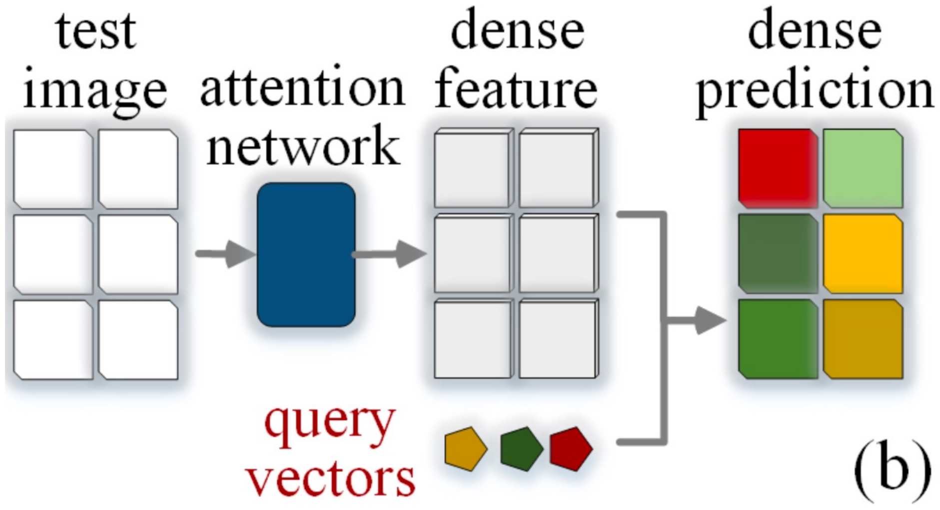
1. What are the relation and difference between FCN based and attention based mask decoding strategies?
2. If the learnable query vectors indeed implicitly capture some intrinsic properties of data, is there any better way to achieve this?
3. What are the limitations of this learnable prototype based parametric paradigm?
4. How to address these limitations?

Parametric Prototype Learning



$$\boldsymbol{W}=[\boldsymbol{w}_1,\cdots,\boldsymbol{w}_C]\in\mathbb{R}^{C\times D};\ \boldsymbol{w}_c\in\mathbb{R}^D$$

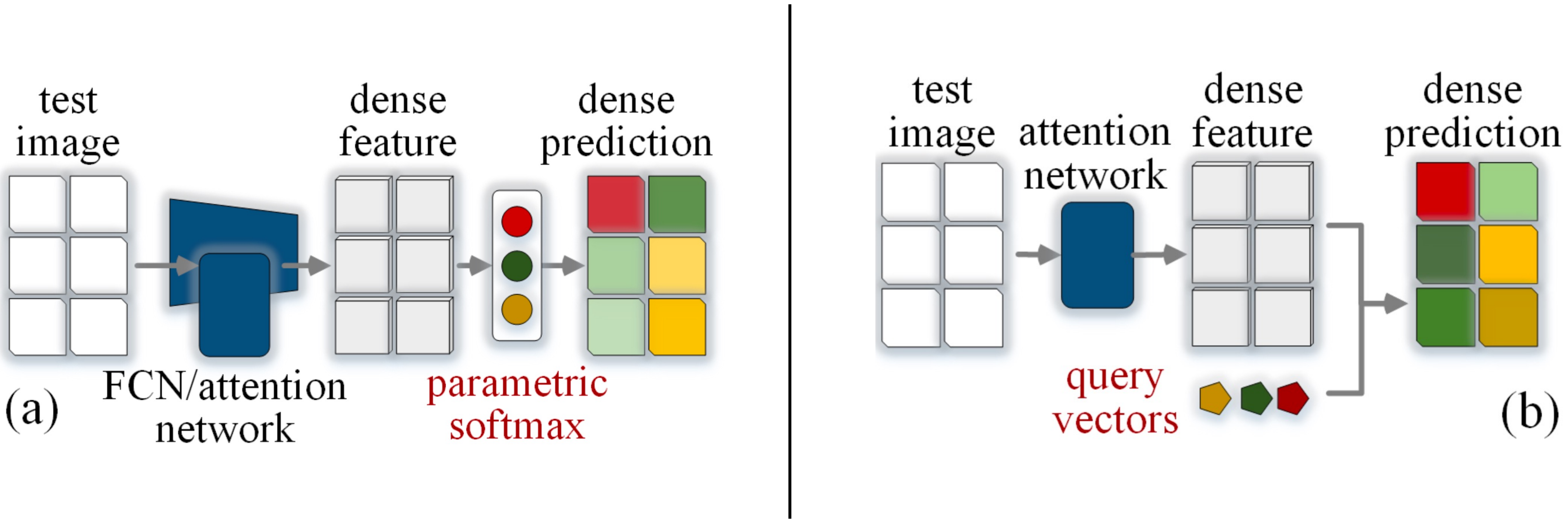
$$p(c|\boldsymbol{i})=\frac{\exp(\boldsymbol{w}_c^\top\boldsymbol{i})}{\sum_{c'=1}^{C}\exp(\boldsymbol{w}_{c'}^\top\boldsymbol{i})},$$



$$\boldsymbol{E}=[\boldsymbol{e}_1,\cdots,\boldsymbol{e}_C]\in\mathbb{R}^{C\times D}$$

$$p(c|\boldsymbol{i})=\frac{\exp(\boldsymbol{e}_c*\boldsymbol{i})}{\sum_{c'=1}^{C}\exp(\boldsymbol{e}_{c'}*\boldsymbol{i})},$$

Parametric Prototype Learning



$$p(c|\boldsymbol{i}) = \frac{\exp(-\langle \boldsymbol{i}, \boldsymbol{g}_c \rangle)}{\sum_{c'=1}^{C} \exp(-\langle \boldsymbol{i}, \boldsymbol{g}_{c'} \rangle)},$$

<·,·> distance measure

Introduction

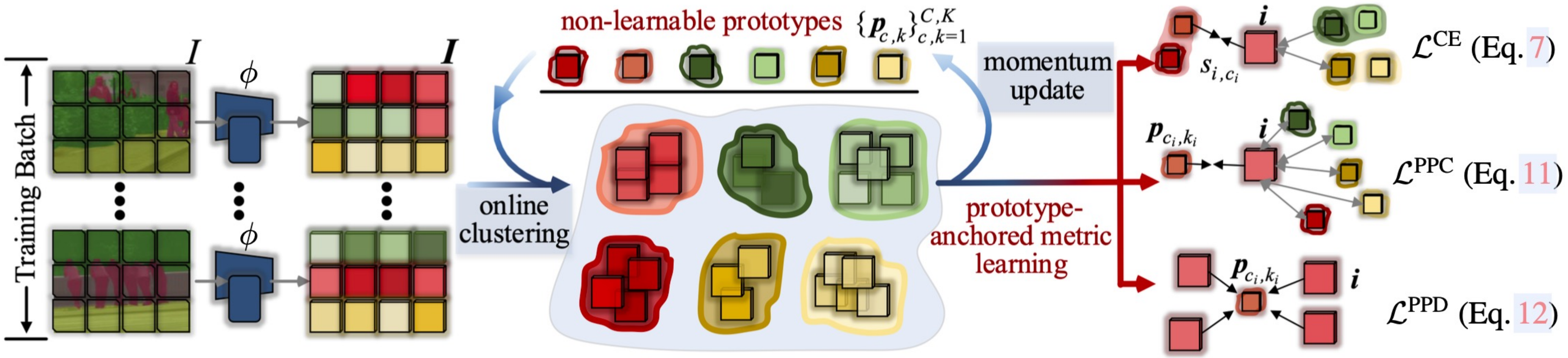
Limitations:

1. Single learned prototype per class, insufficient to rich intra-class variance.

2. Much parameters needed for prototype learning, hurting generalizability.

3. Ignoring known inductive biases, intra-class compactness about feature distribution.

Architecture illustration



$I \in R^{H \times W \times D}$ $\qquad$ $p_{c,k} \in R^D$

Non-Learnable Prototype based pixel classification

Pixel features, $I \in R^{H \times W \times D}$, CK non-learnable prototypes $\{p_{c,k} \in R^D\}_{c,k=1}^{C,K}$.

The category prediction of each pixel $i \in I$.

$$\hat{c}_i = c^*, \text{ with } (c^*, k^*) = \underset{(c,k)}{\arg\min} \{\langle \boldsymbol{i}, \boldsymbol{p}_{c,k} \rangle\}_{c,k=1}^{C,K},$$

Probability of Pixel i over the C class,

$$p(c|\boldsymbol{i}) = \frac{\exp(-s_{i,c})}{\sum_{c'=1}^{C} \exp(-s_{i,c'})}, \text{ with } s_{i,c} = \min\{\langle \boldsymbol{i}, \boldsymbol{p}_{c,k} \rangle\}_{k=1}^{K},$$

update prototypes,

$$\boldsymbol{p}_{c,k} \leftarrow \mu \boldsymbol{p}_{c,k} + (1 - \mu)\bar{\boldsymbol{i}}_{c,k},$$

Within-Class Online Clustering

Given pixels $I^c = \{i_n\}_{n=1}^{N}$ in a training batch that belong to class c.

K prototypes $\{p_{c,k}\}_{\mathrm{k}=1}^{K}$ of class c.

Pixel-Prototype mapping, $L^c = \left[l_{i_n}\right]_{n=1}^{N} \in \{0,1\}^{K \times N}$,

$$l_{i_n} = \left[l_{i_n,k}\right]_{k=1}^{K} \in \{0,1\}^{K}$$

Pixel embedding X^c, Prototypes P^c

$$\max_{\boldsymbol{L}^c} \mathtt{Tr}(\boldsymbol{L}^{c\top} \boldsymbol{P}^{c\top} \boldsymbol{X}^c),$$
$$s.t. \quad \boldsymbol{L}^c \in \{0,1\}^{K \times N}, \boldsymbol{L}^{c\top} \mathbf{1}^K = \mathbf{1}^N, \boldsymbol{L}^c \mathbf{1}^N = \frac{N}{K} \mathbf{1}^K,$$

Within-Class Online Clustering

$$\max_{\boldsymbol{L}^c} \mathtt{Tr}(\boldsymbol{L}^{c\top}\boldsymbol{P}^{c\top}\boldsymbol{X}^c) + \kappa h(\boldsymbol{L}^c),$$
$$\textit{s.t.} \quad \boldsymbol{L}^c \in \mathbb{R}_+^{K\times N},\ \boldsymbol{L}^{c\top}\mathbf{1}^K = \mathbf{1}^N, \boldsymbol{L}^c\mathbf{1}^N = \frac{N}{K}\mathbf{1}^K,$$

Solution using Sinkhorn-Knopp iteration.

$$\boldsymbol{L}^c = \mathtt{diag}(\boldsymbol{u})\exp\big(\frac{\boldsymbol{P}^{c\top}\boldsymbol{X}^c}{\boldsymbol{\kappa}}\big)\mathtt{diag}(\boldsymbol{v}),$$

Training Objects

CE loss

$$\mathcal{L}_{\text{CE}} = -\log p(c_i|\boldsymbol{i}) = -\log \frac{\exp(-s_{i,c_i})}{\exp(-s_{i,c_i}) + \sum_{c' \neq c_i} \exp(-s_{i,c'})}.$$

Pixel-Prototype Contrastive Learning

$$\mathcal{L}_{\text{PPC}} = -\log \frac{\exp(\boldsymbol{i}^\top \boldsymbol{p}_{c_i,k_i}/\tau)}{\exp(\boldsymbol{i}^\top \boldsymbol{p}_{c_i,k_i}/\tau) + \sum_{\boldsymbol{p}^- \in \mathcal{P}^-} \exp(\boldsymbol{i}^\top \boldsymbol{p}^- / \tau)},$$

Pixel-Prototype Distance Optimization

$$\mathcal{L}_{\text{PPD}} = (1 - \boldsymbol{i}^\top \boldsymbol{p}_{c_i,k_i})^2.$$

$$\mathcal{L}_{\text{SEG}} = \mathcal{L}_{\text{CE}} + \lambda_1 \mathcal{L}_{\text{PPC}} + \lambda_2 \mathcal{L}_{\text{PPD}}.$$

Experiments

Method	Backbone	# Param (M)	mIoU (%)
DeepLabV3+ [ECCV18] [16]	ResNet-101 [46]	62.7	44.1
OCR [ECCV20] [131]	HRNetV2-W48 [110]	70.3	45.6
MaskFormer [NeurIPS21] [20]	ResNet-101 [46]	60.0	46.0
UperNet [ECCV20] [119]	Swin-Base [79]	121.0	48.4
OCR [ECCV20] [131]	HRFormer-B [132]	70.3	48.7
SETR [CVPR21] [141]	ViT-Large [31]	318.3	50.2
Segmenter [ICCV21] [102]	ViT-Large [31]	334.0	51.8
†MaskFormer [NeurIPS21] [20]	Swin-Base [79]	102.0	52.7
FCN [CVPR15] [80]	ResNet-101 [46]	68.6	39.9
Ours		68.5	**41.1** ↑ 1.2
HRNet [PAMI20] [110]	HRNetV2-W48 [110]	65.9	42.0
Ours		65.8	**43.0** ↑ 1.0
Swin [ICCV21] [79]	Swin-Base [79]	90.6	48.0
Ours		90.5	**48.6** ↑ 0.6
SegFormer [NeurIPS21] [120]	MiT-B4 [120]	64.1	50.9
Ours		64.0	**51.7** ↑ 0.8

†: backbone is pre-trained on ImageNet-22K.

Table 1. **Quantitative results** (§5.2) on ADE20K [142] `val`.

Method	Backbone	# Param (M)	mIoU (%)
PSPNet [CVPR17] [137]	ResNet-101 [46]	65.9	78.4
PSANet [ECCV18] [138]	ResNet-101 [46]	-	78.6
AAF [ECCV18] [60]	ResNet-101 [46]	-	79.1
Segmenter [ICCV21] [102]	ViT-Large [31]	322.0	79.1
ContrastiveSeg [ICCV21] [113]	ResNet-101 [46]	58.0	79.2
MaskFormer [NeurIPS21] [20]	ResNet-101 [46]	60.0	80.3
DeepLabV3+ [ECCV18] [16]	ResNet-101 [46]	62.7	80.9
OCR [ECCV20] [131]	HRNetV2-W48 [110]	70.3	81.1
FCN [CVPR15] [80]	ResNet-101 [46]	68.6	78.1
Ours		68.5	**79.1** ↑ 1.0
HRNet [PAMI20] [110]	HRNetV2-W48 [110]	65.9	80.4
Ours		65.8	**81.1** ↑ 0.7
Swin [ICCV21] [79]	Swin-Base [79]	90.6	79.8
Ours		90.5	**80.6** ↑ 0.8
SegFormer [NeurIPS21] [120]	MiT-B4 [120]	64.1	80.7
Ours		64.0	**81.3** ↑ 0.6

Table 2. **Quantitative results** (§5.2) on Cityscapes [23] `val`.

Experiments

Method	Backbone	# Param (M)	mIoU (%)
SVCNet [CVPR19] [29]	ResNet-101 [46]	-	39.6
DANet [CVPR19] [35]	ResNet-101 [46]	69.1	39.7
SpyGR [CVPR20] [68]	ResNet-101 [46]	-	39.9
MaskFormer [NeurIPS21] [20]	ResNet-101 [46]	60.0	39.8
ACNet [ICCV19] [36]	ResNet-101 [46]	-	40.1
OCR [ECCV20] [131]	HRNetV2-W48 [110]	70.3	40.5
FCN [CVPR15] [80]	ResNet-101 [46]	68.6	32.5
Ours		68.5	**34.0** ↑ **1.5**
HRNet [PAMI21] [110]	HRNetV2-W48 [110]	65.9	38.7
Ours		65.8	**39.9** ↑ **1.2**
Swin [ICCV21] [79]	Swin-Base [79]	90.6	41.5
Ours		90.5	**42.4** ↑ **0.9**
SegFormer [NeurIPS21] [120]	MiT-B4 [120]	64.1	42.5
Ours		64.0	**43.3** ↑ **0.8**

Table 3. **Quantitative results** (§5.2) on COCO-Stuff [10] test.

Ablation Study

parametric v.s. nonparametric

Method	# Proto	150 classes		300 classes		500 classes		700 classes		847 classes	
		mIoU (%)	# Param (M)	mIoU (%)	# Param (M)	mIoU (%)	# Param (M)	mIoU (%)	# Param (M)	mIoU (%)	# Param (M)
parametric	1	45.1	27.48 (0.12)	36.5	27.62 (0.23)	25.7	27.80 (0.39)	19.8	27.98 (0.54)	16.5	28.11 (0.65)
nonparametric (`Ours`)	1	**45.5** ↑ **0.4**	27.37 (0)	**37.2** ↑ **0.7**	27.37 (0)	**26.8** ↑ **1.1**	27.37 (0)	**21.2** ↑ **1.4**	27.37 (0)	**18.1** ↑ **1.6**	27.37 (0)
parametric	10	45.7	28.56 (1.2)	37.0	29.66 (2.3)	26.6	31.26 (3.9)	20.8	32.86 (5.4)	17.7	33.96 (6.5)
nonparametric (`Ours`)	10	**46.4** ↑ **0.7**	27.37 (0)	**37.8** ↑ **0.8**	27.37 (0)	**27.9** ↑ **1.3**	27.37 (0)	**22.1** ↑ **1.3**	27.37 (0)	**19.4** ↑ **1.7**	27.37 (0)

Table 4. **Scalability study** (§5.3) of our nonparametric model against the parametric baseline (*i.e.*, SegFormer [120]) on ADE20K [142]. For each model variant, we report its segmentation mIoU, parameter numbers of the entire model as well as the prototypes (in the bracket).

Ablation Study

Design

$\mathcal{L}_{\text{CE}}$ (Eq. 7)	$\mathcal{L}_{\text{PPC}}$ (Eq. 11)	$\mathcal{L}_{\text{PPD}}$ (Eq. 12)	mIoU (%)
✓			45.0
✓	✓		45.9
✓		✓	45.4
✓	✓	✓	46.4

(a) Training Objective $\mathcal{L}$

# Prototype	mIoU (%)
$K = 1$	45.5
$K = 5$	46.0
$K = 10$	46.4
$K = 20$	46.5
$K = 50$	46.4

(b) Prototype Number K

Coefficient μ	mIoU (%)
$\mu = 0$	44.9
$\mu = 0.9$	45.9
$\mu = 0.99$	46.0
$\mu = 0.999$	46.4
$\mu = 0.9999$	46.3

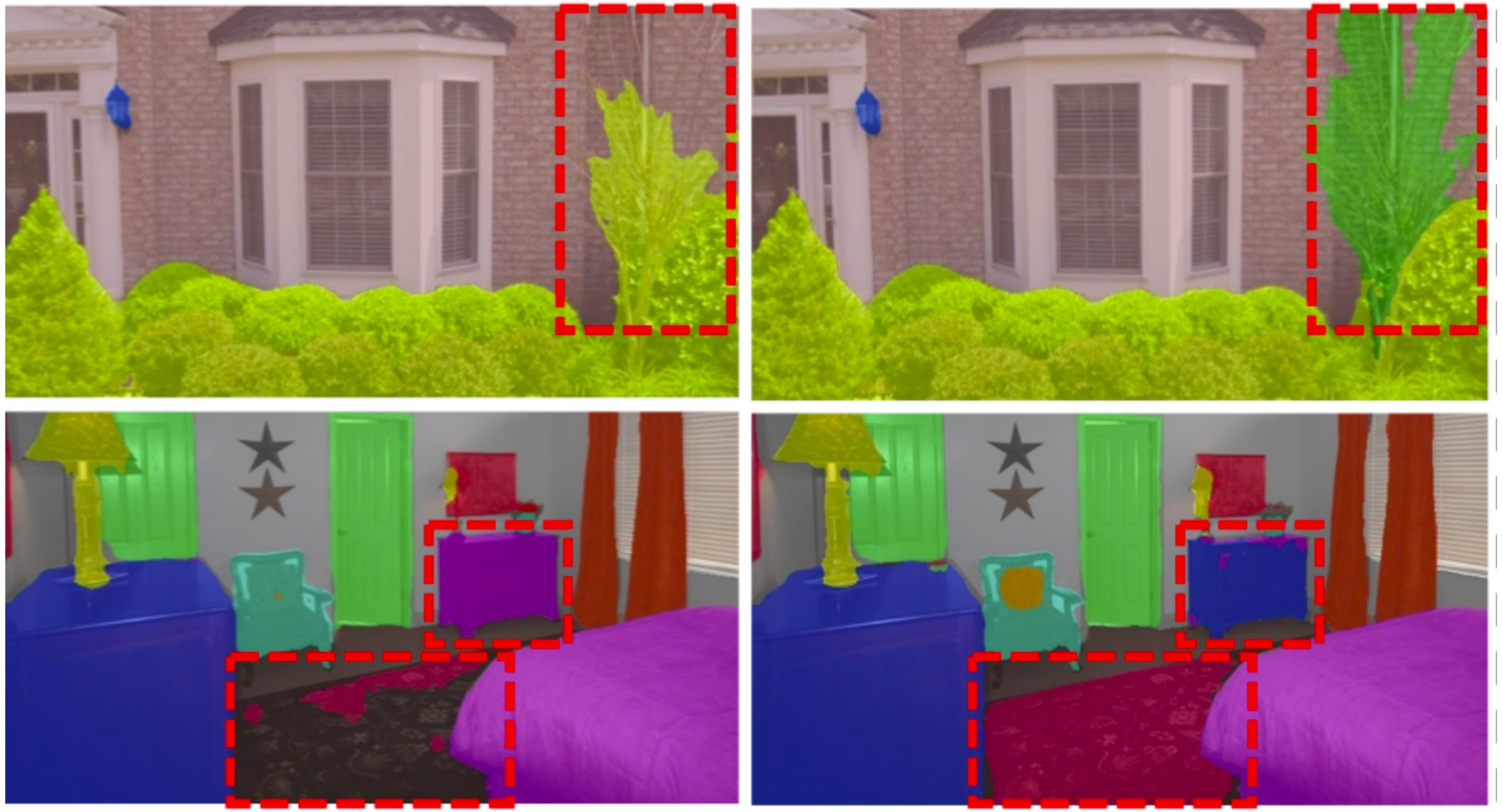
(c) Momentum Coefficient μ

Distance Measure	mIoU (%)
Standard	45.7
Huberized	45.2
Cosine	46.4

(d) Distance Measure

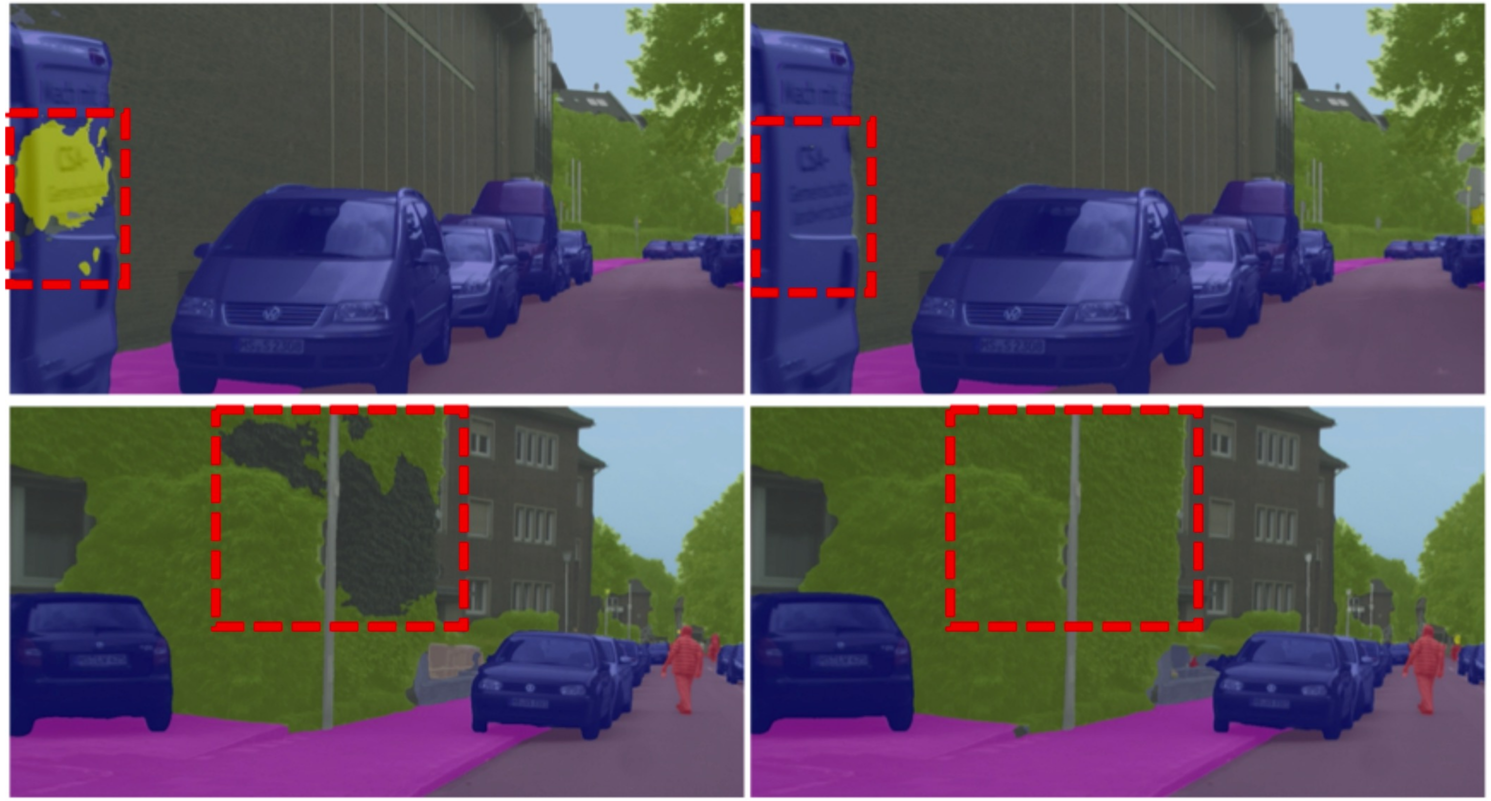
Table 5. A set of **ablative studies** (§5.4) on ADE20K [142] `val`. All model variants use MiT-B2 [120] as the backbone.

Visualization



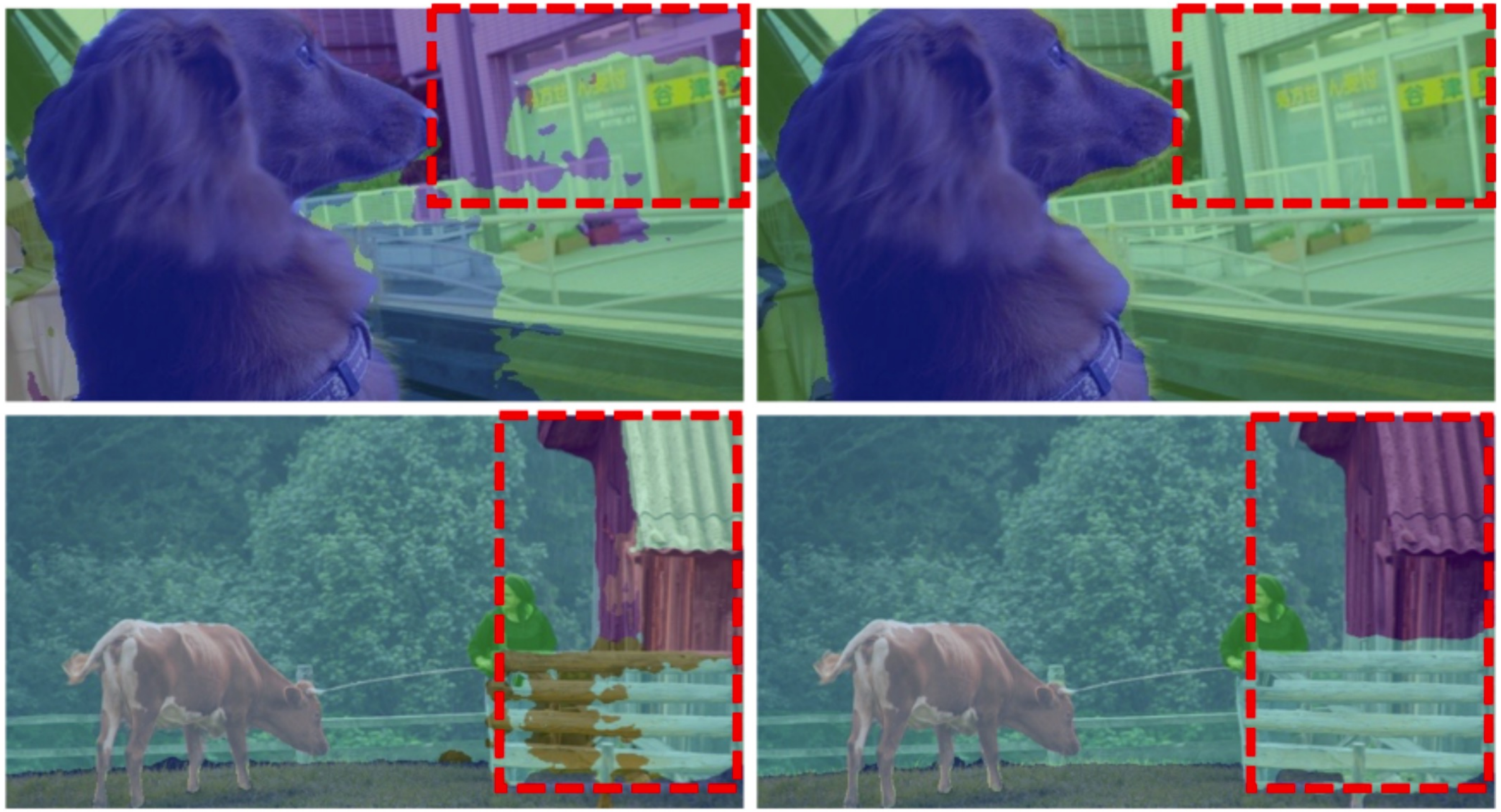
Segformer Ours

Visualization



Segformer　　　　Ours

Visualization



Segformer Ours

Prototype Meaning

Figure 3. **Visualization of pixel-prototype similarity** for *person* (top) and *car* (bottom) classes. Please refer to §3 for details.

Embedding Space

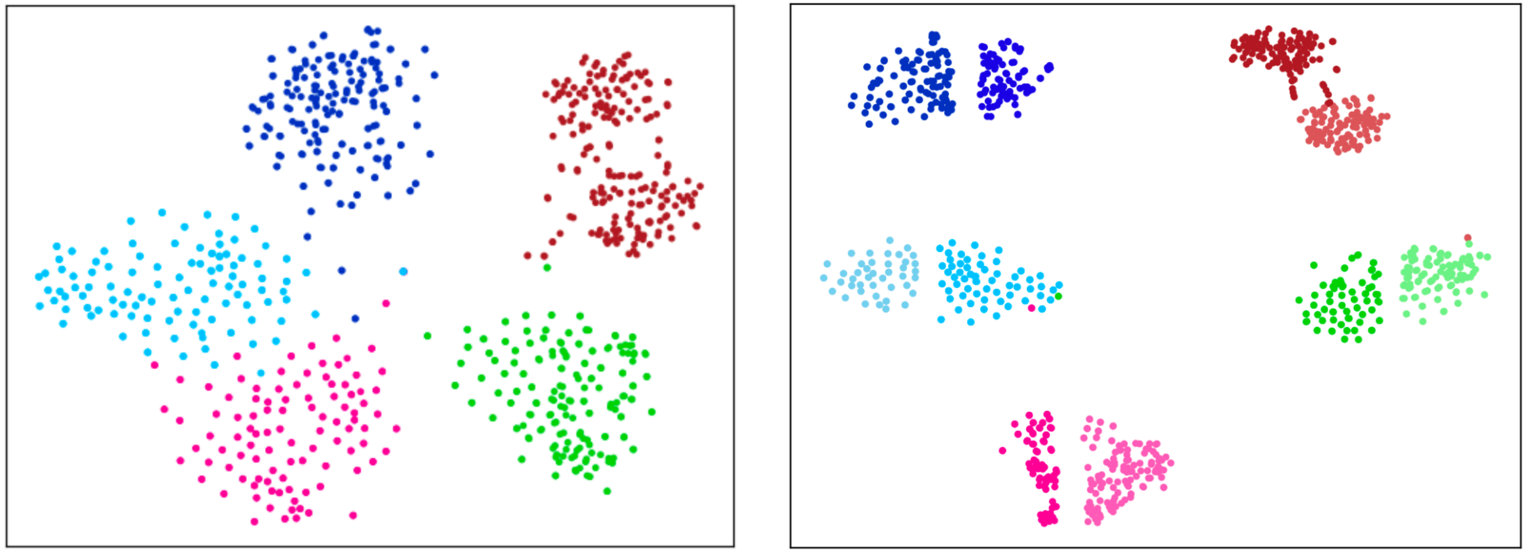


Figure 5. **Embedding spaces** learned by (left) parametric model [120], and (right) our nonparametric model. For better visualization, we show five classes of Cityscapes [23] with two prototypes per class.