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

https://github.com/tfzhou/ProtoSeg

CVPR2022 oral

Exploring Cross-Image Pixel Contrast for Semantic Segmentation

ICCV2021 oral

Motivation:

- Pixel-Wise Cross-Entropy supervision:

1) ignore relationship between pixels
2) cannot supervise the learned representation directly

- Previse sturecture-aware loss ignore correlations between pixels across image

Method:

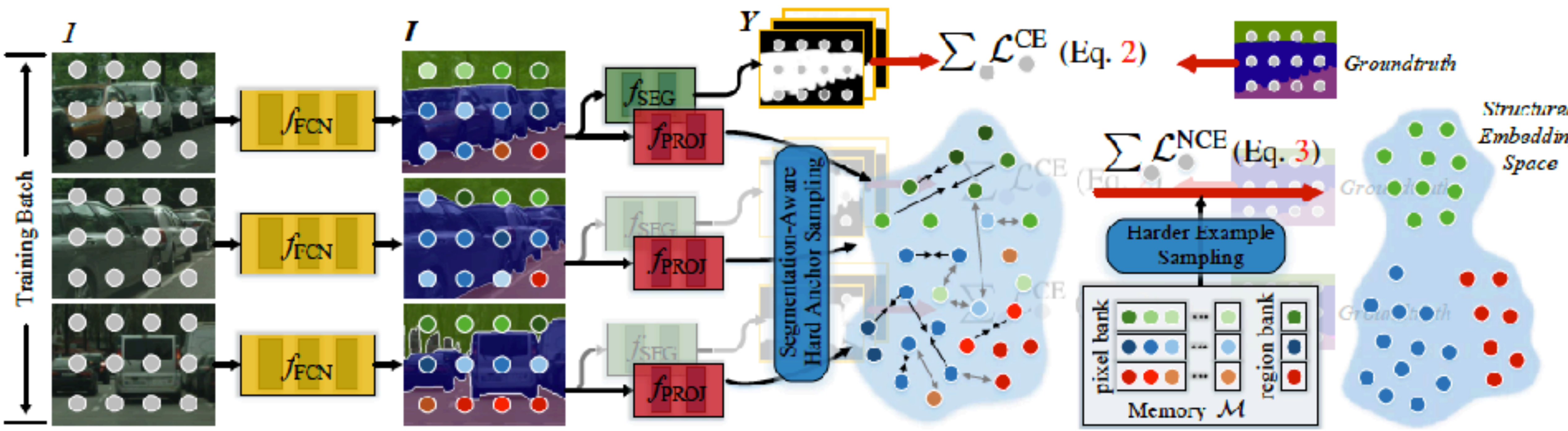


Figure 3: **Detailed illustration** of our pixel-wise contrastive learning based semantic segmentation network architecture.

- Design memory bank for dense pixel embeddings
- hard sample selection

Exploring Cross-Image Pixel Contrast for Semantic Segmentation

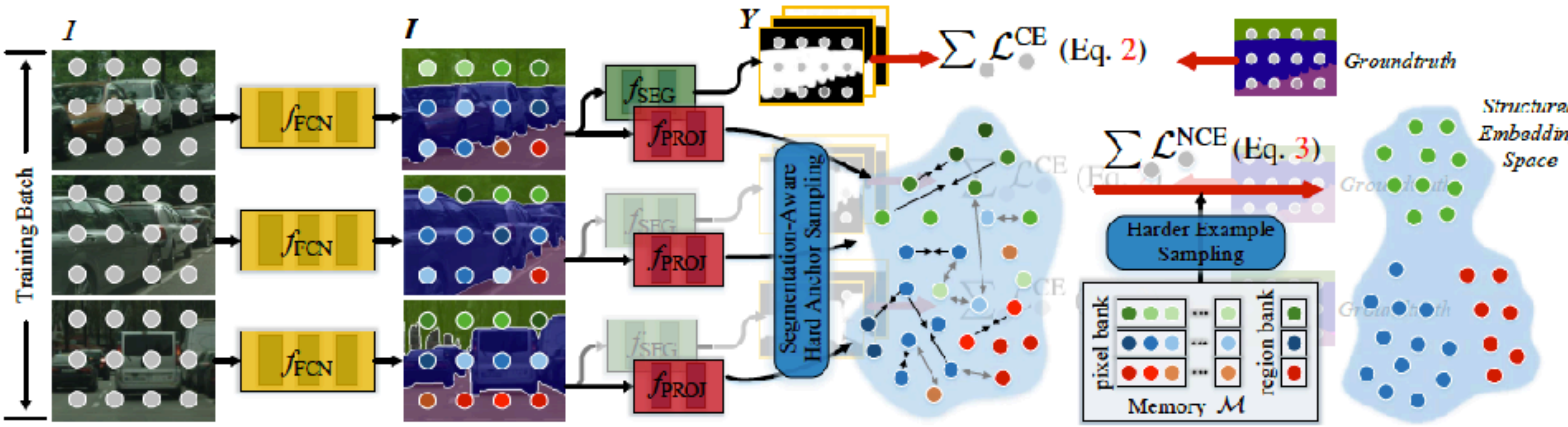


Figure 3: **Detailed illustration** of our pixel-wise contrastive learning based semantic segmentation network architecture.

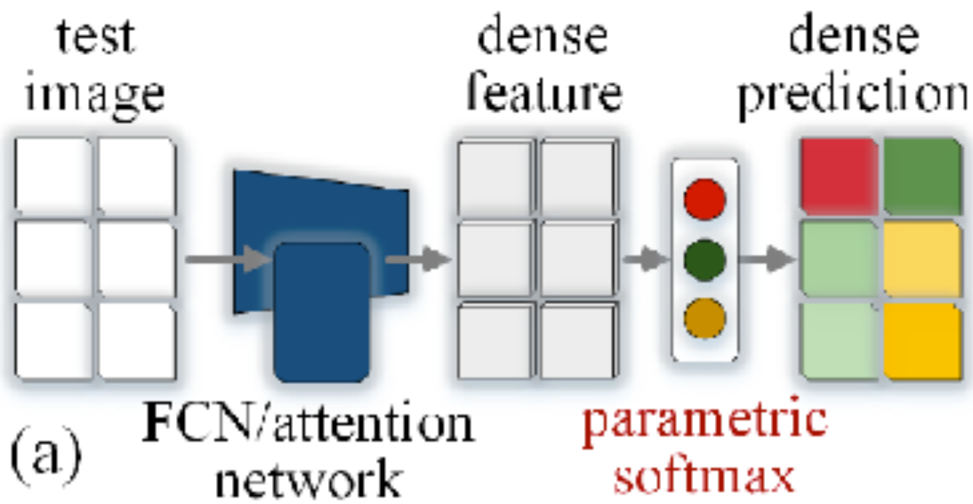
Limitation:

- pull all features $\in C_i$ together
- two stream

Rethinking Semantic Segmentation: A Prototype View

Previous semantic segmentation methods

- **Parametric Softmax Projections**



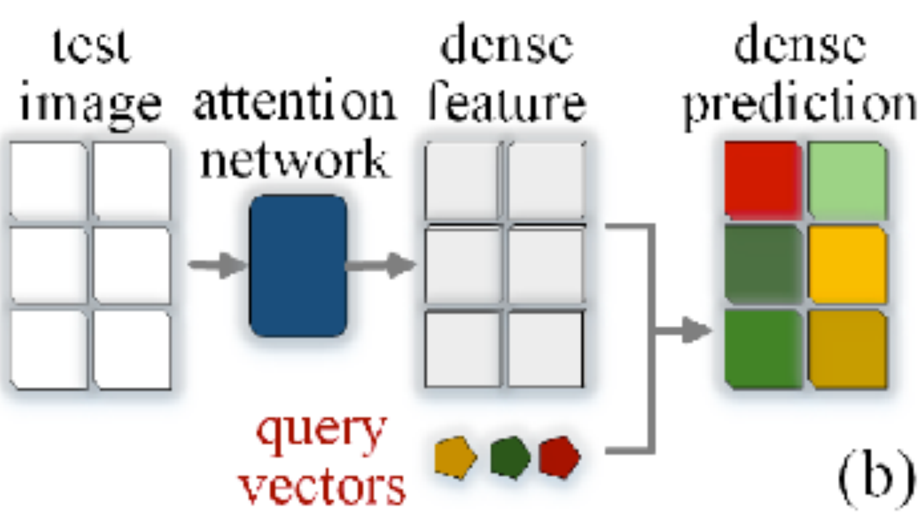
dense visual feature extractor →

+

pixel wise linear layer $\boldsymbol{W} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_C] \in \mathbb{R}^{C \times D}$

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

- **Parametric Pixel-Query**



dense visual feature extractor →

+

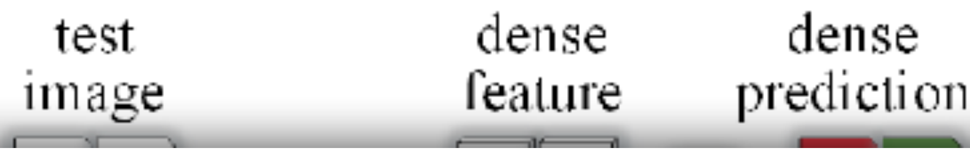
pixel query layer $\boldsymbol{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_C] \in \mathbb{R}^{C \times D}$

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

Rethinking Semantic Segmentation: A Prototype View

Previous semantic segmentation methods

- **Parametric Softmax Projections**



dense visual feature extractor →

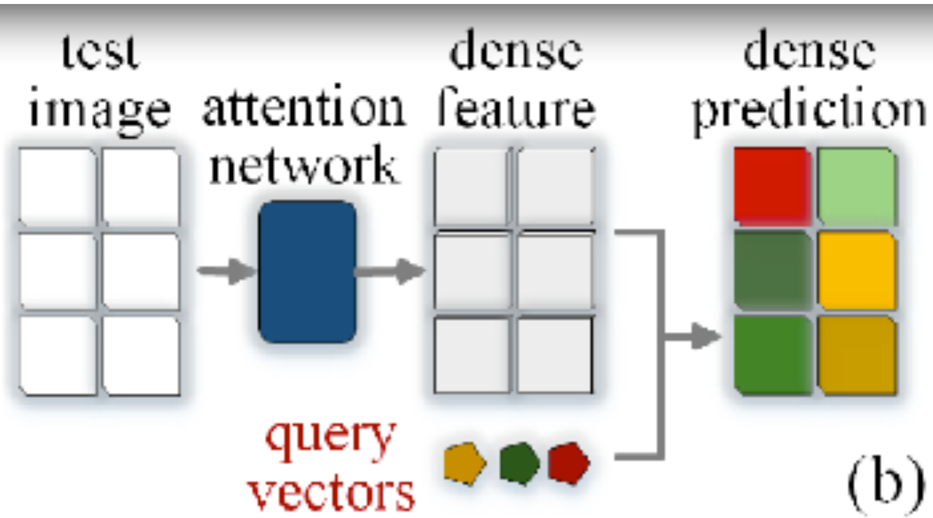
+

pixel wise linear layer $\boldsymbol{W}=[\boldsymbol{w}_1,\cdots,\boldsymbol{w}_C]\in\mathbb{R}^{C\times D}$

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

Limitation

1. single prototype per class
2. parameter: D*C
3. ignore relationship between pixels and prototypes



...al feature extractor →

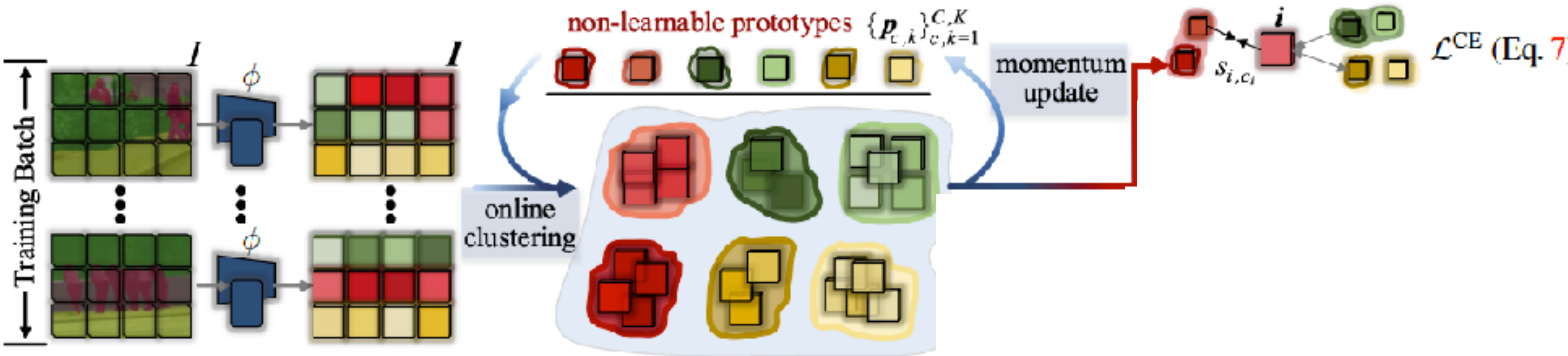
+

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

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

Rethinking Semantic Segmentation: A Prototype View

Proposed Method



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

$$\mathcal{L}_{\mathrm{CE}} = -\log p(c_i|\boldsymbol{i})$$

$$= -\log \frac{\exp(-s_{i}}{\exp(-s_{i,c_i}) + \sum_{c' \neq}}$$

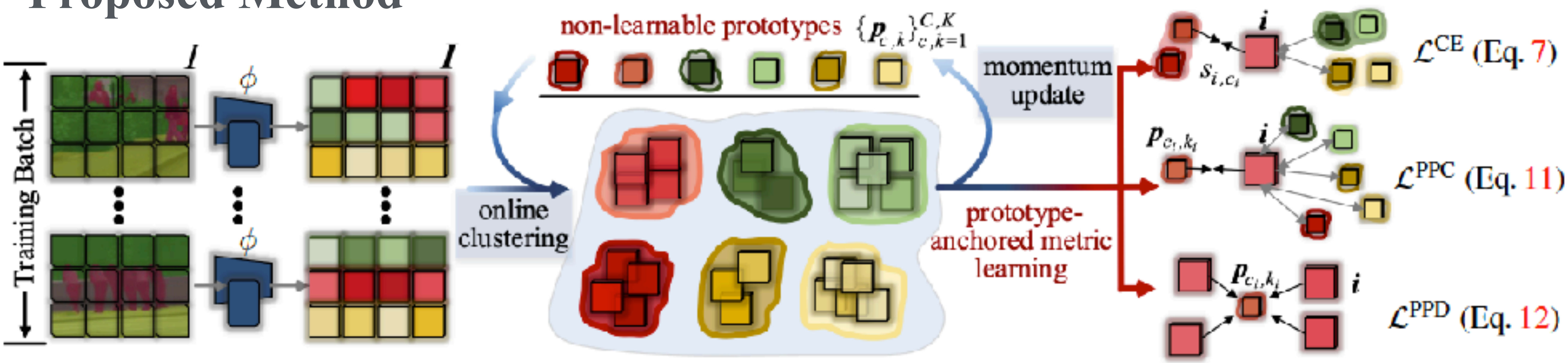
Limitation

1. ignore within-class pixel-prototype relations
2. only consider relative relation between intra-class and inter-class

-> cannot directly supervise the feature distribution

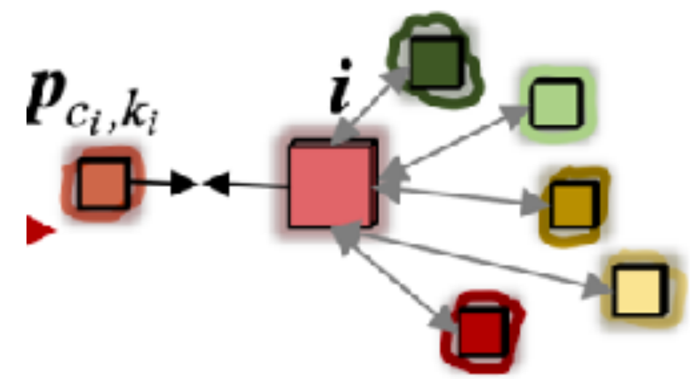
Rethinking Semantic Segmentation: A Prototype View

Proposed Method



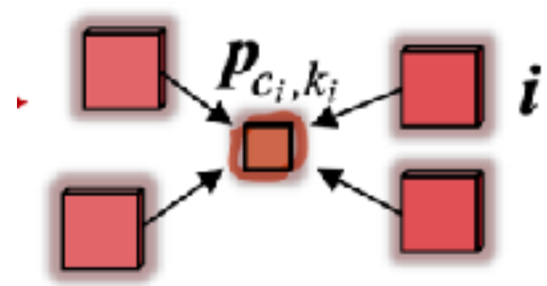
Pixel-Prototype Contrastive Learning.

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



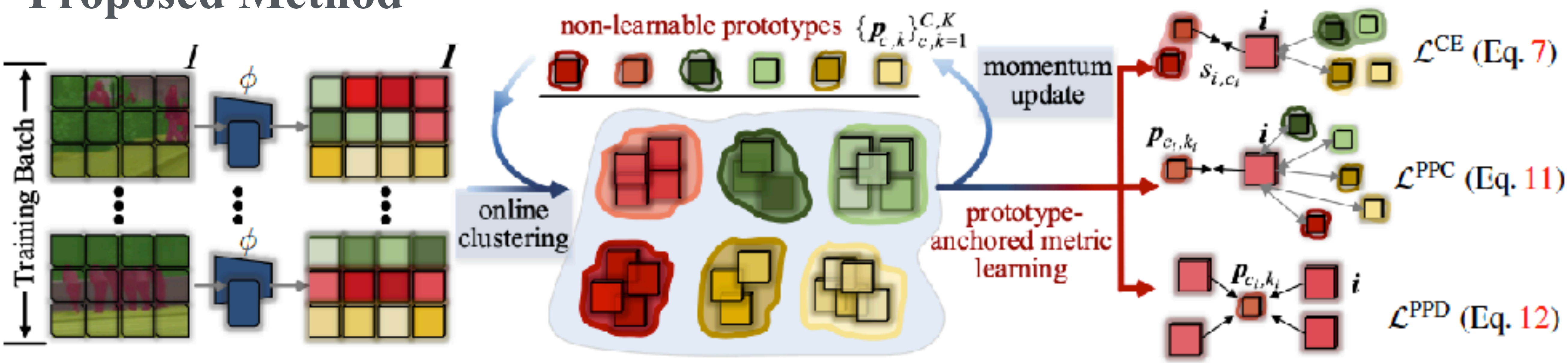
Pixel-Prototype Distance Optimization.

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



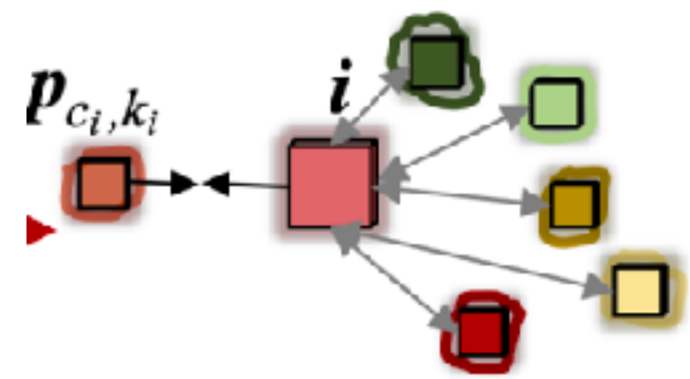
Rethinking Semantic Segmentation: A Prototype View

Proposed Method



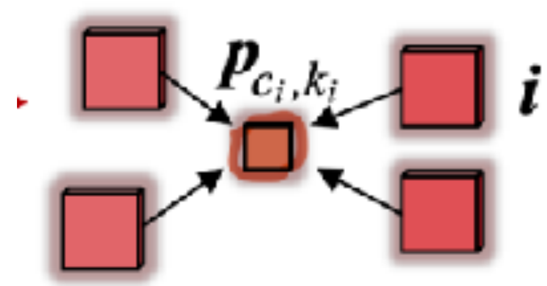
Pixel-Prototype Contrastive Learning.

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



Pixel-Prototype Distance Optimization.

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



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

Rethinking Semantic Segmentation: A Prototype View

Result

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 (%)
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`.

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`.

Rethinking Semantic Segmentation: A Prototype View

prototype explanation

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

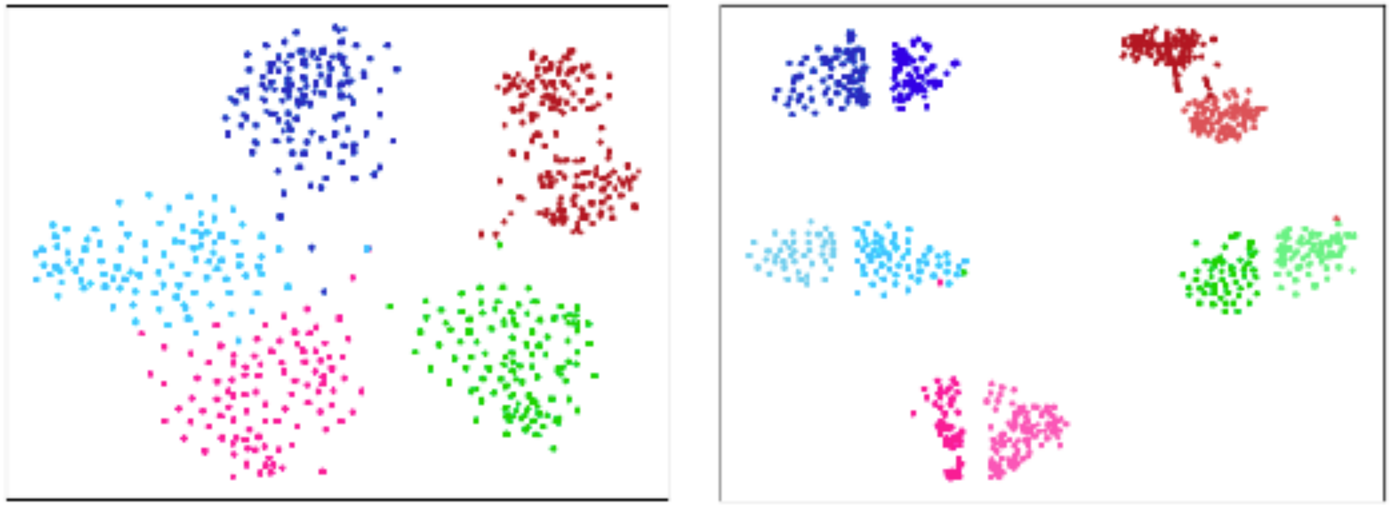


Figure 5. **Embedding spaces** learned by (left) parametric model [120], and (right) our nonparametric model. For better visualization, we show five classes of Cityscapes [23] with two prototypes per class.

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

Deep Hierarchical Semantic Segmentation