Supplementary Materials for "Training Vision Transformers for Semi-Supervised Semantic Segmentation"

Xinting Hu xhu@mpi-inf.mpg.de
Li Jiang lijiangcse@gmail.com
Bernt Schiele schiele@mpi-inf.mpg.de
Max Planck Institute for Informatics, Saarland Informatics Campus, Germany

In this supplementary material, we will provide the detailed implementations of our model in Section S1, more supplementary results in Section S2, and more visualization in Section S3.

S1. More Implementation Details

In Section 4 of the main paper, we deployed our methods on our baseline (S^4Former) together with two previous SOTA works: UniMatch [8] and AugSeg [11]. This section first shows the detailed implementation of our methods and supplements different settings of ablation experiments in Section 4.3 of the main paper.

S1.1. Algorithm.

In a nutshell, the overall framework can be summarized in the Algorithm 1. When implementing our methods on UniMatch [8], we replace the feature dropout head with our Patch-Adaptive Self-Attention (PASA) for feature perturbation. To implement AugSeg [11], we replace the original CutMix in strong augmentations for the student image x^S with the proposed Adaptive CutMix [11], and combine it with our PatchShuffle as the novel strong augmentation.

S1.2. Training Details.

Network. We set the total batch size equal to 16, consisting of 8 labeled images and 8 unlabeled images. We loaded the weights pre-trained on ImageNet-1K for the encoder and randomly initialized the decoder. The EMA update momentum is set to 0.999. For models based on SETR (DeiT-Base) [12], an SGD optimizer with a momentum of 0.9 and a polynomial learning-rate decay with an initial value of 0.001 ($\times$10 for the decoder) are adopted to train the model. For models based on SegFormer (MiT-B4) [7], an Adam optimizer and a polynomial learning-rate decay with an initial value of 6e-5 ($\times$10 for the decoder) are adopted to train the model.

Data. For Pascal VOC 2012, the images are randomly cropped into 512$\times$512 for training, and the total training epoch is 80K. For COCO, the images are randomly cropped into 512$\times$512 for training, and the total training epoch is 160K. For Cityscapes, the images are randomly cropped into 768$\times$768 for training, and the total training epoch is 40K. We also use the sliding evaluation for Cityscapes as previous works [6, 8, 11] to examine the performance on validation images with a resolution of 1024$\times$2048.

S1.3. Different Ablation Settings

As discussed in Section 4.3 of the main paper, we used different attention-mask adjustments and different regularization losses for ablation studies. We first show the details of different attention-mask adjustments, where *Rand* strategy uses random scaled numbers to adjust the attention mask, *Reverse1* emphasizes the self-attention of confident regions within high-confidence areas, and *Reverse2* encourages less confident regions to focus more on high-confidence counterparts. They are shown as follows:

Rand:

$$\mathcal{M}_{ij} = \alpha \cdot \texttt{Random}[0,1], \tag{1}$$

where Random[0,1] means a random number between 0 and 1.

Reverse1:

$$\mathcal{M}_{ij} = \begin{cases} \alpha \cdot \bar{C}_j^T & \text{if } \bar{C}_i^T > \texttt{Median}(\bar{C}^T), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Reverse2:

$$\mathcal{M}_{ij} = \begin{cases} 0 & \text{if } \bar{C}_i^T > \texttt{Median}(\bar{C}^T), \\ \alpha \cdot \bar{C}_j^T & \text{otherwise.} \end{cases} \tag{3}$$

We further tried another implementation where both the confidence of i-th patch and j-th patch can directly adjust M_{ij} as:

$$\mathcal{M}_{ij} = \alpha \cdot (1 - \bar{C}_j^T)(1 - \bar{C}_i^T). \tag{4}$$

After tuning α, we can achieve improved performance with scores of 78.8% and 78.4% for α values of 50 and 100, respectively, against our original 77.9% with 92 labeled images of Pascal VOC.

Algorithm 1 S^{4}Former Training

Input : D^L, D^U ▷ labeled and unlabeled data
Input : model θ (θ_E for encoder, θ_D for decoder) , strong augmentation $\mathcal{A}$
Output : optimized model θ
Initialize $\theta^T = \theta^S = \theta$ randomly at iteration $t = 0$
for $t <$ MaxIter **do**
  $\{X^L, Y^L\} \leftarrow D^L$, $\{X^U\} \leftarrow D^U$ ▷ sample a mini-batch
  # For supervised labeled data:
  $P(Y|X^L;\theta^S) \leftarrow \theta^S(X^L)$ ▷ model prediction (softmax probability)
  $\mathcal{L}^l(X^L, Y^L) \leftarrow \text{CE}(Y^L, P(Y|X^L;\theta^S))$ ▷ cross-entropy loss in Eq. (2)
  # For unlabeled data:
  $P(Y|X^U;\theta^T) \leftarrow \theta^T(X^U)$ ▷ teacher model prediction (softmax probability) without gradient
  $C^T \leftarrow \text{MAX}(P(Y|X^U;\theta^T))$ ▷ generate confidence map
  $Y^T \leftarrow \text{ARGMAX}(P(Y|X^U;\theta^T))$ ▷ generate pseudo labels
  $F^{\mathcal{A}}(X^U,\theta^S) \leftarrow \theta^S_E(\mathcal{A}(X^U))$ ▷ student feature for augmented view
  $F^{\text{PASA}}(X^U,\theta^S) \leftarrow \theta^S_E(X^U, C^T)$ ▷ student feature with PASA in Sec 3.3
  if $\mathcal{A}()$ is PATCHSHUFFLE() **then** $F^{\mathcal{A}}(X^U,\theta^S) \leftarrow \mathcal{A}^{-1}(F(X^U,\theta^S))$
  end if ▷ feature restoration in Sec 3.2
  $P^{\mathcal{A}}(Y|X^U;\theta^S) \leftarrow \theta^S_D(F^{\mathcal{A}}(X^U,\theta^S))$
  $P^{\text{PASA}}(Y|X^U;\theta^S) \leftarrow \theta^S_D(F^{\text{PASA}}(X^U,\theta^S))$
  $\mathcal{L}^u_{\text{CE}}(X^U) \leftarrow \text{CE}(Y^T, P^{\mathcal{A}}(Y|X_U;\theta^S)) + \text{CE}(Y^T, P^{\text{PASA}}(Y|X_U;\theta^S))$
  $\mathcal{L}^u_{\text{NCR}}(X^U) \leftarrow \text{NCR}(P(Y|X^U;\theta^T), P(Y|X_U;\theta^S)) + \text{NCR}(P(Y|X^U;\theta^T), P^{\text{PASA}}(Y|X_U;\theta^S))$ ▷ NCR loss
  $\mathcal{L}^u(X^U) = 1/2(\mathcal{L}^u_{\text{CE}}(X^U) + \mathcal{L}^u_{\text{NCR}}(X^U)$
  # Model update:
  $\theta^S \leftarrow \theta^S - \nabla_{\theta^S}\mathcal{L}^l(X^L, Y^L) + \mathcal{L}^u(X^U)$
  $\theta^T \leftarrow \mu\theta^T + (1-\mu)\theta^S$ ▷ teacher model update in Eq. (1)
end for
Return θ^S

For different regularization losses over student predictions p^S with pseudo-label $\tilde{y}$, *Soft* applies the cross entropy loss with soft labels and *All-CR* denotes the approach of applying L2 distance loss across all classes. They are shown as follows:

Soft:

$$\mathcal{L}_{Soft} = -\log\left(\frac{\exp(p^S_{\tilde{y}}/T)}{\sum_{k=1}^{K}\exp(p^S_k/T)}\right), \tag{5}$$

where T is temperature, we use $T = 2$ to produce a softer probability distribution over classes. K is the number of classes.

All-CR:

$$\mathcal{L}_{All-CR} = \sum_{k\in K}\left(\hat{p}^T_k - \hat{p}^S_k\right)^2, \quad \text{where}$$
$$\hat{p}^{T \text{ or } S}_k = \frac{\exp(p^{T \text{ or } S}_k)}{\sum_{k\in K}\exp(p^{T \text{ or } S}_k)} \tag{6}$$

The All-CR loss is calculated as the L2 distance between normalized probabilities over all the K classes.

S2. Additional Results

Additional results on SOTA methods. We show the detailed numbers of Figure 6 of the main paper in Table S1.

Additional results on the effect of proposed components. We include additional results on the effect of proposed components individually and in different combinations (Table S2). When 1/8 of images are labeled, though "PatchShuffle + NCR" performs worse than using

Dataset Split		Pascal VOC *classic*					Pascal VOC *blend*		
Backbone	Methods	1/16 (92)	1/8 (183)	1/4 (366)	1/2 (732)	Full (1464)	1/16 (662)	1/8 (1323)	1/4 (2646)
DeepLabV3+ (ResNet101)	Sup-Only	50.7	59.1	65.0	70.6	74.1	67.5	71.1	74.2
	U2PL [6]	68.0	69.2	73.7	76.2	79.5	74.4	77.6	78.7
	PS-MT [3]	65.8	69.6	76.6	78.4	–	75.5	78.2	78.7
	iMAS [10]	70.0	75.3	<u>79.1</u>	80.2	<u>82.0</u>	77.2	78.4	<u>79.3</u>
	AugSeg [11]	71.1	75.5	78.8	<u>80.3</u>	81.4	77.0	77.3	78.8
	UniMatch [8]	<u>75.2</u>	<u>77.2</u>	78.8	79.9	81.2	<u>78.1</u>	<u>78.4</u>	79.2
SETR (DeiT-Base)	Sup-Only	67.7	72.8	77.4	80.7	82.5	76.6	77.8	79.9
	S^4Former-Base	75.8	77.4	80.0	81.7	83.9	79.0	80.1	80.7
	+ Ours	80.1 (+4.3)	81.3 (+3.9)	82.4 (+2.4)	83.2 (+1.5)	**85.0** (+1.1)	79.9 (+0.9)	80.5 (+0.4)	81.3 (+0.6)
	UniMatch	76.2	77.6	80.6	82.5	83.6	78.3	80.5	80.5
	+ Ours	79.3 (+3.1)	80.4 (+2.8)	81.9 (+1.3)	**83.4** (+0.9)	84.7 (+1.1)	78.9 (+0.6)	80.7 (+0.2)	80.9 (+0.4)
	AugSeg	77.7	80.1	82.1	82.5	83.9	79.3	80.4	80.9
	+ Ours	**80.3** (+2.6)	**81.3** (+1.2)	**82.8** (+0.7)	83.2 (+0.7)	84.6 (+0.7)	**80.0** (+0.7)	**80.8** (+0.4)	**81.3** (+0.4)

Table S1. Comparison of mIoU (%) between *state-of-the-art* and ours methods on the Pascal VOC 2012 dataset. Results are presented for two dataset splits following previous works [8, 11]: *classic*, with labeled samples drawn from the original dataset, and *blend*, with labeled samples drawn from the augmented dataset inclusive of SBD. The fractions (*e.g.*, 1/16) and numbers (*e.g.*, 92) denote the proportion and number of labeled images. Best performances for DeepLabV3+ and our architecture are highlighted with <u>underline</u> and **bold**, respectively.

PatchShuffle	PASA	NCR	1/16	1/8
			75.8	77.4
✓			78.4	80.7
	✓		77.9	79.4
		✓	77.5	78.7
✓	✓		79.6	81.1
	✓	✓	78.3	79.7
✓		✓	79.6	79.3
✓	✓	✓	80.1	81.3

Table S2. Performance comparison with different components. Results for the Pascal VOC 2012 *classic* settings.

Methods	mIoU
MT [5]	73.1
CutOut [1]	73.5 (+0.4)
CutMix [9]	75.8 (+2.7)
Mix/UnMix [2]	76.0 (+2.9)
ClassMix [4]	77.0 (+3.9)
PatchShuffle (Ours)	76.6 (+3.5)
+ PatchShuffle w Mix/UnMix	78.0 (+4.9)
+ PatchShuffle w ClassMix	77.8 (+4.7)
+ PatchShuffle w CutMix	78.4 (+5.3)

Table S3. Performance comparison with different strong augmentation strategies for student images. Results for the Pascal VOC 2012 *classic* 1/16 settings.

PatchShuffle alone, it performs better when further combined with PASA. Overall, combining all our proposed regularization strategies from all the image, feature and output ends delivers robust results. We further compare our PatchShuffle with other existing image augmentations in Table S3. Those methods either mask out or mix some regions of the current image with another one. One can observe that our method achieves competitive performance in terms of strong image augmentations for student images. Notably, although ClassMix [4] yields a slightly higher IoU (*i.e.*, 0.4%), our method can be synergistically combined with existing techniques for enhanced results.

Additional results on ViT-L. We further implement our methods on ViT-L, which is ImageNet-21K pretrained. The results are shown in Table S4, where one can observe consistent improvements.

Backbone	Methods	1/16	1/8	1/4
SETR (ViT-L)	Sup-Only	74.3	76.8	79.7
	S^4Former-Base	76.2	79.9	80.3
	+ Ours	77.6 (+1.4)	80.5 (+0.6)	81.1 (+0.8)

Table S4. Performance on ViT-L backbone. Results for the Pascal VOC 2012 *classic* settings.

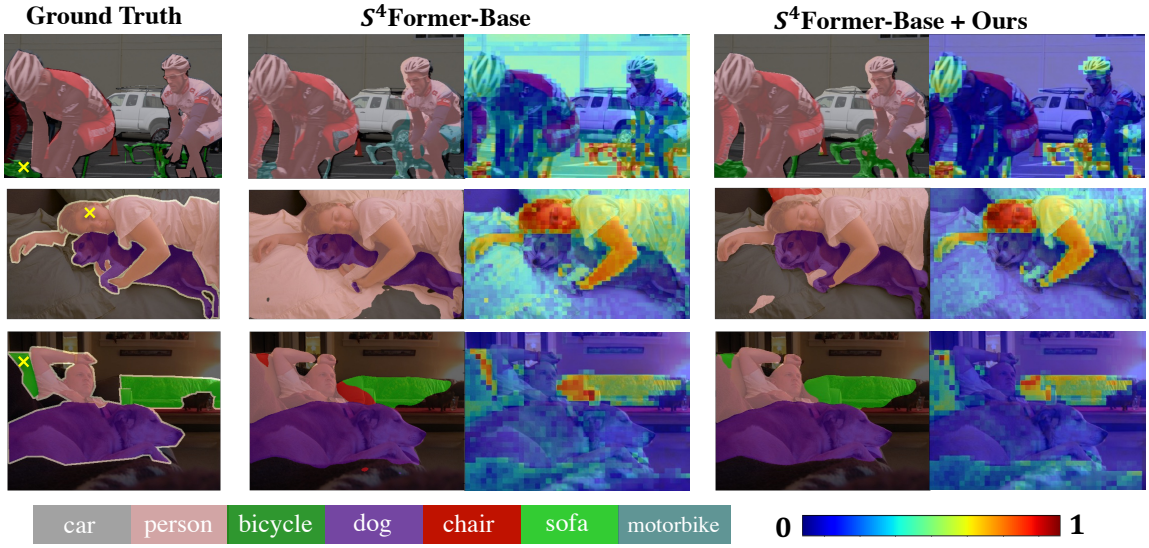

Figure S1. Comparative visual results on Pascal VOC 2012 classic setting with a limited set of 92 labeled images. We show segmentation predictions as well as the normalized attention weights across all image patches for a specific patch of interest (marked by a yellow cross).

Backbone	Methods	1/16	1/8	1/4
DeepLabV3+ (ResNet101)	Base	64.5	73.0	75.6
	+ PatchShuffle	68.5 (+4.0)	72.6 (-0.4)	75.4 (-0.2)
	+ NCR	60.7 (-3.8)	69.4 (-3.6)	74.6 (-1.0)

Table S5. Performance evaluation of integrating PatchShuffle and Negative Class Ranking (NCR) loss with the DeepLabV3+ model using a ResNet101 backbone. Results for the Pascal VOC 2012 *classic* settings. Here, "Base" is the "FixMatch" baseline in UniMatch [8]. These results highlight the differential impact of our proposed components when applied to a ConvNet-based architecture, with PatchShuffle showing mixed outcomes and NCR generally leading to a decrease in performance.

Additional results on ResNet. Our exploration extended to integrating our proposed components with the ResNet backbone. As shown in Table S5, we observe these modifications did not consistently enhance performance and even detrimentally impacted it. We attribute this to the inherent limitations of ConvNet-based frameworks. For the PatchShuffle technique, in contrast to Transformers, which can maintain semantic consistency with shuffled inputs due to their adaptive global receptive fields, the fixed receptive fields of ConvNets hinder their ability to preserve semantic content when patches are shuffled. This inconsistent semantic context between student and teacher predictions results in ineffective regularization. As for the Negative Class Ranking (NCR) loss, the application on ResNet leads to challenges due to the noisier pseudo-labels (*i.e.*, positive labels). Consequently, imposing consistency among negative classes does not bring substantial benefits, highlighting the architecture-specific effectiveness of our proposed methods.

S3. More Visualization

S3.1. More Attention Weights Visualizations

In Figure S1, we further visualize the normalized attention weights across all image patches for a specific patch of interest (marked by a yellow cross). This visualization demonstrates how our proposed components steer the model to focus on patches that are semantically or visually related to the selected patch, thereby facilitating more accurate predictions. For instance, in the first row, the 'Base' model initially misclassifies a "bicycle" as a "motorbike". This error arises because it incorrectly associates the bicycle's patch more closely with the car and other surrounding scene elements. However, our model, by directing attention to relevant areas such as another bicycle and the cycling helmet, successfully avoids such misjudgments.

S3.2. Visualization of Strong Image Augmentations

We illustrate the student images with CutMix, Patchshuffle, and PatchShuffle+CutMix in Figure S3. Color-jittering is always used.

Visualization for NCR. Our NCR loss enhances conventional consistency loss by regularizing over negative classes, aiding the network in differentiating confusing classes or regions. For example, it improves the distinction between confusing class pairs "bicycle" *v.s.* "motorbike"

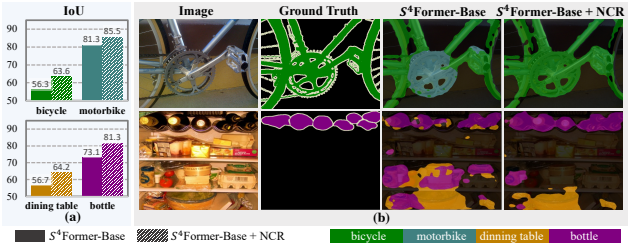

Figure S2. The test set (a) IoU and (b) example images to show the improvements over confusing classes and regions with our NCR.

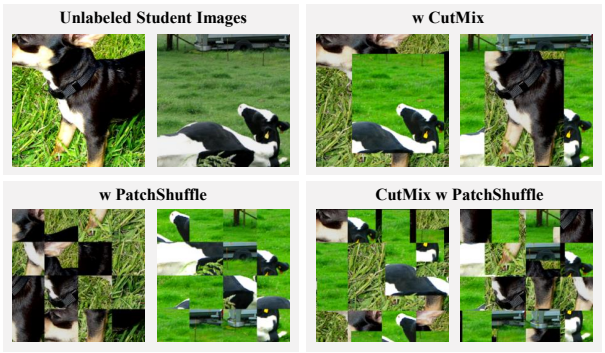

Figure S3. Visualization of CutMix [9], our proposed PatchShuffle as well as the combination of them.

and regions of "dining table" *v.s.* "bottle" (see Figure S2).

S3.3. More Semantic Segmentation Visualizations

n Figure S4, we present more examples for visual comparisons on Pascal VOC 2012, COCO, and Cityscapes. It is evident that our proposed S^{4}Former, with its innovative components, consistently delivers superior segmentation quality across these datasets.

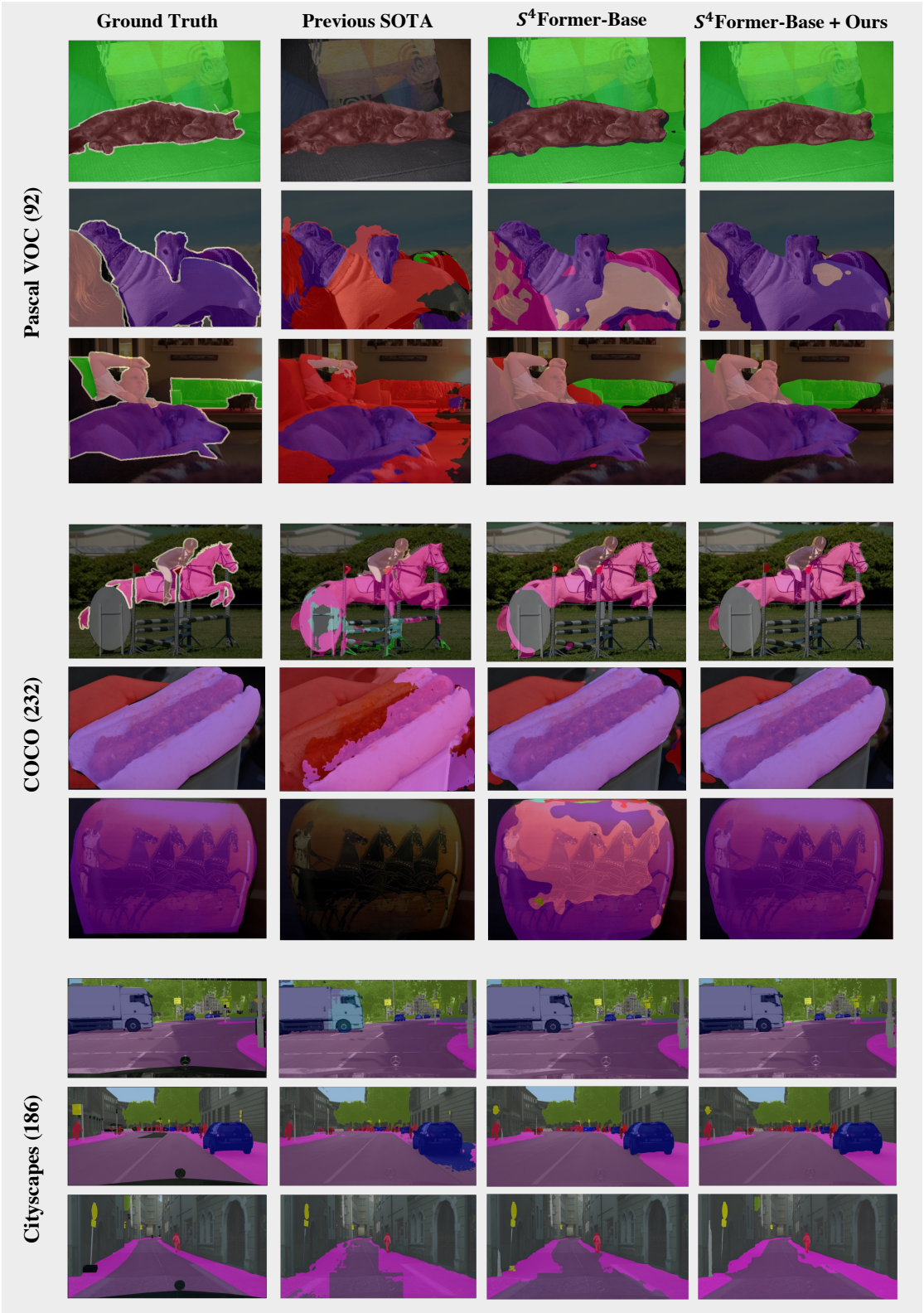

Figure S4. Qualitative comparisons on benchmark datasets. The number in brackets (*e.g.*, 92) represents how many labeled images are used. Here, we used open-sourced UniMatch models as "Previous SOTA".

References

[1] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[2] JongMok Kim, JooYoung Jang, Seunghyeon Seo, Jisoo Jeong, Jongkeun Na, and Nojun Kwak. MUM: Mix Image Tiles and UnMix Feature Tiles for Semi-Supervised Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[3] Yuyuan Liu, Yu Tian, Yuanhong Chen, Fengbei Liu, Vasileios Belagiannis, and Gustavo Carneiro. Perturbed and Strict Mean Teachers for Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[4] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. ClassMix: Segmentation-Based Data Augmentation for Semi-Supervised Learning. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3

[5] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 3

[6] 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 International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 3

[7] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 1

[8] Lihe Yang, Lei Qi, Litong Feng, Wayne Zhang, and Yinghuan Shi. Revisiting Weak-to-Strong Consistency in Semi-Supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 3, 4

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 5

[10] Zhen Zhao, Sifan Long, Jimin Pi, Jingdong Wang, and Luping Zhou. Instance-specific and Model-adaptive Supervision for Semi-supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[11] Zhen Zhao, Lihe Yang, Sifan Long, Jimin Pi, Luping Zhou, and Jingdong Wang. Augmentation Matters: A Simple-yet-Effective Approach to Semi-supervised Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 3

[12] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1