A Pseudo-labels quality analysis

The quality improvement and the quantity increase of pseudo-labels are shown in Fig. 4. Further analysis of the quality improvement of our method is demonstrated in Fig. A.1 by separating the *true positive* and *false positive*.

Within the initial phase of the learning process, the enhancement in the quality of pseudo-labels can be primarily attributed to the advancement in true positive labels. In our method, the refinement not only facilitates the inclusion of a larger number of pixels surpassing the threshold but also ensures that a significant majority of these pixels are of high quality.

As the learning process progresses, most improvements are obtain from a decrease in false positives pseudo-labels. This analysis shows that our method effectively minimizes the occurrence of incorrect pseudo-labeled, particularly when the threshold is set to a lower value. In other words, our method reduce conformation bias that comes from the decaying of the threshold as the learning process progress.

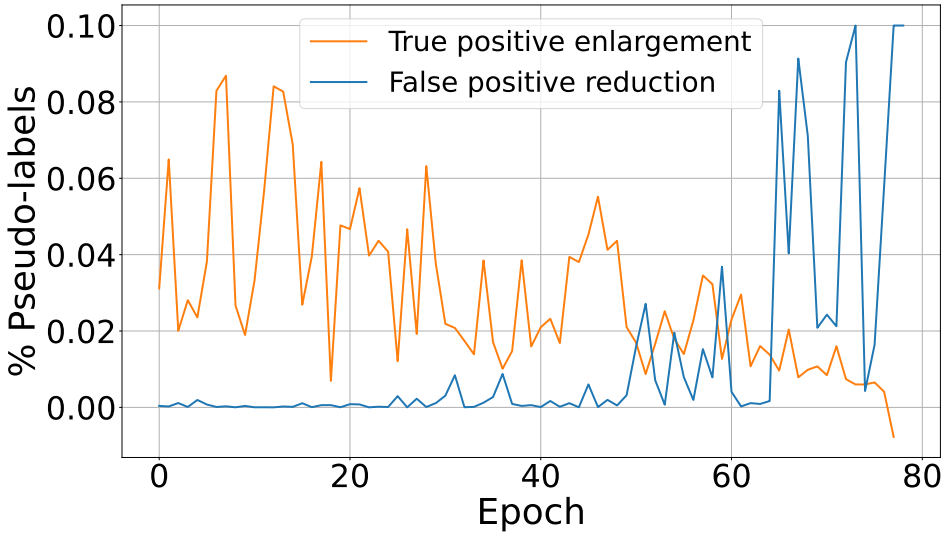


Figure A.1: **Quality of pseudo-labels**, on PASCAL VOC 2012 (Everingham et al., 2010) over training iterations. Fig. 4 separated to *True positive* and *False positive* analysis. *True positive* are the bigger part of improvement at early stage of the training process, while reduction of *false positive* is the main contribution late in the training process

B Confidence function alternatives

In this paper, we introduce a confidence function to determine pseudo-label propagation. We introduced $\kappa_{\text{margin}}(x_{i,j})$ and mentioned other alternatives have been examined.

Here we define several options for the confidence function.

The simplest option is to look at the probability of the dominant class,

$$\kappa_{\max}(x^i_{j,k}) = \max_c p_c(x^i_{j,k}), \tag{B.1}$$

which is commonly used to generate pseudo-labels.

The second alternative is negative entropy, defined as

$$\kappa_{\text{ent}}(x^i_{j,k}) = \sum_{c \in C} p_c(x^i_{j,k}) \log\left(p^c_{i,j}\right). \tag{B.2}$$

Table B.1: Ablation study on the confidence function κ, over Pascal VOC 12 with partition protocols

Function	1/4 (366)	1/2 (732)	Full (1464)
κ_{max}	74.29	76.16	79.49
κ_{ent}	75.18	77.55	79.89
κ_{margin}	75.41	77.73	80.58

Note that this is indeed a confidence function since high entropy corresponds to high uncertainty, and low entropy corresponds to high confidence.

The third option is for us to define the margin function (Scheffer et al., 2001; Shin et al., 2021) as the difference between the first and second maximal values of the probability vector and also described in the main paper:

$$\kappa_{\text{margin}}(x_{i,j}) = \max_c(p_c(x^i_{j,k})) - \max 2_c(p_c(x^i_{j,k})), \tag{B.3}$$

where $\max 2$ denotes the vector's second maximum value. All alternatives are compared in Table B.1.

Table B.1 studies the impact of different confidence functions on pseudo-label refinement. We found that using a margin to describe confidence is a suitable way when there is a contradiction in smooth regions.

C Bounding the joint probability

In this paper, we had the union event estimation with the independence assumption, defined as

$$p^1_c(x^i_{j,k}, x^i_{\ell,m}) \approx p_c(x^i_{j,k}) \cdot p_c(x^i_{\ell,m}) \tag{C.1}$$

In addition to the independence approximation, it is possible to estimate the unconditional expectation of two neighboring pixels belonging to the same class based on labeled data:

$$p^2_c(x^i_{j,k}, x^i_{\ell,m}) = \frac{1}{|\mathcal{N}_l| \cdot H \cdot W \cdot |\mathbf{N}|} \sum_{i \in \mathcal{N}_l} \sum_{j,k \in H \times W} \sum_{\ell,m \in \mathbf{N}_{j,k}} \mathbb{1}\{y^i_{j,k} = y^i_{\ell,m}\}. \tag{C.2}$$

To avoid overestimating that could lead to overconfidence, we set

$$p_c(x^i_{j,k}, x^i_{\ell,m}) = \max(p^1_c(x^i_{j,k}, x^i_{\ell,m}), p^2_c(x^i_{j,k}, x^i_{\ell,m})) \tag{C.3}$$

That upper bound of joint probability ensures that the independence assumption does not underestimate the joint probability, preventing overestimating the union event probability. Using Eq. (C.3) increase the mIOU by **0.22** on average, compared to non use of S4MC refinement, using 366 annotated images from PASCAL VOC 12 Using only Eq. (C.2) reduced the mIOU by **-14.11** compared to non use of S4MC refinement and actually harmed the model capabilities to produce quality pseudo-labels.

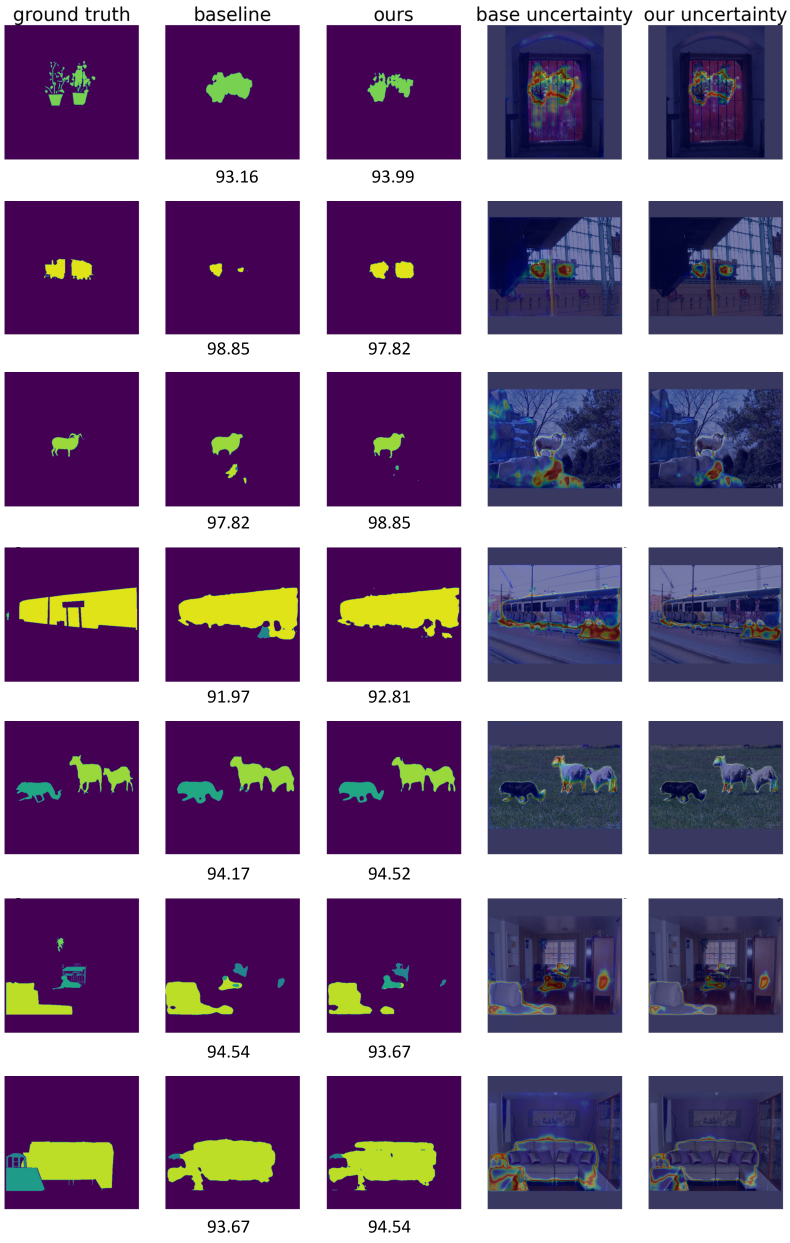
D Implementation Details

All experiments were conducted for 80 training epochs with the simple stochastic gradient descent (SGD) optimizer with a momentum of 0.9 and learning rate policy of $lr = lr_{\text{base}} \cdot \left(1 - \frac{\text{iter}}{\text{total iter}}\right)^{\text{power}}$. With the probability of 0.5, we apply CutMix (Yun et al., 2019) augmentation on the unlabeled data.

For PASCAL VOC 2012 $lr_{\text{base}} = 0.001$ and the decoder only $lr_{\text{base}} = 0.01$, the weight decay is set to 0.0001 and all images are cropped to 513×513 and $\mathcal{B}_l = \mathcal{B}_u = 3$.

For Cityscapes, all parameters use $\text{lr}_{\text{base}} = 0.01$, and the weight decay is set to 0.0005. The learning rate decay parameter is set to $\text{power} = 0.9$. Due to memory constraints, all images are cropped to 769×769 and $\mathcal{B}_\ell = \mathcal{B}_u = 2$. All experiments are conducted on a machine with 8 Nvidia RTX A5000 GPUs.

Figure E.1: **Example of refined pseudo-labels**, the structure is as in Fig. 3, the numbers under the predictions show the pixel-wise accuracy of the prediction map.



E More visual results

We present in Appendix E an extension of Fig. 3, showing more instances from the unlabeled data and the corresponding pseudo-labeled with the baseline model and S4MC.

Through our method, we can achieve more accurate predictions during the inference phase without any refinements. This results in the generation of more seamless and continuous predictions, which depict the spatial configuration of objects more accurately.

Table F.1: Comparison between our method and prior art on the PASCAL VOC 2012 `val` on different partition protocols. the caption describes the share of the training set used as labeled data and, in parentheses, the actual number of labeled images. Larger improvement can be observed for partitions of extremely low annotated data, where other methods suffer from starvation due to poor teacher generalization.

Method	**1/16 (92)**	**1/8 (183)**	**1/4 (366)**	**1/2 (732)**	**Full (1464)**
Supervised Only	45.77	54.92	65.88	71.69	72.50
CutMix-Seg (French et al., 2020)	52.16	63.47	69.46	73.73	76.54
PseudoSeg (Zou et al., 2021)	57.60	65.50	69.14	72.41	73.23
PC^2Seg (Zhong et al., 2021)	57.00	66.28	69.78	73.05	74.15
CPS (Chen et al., 2021)	64.10	67.40	71.70	75.90	-
ReCo (Liu et al., 2022a)	64.80	72.0	73.10	74.70	-
ST++ (Yang et al., 2022b)	65.2	71.0	74.6	77.3	79.1
U^2PL (Wang et al., 2022)	67.98	69.15	73.66	76.16	79.49
PS-MT (Liu et al., 2022b)	65.8	69.6	76.6	78.4	80.0
FixMatch* (Martí i Rabadán et al., 2022)	65.93	72.72	75	77.8	78.35
S4MC + CutMix-Seg (Ours)	70.96	71.69	75.41	77.73	80.58
S4MC + FixMatch (Ours)	**74.32**	**75.62**	**77.84**	**79.72**	**81.51**

Table F.2: Comparison between our method and prior art on the 'coarse' PASCAL VOC 2012 `val` dataset under different partition protocols, using additional unlabeled data from (Hariharan et al., 2011). For each partition ratio we included the number of labeled images in parentheses. As in Table 1, larger improvements are observed for partitions with less annotated data.

Method	**1/16 (662)**	**1/8 (1323)**	**1/4 (2646)**	**1/2 (5291)**
Supervised Only	67.87	71.55	75.80	77.13
CutMix-Seg (French et al., 2020)	71.66	75.51	77.33	78.21
CCT (Ouali et al., 2020)	71.86	73.68	76.51	77.40
GCT (Ke et al., 2020)	70.90	73.29	76.66	77.98
CPS (Chen et al., 2021)	74.48	76.44	77.68	78.64
AEL (Hu et al., 2021)	77.20	77.57	78.06	80.29
PS-MT (Liu et al., 2022b)	75.5	78.2	78.7	-
U^2PL (Wang et al., 2022)	77.21	79.01	79.3	80.50
FixMatch* (Martí i Rabadán et al., 2022)	76.5	77.19	78.07	78.13
S4MC + CutMix-Seg (Ours)	78.49	79.67	79.85	81.11
S4MC + FixMatch (Ours)	**80.77**	**81.9**	**82.3**	**83.3**

F Additional experiments

More recent research gain popularity using FixMatch for Semantic segmentation (Martí i Rabadán et al., 2022) We additionally compared our method with them, denote by $*$ a re-implementation that achieve better results then reported in the paper. All setups for different datasets and partition protocols are reported in Tables F.1 to F.3, similar to all experiments in the main paper.

Table F.3: Comparison between our method and prior art on the Cityscapes `val` dataset under different partition protocols. Labeled and unlabeled images are selected from the Cityscapes `training` dataset. For each partition protocol, the caption gives the share of the training set used as labeled data, in parentheses, the number of labeled images.

Method	**1/16 (186)**	**1/8 (372)**	**1/4 (744)**	**1/2 (1488)**
Supervised Only	62.96	69.81	74.08	77.46
CutMix-Seg (French et al., 2020)	69.03	72.06	74.20	78.15
CCT (Ouali et al., 2020)	69.32	74.12	75.99	78.10
GCT (Ke et al., 2020)	66.75	72.66	76.11	78.34
CPS (Chen et al., 2021)	69.78	74.31	74.58	76.81
AEL (Hu et al., 2021)	74.45	75.55	77.48	79.01
U^2PL (Wang et al., 2022)	70.30	74.37	76.47	79.05
PS-MT (Liu et al., 2022b)	-	76.89	77.6	<u>79.09</u>
FixMatch* (Martí i Rabadán et al., 2022)	72.6	76.15	76.93	78.22
S4MC + CutMix-Seg (Ours)	<u>75.03</u>	<u>77.02</u>	<u>78.78</u>	78.86
S4MC + FixMatch (Ours)	**76.3**	**78.25**	**78.95**	**79.13**