1. Overview

In this appendix, we provide the detailed setting about T in Sec. 2. For more analyses about our method, the difference between FPL and negative learning is in Sec. 3.1, gradient vanishing in K value selection strategy is in Sec. 3.2, details of positive gradient score are in Sec. 3.3, and gradient similarity between G^f and G^{ide} is in Sec. 3.4. For more empirical results, the ablation study about the K value selection strategy is provided in Sec. 4.1, and class-wise analysis is in Sec. 4.2. Besides, we discuss the limitation of our FPL in Sec. 5 and illustrate more examples in Sec. 6.

2. Experimental Details

We provide the detailed setting about the cumulative probability upper bound T in our experiments in Table 1, Table 2, Table 3.

Table 1. **The setting of T on Cityscapes.**

Method	ResNet 50				ResNet 101			
	1/32	1/16	1/8	1/4	1/32	1/16	1/8	1/4
FPL+CPS w/o cutmix	0.95	0.9	0.9	0.9	0.95	0.95	0.95	0.9
FPL+CPS w/ cutmix	0.9	0.85	0.85	0.85	0.9	0.85	0.85	0.85
FPL+AEL	0.95	0.95	0.9	0.9	0.9	0.9	0.85	0.85

Table 2. **The setting of T on VOC2012.**

Method	ResNet 50			ResNet 101		
	1/16	1/8	1/4	1/16	1/8	1/4
FPL+CPS w/o cutmix	0.9	0.9	0.9	0.95	0.9	0.9
FPL+CPS w/ cutmix	0.95	0.9	0.9	0.9	0.9	0.9
FPL+AEL	0.95	0.95	0.95	0.95	0.9	0.9

Table 3. **The setting of T on VOC2012 LowData.**

Method	1/32	1/16	1/8	1/4
FPL+CPS w/ cutmix	0.95	0.85	0.85	0.85

3. More Analysis

3.1. Difference between FPL and negative learning

For uncertain unlabeled pixels, negative learning-based methods find their models always predict certainly that these pixels do not belong to some categories. Hence, they treat the uncertain pixels as negative samples to those unlikely categories. A commonly used paradigm sets a threshold (e.g., 0.2), and considers the classes for which the predicted probabilities are less than the threshold as negative categories [3]. For clarity, we take the negative learning loss based on cross-entropy loss as the comparison object, since our method is also an extension of cross-entropy loss. To unify the form, we denote the categories that do not belong to the negative categories as $\mathbb{Y}_{us}$. Formulately, the negative loss $\mathcal{L}^n$ is:

$$\mathcal{L}^n(x_{us}) = \mathcal{L}^n_{us} = -\sum_{j \notin \mathbb{Y}_{us}} \log(1 - p^j_{us}). \tag{1}$$

We see that this loss function requires the probabilities for negative categories to be small. To further show the difference between $\mathcal{L}^n$ and our $\mathcal{L}^f$, we convert the $\mathcal{L}^n$ as:

$$\begin{aligned}\mathcal{L}^n_{us} &= -\sum_{j \notin \mathbb{Y}_{us}} \log(1 - p^j_{us}) = \sum_{j \notin \mathbb{Y}_{us}} \log(\frac{1}{1 - p^j_{us}}) \\ &= \sum_{j \notin \mathbb{Y}_{us}} \log(1 + \frac{p^j_{us}}{1 - p^j_{us}}) = \sum_{j \notin \mathbb{Y}_{us}} \log(1 + \frac{e^{z^j_{us}}}{\sum_{i \neq j} e^{z^i_{us}}}) \\ &\approx \sum_{j \notin \mathbb{Y}_{us}} ReLU(z^j_{us} - \max_{i \neq j}(z^i_{us})).\end{aligned} \tag{2}$$

Eq. 2 shows that the negative loss implicitly increases the prediction for the top-1 pseudo label $\max_{i \neq j}(z^i_{us})$, indicating that it still corrupts the training of the model when pseudo labels are wrong. Differently, our FPL desires to increase the predictions for all fuzzy positive categories in $\{z^i_{us}, i \in \mathbb{Y}_{us}\}$, hence we encourage their minimum $\min(z^i_{us})$ to learn the semantics of possible GT in them:

$$\mathcal{L}^f_{us} \approx ReLU(\max_{j \notin \mathbb{Y}_{us}}(z^j_{us}) - \min_{i \in \mathbb{Y}_{us}}(z^i_{us})). \tag{3}$$

Furthermore, we empirically demonstrate the superiority of FPL over the negative learning-based method. Besides, we also evaluate the performance using a soft loss $\mathcal{L}^s$ with the soft label since it has similarities to FPL in softening pseudo labels, which is computed as:

$$\mathcal{L}^s(x_{us}) = \mathcal{L}^s_{us} = \sum_i^C q^i_{us} \log \frac{q^i_{us}}{p^i_{us}}, \tag{4}$$

where p_{us} is the predicted probability, and q_{us} is the learning target. Segmentation performances are shown in Table 4, where 'Nega.' represents the results obtained by negative loss $\mathcal{L}^n$, and 'Soft.' represents the results obtained by soft loss $\mathcal{L}^s$. In addition, U2PL [4] introduces the idea of negative learning in the manner of contrastive learning, hence we also provide its performance here. From Table 4, we see our FPL model achieves the best performance, reflecting the superiority of FPL over other alternatives.

Table 4. These results are obtained on Cityscapes using ResNet 101 as the backbone.

Method	1/16	1/8	1/4
CPS w/ cutmix	74.72	77.62	78.93
Soft.+ CPS w/ cutmix	73.19	77.43	78.75
Nega.+ CPS w/ cutmix	75.34	77.15	78.31
U2PL [4]	74.90	76.48	78.51
FPL+CPS w/ cutmix	**75.74**	**78.47**	**79.19**

3.2. Gradient vanishing in K value selection strategy

In the K value selection strategy, we select K=n-1 instead of K=n. This practice is to alleviate the problem of gradient

vanishing. To explain this, we first perform an analysis in a simplified case where no perturbations are added in training, that is, the prediction that generates pseudo labels has the same distribution as the training prediction. We further illustrate the actual training gradient in Fig. 1.

Analysis in simplified case. To discuss training gradient, we convert the gradients of $\mathcal{L}_{us}^{f}$ to probabilistic form:

$$\begin{aligned}\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{i}} &= \frac{-\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}}{1+\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}\times\sum_{i\in\mathbb{Y}_{us}}\frac{1}{p_{us}^{i}}}\times\frac{1}{p_{us}^{i}}\\ \frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{j}} &= \frac{\sum_{i\in\mathbb{Y}_{us}}\frac{1}{p_{us}^{i}}}{1+\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}\times\sum_{i\in\mathbb{Y}_{us}}\frac{1}{p_{us}^{i}}}\times p_{us}^{j}.\end{aligned} \tag{5}$$

Here we only need to analyze the gradients of positive categories, because the absolute value of the gradient sum on the positive and negative categories are equal:

$$\begin{aligned}|\sum_{i\in\mathbb{Y}_{us}}\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{i}}| &= \sum_{j\notin\mathbb{Y}_{us}}\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{j}}\\ &= \frac{\sum_{j\notin\mathbb{Y}_{us}} e^{z_{us}^{j}}\times\sum_{i\in\mathbb{Y}_{us}} e^{-z_{us}^{i}}}{1+\sum_{j\notin\mathbb{Y}_{us}} e^{z_{us}^{j}}\times\sum_{i\in\mathbb{Y}_{us}} e^{-z_{us}^{i}}}.\end{aligned} \tag{6}$$

From Eq. 5, we see that the $\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{i}}$ is close to 0 when its numerator (i.e. $\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}$) is close to 0. According to our K value selection strategy, the lower bound of $\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}$ can be easily obtained. If we choose $K_{us}=n$, then we get:

$$\inf(\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}) = 0, \tag{7}$$

where inf means the lower bound. Eq. 7 shows that it is possible for $\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}$ to approach 0 causing the problem of gradient vanishing. When setting K_{us} an integer less than n (i.e., $K_{us}=[\alpha\cdot n], 0<\alpha<1$)), we derive that:

$$\inf(\sum_{j\notin\mathbb{Y}_{us}} p_{us}^{j}) = \frac{[\alpha\cdot n]}{n-1}(1-T). \tag{8}$$

Eq. 8 provides a lower bound for the numerator of $\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{i}}$, which alleviates the problem of gradient vanishing. In practice, we use $K_{us}=n-1$ for all our experiments.

Actual gradients in training. In actual training, the above inference will be deviated due to the influence of disturbance (e.g., data augmentation), but the conclusion still holds. Considering that our model is also subject to a supervised loss $\mathcal{L}^{sup}$ except for the fuzzy positive loss $\mathcal{L}^{f}$. A too-small gradient from $\mathcal{L}^{f}$ will lead the information of unlabeled data to be overwhelmed by the supervised loss. We illustrate the actual gradients selecting $K=n-1$ and $K=n$ in Fig. 1. It can be seen that $K=n$ brings a small training gradient while $K=n-1$ obtains a larger gradient in most mini-batches.

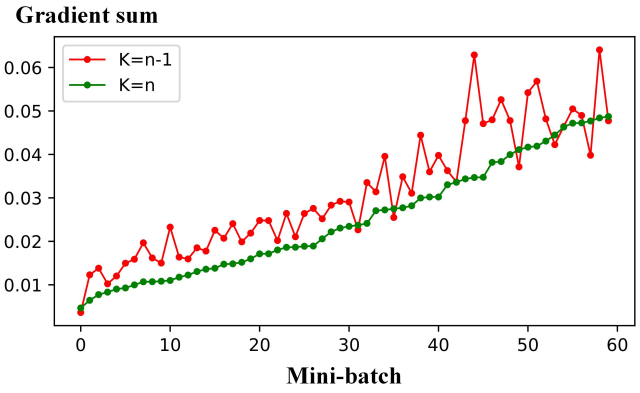

Figure 1. The gradient sum is $\sum_{j\notin\mathbb{Y}_{us}}\frac{\partial \mathcal{L}_{us}^{f}}{\partial z_{us}^{j}}$, and we sort these mini-batches by their gradient sum using $K=n$. Here we only present examples with small (i.e., prone vanishing) gradients.

3.3. More details of positive gradient score

As shown in Fig. 2 (a) and (b), we see that in Case 1, most pixels (>85%) have $K=1$ and positive gradient score R^{f} is very close to 1. Besides, we see that R^{f} is slightly lower than R^{v} in Case 3.

3.4. Gradient similarity between G^{f} and G^{ide}

In Case 2 of Sec. 3.4, though $\mathcal{L}^{f}$ encourages the GT prediction to increase which is better than existing $\mathcal{L}^{v}$, it also encourages the predictions for other positive categories to increase. Ideally, the cross-entropy loss only increases the GT prediction and suppresses the predictions for all other categories. We name the gradient computed in this ideal situation as the ideal gradient $\boldsymbol{G^{ide}}$.

Here, we propose to use the cosine similarity between the ideal gradient vector $\boldsymbol{G^{ide}}$ and our fuzzy gradient vector $\boldsymbol{G^{f}}$ brought by $\mathcal{L}^{f}$ to further analyze our FPL in Case 2. If the cosine similarity is greater than 0, it means the projection of $\boldsymbol{G^{f}}$ on $\boldsymbol{G^{ide}}$ is positive, indicating $\boldsymbol{G^{f}}$ makes our model go further in the ideal direction. For comparison, we also present the cosine similarity between the gradient vector of the vanilla method $\boldsymbol{G^{v}}$ and the ideal gradient $\boldsymbol{G^{ide}}$. Due to the complexity of predicted probability, the relationship between the cosine similarity $sim(\boldsymbol{G^{f}},\boldsymbol{G^{ide}}) = \frac{\boldsymbol{G^{f}}\cdot\boldsymbol{G^{ide}}}{|\boldsymbol{G^{f}}||\boldsymbol{G^{ide}}|}$ and 0 is not mathematically absolute. Therefore, we count $sim(\boldsymbol{G^{f}},\boldsymbol{G^{ide}})$ and $sim(\boldsymbol{G^{v}},\boldsymbol{G^{ide}})$ quantitatively. As shown in Fig. 3, we first observe that the positive rates of $sim(\boldsymbol{G^{f}},\boldsymbol{G^{ide}})$ are more than 90% in all mini-batches, which indicates that $\boldsymbol{G^{f}}$ makes our model go further in the ideal direction in most cases. Second, we see that the $sim(\boldsymbol{G^{f}},\boldsymbol{G^{ide}})$ is greater than the $sim(\boldsymbol{G^{v}},\boldsymbol{G^{ide}})$, which means our fuzzy gradient $\boldsymbol{G^{f}}$ is closer to the ideal gradient $\boldsymbol{G^{ide}}$ than the gradient from vanilla method $\boldsymbol{G^{v}}$.

The norms of $\boldsymbol{G^{f}}$ and $\boldsymbol{G^{ide}}$. The $sim(\boldsymbol{G^{f}},\boldsymbol{G^{ide}})$ only

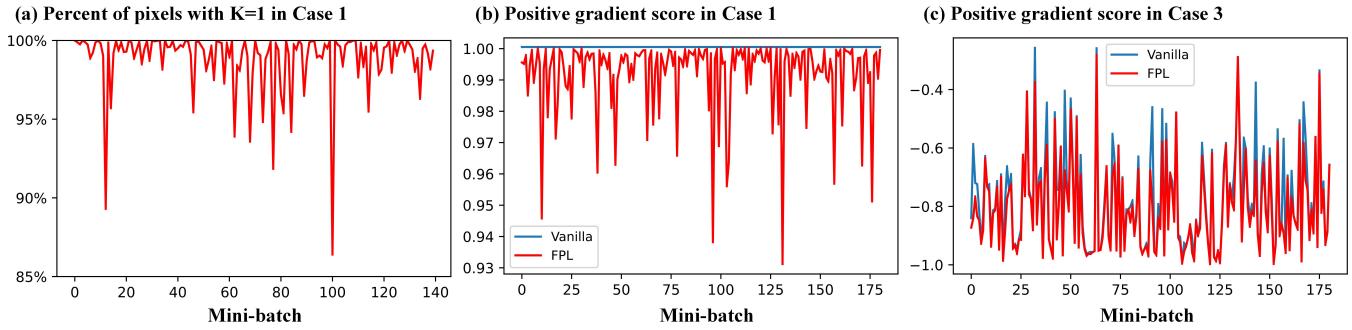

Figure 2. (a) The proportion of $K = 1$ pixels in Case 1. (b) The positive gradient score of Case 1. (c) The positive gradient score of Case 3. This figure is plotted on VOC2012 with 1/8 labeled data.

reflects that the angle between our fuzzy gradient and the ideal gradient is a mostly acute angle. But the norms of $\boldsymbol{G^f}$ and $\boldsymbol{G^{ide}}$ also affects optimization of our model. If the norm of $\boldsymbol{G^f}$ is much larger than that of $\boldsymbol{G^{ide}}$, it will cause $\boldsymbol{G^f}$ over-optimize our model, hence even if their angle is small, it will also be detrimental to optimization. We prove that the norms of $\boldsymbol{G^f}$ and $\boldsymbol{G^{ide}}$ are both range of $[0, \sqrt{2}]$:

$$
\begin{aligned}
&|N^{ide}| = \sqrt{(g^1)^2 + (g^2)^2 + ... + (g^C)^2} \\
&= \sqrt{(g^y)^2 + \sum_{i \neq y}(g^i)^2} = \sqrt{(\sum_{i \neq y} g^i)^2 + \sum_{i \neq y}(g^i)^2} \\
&\leq \sqrt{2(\sum_{i \neq y} g^i)^2} \leq \sqrt{2}
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
&|N^f| = \sqrt{(g^1)^2 + (g^2)^2 + ... + (g^C)^2} \\
&= \sqrt{\sum_{i \in \mathbb{Y}}(g^i)^2 + \sum_{i \notin \mathbb{Y}}(g^j)^2} \leq \sqrt{(\sum_{i \in \mathbb{Y}} g^i)^2 + (\sum_{i \notin \mathbb{Y}} g^j)^2} \\
&= \sqrt{2 \times (\sum_{i \notin \mathbb{Y}} g^j)^2} \leq \sqrt{2}.
\end{aligned} \tag{10}
$$

Quantitatively, we provide the norms of $\boldsymbol{G^f}$ and $\boldsymbol{G^{ide}}$ in Fig. 4. We see that the two norms are close and the norm of $\boldsymbol{G^f}$ is smaller than that of $\boldsymbol{G^{ide}}$, which means that our FPL won't bring the problem of over-optimization.

Table 5. **Ablation study on K value selection strategy.** Results are obtained on VOC2012 and Cityscapes with 1/16 labeled data.

K strategy	K=3	K=2	Step	Ours(K=n)	Ours(K=n-1)
FPL+CPS	56.71	61.09	66.39	65.98	**68.67**

4. More Empirical Studies

4.1. Ablation study for K value selection strategy

Here we evaluate the superiority of the proposed K value selection strategy by comparing our strategy with a fixed K value strategy and a step-decay K value strategy. The step decay strategy is to initialize the K value to 3 and decrease K by one every 1/3 of the total training epochs. In addition, we also verify that $K = n - 1$ is better than $K = n$ in our K value selection strategy. The results are shown in Table 5. We see that fixed K values result in a large degradation in the performance of FPL since a fixed K value causes the model to produce high-entropy predictions, making it difficult to obtain accurate classifications. For the step decay K value strategy, it achieves better results than fixed K values, because it could reduce the K value during training to obtain

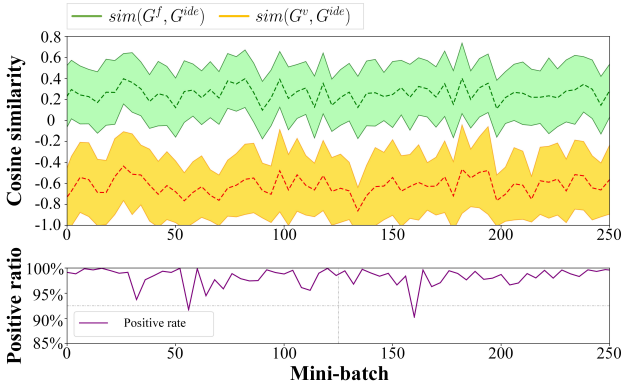

Figure 3. The cosine similarity between ideal gradient $\boldsymbol{G^{ide}}$, fuzzy gradient $\boldsymbol{G^f}$, and vanilla gradient $\boldsymbol{G^v}$. Also, we present the positive rate of the similarity between $\boldsymbol{G^{ide}}$ and $\boldsymbol{G^f}$ computed on pixels in each minibatch. This figure is counted on VOC2012 with 1/8 labeled data using the CPS [1] framework.

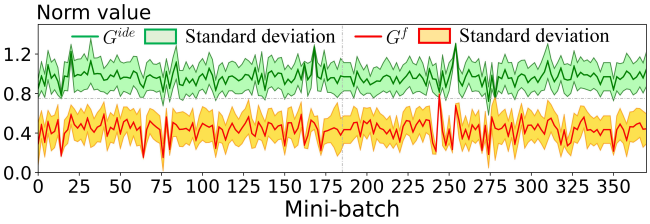

Figure 4. The norms of $\boldsymbol{G^{ide}}$ and $\boldsymbol{G^f}$ counted on VOC2012 with 1/8 labeled data using the CPS [1] framework.

Class	Wall	Fence	Truck	Terrain	Rider	Pole	Motorcycle	Train	Light	Bicycle	Sign	Sidewalk	Person	Bus	Building	Vegetation	Car	Sky	Road
AEL [2]	44.40	45.26	51.35	56.46	59.32	62.81	65.82	69.67	70.71	76.87	76.92	77.85	80.95	82.24	90.79	91.64	93.39	93.59	97.03
AEL+FPL	51.71	48.26	71.88	60.67	59.78	62.97	65.28	71.69	70.84	76.99	76.74	77.93	81.20	80.99	91.10	92.00	94.27	94.01	96.97
Improvement	7.31	3.00	20.53	4.21	0.46	0.16	-0.54	2.02	0.13	0.12	-0.18	0.08	0.25	-1.25	0.31	0.36	0.88	0.42	-0.06

Table 6. Results for each category on Cityscapes with 1/32 labeled data using AEL (ResNet 101) as the baseline.

low-entropy classifications. However, it is still worse than our proposed strategy since it makes K values the same for all pixels, ignoring their difference in the learning progress. In contrast, our method adaptively chooses the K value for each pixel according to its predicted probability distribution. We also see that $K = n - 1$ is better than $K = n$ in our K value selection strategy. This is because selecting $K = n - 1$ alleviates the gradient vanishing problem.

4.2. Class-wise improvements

The improvements that FPL brings to each category are shown in Table 6. Statistically, FPL improves hard classes significantly, e.g., Wall (7.31), Fence (3.00), Truck (20.53), and Terrain (4.21), while achieving results on par with the baseline model in other medium and easy classes. This phenomenon reflects that FPL mainly rectifies the learning of pixels in hard classes. This is because there are more wrong pseudo-labels in hard classes than in easy classes and the advantage of FPL is that it learns these wrong predicted pixels more accurately.

5. Limitations

Though works well, FPL has the limitation of high time complexity since it requires assigning a K value to each pixel. From Eq. (7) of our manuscript, we see that the time complexity of computing $\mathcal{L}^f$ is $O(C)$ when the $\mathbb{Y}$ is determined, where C is the number of classes. For vanilla $\mathcal{L}^v$, it is a special case of $\mathcal{L}^f$ when the K is fixed to 1, hence the time complexity of original $\mathcal{L}^v$ for one pixel is $O(C)$. When it comes to $\mathcal{L}^f$, we additionally need to decide the K value for each pixel of which the time complexity is $O(K)$ since it needs K times additions and K times comparisons. Hence, the time complexity of computing $\mathcal{L}^f$ is $O(KC)$ which is K times of computing the original $\mathcal{L}^v$. We also quantitatively provide the seconds of training our FPL in practice. As shown in Table 7, FPL brings about 15% additional training cost.

Table 7. **Seconds per epoch.** These statistics are measured using 8 Tesla V100 GPUs under the setting of 1/8 labeled data with ResNet 101 baseline.

Method	Cityscapes	VOC2012
AEL [2]	730s	835s
FPL+AEL	820s	985s

6. Visualization

We present more samples of K value maps during training in Fig. 5. And we illustrate some examples of our segmentation results in Fig. 6.

References

[1] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *CVPR*, pages 2613–2622, 2021. 3

[2] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *NeurIPS*, 34, 2021. 4

[3] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *ICLR*, 2020. 1

[4] 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 *CVPR*, pages 4248–4257, 2022. 1

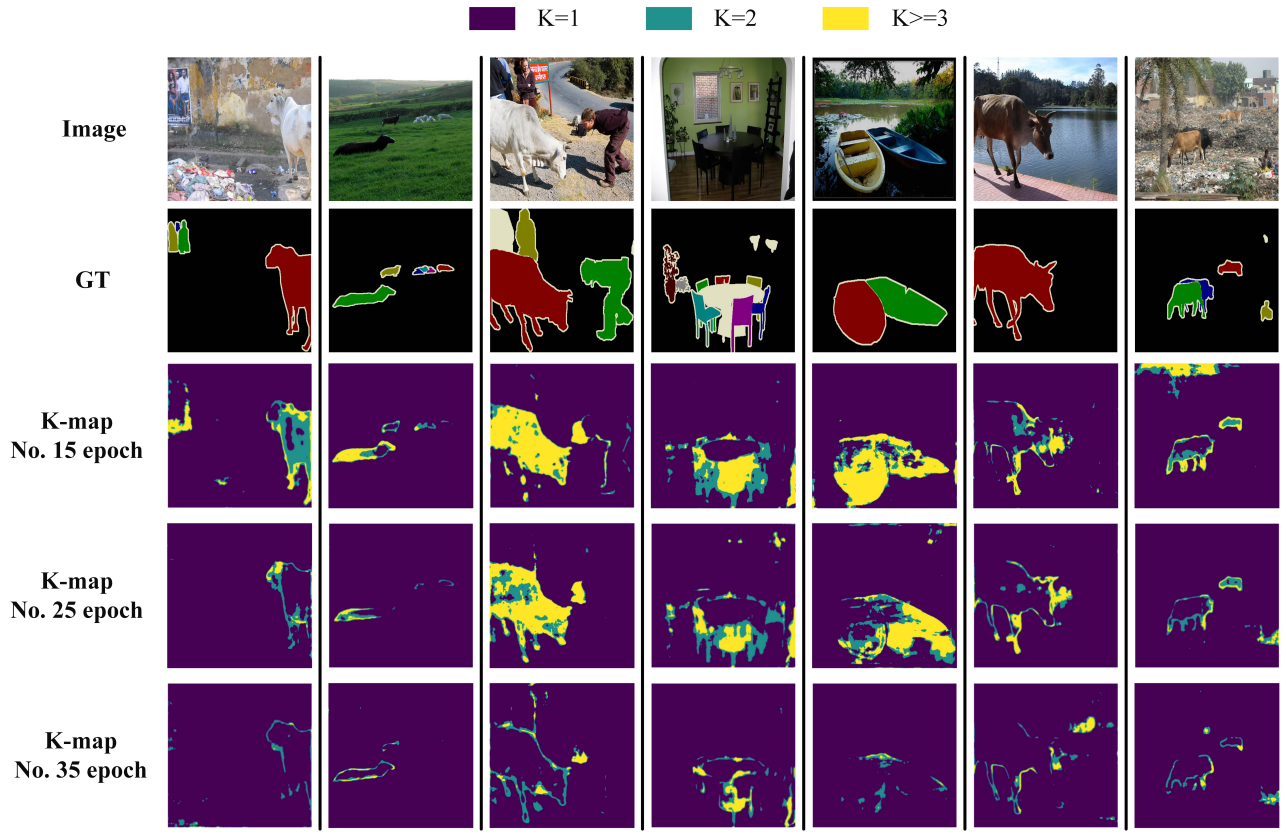

Figure 5. The visualization of K values. This figure is plotted on the VOC2012 with 1/16 labeled data using FPL+CPS w/ CutMix as the training method.

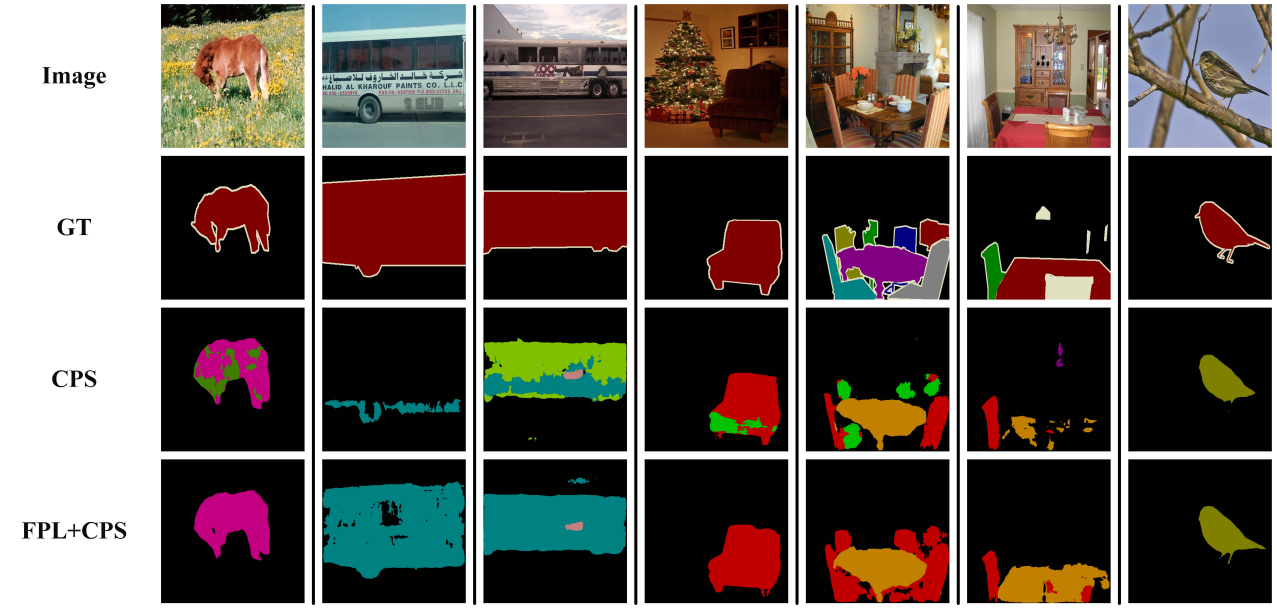

Figure 6. These segmentation results are obtained on the VOC2012 dataset with 1/16 labeled data. For clarity, we show ground truth (GT) in the form of instance labels, and the predictions of CPS and FPL+CPS are presented in the form of semantic labels.