Letter

Domain Adaptive Semantic Segmentation via Entropy-Ranking and Uncertain Learning-Based Self-Training

Chengli Peng and Jiayi Ma

Dear Editor,

This letter develops two new self-training strategies for domain adaptive semantic segmentation, which formulate self-training into the processes of mining more training samples and reducing influence of the false pseudo-labels. Particularly, a self-training strategy based on entropy-ranking is proposed to mine intra-domain information. Thus, numerous false pseudo-labels can be exploited and rectified, and more pseudo-labels can be involved in training. Meanwhile, another novel self-training strategy is developed to handle the regions that may possess false pseudo-labels. In detail, a specific uncertain loss, that makes the network automatically decide whether the pseudo-labels are true, is proposed to improve the network optimization. Consequently, the influence of false pseudo-labels can be reduced. Experimental results prove that, compared with the baseline, the average mIoU performance gain brought by our method can attain 4.3%. Extensive benchmark experiments further highlight the effectiveness of our method against existing state-of-the-arts.

Learning-based image semantic segmentation requires numerous labeled images. However, annotating pixel-wise image semantic segmentation labels is extremely time-consuming [1]. To this end, several unsupervised methods [2], [3] have been investigated and achieved competitive results compared with supervised methods. Specifically, some recent works [3], [4] use data collected from simulators and game engines or similar real-world scenes with precise pixel-level semantic annotations to train segmentation networks. However, the trained model often suffers significant performance degradation when handling unseen data from another new scene due to the cross-domain difference. To alleviate the gap between different domain data, several methods based on the unsupervised domain adaptation (UDA) technique have been developed. Normally, we call the data with pixel-level semantic annotations as the source domain data and the data from the new scene (e.g., without annotation) as the target domain data. UDA aims to use the source and target domain data to produce a model that has favorable segmentation performance on the target domain. Among the UDA methods, the ones based on self-training [5], [6] can achieve better segmentation performance on the target domain due to the consideration of the intra-domain relation (e.g., important information for improving segmentation accuracy [7], [8]) of the target domain. These methods first train a network to align the distribution shift between the source and target domain data. Then, they generate pseudo-labels of the target domain images from the trained network. Finally, they select pseudo-labels with high confidence as the training samples to implement self-training. Generally, the pseudo-labels with high confidence are referred to easy samples while those with low confidence are referred to hard samples. However, the hard samples inevitably contain numerous true pseudo-labels, leading to waste of the training samples. And the easy samples also contain false pseudo-labels, influencing optimization efficiency of the network. To solve above mentioned problems, this letter investigates two varied self-training strategies for mining more training samples and reducing influence of false pseudo-labels, respectively.

Related Work: Due to the domain otherness, the semantic segmentation model obtained from source domain usually suffer obvious performance decrease when processing images from target domain. The UDA technology can address this problem by aligning the distribution shift between the source and the target domain data. For example, AdvEnt [3] uses a generator to generate the predicted feature maps of the source and target domain data, and uses a discriminator to distinguish which domain are the feature maps from. The distribution shift between the source and the target domain data can be aligned by confusing the discriminator. In [9], a maximum squares loss is proposed to balance the gradient of well-classified target samples, which can prevent the training process being dominated by easy-to-transfer samples in the target domain.

The above UDA methods just consider the distribution shift between the source and target domain data while ignoring the distribution shift between intra-target domain data. To address this issue, self-training has been adopted. For instance, Pan *et al.* [6] used an image-wise entropy-based ranking function to separate the target domain images into easy and hard samples and then used the easy samples as the source domain and the hard samples as the target domain to implement a new round of UDA training, which can be regarded as a round of self-training inside of the target domain. MRNet [10] proposes a memory regularization in vivo to exploit the intra-domain knowledge and regularize the model training, benefiting the initialization of pseudo-labels and reducing the influence of false pseudo-labels. Based on MRNet, RPLUE [11] leverages uncertainty estimation to integrate the memory regularization in vivo, which significantly enhances the network optimization efficiency. However, these methods rarely consider the influence of true hard samples and false easy samples, leading to their limited performance. In this letter, our proposed self-training not only involves more samples for training but also limits the influence of false samples, resulting in more efficient self-training and higher segmentation accuracy.

Self-training with entropy-ranking: Due to that the hard samples contain a large number of false pseudo-labels, they can not be considered as the training samples even they have several true pseudo-labels. But, if the numbers of the false pseudo-labels in the hard samples can be reduced, they will have positive effect for the network optimization and can be considered as the training samples. According to this analysis, we use first round of self-training to refine the false pseudo-labels. How to select and rectify the false pseudo-labels are critical in this stage. Previous works prove that, if a pixel has a high entropy value, it will have high uncertainty and a high probability of getting a false pseudo-label. Considering this property, a pixel-wise entropy-ranking method is developed as a coarse pseudo-label filter to remove false pseudo-labels, namely hard samples. Meanwhile, the rest pixels after excluding the hard samples can be regarded as the pixels with true pseudo-labels, namely easy samples. Although there are still some easy samples whose pseudo-labels are false, the ratio of false samples is greatly reduced. By mining the relationship of easy samples, we could get beneficial information to rectify the false pseudo-labels.

For getting the pseudo-labels and their corresponding entropy maps, we use recent proposed popular UDA method to generate an optimized model. When we input image $I_t \in \mathbb{R}^{H\times W}$ from the target domain as the input of the optimized model, we can get a soft predicted feature map $P_t \in \mathbb{R}^{H\times W\times C}$, where C is the number of classes. With the soft predicted feature map, the pseudo-label $PL_t \in (1, C)^{H\times W}$ and the entropy map $EM_t \in \mathbb{R}^{H\times W}$ of I_t can be

Corresponding author: Jiayi Ma.

Citation: C. L. Peng and J. Y. Ma, "Domain adaptive semantic segmentation via entropy-ranking and uncertain learning-based self-training," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1524–1527, Aug. 2022.

The authors are with the Electronic Information School, Wuhan University, Wuhan 430072, China (e-mail: Pengcl@whu.edu.cn; jyma2010@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2022.105767

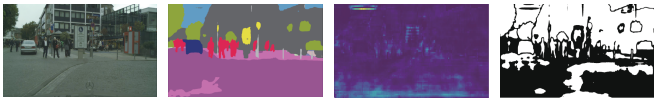
produced. Particularly, the pseudo-label is obtained as

$$PL_t = \arg\max_c P_t \tag{1}$$

where arg max is operated on the channel-dimension. EM_t can be calculated as follows:

$$EM_t = \sum_{c=1}^{C} -P_t^{(c)} \log P_t^{(c)}. \tag{2}$$

Through traversing all I_t of the target domain, we can obtain the pseudo-labels of the target domain images. Figs. 1 (b) and 1(c) display the pseudo-label and entropy map of a typical target image.



(a) Original image (b) Pseudo-label (c) Entropy map (d) Mask

Fig. 1. Pseudo-label, entropy map and mask after entropy-ranking of a target image. In the entropy map, bright color represents high entropy regions and dark color means low entropy regions. In the mask, white regions represent easy samples and black regions represent hard samples.

Subsequently, an entropy-ranking method is developed to select the hard samples. Particularly, we generate a mask $M_t \in \mathbb{R}^{H\times W}$ to distinguish the easy and hard samples according to the values of pixels in the entropy map. For the hard samples that possess high entropy values in the entropy map, their values are set to 0 in M_t. On the contrary, for the easy samples with low entropy values, their values are set to 1 in M_t. In this letter, we leverage a hyper-parameter $\lambda_r \in [0,1]$ to judge a high or low entropy value. Particularly, we first sort the pixel values of the entropy maps in descending order and get a vector $V_r \in \mathbb{R}^{1\times N_t}$, where N_t is the pixel number of the whole target domain images. Subsequently, we select the $(1-\lambda_r)\cdot N_t$-th element of V_r as the boundary. For the pixels whose entropy values surpass this element, we set them as the hard samples, and vice versa. As a result, we can get $\lambda_r \cdot N_t$ easy samples and $(1-\lambda_r)\cdot N_t$ hard samples.

As the hard samples contain more false pseudo-labels compared with the easy samples, using them as training samples to exploit intra-domain information is improper. Hence, a round of self-training just involving the easy samples is developed, as illustrated in the left part of Fig. 2. During the training stage, the network takes target image I_t as the input and produces a soft segmentation map $P_t \in \mathbb{R}^{H\times W\times C}$. The corresponding pseudo-label of I_t is $PL_t \in (1,C)^{H\times W}$. Then, the network uses PL_t to generate a one-hot vector $OH_{plt} \in \mathbb{R}^{H\times W\times C}$. The segmentation network is optimized in a supervised way by minimizing the cross-entropy loss

$$L_{seg1}(P_t, OH_{plt}, M_t) = \frac{-1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} M_t^{(h,w)} \sum_{c=1}^{C} OH_{plt}^{(h,w,c)} \log P_t^{(h,w,c)}. \tag{3}$$

With the introduction of masks, the hard samples will not contribute to model optimization, which ensures the stability of training stage. As a result, the intra-domain relationship and false pseudo-labels can be better exploited. Subsequently, we input the original training images of the target domain into the optimized model of the first round of self-training, and we can obtain the updated pseudo-labels. In the updated pseudo-labels, the false samples of hard samples are much fewer than the true samples, similar to the easy samples. Hence, each pixel in the updated pseudo-labels can be considered as an easy sample, avoiding waste of pseudo-labels.

Self-training based on uncertain learning: Since the first round of self-training can tackle the problem of easy sample shortage, the existence of false pseudo-labels still has a non-negligible influence on network optimization. We aim to address the main problem that causes the false pseudo-labels, e.g., intra-domain gap problem [6]. In general, different objects always show disparate features even they belong to the same class. Sometimes the difference between objects

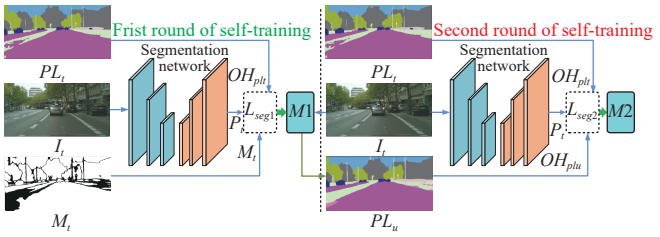


Fig. 2. The proposed self-training strategy. In the first round of self-training, the network takes original target domain image I_t, pseudo-label PL_t and mask M_t as the inputs. The network is optimized by the loss L_{seg1}. After the first round of self-training, the model (i.e., $M1$) used to update the pseudo-labels can be obtained. We input original target domain image I_t into $M1$ to get the updated pseudo-label PL_u. In the second round of self-training, the network takes original target domain image I_t, original pseudo-label PL_t and updated pseudo-label PL_u as the inputs. The network is optimized by the loss L_{seg1}. The model (i.e., $M2$) leveraged to evaluate network performance will be produced after the second round of self-training.

of the same class inside the target domain is more obvious than that across the source and target domains. However, the first round of self-training enhances the intra-domain relationship without considering this phenomenon and hence may provide some false update information. In other words, some pixels with true pseudo-labels may possess false ones after the first round of self-training. Hence, the second round of self-training should not only consider the updated pseudo-labels but also the original pseudo-labels. To achieve this goal, we propose a self-training based on uncertain learning. Particularly, the updated pseudo-labels will play the main role and the original pseudo-labels play an auxiliary role in the loss function.

The detailed process of the second round of self-training is shown in the right part of Fig. 2. The segmentation network takes the target domain image $I_t \in \mathbb{R}^{H\times W\times 3}$ as the input and produces a soft predicted feature map $P_t \in \mathbb{R}^{H\times W\times C}$. Subsequently, we use the original pseudo-label $PL_t \in (1,C)^{H\times W}$ and the updated pseudo-label $PL_u \in (1,C)^{H\times W}$ to produce two one-hot vectors $OH_{plt} \in \mathbb{R}^{H\times W\times C}$ and $OH_{plu} \in \mathbb{R}^{H\times W\times C}$, respectively. With P_t, OH_{plt} and OH_{plu}, the loss of the second round of self-training can be calculated as follows:

$$\begin{aligned} &L_{seg2}(P_t, OH_{plt}, OH_{plu}) \\ &= \frac{-1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} e^{-UC^{(h,w)}} \sum_{c=1}^{C} OH_{plu}^{(h,w,c)} \log(P_t^{(h,w,c)}) \\ &\quad + \lambda_{uc}\frac{1}{NUM_{ue}}\sum_{h=1}^{H}\sum_{w=1}^{W} UC^{(h,w)} \end{aligned} \tag{4}$$

$$UC^{(h,w)} = \begin{cases} 0, & OH_{plt} = OH_{plu} \\ Ent(h,w) + Dis(h,w), & \text{otherwise} \end{cases}$$

where λ_{uc} is a hyper-parameter used to balance every loss term and NUM_{ue} is the pixel number of the regions that possess different pseudo-labels. The uncertain term UC is an important term of this loss function. If a region has the same label in original and updated pseudo-labels, UC will be set to 0. Otherwise, UC will be obtained by computing an entropy term Ent and a distance term Dis

$$\begin{aligned} Ent(h,w) &= \sum_{c=1}^{C} -P_t^{(h,w,c)} \log(P_t^{(h,w,c)}) \\ Dis(h,w) &= \left\| \sum_{c=1}^{C} P_t^{(h,w,c)} OH_{plt} - \sum_{c=1}^{C} P_t^{(h,w,c)} OH_{plu} \right\|. \end{aligned} \tag{5}$$

Specifically, Ent can control the prediction results not tend to the classes that do not belong to the original or updated pseudo-labels, and Dis can limit the response difference between P_t to OH_{plt} and OH_{plu}. If P_t of a region satisfies the above requirements, the weight term e^{-UC} will tend to 1 and the uncertain term UC will tend to 0. In other words, this region has small uncertainty, and the loss provided

Table 1. The Performance of Network With Different Settings of λ_r on the GTA5→Cityscapes Task

λ_r	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
mIoU (%)	32.1	37.5	39.4	41.2	44.7	45.3	46.3	45.6	45.3	44.9

by this region is dependable. By minimizing this loss function, the prediction results will be close to updated pseudo-labels without ignoring the influence of original pseudo-labels. Thus the false pseudo-labels brought the intra-domain gap problem can be restrained greatly.

Experimental setup: We choose DeepLab V2 [12] as the segmentation network, which can provide a fair comparison. We leverage two representative UDA methods (i.e., AdvEnt [3] and MRNet [10]) to align the distribution shift between the source and target domain data, which can provide optimized models to generate the coarse pseudo-labels. We select four typical road scene datasets to perform our experiments, including GTA5, SYNTHIA, Cityscapes and Oxford RobotCar benchmarks. When performing GTA5→Cityscapes and SYNTHIA5→Cityscapes tasks, the Cityscapes dataset is adopted as a target domain dataset, and the GTA5 and SYNTHIA datasets are selected as the source domain data. When performing Cityscapes→Oxford RobotCar task, the Oxford RobotCar dataset is adopted as the target domain dataset, and the Cityscapes dataset is selected as the source domain data. For GTA5→Cityscapes task, we consider 19 categories for evaluation. For SYNTHIA→Cityscapes task, 13-class and 16-class subsets are used to evaluate the performance. For Cityscapes→Oxford RobotCar task, we consider 9 categories for test. Since AdvEnt and MRNet adopt different training settings, for a fair comparison, we adopt the setting provided by original papers to optimize the network. When selecting AdvEnt to generate the pseudo-labels, we train our model in 120k iterations and use mini-batch stochastic gradient descent with a batch size of 1, input size of 1028×720, momentum of 0.9, and weight decay of $5\exp(-4)$. For MRNet, we train our model in 100k iterations and use mini-batch stochastic gradient descent with a batch size of 9, input size of 1024×512, momentum of 0.9, and weight decay of $5\exp(-4)$.

Ablation experiments:

1) Self-training based on entropy-ranking: We perform ablation experiments on GTA5→Cityscapes task. We input the original target domain image into the optimized model provided by AdvEnt to obtain the original pseudo-label. The segmentation performance is evaluated by the mIoU metric. To evaluate the performance gain brought by our proposed self-training strategy, we first input the validation images of the Cityscapes dataset into the optimized model provided by AdvEnt directly, and it is worth noting that the mIoU of the original model is 43.8%. Subsequently, we input the training images of the Cityscapes dataset into the optimized model provided by AdvEnt to produce the pseudo-labels and entropy maps. According to the values of the entropy maps, we can get the easy and hard samples. Next, we conduct an ablation study to select a proper value for hyper-parameter λ_r. Particularly, we adopt λ_r among $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$ and test the valid mIoU index after the first round of self-training, and the results are illustrated in Table 1. It can be seen that the first round of self-train can obtain the mIoU of 46.3% when setting $\lambda_r = 0.7$. Compared with the optimized model provided by AdvEnt, it is a significant improvement. Therefore, in the following experiments, $\lambda_r = 0.7$ will be adopted. The ablation study regarding hyper-parameter λ_r proves that involving too many hard samples or discarding too many easy samples both have negative influences on self-training efficiency. Finally, we input the training images into the optimized model produced in the first round of self-training and generate the updated pseudo-labels. Compared with original pseudo-labels, the updated pseudo-labels have fewer false samples and every pixel can be considered as the easy sample for further optimization of the network, avoiding waste of pseudo-labels. Some typical evaluation results before and after the first round of self-training are illustrated in Figs. 3(c) and 3(d), respectively.

2) Self-training based on uncertain learning: Since we introduce the original pseudo-labels into the second round of self-training, the negative influence of the false update information can be limited. To prove this advantage, in the second round of self-training, we use the same training strategy as the first round. In other words, during the second round of self-training, we first split the updated pseudo-labels into the easy and hard samples according to the entropy-ranking, and then use the easy samples to train the network. We set different λ_r to maximize the performance of the network, and find that the performance gain brought by the second round of self-training is small (e.g., from 46.3% to 46.6%), which proves the negative effect of the false update information brought by the intra-domain problem. To limit this negative effect, we use our proposed self-training strategy to perform the second round of self-training. During the training stage, we set λ_{uc} to 1. After the second round of self-training, the mIoU on the validation dataset can reach up to 47.4%. Compared with the self-training based on entropy-ranking, our proposed method can result in higher segmentation accuracy due to that our proposed loss function can eliminate the misleading information produced in the first round of self-training and maintain beneficial information provided by the source domain. Some typical results after the second round of self-training are shown in Fig. 3(e).

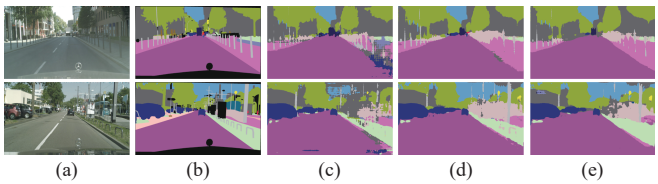


Fig. 3. Several typical segmentation results of different stages. (a) Original images; (b) Ground truth; (c) Results before the first round of self-training; (d) Results after the first round of self-training; (e) Results after the second round of self-training.

Additionally, we leverage another ablation study to demonstrate the influence of every term in uncertain term UC. Particularly, we remove entropy term Ent and distance term Dis from UC, respectively. The experimental results are reported in Table 2. It can be seen that the Ent and Dis terms can both contribute to the segmentation accuracy. Particularly, when removing Ent and Dis terms from UC simultaneously, the training loss will degenerate to a normal segmentation loss, and the second round of the self-training will be the same as the first round of self-training (e.g., $\lambda_r = 1$).

Comparison with state-of-the-art: We first validate the superiority of our proposed method on the GTA5→Cityscapes task. For achieving higher segmentation accuracy and comprehensive comparison, we further implement experiments based on MRNet. Particularly, we set λ_r and λ_{uc} to 0.7 and 0.05, respectively. The experimental results are summarized in Table 3. Clearly, our proposed method achieves better results on mIoU. Some typical results are shown in Fig. 4.

Subsequently, we compare our method with state-of-the-arts on SYNTHIA → Cityscapes task. Because AdvEnt does not provide an optimized model for the initialization of the pseudo-labels, we only implement experiments based on MRNet. We set λ_r and λ_{uc} to 0.8 and 0.1, respectively. The quantitative results are reported in Table 4 and several typical qualitative results are shown in Fig. 5. Clearly, our proposed method still achieves the best performance on the mIoU.

Finally, we compare our method with state-of-the-arts on the Cityscapes→Oxford RobotCar task. Similarly, the AdvEnt does not provide an optimized model for the initialization of the pseudo-labels. Hence, we only perform experiments based on MRNet. We set λ_r and λ_{uc} to 0.9 and 0.1, respectively. The quantitative results are reported in Table 5 and several typical qualitative results are given in Fig. 6. It can be observed that, compared with the baseline, the performance gain brought by our proposed method is much smaller than the other two tasks, which is caused by the following reasons. On the one hand, the Oxford RobotCar dataset is a real scene dataset

Table 2. Illustration of the Effect of Different Terms in *UC*. None Means That Neither *Ent* nor *Dis* is Introduced Into *UC*

	None	*Ent*	*Dis*	*Ent* + *Dis*
mIoU (%)	46.4	47.0	46.8	47.4

Table 3. The mIoU on GTA5→Cityscapes. *M*1, *M*2, *M*3, *M*4, *M*5, *M*6, *M*7 Represent PatchAlign [13], AdvEnt [3], MRKLD [14], MRNet [10], RPLUE [11], AdvEnt + Ours and MRNet + Ours, Respectively. Bold Indicates the Best

Method	*M*1	*M*2	*M*3	*M*4	*M*5	*M*6	*M*7
mIoU (%)	46.5	45.5	47.1	45.5	50.3	47.4	**50.5**

Table 4. Quantitative Results on SYNTHIA→Cityscapes. We Present Per-Class IoU, mIoU and mIoU*. mIoU and mIoU* are Averaged Over 16 and 13 Categories, Respectively. *M*1, *M*2, *M*3, *M*4, *M*5, *M*6 Represent AdvEnt [3], MRNet [10], CBST [5], MRKLD [14], RPLUE [11] and MRNet + Ours, Respectively. Bold Indicates the Best

Method	*M*1	*M*2	*M*3	*M*4	*M*5	*M*6
mIoU* (%)	48.0	50.2	48.9	50.1	54.9	**55.1**
mIoU (%)	41.2	43.2	42.6	43.8	47.9	**48.9**

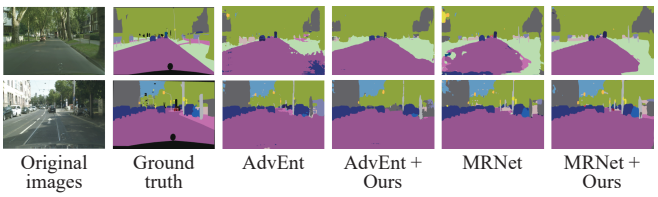


Fig. 4. Qualitative results of semantic segmentation adaptation on GTA5→Cityscapes task.

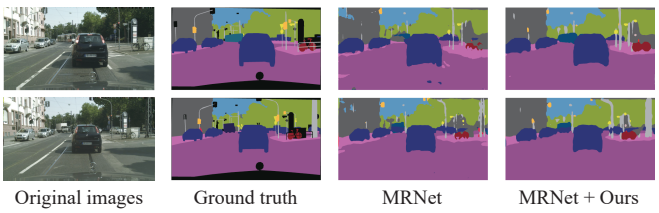


Fig. 5. Qualitative results of semantic segmentation adaptation on the SYNTHIA→Cityscapes task.

as the same as Cityscapes, leading to their small domain discrepancy and limited effect of the first round of self-training. On the other hand, the images from Oxford RobotCar have a more similar style compared with the Cityscapes dataset, resulting in weak influence of the second round of self-training. Hence, our proposed method is more suitable for images that possess obvious style differences.

Table 5. mIoU on the Cityscapes→Oxford RobotCar Task. *M*1, *M*2, *M*3, *M*4 Represent PatchAlign [13], MRNet [10], PKA [15] and MRNet + Ours, Respectively. Bold Indicates the Best

Method	*M*1	*M*2	*M*3	*M*4
mIoU (%)	72.0	72.5	73.9	**74.2**

Conclusions: In this work, we proposed a novel self-training strategy for domain adaptive semantic segmentation. Particularly, we used two rounds of self-training to address the pseudo-label waste and false pseudo-label problems, respectively. We first developed a round of self-training based on entropy-ranking to generate more easy samples. Thus, more pseudo-labels can be involved in the training, avoiding the waste of pseudo-labels. Subsequently, we developed another round of self-training based on uncertainly learning to reduce the influence of misleading information for network optimization. Experimental results proved that our proposed method can increase the performance of baseline methods significantly. Meanwhile, our proposed method can outperform several state-of-the-arts.

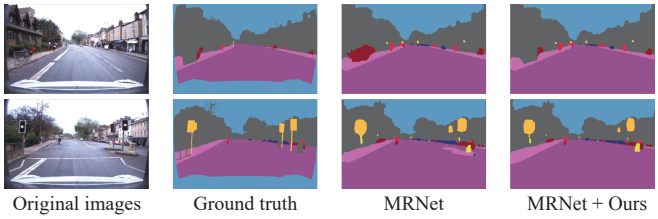


Fig. 6. Qualitative results of semantic segmentation adaptation on the Cityscapes→Oxford RobotCar task.

Acknowledgments: This work was supported by the Key Research and Development Program of Hubei Province (2020BAB113), and the Natural Science Fund of Hubei Province (2019CFA037).

References

[1] C. Sun, J. M. U. Vianney, Y. Li, L. Chen, L. Li, F.-Y. Wang, A. Khajepour, and D. Cao, "Proximity based automatic data annotation for autonomous driving," *IEEE/CAA J. Autom. Sincia*, vol. 7, no. 2, pp. 395–404, 2020.

[2] W. Wang, J. Shen, R. Yang, and F. Porikli, " Saliency-aware video object segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* , vol. 40, no. 1, pp. 20–33, 2018.

[3] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2517–2526.

[4] K. Liu, Z. Ye, H. Guo, D. Cao, L. Chen, and F.-Y. Wang, "FISS GAN: A generative adversarial network for foggy image semantic segmentation," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 8, pp. 1428–1439, 2021.

[5] Y. Zou, Z. Yu, B. Kumar, and J. Wang, " Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 289–305.

[6] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intradomain adaptation for semantic segmentation through self-supervision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3764–3773.

[7] T. Zhou, S. Qi, W. Wang, J. Shen, and S.-C. Zhu, "Cascaded parsing of human-object interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2827–2840, 2022.

[8] X. Lu, W. Wang, J. Shen, D. Crandall, and L. Van Gool, "Segmenting objects from relational visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. DOI: 10.1109/TPAMI.2021.3115815

[9] M. Chen, H. Xue, and D. Cai, " Domain adaptation for semantic segmentation with maximum squares loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2090–2099.

[10] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 1076–1082.

[11] Z. Zheng and Y. Yang, " Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1106–1120, 2021.

[12] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[13] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative patch representations," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1456–1465.

[14] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5982–5991.

[15] H. Tian, S. Qu, and Payeur, " A prototypical knowledge oriented adaptation framework for semantic segmentation," *IEEE Trans. Image Process.*, vol. 31, pp. 149–163, 2021.