Bi-directional Contrastive Learning for Domain Adaptive Semantic Segmentation - Supplementary materials

Geon Lee, Chanho Eom, Wonkyung Lee, Hyekang Park, and Bumsub Ham*

Yonsei University

1 Training

Hyperparameters. Following [3], we set $\lambda^{\mathrm{S}}_{seg} = \lambda^{\mathrm{T}}_{seg} = 1$, and $\lambda^{\mathrm{S}}_{ent} = \lambda^{\mathrm{T}}_{ent} = 0.4$. To set the other hyperparameters, we divide the training set of Cityscapes [2] into two subsets. Specifically, we divide images in the training set into two subsets of sizes 2380/595, and use them for training/cross-validation splits. We find all hyperparameters on the case of GTA5 [4] $\rightarrow$ Cityscapes [2]. For the hyperparameters λ_{FC} and λ_{BC}, we perform a grid search over λ_{FC}, λ_{BC} $\in \{0.1, 0.3, 0.5, 0.7\}$. For the momentum parameter λ and the threshold value $\mathcal{T}$, we use a grid search over $\lambda \in \{0.99, 0.999, 0.9999\}$, and $\mathcal{T} \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, respectively. The results are shown in Fig. 1. We set $\lambda_{FC} = 0.5$, $\lambda_{BC} = 0.5$, $\lambda = 0.999$, and $\mathcal{T} = 0.7$. We fix hyperparameters, and train our model on all cases including GTA5 [4] $\rightarrow$ Cityscapes [2] and SYNTHIA [5] $\rightarrow$ Cityscapes [2].

Weighted sampling. Although a pair-wise training scheme in our framework is useful for aligning cross-domain features, it may lead to a class imbalance problem due to low co-occurrence rates of rare object classes, when source and target images are sampled randomly. To mitigate this issue, we use a weighted sampling strategy that samples more images from rare object categories in a source domain. Specifically, we calculate the occurrence ratios for all labels in the source domain before training, and then compute sampling probabilities P for all labels by inverting the occurrence ratios as follows:

$$P(c) = \frac{1}{\sum_{c=1}^{C} p(c)^{-1}} p(c)^{-1}, \tag{1}$$

where we denote by $P(c)$ and $p(c)$ a sampling probability and an occurrence ratio of the c-th class, respectively. We divide the number of images containing the c-th category by the total number of images in a source dataset to obtain $p(c)$. C is the number of classes. For sampling a source-target pair in training time, we first sample a category from the sampling probability distribution P, and then pick a source image that contains the category in its ground-truth label.

* Corresponding author.

Self-distillation. Following [6, 7], we additionally apply a self-distillation technique to our final model. Specifically, we train the model using the method of [1], and use it as an initial model for the student model. We then apply a knowledge distillation, transferring the knowledge of our final model to the student which is pre-trained in a self-supervised manner. The loss for a distillation stage is as follows:

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{base}} + \text{KL}(q_t||q_s), \tag{2}$$

where q_t, and q_s denote the outputs of the teacher and student models, respectively. $\text{KL}(\cdot||\cdot)$ computes KL divergence of two predictions.

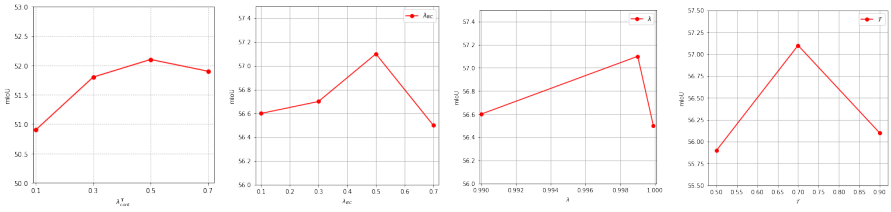
2 More experimental results

Accuracy-density plots. Static pseudo labels can be densified by lowering a threshold, but the label accuracy decreases accordingly. We provide an accuracy-density plot for static/dynamic pseudo labels in Fig. 2. We can see that dynamic pseudo labels are denser than static ones, while providing higher accuracies. We can establish more correct correspondences between source and target domains by using the calibration process. The approach of [8] neglects the biases between source and target domains. Different from [8], ours compensate for the class-wise domain biases and generate more accurate and denser labels than [8].

Domain biases. We estimate domain biases when obtaining dynamic pseudo labels. In Fig. 3, we visualize the magnitude of class-wise domain biases during training. The estimated domain biases gradually decrease during training, indicating that we successfully perform class-wise alignment between source and target domains.

Quality of pseudo labels and mIoU performance. We show the relationships between the quality of pseudo labels and mIoU performance in Table 1. By using static labels for contrastive learning, we obtain the mIoU performance of 53.5% for GTA5 $\rightarrow$ Cityscapes. And using dynamic ones for our learning leads to the mIoU improvements of 2%. By combining two labels, we generate hybrid labels, and using them leads to the mIoU performance of 57.1%. Using denser and more accurate labels contributes to the improvements of mIoU performance.

Dynamic pseudo labels. In Fig. 4, we visualize how dynamic pseudo labels change with iterations. We show the results at iteration $10k, 50k, 100k$, respectively. Dynamic pseudo labels are sparse and include vegetation, sky, and road classes at iteration $10k$. At iteration $50k$, our pseudo labels become denser and include additional classes of car and building. At iteration $100k$, our pseudo labels become more dense and accurate than labels at iteration $50k$, guiding better class-wise alignment.



(a) λ_{FC}. (b) λ_{BC}. (c) λ. (d) $\mathcal{T}$.

Fig. 1: (a-d) Quantitative comparisons for the hyperparameters, λ_{FC}, λ_{BC}, λ, and $\mathcal{T}$. We measure mIoU for all cases.

Comparison with ProDA. We compare the quality of pseudo labels for ours and ProDA in Table 2. ProDA [7] focuses on removing false-positives of pseudo labels [8] and uses sparse labels. Different from ProDA [7], we are interested in obtaining more true-positives and generating denser labels using pixel-prototype correspondences. By exploiting denser labels for the bi-directional contrastive losses, we achieve mIoU gains of 0.6% and 1.6% for GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes, respectively, compared to ProDA [7].

Performances on a different backbone. We show in Table 3 the results obtained using FCNs with VGG16 as a backbone network. We compare the segmentation performance of a baseline and our model on GTA5 $\rightarrow$ Cityscapes and SYNTHIA $\rightarrow$ Cityscapes. We show mIoU scores for both models on the validation split of Cityscapes. We can clearly see that our method boosts the performance of the baseline by significant margins.

Memory and runtime for training and testing. In Table 4, we report a total amount of memory usage and runtime for a baseline and our model at training and test times. We report the results for GTA5 $\rightarrow$ Cityscapes. Compared with the baseline, our model requires additional memory and time, but only at training time, which brings significant performance gains. For example, our method can boost the segmentation performance of the baseline by 7.6% for GTA5 $\rightarrow$ Cityscapes. Note that ours do not require extra memory and runtime at test time.

Segmentation results. We provide more visual results for GTA5 [4] $\rightarrow$ Cityscapes [2] in Fig. 5 and SYNTHIA [5] $\rightarrow$ Cityscapes [2] in Fig. 6. We compare the results of our model and those of the baseline. We can see that our model provides better segmentation outputs than the baseline for the classes of car, road, sidewalk, and rider. Our baseline model uses pseudo labels to perform the class-aware alignment. However, it does not consider intra-/inter-class variations of features. Our model learns discriminative features using the contrastive framework, and yields better segmentation results than the baseline.

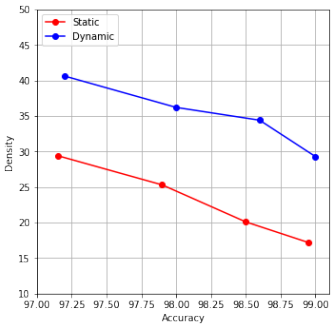


Fig. 2: An accuracy-density plot.

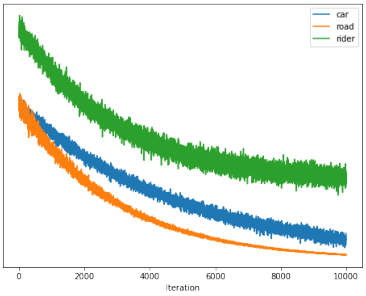


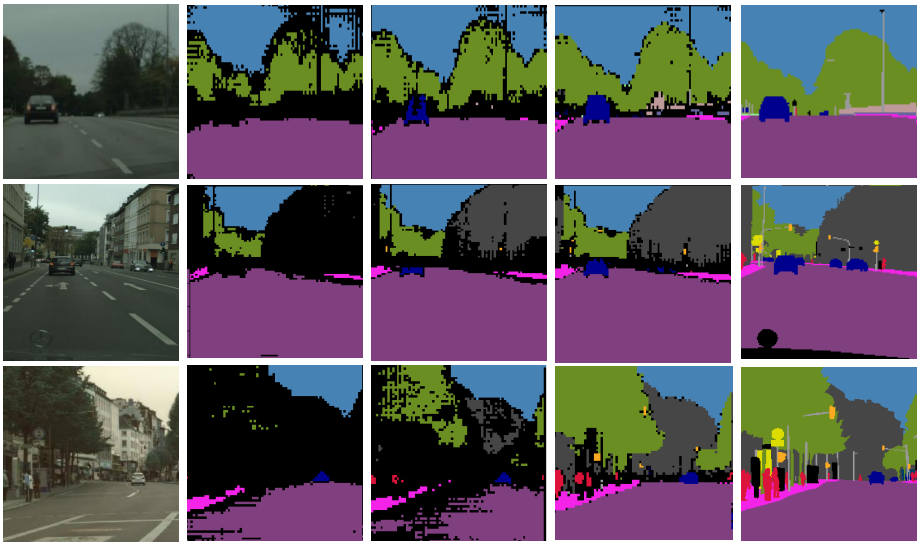
Fig. 3: Magnitude of domain biases for the classes of car, road, and rider.

Pseudo labels	Density(%)	Accuracy(%)	mIoU(%)
Static [8]	20.1	98.5	53.5
Dyn. (w/o cal.)	22.2	98.6	54.4
Dyn. (w/ cal.)	34.3	98.6	55.5
Hybrid	42.3	98.8	57.1

Table 1: Quantitative comparisons of pseudo labels (in terms of density and accuracy), and segmentation results (in terms of mIoU scores) using each label.

References

1. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
2. Cordts, M., Omran, M., Ramos, S., Scharwächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset. In: CVPR Workshop (2015)
3. Li, G., Kang, G., Liu, W., Wei, Y., Yang, Y.: Content-consistent matching for domain adaptive semantic segmentation. In: ECCV (2020)
4. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: ECCV (2016)
5. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: CVPR (2016)
6. Wang, H., Shen, T., Zhang, W., Duan, L.Y., Mei, T.: Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In: ECCV (2020)
7. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: CVPR (2021)
8. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018)



(a) Tgt. images. (b) Iter $10k$. (c) Iter $50k$. (d) Iter $100k$. (e) GT labels.

Fig. 4: Visualizations of dynamic pseudo labels. (a) Target images. (b-d) Dynamic pseudo labels at iteration $10k$, $50k$, and $100k$. (e) Ground-truth labels.

Methods	Density(%)	Accuracy(%)
ProDA [7]	18.5	99.3
Ours	42.3	98.8

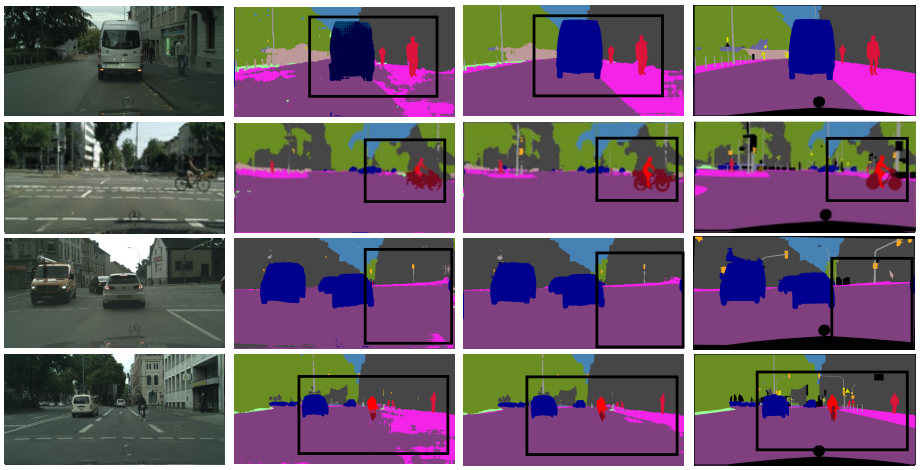
Table 2: Quantitative comparisons of pseudo labels (in terms of density and accuracy) for our method and ProDA [7].

	GTA5 → Cityscapes	SYNTHIA → Cityscapes
Baseline	41.1	43.2
Ours	48.5	52.4

Table 3: mIoU scores for a baseline and ours using the FCNs with VGG16 as a backbone network.

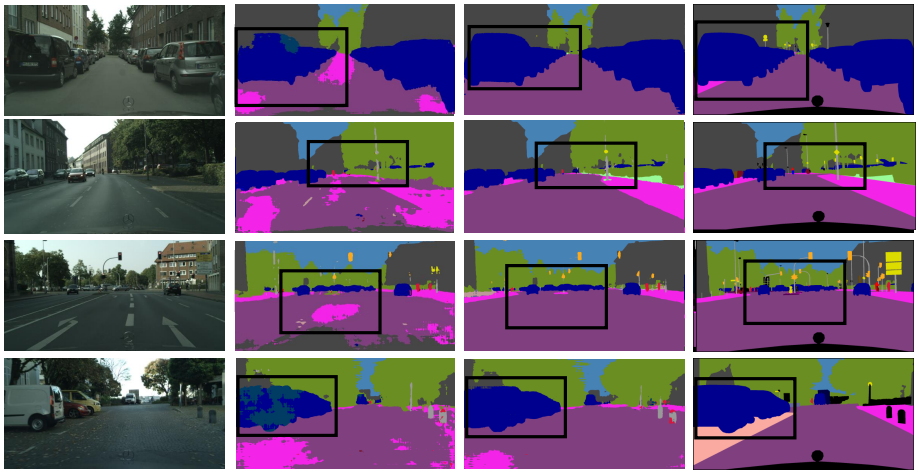
	Memory usage (Train/Test)	Runtime (Train/Test)
Baseline	24G/5G	24 hours/13 minutes
Ours	30G/5G	32 hours/13 minutes

Table 4: Quantitative comparisons of baseline and our models in terms of memory usage and runtime.



(a) Target images. (b) Our baseline. (c) Our model. (d) GT labels.

Fig. 5: Qualitative comparisons on GTA5 → Cityscapes.



(a) Target images. (b) Our baseline. (c) Our model. (d) GT labels.

Fig. 6: Qualitative comparisons on SYNTHIA → Cityscapes.