Open-Set Domain Adaptation for Semantic Segmentation

Supplementary Material

A. Implementation Details

In this section, we provide further implementation details of the proposed method. For DECON loss, we crop the target private map to a size of 64×64. Then, we apply the dilation and erosion function. The results of the dilation and erosion functions vary depending on the kernel size and iterations. In this study, we utilize 3×3 kernel size and 1 iteration. For OpenReMix, we select one thing class from the source image, resize it, and paste it to the random location of the target image. Here, we resize the selected class by a ratio of 0.5 with bilinear downsampling. We represent an example of OpenReMix in Figure 1. The resized thing class is marked with a yellow mask. In the attaching private process, the parts predicted as unknown from the target image are attached to the source image. That parts are indicated with a red mask. Additionally, we utilize MobileSAM [1] as a refinement network, which is a lightweight version of the Segment Anything Model (SAM) [2] for image segmentation. MobileSAM is a highly generalized image segmentation model that can provide reasonable masks for objects in an image even in zero-shot scenarios, but it cannot provide labels. Leveraging these label-less but precise masks, we refine the pseudo-labels. For each generated mask, the pixel count for each class is calculated, and the region of the mask is replaced entirely with the most frequent class. We apply the last 3k iterations every 10k iterations, resulting in a total of 12k iterations out of 40k iterations. And, we also apply the attaching private process in OpenReMix only when pseudo-label refinement is applied.

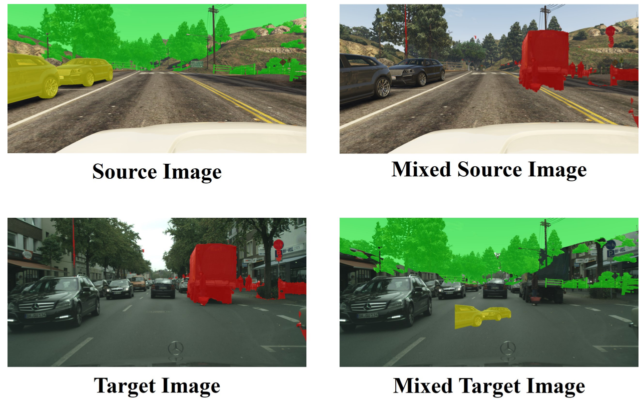

Figure 1. Example of OpenReMix. The source image is mixed with private classes from the target image (red mask). The target image is mixed by Classmix [3] (green mask) and is additionally mixed with an additional resized thing class from the source image (yellow mask).

Crop Size	GTA5 → Cityscapes	SYNTHIA → Cityscapes
32 × 32	61.22	39.39
64 × 64	**62.81**	**44.01**
128 × 128	61.30	37.62

Table 1. Sensitivity of crop size in DECON loss.

Kernel Size	GTA5 → Cityscapes	SYNTHIA → Cityscapes
3 × 3	**62.81**	**44.01**
5 × 5	60.40	35.26
7 × 7	58.57	33.37

Table 2. Sensitivity of kernel size in DECON loss.

s_r	GTA5 → Cityscapes	SYNTHIA → Cityscapes
$\mathbf{[0.5, 0.5]}$	**62.81**	**44.01**
[0, 2] (All)	59.73	36.77
[1, 2] (Upscale)	54.36	37.73
[0, 1] (Downscale)	59.84	38.90

Table 3. Sensitivity of resizing scale in OpenReMix.

B. Hyperparameter Sensitivity

B.1. Crop Size in DECON Loss

We randomly crop the target private mask and apply the dilation and erosion operation for DECON loss. Table 1 shows the experimental results on the effect of the crop size. In terms of the H-Score, we confirm the robust performance across different crop sizes. And it shows the best performance when cropped to a size of 64×64. Additionally, we observe that the performance significantly decrease in the case of 128×128. This is because, when too much target private information is included in the mask, the anchor cannot reflect the specific characteristics of a particular target private class.

B.2. Kernel Size in DECON Loss

We examine the influence of different kernel sizes when applying dilation and erosion functions for DECON loss. In Table 2, we increase the size from 3 × 3 to 7 × 7. We confirm that as the kernel size increases, the performance decreases for both scenarios. As the kernel size increases, it considers features further away from the boundary. Therefore, it hinders the model from focusing on the boundary regions where it is difficult to distinguish between known and unknown classes.

Method	Road	S.walk	Build.	Wall	Fence	Light	Veget.	Terrain	Sky	Car	Bus	M.bike	Bike	Common	Private	H-Score
								GTA5 → Cityscapes								
DAF [4]	95.80	65.37	87.12	54.08	45.81	51.78	89.20	42.93	91.03	89.19	37.93	50.54	48.49	66.09	29.23	40.53
DAF + BUS	91.90	41.06	88.04	48.65	48.74	48.94	89.59	44.37	91.61	89.99	46.09	48.49	62.47	64.61	39.23	48.82
HRDA [5]	95.31	37.70	89.26	57.41	37.00	61.16	90.96	46.86	94.39	93.39	62.45	58.13	65.71	68.44	31.02	42.70
HRDA + BUS	88.07	39.59	88.57	55.12	48.29	56.24	90.02	46.30	91.76	92.03	46.96	57.10	66.02	66.62	42.50	51.89
MIC [6]	97.14	79.45	88.78	55.6	53.92	26.11	89.94	50.98	93.54	92.46	69.09	54.53	63.43	70.38	31.78	43.79
MIC + BUS (Ours)	95.06	66.65	90.53	55.37	55.38	57.20	91.12	49.69	92.96	93.50	68.81	58.73	67.04	72.47	55.42	62.81

Table 4. Comparison with some self-training-based UDA methods. White row denotes the head-expansion baseline and gray row means our proposed BUS.

τ_p	0.3	0.4	**0.5**	0.6	0.7
H-Score	17.91	32.21	**62.81**	26.74	23.36

Table 5. Sensitivity of threshold τ_p in GTA5 → Cityscapes scenario.

Method	Common	Private	H-Score
BUDA	37.3	18.5	24.7
MIC	54.3	24.1	34.4
BUS	**55.6**	**39.7**	**46.3**

Table 6. Comparison with BUDA in Cityscapes → IDD scenario.

GTA5 → CityScapes			
# of Unknown	Config. A	Config. B	**Ours**
6	19.71	43.79	62.81
8	20.06	52.76	62.01
10	18.89	48.88	55.56

Table 7. Experiments of different private classes. Config. A denotes confidence-based MIC and config. B denotes MIC with head-expansion.

Method	Common	Private	H-Score
MIC	60.35 ± 6.55	61.38 ± 10.61	59.66 ± 3.33
BUS	$\mathbf{64.16 \pm 7.07}$	$\mathbf{66.22 \pm 11.89}$	$\mathbf{64.33 \pm 3.45}$

Table 8. Experiments on randomly selected private categories. We conducted three experiments and presented the average deviation.

B.3. Resizing Scale in OpenReMix

We provide the results on various resizing factors for OpenReMix in Table 3. For each iteration, we randomly select the scale factor from a uniform distribution within a specified range. From this result, we confirm that the proposed OpenReMix is robust to scale factors, and a simply fixed scale factor of 0.5 is enough to learn size-invariant features for our model.

B.4. Threshold in Pseudo-Label Generation

We study the influence of different thresholds τ_p for assignment of unknown classes during pseudo label generation. Table 5 shows the results under the various values of τ_p in GTA5 → Cityscapes scenario. We observe that for any value other than $\tau_p = 0.5$, the performance degrades significantly. Therefore, our method is sensitive to τ_p, so selecting an appropriate threshold is important.

C. Comparison with Other Baselines

Our proposed methods can be applied to existing self-training-based UDA methods. Therefore, we present the results applying the head expansion baseline to existing UDA methods, as well as the results incorporating the two components we propose, which are DECON loss and OpenReMix. In Table 4, we confirm the increase of H-Score for DAF [4] by $+8.29\%$ and for HRDA [5] by $+9.19\%$. Particularly, in the case of MIC [6], there was a substantial increase of $+19.02\%$. We confirm that the better the performance of UDA, the better the performance when applying our proposed methods. This is because DECON loss and attaching private process are based on the quality of pseudo labels. Therefore, the models that generate more accurate pseudo-labels have an advantage.

We also compare with the most similar work BUDA [7] to our method. In BUDA, models have access to *private category definitions*, a crucial assumption not shared by OSDA-SS. In our OSDA-SS setting, there is no provision for such private category definitions. In OSDA-SS, one should devise a method that *rejects* novel classes without needing to know any information about their definition. In BUDA, one should devise a method that *predicts* novel classes explicitly *at the expense of* predefined class definitions. Given the fundamental differences between OSDA-SS and BUDA, direct comparison is not practical. Nonetheless, we offer a comparative analysis in Table 6. To demonstrate the applicability of our proposed methods to various datasets, we conduct experiments on a new dataset called IDD (India Driving Dataset). Please note that BUDA has the *privilege to access novel class definitions* while BUS do not.

N	**1**	3	6	10
H-Score	**62.81**	46.02	38.01	30.56

Table 9. Influence of the number of expanded head in GTA5 $\rightarrow$ Cityscapes.

D. More Experiments about Private Classes

In the main paper, we experimented with a total of six private classes in the GTA $\rightarrow$ Cityscapes scenario and included results for scenarios where the number of private classes decreases. In Table 7, we further present the comparison results when the number of private classes increase. For 8 private classes, we include ("M.bike", "Bike"), and for 10 unknown classes, we additionally add ("Light", "Bus"). Despite an increase of the number of private classes, our method still outperform the other baselines.

As we mentioned in the scenario construction section in main paper, we selected private classes from the thing categories. While it is rare for stuff classes to emerge in the real world, we conduct experiments on cases where stuff classes are also treated as private classes. We randomly select 6 privates out of 19 classes regardless of thing and stuff categories in GTA5 $\rightarrow$ Cityscapes scenario. Table 8 demonstrates that BUS still outperforms the previous baseline in various settings with a significant margin.

E. What if using $(C+N)$ heads?

In our OSDA-SS task, the number of private classes N is unknown since target private labels are absent. In that sense, setting $N = 1$ for the private class is a reasonable option. Despite this, we experiment using $(C+N)$ heads with random pseudo-labeling. Understandably, Table 9 demonstrates that our BUS shows the best performance when N is set to 1.

References

[1] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 1

[2] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[3] Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 1

[4] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[5] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[6] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2

[7] Maxime Bucher, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Handling new target classes in semantic segmentation with domain adaptation. *Computer Vision and Image Understanding*, 2021. 2