SimT: Handling Open-set Noise for Domain Adaptive Semantic Segmentation
(Supplementary material)

Xiaoqing Guo[1] Jie Liu[1] Tongliang Liu[2] Yixuan Yuan[1*]
[1]City University of Hong Kong [2]University of Sydney
{xqguo.ee, jliu.ee}@my.cityu.edu.hk tongliang.liu@sydney.edu.au yxyuan.ee@cityu.edu.hk

In the supplementary material, we first summarize notations used in the manuscript. Then we provide extensive implementation details with additional qualitative and quantitative performance analysis. Towards reproducible research, we will release our code and optimized network weights. This supplementary is organized as follows:

- Section 1: Notations
- Section 2: Implementation details
  - Experimental settings (Sec. 2.1)
  - Reparameterization of SimT (Sec. 2.2)
  - Reparameterization of weighting matrix (Sec. 2.3)
- Section 3: Experimental results
  - Analysis on open-set classes (Sec. 3.1)
  - Segmentation visualization of SFDA (Sec. 3.2)

1. Notations

We summarize the notations utilized in the manuscript, as listed in Table 1.

2. Implementation details

2.1. Experimental settings

Hyper-parameters in Endovis17→Endovis18 scenario. We adopt polynomial learning rate scheduling to optimize the feature extractor with the initial learning rate of 1e-4, while it is set as 1e-3 for the optimization of classifier and SimT. The batch size is set as 8, and the maximum epoch number is 30. Hyper-parameters of n, λ, α, β, γ are set as 5, 0.1, 1.0, 1.0, 0.1 in our implementation.

Detailed training and inference procedure. We adopt DeepLab-v2 [3] backbone with encoder of ResNet-101 [5] as our segmentation model. Given the pseudo-labeled target domain data derived from a given black-box model $f_b(\cdot)$, we calculate the class distribution of pseudo labels $\boldsymbol{C}$. During training phase, we first warm up the whole segmentation model to obtain $f(\cdot)_{\mathbf{w}^{fixed}}$ with C-way output probabilities using the generated pseudo-labeled target domain data. The warm-up model is utilized to produce noisy class posteriors for anchor points and derive confidence score for each pixel. The classifier of segmentation model $f(\cdot)_{\mathbf{w}^{fixed}}$ is then extended to output $(C+n)$-way clean class posterior

Table 1. Notation Table.

Symbol	Description
$\boldsymbol{T}$	SimT
$\boldsymbol{x}$	Pixel
$\widetilde{\boldsymbol{y}}$	Pseudo label of pixel $\boldsymbol{x}$
$\boldsymbol{y}$	Ground truth label of pixel $\boldsymbol{x}$
X_t	Target domain image
$\widetilde{Y}_t$	Pseudo label of image X_t
Y_t	Ground truth label of image X_t
$\mathcal{X}_{\mathcal{T}}$	Set of target domain images
$\mathcal{Y}_{\mathcal{T}}$	Set of target domain pseudo segmentation labels
$p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$	Noisy class posterior probability
$p(\boldsymbol{y} \mid \boldsymbol{x})$	Clean class posterior probability
$f_b(\cdot)$	Balck-box model
$f(\cdot)_{\mathbf{w}}$	Segmentation model parameterized by $\mathbf{w}$
$f(\cdot)_{\mathbf{w}^{fixed}}$	Warm-up model parameterized by $\mathbf{w}^{fixed}$
$\boldsymbol{x}^c$	Anchor point of class c
$\boldsymbol{x}_k$	Confident closed-set pixel
$\widetilde{\boldsymbol{y}}_k$	Label of confident closed-set pixel $\boldsymbol{x}_k$
$\boldsymbol{x}_u$	Confident open-set pixels
$\widetilde{\boldsymbol{y}}_u$	Label of confident open-set pixel $\boldsymbol{x}_u$
$X_{\mathcal{K}}$	Set of confident closed-set pixels
$X_{\mathcal{U}}$	Set of confident open-set pixels
$\boldsymbol{u}$	Weighting matrix
$\mathcal{L}_{ST}$	Self-training loss
$\mathcal{L}_{LC}$	Loss correction
$\mathcal{L}_{Volume}^{SimT}$	Volume regularization
$\mathcal{L}_{Anchor}^{SimT}$	Anchor guidance
$\mathcal{L}_{Convex}^{SimT}$	Convex guarantee
α, β, γ	Regularization coefficients

*Yixuan Yuan is the corresponding author.

This work was supported by Hong Kong Research Grants Council (RGC) General Research Fund 11211221(CityU 9043152).

probabilities, and the extended model is denoted as $f(\cdot)_{\mathbf{w}}$. Incorporating SimT, we only update $conv3$ and $conv4$ layer in feature extractor and $(C+n)$-way classifier through gradient decent derived from corrected loss. During inference phase, we obtain final predictions from first C-way of the extended model $f(\cdot)_{\mathbf{w}}$ directly.

2.2. Reparameterization of SimT ($\boldsymbol{T}$)

To make SimT ($\boldsymbol{T}$) differentiable and satisfy the condition of $\boldsymbol{T} \in [0,1]^{(C+n)\times C}$, $\sum_{k=1}^{C} \boldsymbol{T}_{jk} = 1$, we utilize the reparameterization method, and the code is shown in Listing 1. Specifically, we randomly initialize a matrix $\boldsymbol{U} \in \mathbb{R}^{(C+n)\times C}$, as in lines 8-14. To preserve the diagonally dominant property of closed-set part of SimT ($\boldsymbol{T}_{1:C,:}$), the diagonal prior $\boldsymbol{I}$ is introduced in line 16, which has been widely used in the literature of NTM estimation [7, 10]. Considering segmentation model tends to classify samples into majority categories, the class distribution of pseudo labels $\boldsymbol{C}$ is involved in lines 18-20. Incorporating these prior informations, we obtain $\boldsymbol{V} = \boldsymbol{C} \cdot \sigma(\boldsymbol{U}) + \boldsymbol{I}$ in lines 23-24, where $\sigma(\cdot)$ is the sigmoid function to avoid negative value in SimT. Then we do the normalization of $\boldsymbol{T}_{jk} = \frac{\boldsymbol{V}_{jk}}{\sum_{k=1}^{C} \boldsymbol{V}_{jk}}$ to derive SimT ($\boldsymbol{T}$) in line 25. Since both sigmoid function and normalization operation are differentiable, SimT can be updated through gradient descent on $\boldsymbol{U}$.

```
1  import torch
2  import torch.nn as nn
3  import torch.nn.functional as F
4  import numpy as np
5  class SimT(nn.Module):
6      def __init__(self, num_classes, open_classes
       =0):
7          super(SimT, self).__init__()
8          T = torch.ones(num_classes+open_classes,
       num_classes)
9
10         self.register_parameter(name='NTM', param
       =nn.parameter.Parameter(torch.FloatTensor(T))
       )
11
12         self.NTM  # U
13
14         nn.init.kaiming_normal_(self.NTM, mode='
       fan_out', nonlinearity='relu')
15
16         self.Identity_prior = torch.cat([torch.
       eye(num_classes, num_classes), torch.zeros(
       open_classes, num_classes)], 0) # I
17
18         Class_dist = np.load('../ClassDist/
       AdaptSegNet_CD.npy') # C
19
20         self.Class_dist = torch.FloatTensor(np.
       tile(Class_dist, (num_classes + open_classes,
        1)))
21
22     def forward(self):
23         T = torch.sigmoid(self.NTM).cuda()
24         T = T.mul(self.Class_dist.cuda().detach()
       ) + self.Identity_prior.cuda().detach() # V
25         T = F.normalize(T, p=1, dim=1) # SimT
26         return T
```

Listing 1. Reparameterization of SimT ($\boldsymbol{T}$)

2.3. Reparameterization of weighting matrix ($\boldsymbol{u}$)

In the proposed convex guarantee of SimT, we introduce a weighting matrix $\boldsymbol{u} \in [0,1]^{(C+n)\times(C+n)}$ with constraints of $\boldsymbol{u}_{j,k=j} = -1$ and $\sum_k \boldsymbol{u}_{j,k\neq j} = 1$. To make the weighting matrix $\boldsymbol{u}$ differentiable and satisfy its constraints, we utilize the reparameterization method, as shown in Listing 2. To be specific, we uniformly initialize a matrix $\boldsymbol{W} \in \mathbb{R}^{(C+n)\times(C+n)}$ except diagonal entries in lines 9-15. The softmax operator is introduced to ensure the summation of non-diagonal entries in each row of $\boldsymbol{u}$ to be 1, as shown in line 24. Diagonal entries are all detached and set as -1, as in lines 17-25. Since the softmax operation is differentiable, the weighting matrix $\boldsymbol{u}$ can be updated through gradient descent on $\boldsymbol{W}$.

```
1  import torch
2  import torch.nn as nn
3  import torch.nn.functional as F
4  import numpy as np
5  class sig_u(nn.Module):
6      def __init__(self, num_classes, open_classes
       =0):
7          super(sig_u, self).__init__()
8
9          self.classes = num_classes+open_classes
10
11         init = 1./(self.classes-1.)
12
13         self.register_parameter(name='weight',
       param=nn.parameter.Parameter(init*torch.ones(
       self.classes, self.classes)))
14
15         self.weight # W
16
17         self.identity = torch.zeros(self.classes,
        self.classes) - torch.eye(self.classes)
18
19     def forward(self):
20         ind = np.diag_indices(self.classes)
21         with torch.no_grad():
22             self.weight[ind[0], ind[1]] = -10000.
        * torch.ones(self.classes).detach()
23
24         w = torch.softmax(self.weight, dim = 1).
       cuda()
25         weight = self.identity.detach().cuda() +
       w # u
26         return weight
```

Listing 2. Reparameterization of weighting matrix ($\boldsymbol{u}$)

3. Experimental results

3.1. Analysis on open-set classes

In UDA of GTA5→Cityscapes scenario, the compatible label set shared between GTA5 [11] and Cityscapes [4] includes 19 classes, *i.e.* 'road', 'sidewalk', 'building', 'wall',

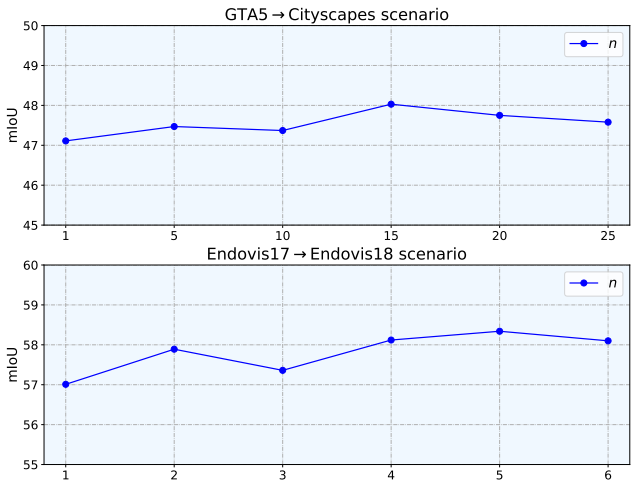

Figure 1. Qualitative results of SFDA in Endovis17→Endovis18 scenario. (a) Target image, (b) Ground truth, Predictions from (c) source only model, (d) IGNet [8], (e) ours (SimT).

'fence', 'pole', 'traffic-light', 'traffic-sign', 'vegetable', 'terrain', 'sky', 'person', 'rider', 'car', 'truck', 'bus', 'train', 'motor', 'bike'. In target domain (Cityscapes), 15 open-set classes, including 'ego vehicle', 'rectification border', 'out of roi', 'static', 'dynamic', 'ground', 'parking', 'rail track', 'guard rail', 'bridge', 'tunnel', 'polegroup', 'caravan', 'trailer', 'license plate' are unknown in source domain (GTA5)[a]. In the proposed SimT, n is a hyper-parameter that indicates the potential open-set class number, and we set $n = 15$ to implicitly model the diverse semantics within open-set classes. Considering the prior of open-set class number is not available in real-word deployment, we tune it in the range of $\{1, 5, 10, 15, 20, 25\}$ and show the influence of open-set class number n, as illustrated in the upper row of Figure 1. In this experiment, we adopt AdaptSegNet [12] as the black-box model to generate pseudo labels for target domain data. It is observed that $n = 1$ shows the inferior performance. This result validates that multiple open-set ways of classifier capacitate the segmentation model to encode the diverse feature representations within open-set regions. Moreover, we also observe that the performance of segmentation model remains stable across a wide range of n, indicating the robustness of the proposed SimT.

In UDA of Endovis17→Endovis18 scenario, the compatible label set of instrument types shared between Endovis17 [2] and Endovis18 [1] includes 3 classes, *i.e.* 'scissor', 'needle driver', 'forceps'. In target domain (Endovis18), 3 open-set instrument type classes, including 'ultrasound probe', 'suction instrument', 'clip applier' are as the unknown classes in source domain (Endovis17). We

[a] https://github.com/mcordts/cityscapesScripts/blob/master/cityscapesscripts/helpers/labels.py

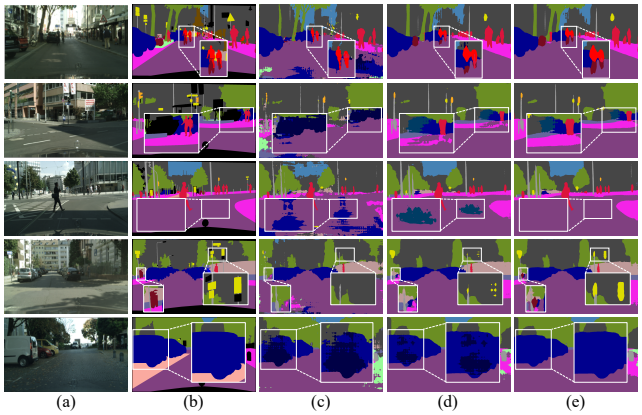

Figure 2. Qualitative results of SFDA in GTA5→Cityscapes scenario. (a) Target image, (b) Ground truth, Predictions from (c) source only model, (d) SFDASeg [6], (e) ours (SimT).

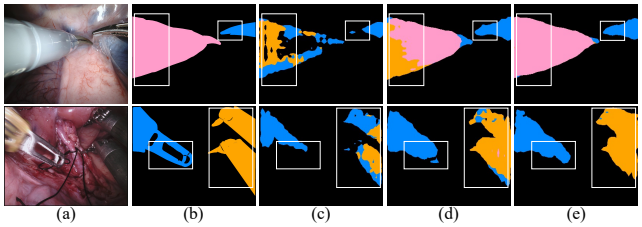

Figure 3. Qualitative results of SFDA in Endovis17→Endovis18 scenario. (a) Target image, (b) Ground truth, Predictions from (c) source only model, (d) SFDASeg [6], (e) ours (SimT).

tune the open-set class number n in the range of $\{1, 2, 3, 4, 5, 6\}$, and the comparison results are shown in the lower row of Figure 1. In this experiment, the black-box model is borrowed from IGNet [8] to generate pseudo labels for target domain data. It is clear that the verification performance of segmentation model with various n settings remains stable within a wide range.

3.2. Segmentation visualization of SFDA

In Figure 2, we present segmentation results on the SFDA semantic segmentation task in GTA5→CityScapes scenario. As shown in the figure, the source only model (c) produces the worst segmentation results, since the extracted features for target data are not discriminative enough. The baseline SFDA model (d) [6] obtains more precise segmentation predictions in comparison to the source only model, but is error-prone in some ambiguous categories (*e.g.*, 'rider' and 'bike', 'road' and 'sidewalk') and small-scale objects (*e.g.*, 'traffic sign'). In the results of our SimT approach, these mistakes are effectively mitigated, resulting in more reasonable segmentation predictions. We conjecture the reason is that our method can adaptively model the noise distribution of pseudo labels in target domain and learn the discriminative feature representation of target data

with the corrected supervision signals.

We provide segmentation results on the SFDA semantic segmentation task of Endovis17→Endovis18 in Figure 3. The qualitative results show that without adaptation (source only model), it is difficult to correctly identify the surgical instruments due to the limited discriminative capability on the target data. The predicted instrument regions are fragmentary, deteriorating the segmentation performance. Compared with the baseline SFDA model (d) [6], the proposed approach (e) generates more semantically meaningful segmentation results, demonstrating the superior property of SimT in alleviating the noise issues in SFDA task.

References

[1] Max Allan, Satoshi Kondo, Sebastian Bodenstedt, Stefan Leger, Rahim Kadkhodamohammadi, Imanol Luengo, Felix Fuentes, Evangello Flouty, Ahmed Mohammed, Marius Pedersen, et al. 2018 robotic scene segmentation challenge. *arXiv preprint arXiv:2001.11190*, 2020. 3

[2] Max Allan, Alex Shvets, Thomas Kurmann, Zichen Zhang, Rahul Duggal, Yun-Hsuan Su, Nicola Rieke, Iro Laina, Niveditha Kalavakonda, Sebastian Bodenstedt, et al. 2017 robotic instrument segmentation challenge. *arXiv preprint arXiv:1902.06426*, 2019. 3

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017. 1

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[6] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. 2021. 3, 4

[7] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. In *ICML*, pages 6403–6413, 2021. 2

[8] Jie Liu, Xiaoqing Guo, and Yixuan Yuan. Graph-based surgical instrument adaptive segmentation via domain-common knowledge. *IEEE Trans. Med. Imag.*, 2021. 3

[9] Yahao Liu, Jinhong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *ICCV*, pages 8801–8811, 2021.

[10] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017. 2

[11] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, pages 102–118. Springer, 2016. 2

[12] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018. 3

[13] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, pages 5982–5991, 2019.