FREST: Feature RESToration for Semantic Segmentation under Multiple Adverse Conditions
—Supplementary Material—

Sohyun Lee[1], Namyup Kim[2], Sungyeon Kim[2], and Suha Kwak[1,2]

[1] Graduate School of Artificial Intelligence, POSTECH, Korea
[2] Department of Computer Science and Engineering, POSTECH, Korea
https://sohyun-l.github.io/frest

This supplementary material presents additional experimental details and results that are omitted from the main paper due to the space limit. Firstly, we include more implementation details in Sec. A. Then, Sec. B shows the algorithm of FREST and Sec. C presents a thorough analysis of FREST, covering aspects such as the analysis on FREST via t-SNE and the impact of the adverse condition discriminating loss. Finally, we offer an additional ablation study in Sec. D, containing combinations of losses, patch confidence threshold, and extended quantitative and qualitative results in Sec. E and Sec. J.

A Implementation Details

In this section, we present the implementation settings that are omitted from the main paper. All experiments were conducted using a single A6000 GPU. We train both the segmentation network and the condition strainer for 8 epochs while training only the condition strainer for 2 epochs to train a stable condition strainer. Each mini-batch consists of one image for each adverse and normal image. During training, images are cropped to 1080×1080 and flipped horizontally at random. Additionally, we utilize the exponential moving average (EMA) model for implementing a model with a stopping gradient in our framework, the weight parameter is set as 0.9999 to preserve the source knowledge. For the condition strainer, we initialize up-projecting parameters with He uniform initialization and down-projecting parameters with zero initialization. This initialization strategy aims to optimize the learning efficiency and model stability in handling condition-specific features.

B Algorithm of FREST

We present the training procedure of FREST in Algorithm 1.

C Empirical Analysis

C.1 Analysis on FREST

To investigate the impact of FREST, we show that it reduces the condition gaps between adverse and normal conditions. To this end, visualize t-SNE [7] using

Algorithm 1 : Training Algorithm of FREST

Input: Condition strainer: ψ_{strainer}, Projection head: ψ_{proj}, l^{th} layer of encoder: ϕ^l_{enc}, Decoder: ϕ_{dec}, Number of layers: L, Prediction: P, Pseudo Label: $\hat{Y}$, Input images: $\{x_{\text{adv}}, x_{\text{norm}}\}$

Output: Optimized encoder ϕ_{enc} and decoder ϕ_{dec}.

for $\{1, \ldots, \#$ of training iterations$\}$ **do**
  Sample mini-batch $\{x_{\text{adv}}, x_{\text{norm}}\}$
  for $\{l \longleftarrow 1$ to $L\}$ **do**
    $\mathbf{c}^l_{\text{adv}} = \phi^l_{\text{enc}}(\mathbf{c}^{l-1}_{\text{adv}}) + \psi^l_{\text{strainer}}(\mathbf{c}^{l-1}_{\text{adv}})$
    $\mathbf{c}^l_{\text{norm}} = \phi^l_{\text{enc}}(\mathbf{c}^{l-1}_{\text{norm}}) + \psi^l_{\text{strainer}}(\mathbf{c}^{l-1}_{\text{norm}})$
  end for
  $\mathcal{L}_{\text{spec}} = \mathcal{L}_{\text{spec}}(\psi_{\text{proj}}(\mathbf{c}_{\text{adv}}), \psi_{\text{proj}}(\mathbf{c}_{\text{norm}}))$ ▷ Eq. (1)&(2)
  Update projection head ψ_{proj}
  $\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{self}}(P_{\text{adv}}, \hat{Y}_{\text{adv}})$
  $\mathcal{L}_{\text{strainer}} = \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \mathcal{L}_{\text{self}}$
  Update condition strainer ψ_{strainer} with $\mathcal{L}_{\text{strainer}}$ ▷ Step 1
  for $\{l \longleftarrow 1$ to $L\}$ **do**
    $\mathbf{f}^l_{\text{adv}} = \phi^l_{\text{enc}}(\mathbf{f}^{l-1}_{\text{adv}})$
    $\mathbf{c}^l_{\text{adv}} = \phi^l_{\text{enc}}(\mathbf{f}^{l-1}_{\text{adv}}) + \psi^l_{\text{strainer}}(\mathbf{f}^{l-1}_{\text{adv}})$
    $\mathcal{L}^l_{\text{dis}} = -\mathcal{L}^l_{\text{dis}}(\mathbf{f}^l_{\text{adv}}, \mathbf{c}^l_{\text{adv}})$ ▷ Eq. (4)
  end for
  $\mathcal{L}_{\text{self}} = \mathcal{L}_{\text{self}}(P_{\text{adv}}, \hat{Y}_{\text{adv}})$
  $\mathcal{L}_{\text{resto}} = \mathcal{L}_{\text{resto}}(\psi_{\text{proj}}(\mathbf{f}_{\text{adv}}), \psi_{\text{proj}}(\mathbf{c}_{\text{norm}}))$ ▷ Eq. (3)
  $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{self}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{resto}} + \lambda_{\text{dis}} \sum_{l=1} \mathcal{L}^l_{\text{dis}}$
  Update ϕ_{enc} and ϕ_{dec} of the model with $\mathcal{L}_{\text{total}}$ ▷ Step 2
end for

our segmentation features under adverse conditions (*i.e.* fog, night, rain, and snow) and condition-specific features under the normal condition (*i.e.* normal) in the condition embedding space learned by FREST. In detail, we utilize the validation images from the ACDC dataset [9] input images, and we compute their condition embeddings using both the condition strainer and the projection head. Fig. a1 shows that FREST effectively reduces the condition gaps between adverse and normal conditions well, which suggests that FREST achieves condition-invariance through feature restoration.

C.2 Additional Analysis on Fig. 6

We present a detailed analysis on FREST. Please note that the goal of FREST is not image restoration but feature restoration, which means FREST aims to train a model to robustly recognize adverse condition images as if they were in normal conditions, not to convert them into normal images. As shown in Fig. 6 and Fig. a2, we reconstructed images only to qualitatively investigate the impact of feature restoration. They show the favorable impact of FREST on recognition,

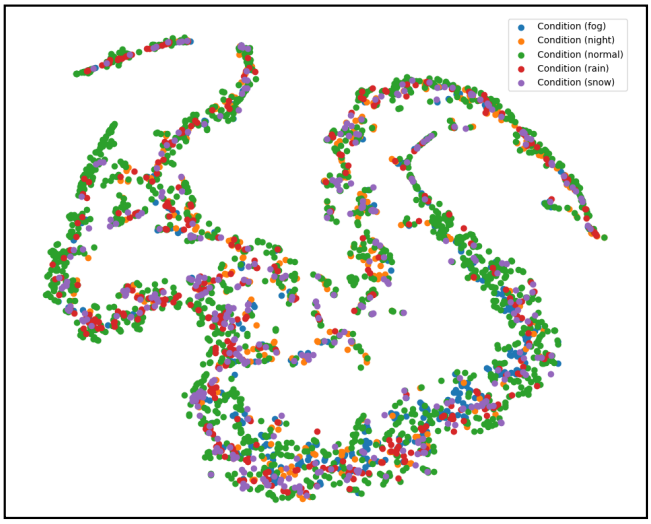

Fig. a1: t-SNE visualization of the distribution of condition embeddings in the condition embedding space.

particularly in enhancing the boundaries of buildings, compared to the baseline results.

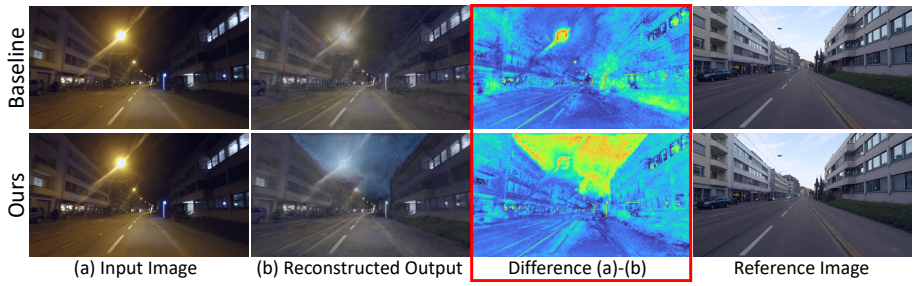

Fig. a2: Qualitative analysis on FREST.

C.3 Analysis on Adverse Condition Discriminating Loss

We devised the adverse condition discriminating loss to further mitigate the adverse effects of our segmentation feature, which is computed from the segmentation encoder. To implement this strategy, we introduced a condition discriminator which classifies each condition (*i.e.*, adverse and normal conditions). For in-depth analysis, we introduce other possible solutions to remove the adverse effects.

Our initial approach is employing residual learning [4] to eliminate condition-specific information from the encoder features $\mathbf{f}_{\text{adv}}$ as illustrated in Fig. a3(a). Additionally, we implement a strategy of minimizing mutual information [3] between the encoder feature $\mathbf{f}_{\text{adv}}$ and condition-specific feature $\mathbf{c}_{\text{adv}}$ as shown in Fig. a3(b). Furthermore, we maximize the feature distance between the encoder feature $\mathbf{f}_{\text{adv}}$ and condition-specific feature $\mathbf{c}_{\text{adv}}$ by utilizing L1 loss as Fig. a3(c). Finally, to utilize the domain information more effectively, we maximize the feature statistics distance. For this, we calculate the mean and standard deviation of features as feature statistics and maximize feature statistics between $\mathbf{f}_{\text{adv}}$ and $\mathbf{c}_{\text{adv}}$ as Fig. a3(d). Subsequently, through empirical analysis, we demonstrate empirically that our adverse condition discriminating loss, implemented through feature classification, is the most effective method when compared to the aforementioned possible solutions.

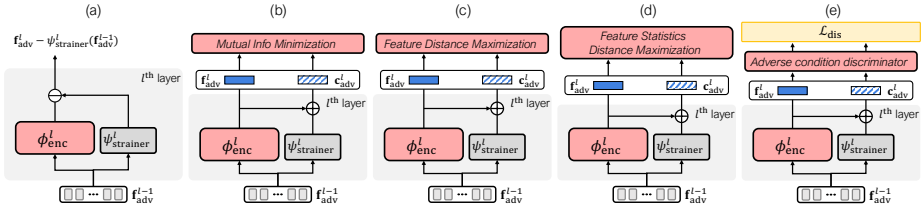

Fig. a3: Illustrations for the possible solutions of removing adverse effects from the segmentation feature computed by the segmentation encoder. (a) Residual learning (b) Minimizing mutual information (c) Maximizing feature distance (d) Maximizing feature statistics distance (e) Feature classification (Ours)

Table a1: Analysis of possible solutions for adverse condition distancing loss. The results are reported in mIoU on the ACDC validation set.

Possible Solution	mIoU
(a) Residual learning	62.8
(b) Minimizing mutual information	65.6
(c) Maximizing feature distance	66.5
(d) Maximizing feature statistics distance	67.4
(e) Feature classification (Ours)	**68.6**

D Additional Ablation Study

D.1 Effect of Each Loss

We investigate contributions of entropy loss $\mathcal{L}_{\text{ent}}$, self-training loss $\mathcal{L}_{\text{self}}$, and our proposed losses (*i.e.* feature restoration loss $\mathcal{L}_{\text{resto}}$ and adverse condition discriminating loss $\mathcal{L}_{\text{dis}}$) on ACDC [9] validation performance. To this end, we evaluate our model without each loss. As presented in Table a2, all the losses contribute to the performance. Especially, each impact of $\mathcal{L}_{\text{self}}$ and our proposed losses (*i.e.* feature restoration loss $\mathcal{L}_{\text{resto}}$ and adverse condition discriminating loss $\mathcal{L}_{\text{dis}}$) is larger than another component.

Table a2: Effect of additional losses as self-training and entropy minimization loss. The results are reported in mIoU on the ACDC validation set.

w/o $\mathcal{L}_{\text{self}}$	w/o $\mathcal{L}_{\text{ent}}$	w/o $\mathcal{L}_{\text{resto}}$&$\mathcal{L}_{\text{dis}}$	mIoU
✓			59.1
	✓		67.8
		✓	62.7
			68.6

D.2 Effect of Patch Confidence Threshold

Table a3 presents the sensitivity of FREST performance to the confidence threshold value in Eq. (3) and Eq. (4). These results show that our method is insensitive to the hyperparameter of confidence.

Table a3: Effect of the confidence threshold on FREST. The results are reported in mIoU on the ACDC validation set.

Confidence Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6
mIoU	67.5	68.2	**68.6**	68.0	67.7	67.3	67.4

E Condition-Wise Performance

In this section, we present the condition-wise test results on ACDC Fog, ACDC Night, ACDC Rain, and ACDC Snow [9]. In Table a6, a7, a8, a9, FREST outperforms all the competitors in the four condition splits.

F Application to a Different Backbone.

We employed DeepLab-v2 [2] as the segmentation backbone, and modified our condition strainer structure by substituting its linear layers with 1×1 convolution layers for this ConvNet architecture. As shown in Table a4, FREST significantly outperformed both the source model and CMA in this setting as well, suggesting that it is generic enough.

Table a4: Results of application to a different backbone. The results are reported in mIoU on the ACDC validation set.

Source model	CMA	FREST
37.6	46.6	**48.4**

G Impact of the Length of the Positive Queue.

We conducted an ablation study to investigate the impact of the length of the positive queue. As shown in Table a5, FREST is insensitive to the length.

Table a5: Effect of the length of the positive queue. The results are reported in mIoU on the ACDC validation set.

Length	50K	55K	60K	65K (Ours)	70K	75K	80K
	68.3	68.0	68.4	68.6	68.1	68.2	67.9

H Additional Explanation for Fig. 7.

This section presents an additional explanation for the meaning of Fig. 7 in the main paper. It is natural for the intra-domain distance to increase since, as a model updates, the feature distribution changes from the initial distribution during training. Notably, as seen in Fig. 7 and Fig. a4, we included for clarity, d_{adv} increases significantly more than d_{normal}, while d_{inter} between normal and adverse conditions decreases. This suggests the desired feature restoration: adverse condition features shift towards normal condition features during training with FREST.

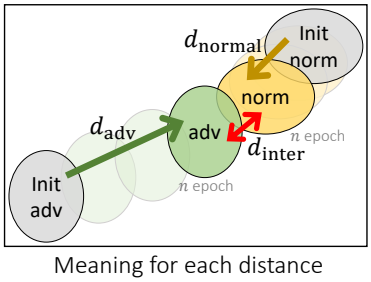

Fig. a4: Meaning of each distance in Fig. 7 of the main paper.

I Computational resource and training time.

FREST was trained using a single NVIDIA RTX 3090 GPU, taking 9 hours and 45 minutes, significantly faster than UDA methods [11, 12] which typically take about 4 days. Our method is particularly efficient during inference as it uses only the segmentation backbone without any auxiliary modules.

J Additional Qualitative Results

We present more qualitative results on ACDC [9] and Robotcar [6, 8] in this section. Fig. a5 shows the results of SegFormer, CMA, and FREST (Ours). This demonstrates that SegFormer and CMA often fail to predict detailed objects, while our method surpasses them.

Table a6: Comparison with source-free DA methods on Cityscapes→ACDC. The results are reported in mIoU (%) on the ACDC Fog test set.

Method	ACDC Fog IoU																			
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
Source model [13]	87.8	60.7	73.1	44.5	30.1	42.1	52.3	64.4	81.4	68.8	93.4	51.1	53.2	78.4	66.0	39.7	75.1	43.2	47.4	60.7
HCL [5]	88.5	63.2	79.8	45.3	30.6	44.7	53.7	65.9	81.8	69.6	95.5	52.5	55.0	79.4	68.0	40.7	74.0	40.7	46.9	61.9
URMA [10]	89.3	61.8	87.9	51.4	36.3	52.3	58.1	67.9	85.7	71.8	97.2	54.5	**62.5**	82.3	70.6	62.0	82.0	52.9	36.2	66.5
CMA [1]	**93.5**	**75.3**	**88.6**	**53.4**	33.0	52.2	58.2	67.0	86.9	71.5	**97.8**	55.6	42.0	80.4	70.0	54.8	83.3	43.0	37.4	65.5
FREST	**93.5**	74.4	87.4	51.5	**36.7**	**54.1**	**59.1**	**69.6**	**87.2**	**72.1**	97.6	**59.8**	60.1	**85.1**	**73.8**	**77.2**	**84.7**	**63.6**	**45.6**	**70.2**

Table a7: Comparison with source-free DA methods on Cityscapes→ACDC. The results are reported in mIoU (%) on the ACDC Night test set.

Method	ACDC Night IoU																			
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
Source model [13]	87.9	52.7	64.1	34.0	20.2	37.2	34.5	40.2	51.8	32.4	6.6	54.5	31.4	72.8	49.6	65.2	54.1	34.0	41.4	45.5
HCL [5]	88.2	54.3	64.4	35.3	20.7	39.1	36.8	40.4	52.0	32.1	2.8	55.2	33.7	73.5	49.2	66.5	58.1	35.4	41.7	46.3
URMA [10]	90.6	60.1	71.9	42.6	26.7	47.5	47.5	47.4	46.7	42.9	0.4	54.4	34.6	76.8	42.1	65.6	71.0	38.0	37.2	49.7
CMA [1]	**95.2**	**77.5**	**84.3**	43.9	30.9	49.4	**52.0**	**49.6**	**74.2**	**51.2**	78.4	61.4	41.2	79.2	63.6	75.1	**75.8**	34.6	47.3	61.3
FREST	94.6	75.1	82.5	**44.2**	**32.8**	**53.2**	48.5	49.2	71.1	48.5	**78.5**	**63.0**	**41.5**	**82.7**	**67.1**	**75.5**	74.6	**48.3**	**50.7**	**62.2**

Table a8: Comparison with source-free DA methods on Cityscapes→ACDC. The results are reported in mIoU (%) on the ACDC Rain test set.

Method	ACDC Rain IoU																			
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
Source model [13]	83.1	46.7	89.5	40.5	47.2	54.0	67.0	66.9	92.6	40.2	97.6	63.5	24.6	87.8	65.1	72.7	81.0	42.8	58.0	64.3
HCL [5]	84.2	50.5	90.1	42.7	48.9	57.0	68.5	69.0	93.0	40.9	97.8	65.4	26.1	88.7	68.1	74.4	80.4	43.8	58.0	65.6
URMA [10]	87.2	61.0	92.4	52.0	51.9	57.2	72.0	73.1	93.8	**46.1**	98.1	68.8	31.8	90.6	73.2	85.9	**86.9**	51.7	51.9	69.8
CMA [1]	**93.3**	**76.3**	92.8	58.1	**58.2**	61.2	70.4	71.8	93.8	45.0	97.9	67.4	36.8	89.7	72.2	**88.5**	86.4	50.5	**66.7**	72.5
FREST	92.1	73.5	**93.9**	**62.3**	57.8	**65.4**	**72.7**	**75.8**	**93.9**	42.2	**98.4**	**72.4**	**39.0**	**92.6**	**79.4**	84.5	84.9	**55.6**	65.4	**73.8**

Table a9: Comparison with source-free DA methods on Cityscapes→ACDC. The results are reported in mIoU (%) on the ACDC Snow test set.

Method	ACDC Snow IoU																			
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
Source model [13]	82.0	44.9	80.5	30.4	45.4	46.8	65.6	63.1	86.8	5.2	93.6	67.8	40.8	87.1	56.4	76.7	83.1	32.8	60.3	60.5
HCL [5]	82.9	47.4	83.2	35.4	46.8	50.1	67.8	64.9	87.7	5.3	95.6	69.8	43.9	87.6	60.1	76.9	83.2	35.3	63.4	62.5
URMA [10]	88.0	58.9	87.2	52.0	51.7	57.8	75.6	70.3	88.8	5.8	97.1	75.0	63.6	89.0	69.6	79.0	**89.8**	50.1	65.4	69.2
CMA [1]	**92.4**	**70.5**	88.3	50.4	**55.6**	56.3	74.8	71.1	**90.8**	**29.4**	96.9	77.4	63.5	90.1	63.5	79.6	89.0	45.6	**73.9**	71.5
FREST	91.3	65.0	**88.4**	**54.5**	55.3	**60.8**	**76.6**	**73.9**	89.6	10.6	**97.4**	**79.6**	**66.3**	**91.8**	**72.4**	**80.4**	88.3	**53.6**	72.8	**72.0**

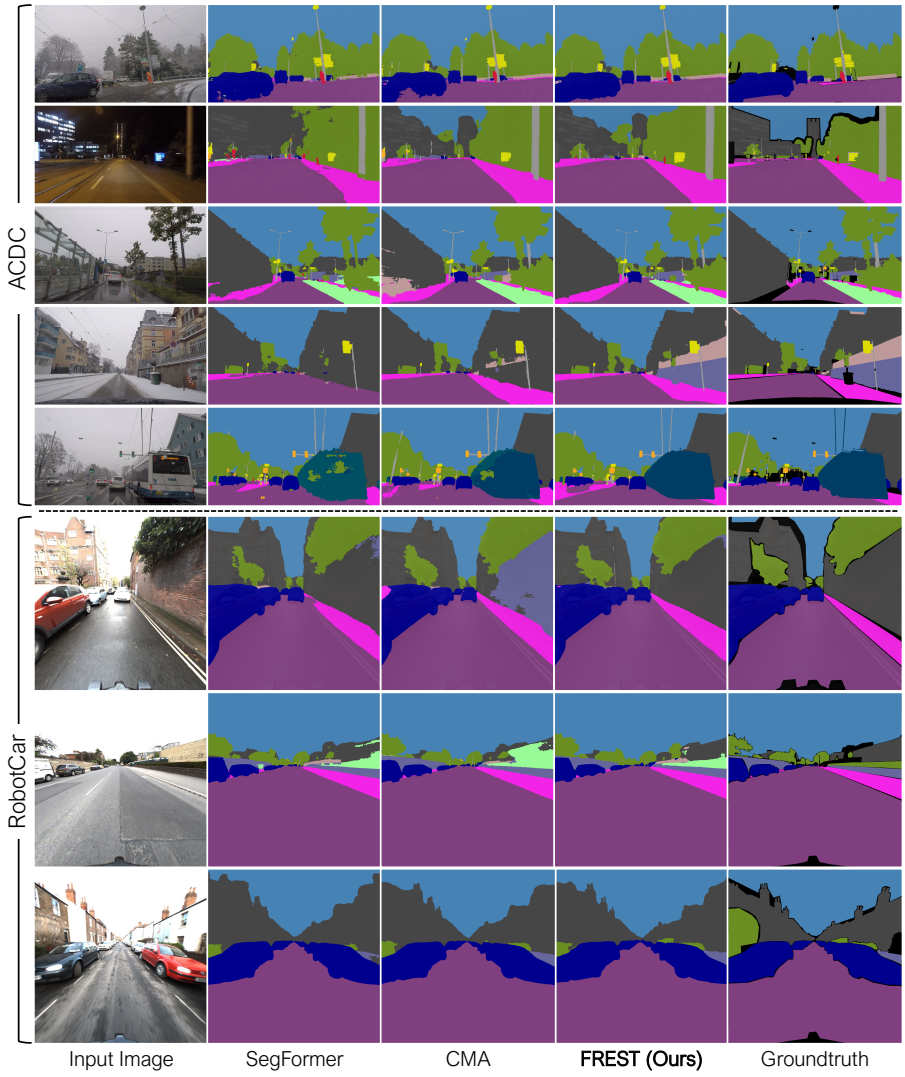

Fig. a5: Qualitative segmentation results for ACDC and RobotCar.

References

1. Brüggemann, D., Sakaridis, C., Brödermann, T., Van Gool, L.: Contrastive model adaptation for cross-condition robustness in semantic segmentation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2017)
3. Cheng, P., Hao, W., Dai, S., Liu, J., Gan, Z., Carin, L.: Club: A contrastive log-ratio upper bound of mutual information. In: Proc. International Conference on Machine Learning (ICML) (2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
5. Huang, J., Guan, D., Xiao, A., Lu, S.: Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In: Proc. Neural Information Processing Systems (NeurIPS) (2021)
6. Larsson, M., Stenborg, E., Hammarstrand, L., Pollefeys, M., Sattler, T., Kahl, F.: A cross-season correspondence dataset for robust semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
7. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research (2008)
8. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research (2017)
9. Sakaridis, C., Dai, D., Van Gool, L.: ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
10. Teja S, P., Fleuret, F.: Uncertainty reduction for model adaptation in semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
11. Tsai, Y.H., Hung, W.C., Schulter, S., Sohn, K., Yang, M.H., Chandraker, M.: Learning to adapt structured output space for semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
12. Vu, T.H., Jain, H., Bucher, M., Cord, M., Pérez, P.: Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
13. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and efficient design for semantic segmentation with transformers. In: Proc. Neural Information Processing Systems (NeurIPS) (2021)