FREST: Feature RESToration for Semantic Segmentation under Multiple Adverse Conditions

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

Abstract. Robust semantic segmentation under adverse conditions is crucial in real-world applications. To address this challenging task in practical scenarios where labeled normal condition images are not accessible in training, we propose FREST, a novel feature restoration framework for source-free domain adaptation (SFDA) of semantic segmentation to adverse conditions. FREST alternates two steps: (1) learning the condition embedding space that only separates the condition information from the features and (2) restoring features of adverse condition images on the learned condition embedding space. By alternating these two steps, FREST gradually restores features where the effect of adverse conditions is reduced. FREST achieved a state of the art on two public benchmarks (*i.e.*, ACDC and RobotCar) for SFDA to adverse conditions. Moreover, it shows superior generalization ability on unseen datasets.

Keywords: semantic segmentation, feature restoration, robustness, source-free domain adaptation

1 Introduction

The advent of deep neural networks has brought significant advancement of semantic segmentation [4, 36, 38, 44, 47]. Although most existing models for semantic segmentation demonstrate outstanding performance under normal conditions, they often fail under real-world adverse conditions like fog, rain, snow, and nighttime that significantly degrade the quality of input images [5, 7, 20, 21, 31–33, 35, 46]. This lack of robustness limits the applicability of semantic segmentation, especially to high-stakes tasks like autonomous driving.

An obstacle in enhancing the robustness of semantic segmentation models is the difficulty of collecting labeled data for every possible adverse condition. This issue has steered the computer vision community towards unsupervised domain adaptation (UDA) [2, 10, 12–14, 18, 27, 39, 40, 42, 48, 51]. UDA is the task of training a model using labeled data from a *source* domain (*i.e.* clear weather) and unlabeled data from a *target* domain (*i.e.* adverse weather conditions) while bridging the gap between these domains. This approach mitigates the need for labeling target domain data while improving performance in that domain.

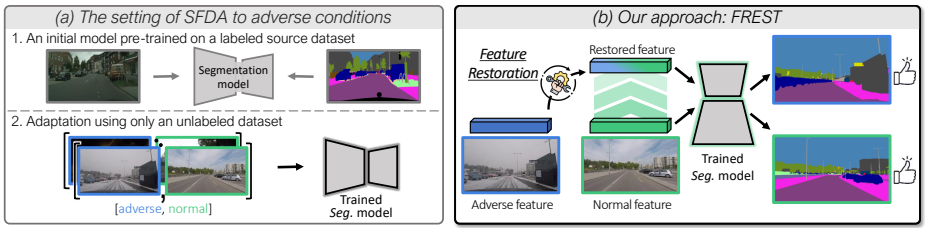

Fig. 1: (a) The setting of SFDA to adverse conditions. A segmentation model, initially pre-trained on a labeled source dataset, is adapted to adverse conditions using pairs of unlabeled adverse and normal images. (b) Following the SFDA setting, FREST restores features of adverse condition images to simulate the normal condition.

More recent research has explored source-free domain adaptation (SFDA), a more practical form of UDA where access to the source domain data is not allowed due to privacy leaks or the prohibitively large scale of the data. [1, 9, 24, 37, 43, 49]. In SFDA, a model is first pre-trained with labeled data from the source domain and then fine-tuned using only unlabeled data from the target domain afterwards. In particular, SFDA to adverse conditions has been studied on a specialized setup [1], in which target domain images are taken under various adverse conditions and each target image is paired with a reference image taken in a similar geolocation but under the normal condition, *i.e.*, clear weather. This setting differs from the conventional SFDA as the reference images from the normal condition are available, but following the previous work [1], we refer to this setting as SFDA to adverse conditions. In this setting, a pair of target and reference images are matched by global navigation satellite system (GNSS), and thus are only roughly aligned due to the variations between them in terms of camera pose and shooting time. Also, both target and reference images are unlabeled, and the adverse condition type of each target image is unknown. This setup is illustrated in Fig. 1(a).

The prior work for SFDA to adverse conditions enhances the robustness of semantic segmentation models by learning condition-invariant features [1]. To this end, it encourages features extracted from each pair of adverse (target) and normal (reference) images to be close. However, regarding that normal images could resemble the source domain, updating features of normal images to be close to those of adverse images causes the model pre-trained on the source domain to forget the rich knowledge of the domain, resulting in poor representations for both normal and adverse images in the end. Also, the feature matching relies heavily on the assumption that the alignment between a pair of adverse and normal images is sufficiently accurate, which typically does not hold, unfortunately.

In this work, we introduce a novel framework for *Feature RESToration for multiple adverse conditions*, called **FREST**, which is illustrated in Fig. 1(b). FREST overcomes the aforementioned limitations by restoring features of adverse condition images so that they simulate the normal condition of reference images in feature spaces. This notion of feature restoration is embodied by extracting and leveraging *condition-specific information*, which ideally depends

only on the condition of image and is not affected by its semantic content. The use of condition-specific information enables our method to restore features while considering only the condition of input. Hence, FREST mitigates the catastrophic forgetting of the source domain knowledge through the feature restoration, and is less affected by the content mismatch between a pair of adverse and normal images by utilizing condition-specific information.

FREST operates in two steps as follows. First, it learns a new embedding space that represents only condition-specific attributes of images. To be specific, in this embedding space, images taken under similar conditions are closely aligned while those under different conditions are separate. We consider the embedding vector of an image in this space as its condition-specific information. In the second stage, FREST learns restored features for adverse images, tailored for semantic segmentation, by optimizing the model for both segmentation and feature restoration. In detail, the segmentation objective leverages pseudo segmentation labels as supervision, while the objective for feature restoration enforces condition-specific information of adverse images approximates that of their corresponding normal images so that *the model learns to represent images of any conditions as if they are taken under the normal condition.* FREST alternates between these two stages so that condition-specific information is adapted to reflect the update of the segmentation network in the first stage, which improves the restored features of the model in the second stage consequently. It is empirically demonstrated in Sec. 4.3 that FREST effectively learns restored features of adverse images.

FREST was evaluated on the standard benchmarks for SFDA to adverse conditions based on the Cityscapes [6], ACDC [34], and RobotCar [19, 26] datasets, *i.e.*, Cityscapes → ACDC and Cityscapes → RobotCar settings. FREST achieved a new state of the art in both two settings. Moreover, its superiority in terms of robustness and generalization capability was demonstrated on unseen datasets by applying our model for Cityscapes→ACDC to the ACG [1] and Cityscapes-lindau40 [6] datasets.

2 Related Work

Robustness. Robust recognition has been actively studied due to its relation with crucial safety-critical applications [5, 7, 20, 21, 31–33, 35, 46]. In this context, various UDA methods [2, 12–14] have been proposed to improve robustness across multiple adverse conditions. In particular, Brüggemann *et al.* [2] suggest a method of spatial alignment between target and reference images and adaptive label correction guided by the warping results. Brüggemann *et al.* [1] introduce a method for SFDA under multiple adverse conditions through contrastive learning for condition-invariant learning. In contrast to the previous work, FREST restores features from multiple adverse conditions to those of the normal condition, effectively learning robust features for adverse conditions.

Unsupervised Domain Adaptation. UDA has been widely studied for semantic segmentation with the introduction of synthetic datasets [29, 30], which

provide automatically generated pixel-level labels. UDA allows the use of both labeled synthetic and unlabeled real data for training. Existing UDA methods are mainly categorized into distribution alignment [10, 39, 40, 42, 48] and self-training [18, 22, 27, 50, 51]. Revealing the potential of transformers [8, 41, 45] for semantic segmentation, recent studies have conducted transformer-based approaches [2,12–14]. Despite significant improvements, UDA for semantic segmentation faces the practical limitation of requiring access to labeled source data for adaptation. To address this, source-free domain adaptation is introduced, using only unlabeled target data to adapt a source-trained model.

Source-free Domain Adaptation. SFDA is introduced to adapt a source-trained model to the target domain without accessing source data. The early approaches suggest test-time objectives at the output space of an unlabeled target domain dataset by entropy minimization [37, 43], data-free knowledge distillation [24], and contrastive learning [15]. Recently, Guo *et al.* [9] introduce a plug-and-play method via a noise transition matrix for loss correction on noisy pseudo-labeled target data. Zhao *et al.* [49] suggest enhancing the stability and adaptability of self-training through a dynamic teacher update mechanism and a resampling strategy based on training consistency. Motivated by this line of research, Bruggemann *et al.* [1] propose a practical setup, where target images are taken under multiple adverse conditions and associated with reference images of the normal condition. Following this setup, FREST learns to restore adverse features using normal features for robust semantic segmentation.

3 Configuration of Target Domain Data

Following the problem setting of the previous work [1], we assume that target domain data comprise images taken under various adverse conditions, and that each target image is paired with a reference image captured under the normal condition. Both target and reference images are unlabeled, and it is unknown under what condition each target image was taken. Moreover, a target adverse condition image I_{adv} and its associated reference normal condition image I_{norm} are matched by GNSS so that they are taken in similar geolocations. Note however that a pair of GNSS-matched images will only be roughly aligned due to the variations between them in terms of camera pose and shooting time. To mitigate this issue, following prior work [1,2], we warp I_{norm} onto I_{adv} using the UAWarpC dense matching network [2] pre-trained on the MegaDepth dataset [23]. More details for the warping process can be found in [2]. While the warping alleviates the misalignment issue to some extent, it still leaves nontrivial discrepancy in content due to imperfect warping and dynamic objects.

4 Proposed Method

The overall architecture and training strategy of our model are outlined in Fig. 2; we suppose that the segmentation network is pre-trained with a labeled source

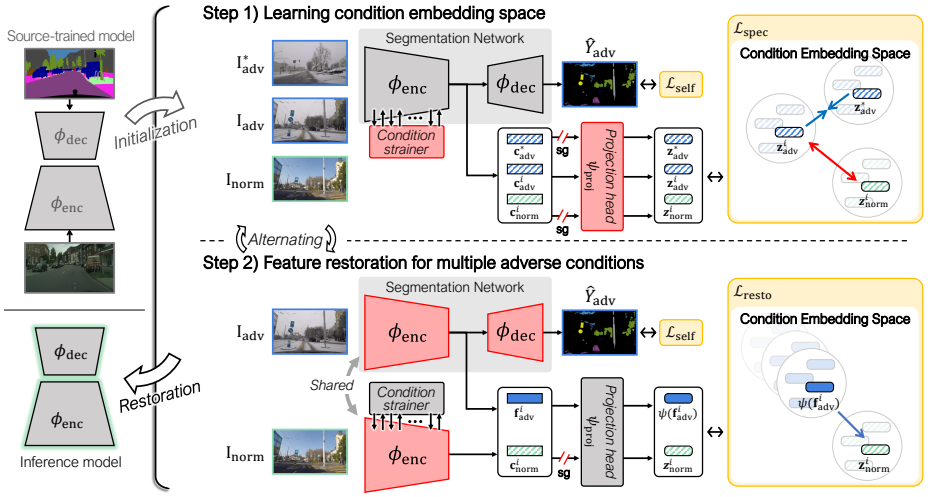

Fig. 2: The overall architecture and training strategy. The segmentation network is pre-trained using a labeled source dataset. For each iteration, the condition strainer and segmentation network are trained alternatingly. The frozen modules are shown in gray, the trainable modules are highlighted in red, and "sg" denotes the stop gradient. (*Step 1*) The condition strainer and projection head are trained to learn the condition embedding space. (*Step 2*) The segmentation network is trained to restore features from adverse to normal conditions on the condition embedding space. For evaluation, only the encoder ϕ_{enc} and decoder ϕ_{dec} of the segmentation network are utilized.

domain dataset following the standard protocol [1]. FREST considers adverse conditions as detrimental, and aims to remove their effects in features of a target adverse image I_{adv} by *feature restoration*, which is the process of learning features of I_{adv} that resemble those of the corresponding normal image I_{norm}, not in the content, but only in the effect of the condition.

However, using features of I_{norm} and I_{adv} directly for feature restoration is less than ideal, as it may lead to a distortion of the semantic content, due to the misalignment between I_{norm} and I_{adv} discussed in Sec. 3. To address this issue, FREST extracts condition-specific information from each feature and utilizes it to guide the feature restoration process. To realize this idea, FREST alternates the following two steps: (1) learning a condition embedding space, and (2) restoring features of I_{adv} so that the features approximate those of I_{norm} on the condition embedding space.

In the first step, FREST learns a condition embedding space to capture the condition-specific information of input. In this space, images are separated and clustered based on whether they were taken under adverse conditions or the normal condition. We consider the embedding vector of an image in this space as its condition-specific information, which is less affected by its semantic content. For learning such an embedding space, we propose to attach a module named *condition strainer*, denoted by ψ_{strainer}, to the frozen segmentation encoder ϕ_{enc}

so that the module retains only condition-specific information that is distinct from the source domain information preserved by the frozen encoder.

The design of the condition strainer is inspired by parameter-efficient fine-tuning [3, 11, 28], which adds a small number of parameters for capturing task-specific information. This structure enables the condition strainer to capture condition-specific information effectively and efficiently. The detailed architecture of the encoder with the condition strainer is presented in Fig. 3. The condition strainer is trained to extract condition-specific information from each layer of the encoder while being separated from the encoder; it is separate from the encoder so that the encoder is not affected by its condition-specific information. We call features extracted by ϕ_{enc} incorporating ψ_{strainer} condition-infused features. Specifically, the condition-infused features $\mathbf{c}^l$ computed at the l^{th} layer of the encoder is given by $\mathbf{c}^l = \phi^l_{\text{enc}}(\mathbf{c}^{l-1}) + \psi^l_{\text{strainer}}(\mathbf{c}^{l-1})$. We indicate $\mathbf{c}$ as the condition-infused feature produced by the last layer of the encoder. Finally, $\mathbf{c}$ is projected on the condition embedding space through a projection head ψ_{proj}. Also, for the sake of brevity, we denote encoder features computed only by ϕ_{enc}, disregarding the strainer ψ_{strainer}, by $\mathbf{f}^l = \phi^l_{\text{enc}}(\mathbf{f}^{l-1})$ where $\mathbf{f}$ is the encoder feature obtained from the last layer. The encoder features $\mathbf{f}^l$ are targeted features for the feature restoration in the second step.

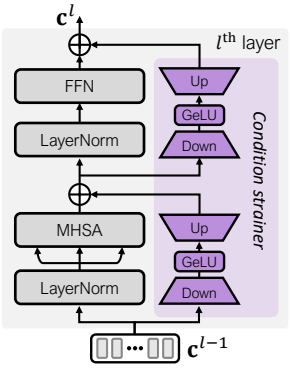

Fig. 3: Detail of the encoder with the condition strainer. Condition strainers are connected to the original feed-forward layer (FFN) and multi-head self-attention layer (MHSA) through the residual connections.

In the second step, we train the segmentation network while conducting the feature restoration with the frozen condition strainer and projection head. FREST restores features of I_{adv} from ϕ_{enc}, denoted by $\mathbf{f}_{\text{adv}}$, to resemble condition-infused features of I_{norm}, denoted by $\mathbf{c}_{\text{norm}}$, where $\mathbf{f}_{\text{adv}}$ and $\mathbf{c}_{\text{norm}}$ are computed from the last layer of the encoder. Specifically, $\mathbf{f}_{\text{adv}}$ is encouraged to approximate $\mathbf{c}_{\text{norm}}$ by a regression objective on the condition embedding space for considering only the condition information during feature restoration.

By alternating the two steps aforementioned, the segmentation network progressively learns restored features for multiple adverse conditions, and the condition strainer is adapted to reflect the update of the segmentation network and facilitates the next feature restoration consequently. The remainder of this section elaborates on the two steps of FREST.

4.1 Learning Condition Embedding Space

In the first step, FREST learns the condition embedding space that only represents condition-specific information of input. To this end, we train the condition

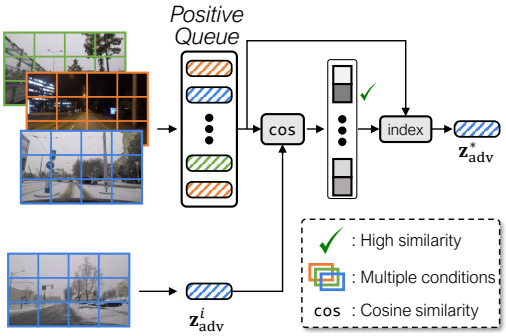

Fig. 4: Detail of positive embedding sampling strategy in the condition-specific learning.

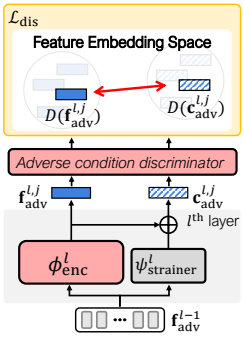

Fig. 5: Detail of the adverse condition discriminator.

strainer and projection head with the frozen segmentation network. Loss functions used in this step are described below.

Condition-specific Learning Loss. The goal of condition-specific learning is to learn the condition embedding space by extracting only condition-specific information from the condition-infused features that contain both content and conditions. We implement this objective by contrastive learning, grouping features from the same condition closely together and distancing those from different conditions. For contrastive learning, the anchor and positive are sampled with different semantics under the same condition, while the anchor and negative are chosen to be semantically similar but under different conditions.

To sample a pair of anchor and negative, we warp normal features to corresponding adverse features as described in Sec. 3. Specifically, based on the confidence scores computed by the warping module [2], we select patch embeddings surpassing a warping confidence threshold of 0.2 for computing the anchor and negative. Given a pair of I_{norm} and I_{adv}, we first sample $\mathbf{c}^i_{\text{norm}}$ and $\mathbf{c}^i_{\text{adv}}$ as condition-infused features for i^{th} patch embedding, for $i \in \mathcal{W}$ where $\mathcal{W} = \{i \mid \text{conf}(i) \geq 0.2\}$ with $\text{conf}(i)$ indicating the warping confidence score. Then, the anchor and negative samples are computed by projecting $\mathbf{c}^i_{\text{adv}}$ and $\mathbf{c}^i_{\text{norm}}$ on the condition embedding space; $\mathbf{z}^i_{\text{adv}} = \psi_{\text{proj}}(\mathbf{c}^i_{\text{adv}})$ and $\mathbf{z}^i_{\text{norm}} = \psi_{\text{proj}}(\mathbf{c}^i_{\text{norm}})$, respectively. To obtain sufficient positive candidates, we stack condition embeddings of adverse images in a batch into a positive queue of length Q for each iteration. As shown in Fig. 4, we choose the representative positive $\mathbf{z}^*_{\text{adv}}$ that is the most similar to the anchor $\mathbf{z}^i_{\text{adv}}$ in the queue, assuming it has the same condition with the anchor; we empirically verify that $\mathbf{z}^*_{\text{adv}}$ is an effective reference point for pushing a negative sample $\mathbf{z}^i_{\text{norm}}$ in Sec. 5.4. For each patch embedding, the condition-specific loss aims to attract the anchor $\mathbf{z}^i_{\text{adv}}$ towards the representative positive $\mathbf{z}^*_{\text{adv}}$ and repel it from the negative $\mathbf{z}^i_{\text{norm}}$.

For $\{\mathbf{z}^i_{\text{adv}}, \mathbf{z}^*_{\text{adv}}, \mathbf{z}^i_{\text{norm}}\}$, the condition-specific loss for patch i is given by:

$$\mathcal{L}_{\text{spec},i} = -\log \frac{\exp({\mathbf{z}^i_{\text{adv}}}^\top \mathbf{z}^*_{\text{adv}}/\tau)}{\exp({\mathbf{z}^i_{\text{adv}}}^\top \mathbf{z}^*_{\text{adv}}/\tau)) + \exp({\mathbf{z}^i_{\text{adv}}}^\top \mathbf{z}^i_{\text{norm}}/\tau)}, \tag{1}$$

where τ serves as a temperature that scales the sensitivity. The condition-specific loss is the average of the losses in (1) applied to individual patches with warping confidence scores larger than the threshold, and formulated as

$$\mathcal{L}_{\text{spec}} = \frac{1}{|\mathcal{W}|} \sum_{i \in \mathcal{W}} \mathcal{L}_{\text{spec},i}. \tag{2}$$

Total Loss for Step 1. Following the previous work [1], we employ a self-training loss $\mathcal{L}_{\text{self}}$, *i.e.*, the pixel-wise cross-entropy loss with pseudo labels generated by class-balanced self-training (CBST) [50]. It is necessary for learning semantic information due to the absence of segmentation labels during training. To preserve the semantic information of condition-infused features, we apply the self-training loss to the predictions computed from condition-infused features. Then, the total training loss of Step 1 is represented as $\mathcal{L}_{\text{step1}} = \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \mathcal{L}_{\text{self}}$.

4.2 Learning Semantic Segmentation with Feature Restoration

In the second step, FREST learns to restore features so that $\mathbf{f}_{\text{adv}}$ from adverse conditions approximates $\mathbf{c}_{\text{norm}}$ from the normal condition in the condition embedding space. To this end, we train the segmentation network with the frozen condition strainer ψ_{strainer}. In addition, for further alleviating adverse effects from $\mathbf{f}^l_{\text{adv}}$, FREST ensures that $\mathbf{f}^l_{\text{adv}}$ is discriminated from the adverse condition-infused feature $\mathbf{c}^l_{\text{adv}}$ for the l^{th} layer of the encoder. The loss functions in this process are described below.

Feature Restoration Loss. Given paired I_{adv} and I_{norm}, we compute $\mathbf{f}_{\text{adv}}$ using ϕ_{enc} only and a condition-infused feature $\mathbf{c}_{\text{norm}}$. Note that we prevent gradient updates from $\mathbf{c}_{\text{norm}}$ to ensure that the adverse condition one-sidedly follows the normal condition as $\mathbf{c}_{\text{norm}}$ is the target for restoration of $\mathbf{f}_{\text{adv}}$. As detailed in Sec. 4.1, we select the pair of adverse and normal patch embeddings, which surpass a warping confidence threshold, denoted as $\mathbf{f}^i_{\text{adv}}$ and $\mathbf{c}^i_{\text{norm}}$ for the i^{th} patch embedding. To guide the condition of our segmentation feature $\mathbf{f}^i_{\text{adv}}$ to resemble the normal condition while only considering condition information, we project our feature $\mathbf{f}^i_{\text{adv}}$ and normal condition-infused feature $\mathbf{c}^i_{\text{norm}}$ on the condition embedding space, *i.e.*, $\psi_{\text{proj}}(\mathbf{f}^i_{\text{adv}})$ and $\mathbf{z}^i_{\text{norm}} = \psi_{\text{proj}}(\mathbf{c}^i_{\text{norm}})$, respectively. Then, we approximate $\psi_{\text{proj}}(\mathbf{f}^i_{\text{adv}})$ to $\mathbf{z}^i_{\text{norm}}$ using an $\ell 1$ regression loss as follows:

$$\mathcal{L}_{\text{resto}} = \frac{1}{|\mathcal{W}|} \sum_{i \in \mathcal{W}} |\psi_{\text{proj}}(\mathbf{f}^i_{\text{adv}}) - \mathbf{z}^i_{\text{norm}}|, \tag{3}$$

where $\mathcal{W} = \{i \mid \text{conf}(i) \geq 0.2\}$ with $\text{conf}(i)$ denoting the warping confidence.

Adverse Condition Discriminating Loss. To further facilitate feature restoration, we present another loss that pushes the encoder feature and condition-infused feature of an adverse condition image apart. To this end, we introduce an MLP-based adverse condition discriminator denoted as D. It discriminates

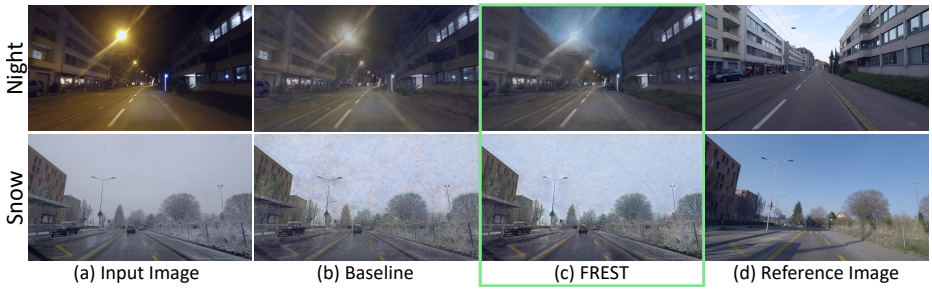

Fig. 6: Image reconstruction results from segmentation features. (a) Input target image. Image reconstructed by the (b) Baseline [45] and (c) FREST. (d) Reference image.

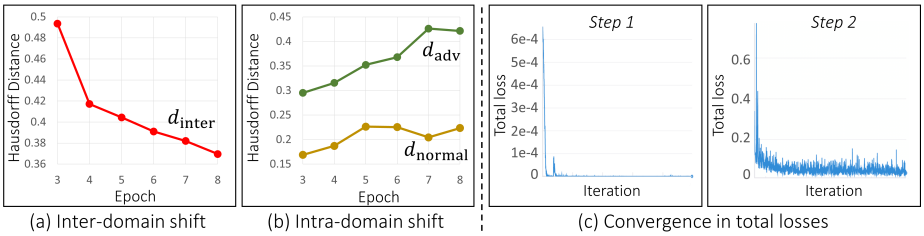

Fig. 7: Empirical analysis on the impact of feature restoration during training FREST. (a) Inter-domain shift between adverse and normal conditions. (b) Intra-domain shift within each condition. (c) Convergence in total losses for both Step 1 and Step 2.

the encoder feature $\mathbf{f}_{\text{adv}}^{l,j}$ and condition-infused feature $\mathbf{c}_{\text{adv}}^{l,j}$ for each l^{th} layer and j^{th} patch embedding for $j \in \mathcal{A}$ where $\mathcal{A}$ is a set of indices of all patch embeddings. Then, $\mathbf{f}_{\text{adv}}^{l,j}$ and $\mathbf{c}_{\text{adv}}^{l,j}$ are vectorized and forwarded to the discriminator, which is trained with the cross-entropy loss, denoted as $\mathcal{L}_{\text{dis}}$, to classify them into two classes: encoder feature and condition-infused feature. The detailed architecture of D is shown in Fig. 5 and the loss is given by

$$\mathcal{L}_{\text{dis}} = -\frac{1}{|\mathcal{A}|}\sum_{j\in\mathcal{A}}\sum_{l=1}^{L}\left\{\lambda \log(D(\mathbf{f}_{\text{adv}}^{l,j})) + (1-\lambda)\log(1-D(\mathbf{c}_{\text{adv}}^{l,j}))\right\}, \tag{4}$$

where L denotes the number of layers of the encoder, and $\lambda = 0$ if the input is the adverse condition-infused feature $\mathbf{c}_{\text{adv}}^{l}$ and $\lambda = 1$ otherwise.

Total Loss for Step 2. Besides the proposed losses, we utilize two conventionally used loss functions: one for self-training $\mathcal{L}_{\text{self}}$ and the other for entropy minimization $\mathcal{L}_{\text{ent}}$, following the previous work [1]. The segmentation network is trained by minimizing $\mathcal{L}_{\text{step2}} = \mathcal{L}_{\text{resto}} + \lambda_{\text{dis}}\mathcal{L}_{\text{dis}} + \mathcal{L}_{\text{self}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}$.

4.3 Empirical Justification

To investigate the effect of FREST, we first conduct a qualitative analysis by reconstructing images from the restored features learned by FREST. Please note

that FREST learns feature restoration for robustly recognizing adverse condition images as if they were in normal condition, does not image reconstruction during both training and testing. For image reconstruction, we adopt SegFormer [45] as a reconstruction model with upsampling layers. The decoder of the reconstruction model is trained for reconstruction on normal images while the encoder is pretrained on Cityscapes [6] and then frozen. The encoder is then replaced with that of FREST (ϕ_{enc}). We also conduct the reconstruction using a baseline [45] in the same manner. Fig. 6 shows the favorable impact of FREST: a night sky turns blue, and a snowy tree appears green.

The impact of feature restoration is also demonstrated by the quantitative analysis in Fig. 7. We measure the inter-domain shift in Fig. 7(a), the distance between feature distributions of adverse and normal domains (d_{inter}), and the intra-domain shift in Fig. 7(b), the distance between feature distributions at before training and after n epochs of training with FREST, for each domain (d_{adv} and d_{normal}). We adopt the Hausdorff distance [16] using cosine distance to measure all such shifts. Our analysis reveals that FREST restores adverse features as desired: the adverse feature distribution gradually approaches to the normal feature distribution during training, while the normal feature distribution changes little.

As shown Fig. 7(c), we investigate the convergence in total losses for both Step 1 and Step 2 during training FREST. The results demonstrate that FREST successfully converges during training while the two steps are performed in each iteration alternatingly. Since $\mathcal{L}_{\text{spec}}$ and $\mathcal{L}_{\text{resto}}$ have opposite objectives conducted in condition embedding space, it might seem optimizing two losses would hinder the convergence of FREST. However, this issue does not occur since we separate the trainable parameters for each step: in Step 1, we train the condition strainer and projection head, while in Step 2, we train the segmentation network. This training strategy ensures that each step is optimized independently, without conflict, enabling FREST to converge as intended.

5 Experiments

5.1 Experimental Setting

Datasets. For our experiments, we utilize Cityscapes [6] as the source domain, and we use ACDC [34] and RobotCar Correspondence [19, 26] as the target domains, respectively. We also use the ACG Benchmark [1] for evaluating the generalization capability of our model on diverse adverse conditions. Lastly, we evaluate our method in the normal condition on Cityscapes-lindau40, a dataset commonly used in robust visual recognition research [7, 21].

Implementation Details. We adopt SegFormer [45] architecture as our segmentation network, which is pre-trained on the Cityscapes dataset [6]. The segmentation network is trained by an AdamW [25] optimizer with weight decay 1e−2. The initial learning rate is set to 1e−5 for the encoder and decoder of the segmentation network and 5e−4 for the condition strainer. In addition, we

Table 1: Comparison with existing methods on Cityscapes → ACDC. The results are reported in mIoU (%) on the ACDC test set.

Method	ACDC IoU																			
	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
Source model [45]	85.7	51.0	76.6	36.4	37.1	45.2	55.7	57.5	77.7	52.0	84.1	60.3	34.8	82.9	61.6	65.4	73.4	37.9	52.5	59.4
HCL [15]	86.4	53.5	78.5	38.8	38.1	48.0	57.8	58.9	78.1	52.4	85.1	61.7	37.1	83.7	64.1	66.6	74.5	39.1	53.3	60.8
URMA [37]	89.2	60.4	84.3	48.7	42.5	53.8	65.4	63.8	76.3	57.3	85.9	63.4	43.9	85.8	68.8	73.2	82.8	46.3	48.4	65.3
URMA + SimT [9]	90.0	65.7	80.6	46.0	41.7	56.3	65.2	62.7	75.9	55.6	84.4	66.4	46.6	85.4	68.4	72.3	80.0	46.8	58.0	65.7
CMA [1]	**94.0**	**75.2**	**88.6**	50.5	45.5	54.9	65.7	64.2	**87.1**	**61.3**	95.2	67.0	45.2	86.2	68.6	76.6	**83.9**	43.3	60.5	69.1
FREST (Ours)	93.3	72.2	88.3	**52.4**	**46.6**	**58.6**	**66.2**	**66.1**	86.1	58.6	**95.3**	**69.9**	**49.2**	**89.1**	**75.1**	**79.4**	83.0	**52.9**	**61.4**	**70.7**

Table 2: Comparison with previous methods on City → RobotCar.

Method	mIoU
Source model [45]	50.0
HCL [15]	50.1
URMA [37]	51.6
URMA + SimT [9]	52.4
CMA [1]	54.3
FREST (Ours)	**58.8**

Table 3: Comparison with UDA methods on City → ACDC. ($\mathcal{SF}$: source-free method).

Method	$\mathcal{SF}$	mIoU
Source model [45]		59.4
Refign [2]		65.5
HRDA [13]		68.0
HRDA + MIC [14]		70.4
CMA [1]	✓	69.1
FREST (Ours)	✓	**70.7**

Table 4: Generalization performance of models adapted from Cityscapes to ACDC on ACG and Cityscapes-lindau40 (C-Lindau).

Method	ACG					C-Lindau
	fog	night	rain	snow	**all**	normal
SegFormer [45]	54.0	27.9	47.5	41.2	40.1	**72.7**
HCL [15]	54.2	28.3	48.2	42.4	40.8	-
URMA [37]	54.1	31.0	51.9	45.5	44.4	-
CMA [1]	59.7	**40.0**	59.6	**52.2**	51.3	71.8
FREST (Ours)	**61.4**	39.9	**61.0**	51.9	**52.6**	72.5

employ a linear learning rate decay with a linear warm-up for the first 1,500 iterations. Finally, the hyper-parameters are set to λ_{spec}, λ_{ent}, λ_{dis}, and τ as 1e−2, 1e−2, 5e−5, and 7e−1, respectively. More details are given in the supplement.

5.2 Quantitative Results

We first compare FREST with existing methods for SFDA under multiple adverse conditions on Cityscapes → ACDC and Cityscapes → RobotCar benchmarks. We extend our comparison to the previous UDA methods on Cityscapes → ACDC benchmark. Lastly, we evaluate the generalization capability of FREST on ACG and Cityscapes-lindau40 compared with existing SFDA methods.

Comparison with SFDA Methods. As summarized in Table 1, FREST achieves state-of-the-art performance with a notable improvement of 1.6% in mIoU over the most recent work [1] on Cityscapes → ACDC benchmark, especially improving on fine-grained objects (*e.g.*, car, truck, bus) by large margins. As shown in Table 2, FREST substantially outperforms all previous SFDA models on Cityscapes → RobotCar benchmark. Considering that the RobotCar dataset includes a wider range of conditions (*i.e.*, dawn, dusk, night, night-rain, overcast, rain, snow, and sun) compared to the ACDC dataset, which has only four conditions (*i.e.*, fog, rain, snow, and night), these results demonstrate that our model becomes increasingly effective in improving robustness as the diversity of adverse conditions increases.

Table 5: Loss analysis on (a) Step 1 and (b) Step 2.

$\mathcal{L}_{\text{self}}$	$\mathcal{L}_{\text{spec}}$	mIoU
		64.3
✓		64.8
✓	✓	**68.6**

(a)

$\mathcal{L}_{\text{resto}}$	$\mathcal{L}_{\text{dis}}$	mIoU
		62.7
✓		67.2
✓	✓	**68.6**

(b)

Table 6: Analysis of the structure and training strategy in FREST.

Trainable Module		Training Strategy		mIoU
Strainer	Seg.	Self-training	FREST	
	✓	✓		62.7
✓		✓		63.1
✓	✓	✓		63.2
✓	✓		✓	**68.6**

Comparison with UDA Methods. As summarized in Table 3, FREST outperforms the existing UDA methods on Cityscapes → ACDC. Note that UDA methods utilize both a labeled source dataset and an unlabeled target dataset during adaptation, while SFDA methods including FREST use only the unlabeled target dataset. The results show that our framework surpasses the UDA methods, even without access to a labeled source domain during the adaptation.
Generalization Capability. Following [1], we evaluate the generalization capabilities of FREST and the competitors, adapted from Cityscapes to ACDC, on the ACG benchmark containing multiple adverse conditions (*i.e.*, fog, night, rain, and snow). As shown in Table 4, FREST outperforms the previous methods for all ACG samples, which shows the robust generalizability of FREST in adverse conditions. Furthermore, we extend our evaluation to the Cityscapes-lindau40 dataset to investigate the generalizability under normal conditions. FREST surpasses CMA [1] and performs on par with SegFormer [45] which is trained on Cityspaces as a labeled source dataset. The results indicate that FREST effectively generalizes on normal conditions as well as adverse conditions by its feature restoration, converting features from adverse to normal conditions.

5.3 Ablation Study

We first investigate the effect of each loss by the ablation study in Table 5. The results show that all the losses contribute to the performance in each step. In particular, the condition-specific loss in Step 1 and the feature restoration loss in Step 2 significantly fulfill the primary objectives of FREST, contributing significantly to improving the robustness of semantic segmentation.

In Table 6, we analyze the effect of the condition strainer, which draws its structural inspiration from the adapter structure [11] for parameter-efficient learning. To this end, we conduct a comparative study of FREST against the naive fine-tuning strategy using the condition strainer (*i.e.*, adapter). As shown in the second row of the table, the conventional fine-tuning scheme using the adapter improves the performance marginally, demonstrating the ineffectiveness of a naive application of the adapter (+0.4%p). In addition, the third row shows that the full fine-tuning of all parameters of both the segmentation network and the adapter results in a slight improvement (+0.5%p). It suggests that our performance improvement does not solely stem from an increase in the number of parameters in the condition strainer. Consequently, the proposed training

Table 7: The number of parameters for additional modules.

	Param.
SegFormer [45]	81.4M (100%)
Condition Strainer	2.1M (2.6%)
Projection Layer	1.2M (1.5%)

Table 8: Impact of restored features $\mathbf{f}_{\text{adv}}$.

Features for Inference	mIoU
Condition-infused feature $\mathbf{c}_{\text{adv}}$	59.0
Restored feature $\mathbf{f}_{\text{adv}}$	**68.6**

Table 9: Performance according to positive embedding selection strategies and loss functions. *Cls.* and *Contra.* denote classification and contrastive loss.

Pos. Emb. Selection	*Cls.*	*Contra.*	mIoU
All	✓		62.9
		✓	63.4
(1) `RANDOM`		✓	62.4
(2) `LOWEST`		✓	56.6
(3) `HIGHEST` (Ours)		✓	**68.6**

scheme, coupled with the condition strainer, contributes to the performance significantly (+5.9%p).

5.4 Analysis on FREST

Parameter Efficiency. As shown in Table 7, the total parameters of the baseline, SegFormer [45], are 81.4M (100%), while the strainers occupy only 2.1M (2.6%), and the projection head has 1.2M (1.5%), which require only a small number of parameters. Also, it's important to note that no additional parameters are required during inference, as only the encoder and decoder are utilized.

Impact of Restored Feature $\mathbf{f}_{\text{adv}}$. We evaluate the efficacy of the restored adverse features $\mathbf{f}_{\text{adv}}$ learned by FREST. To this end, we compare the inference result using the restored features $\mathbf{f}_{\text{adv}}$, as employed in our framework, with that using the condition-infused features $\mathbf{c}_{\text{adv}}$. Table 8 demonstrates that utilizing restored features $\mathbf{f}_{\text{adv}}$ robustly performs as intended, while employing condition-infused features $\mathbf{c}_{\text{adv}}$ leads to inferior performance due to their inclusion of detrimental characteristics of adverse conditions.

Analysis on Condition-specific Learning. To verify our design choice for the objective function learning condition information, we investigate variants of its positive embedding selection strategies and loss functions as summarized in Table 9. In detail, we first evaluate variants using all condition embeddings $\{\mathbf{z}^i_{\text{adv}}\}_{i=0}^{N}$ where N is the number of condition embeddings in a positive queue varying classification and contrastive loss as loss functions. We adopt supervised contrastive learning [17] for multiple positive samples. The results show that the contrastive loss learns better condition embedding space than the classification loss. To address the diverse distribution of positive embeddings from adverse conditions in contrastive learning, we choose the representative positive embedding following variants: (1) `RANDOM` that select an arbitrary embedding in the positive queue, (2) `LOWEST` that pick the lowest similar embedding with the anchor embedding, and (3) `HIGHEST` that pick the highest similar embedding. The results demonstrate that using the highest similar embedding as a positive sample for contrastive learning is the most effective strategy for learning condition embedding space. We suspect the reason is that the most similar positive embedding

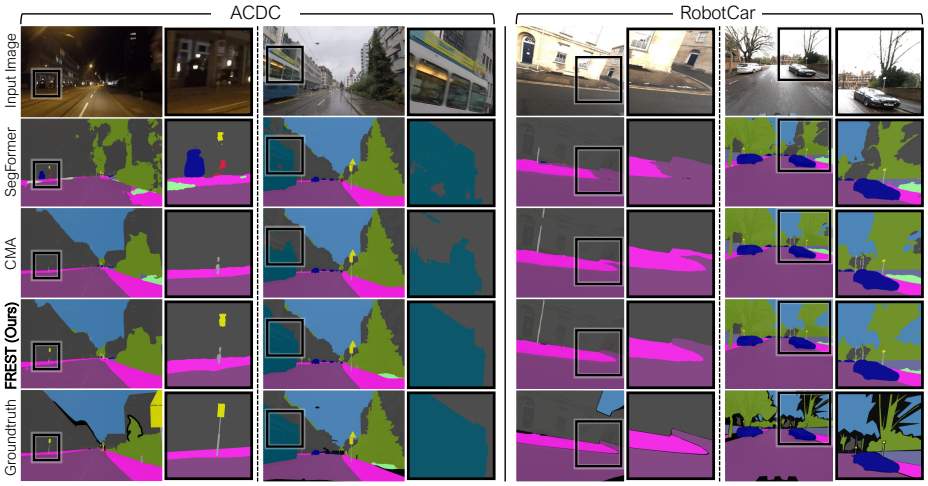

Fig. 8: Qualitative results of FREST (Ours), its baseline (SegFormer [45]), and CMA [1] on ACDC and RobotCar.

is likely to share common adverse conditions with the anchor embedding. This similarity helps to learn condition information, allowing the model to consider the distinct attributes of each adverse condition. Consequently, it helps FREST to learn the condition embedding space effectively.

5.5 Qualitative Results

As illustrated in Fig. 8, we present the qualitative results FREST, its baseline [45], and a previous work [1] on ACDC [34] and RobotCar [19, 26]. The results show that FREST excels in segmenting fine-grained objects such as a pole (1st and 4th column) and classifying ambiguous semantics such as road and sidewalk (2nd and 3rd column) across multiple adverse conditions compared with its baseline and the previous work.

6 Conclusion

We have presented the novel framework of feature restoration for multiple adverse conditions. FREST operates in two stages as follows: (1) learning the condition embedding space that represents only condition-specific information of images and (2) restoring features for adverse conditions on the learned condition embedding space. As a result, FREST achieved a new state of the art on two benchmarks for SFDA under multiple adverse conditions, while it showed superior generalization ability on unseen datasets. In terms of limitations, our framework currently does not cover various adverse conditions, including image degradation and camera artifacts, which will be expanded in our future work.

Acknowledgement. This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1801-52.

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. Brüggemann, D., Sakaridis, C., Truong, P., Van Gool, L.: Refign: Align and refine for adaptation of semantic segmentation to adverse conditions. In: Proc. IEEE Winter Conference on Applications of Computer Vision (WACV) (2023)
3. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. In: Proc. Neural Information Processing Systems (NeurIPS) (2022)
4. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
5. Choi, S., Jung, S., Yun, H., Kim, J.T., Kim, S., Choo, J.: Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
7. Dai, D., Sakaridis, C., Hecker, S., Van Gool, L.: Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. International Journal of Computer Vision (IJCV) (2020)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proc. International Conference on Learning Representations (ICLR) (2021)
9. Guo, X., Liu, J., Liu, T., Yuan, Y.: Simt: Handling open-set noise for domain adaptive semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
10. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
11. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: Proc. International Conference on Machine Learning (ICML). PMLR (2019)
12. Hoyer, L., Dai, D., Van Gool, L.: DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
13. Hoyer, L., Dai, D., Van Gool, L.: HRDA: Context-aware high-resolution domain-adaptive semantic segmentation. In: Proc. European Conference on Computer Vision (ECCV) (2022)
14. Hoyer, L., Dai, D., Wang, H., Van Gool, L.: Mic: Masked image consistency for context-enhanced domain adaptation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11721–11732 (2023)
15. 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)

16. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the hausdorff distance. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (1993)
17. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Proc. Neural Information Processing Systems (NeurIPS) (2020)
18. Kim, M., Byun, H.: Learning texture invariant representation for domain adaptation of semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
19. 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)
20. Lee, S., Rim, J., Jeong, B., Kim, G., Woo, B., Lee, H., Cho, S., Kwak, S.: Human pose estimation in extremely low-light conditions. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
21. Lee, S., Son, T., Kwak, S.: Fifo: Learning fog-invariant features for foggy scene segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
22. Li, Y., Yuan, L., Vasconcelos, N.: Bidirectional learning for domain adaptation of semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
24. Liu, Y., Zhang, W., Wang, J.: Source-free domain adaptation for semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Proc. International Conference on Learning Representations (ICLR) (2019)
26. Maddern, W., Pascoe, G., Linegar, C., Newman, P.: 1 year, 1000 km: The oxford robotcar dataset. The International Journal of Robotics Research (2017)
27. Pan, F., Shin, I., Rameau, F., Lee, S., Kweon, I.S.: Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
28. Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., Gurevych, I.: Adapterfusion: Non-destructive task composition for transfer learning. arXiv preprint arXiv:2005.00247 (2020)
29. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for data: Ground truth from computer games. In: Proc. European Conference on Computer Vision (ECCV) (2016)
30. 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: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
31. Sakaridis, C., Dai, D., Gool, L.V.: Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)

32. Sakaridis, C., Dai, D., Hecker, S., Van Gool, L.: Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In: Proc. European Conference on Computer Vision (ECCV) (2018)
33. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. International Journal of Computer Vision (IJCV) (2018)
34. 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)
35. Son, T., Kang, J., Kim, N., Cho, S., Kwak, S.: Urie: Universal image enhancement for visual recognition in the wild. In: Proc. European Conference on Computer Vision (ECCV) (2020)
36. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
37. 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)
38. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
39. 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)
40. Tsai, Y.H., Sohn, K., Schulter, S., Chandraker, M.: Domain adaptation for structured output via discriminative patch representations. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proc. Neural Information Processing Systems (NeurIPS) (2017)
42. 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)
43. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: Proc. International Conference on Learning Representations (ICLR) (2021)
44. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: Proc. European Conference on Computer Vision (ECCV) (2018)
45. 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)
46. Zendel, O., Honauer, K., Murschitz, M., Steininger, D., Fernandez Dominguez, G.: Wilddash - creating hazard-aware benchmarks. In: Proc. European Conference on Computer Vision (ECCV) (2018)
47. Zhang, F., Zhu, X., Ye, M.: Fast human pose estimation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
48. Zhang, Y., David, P., Gong, B.: Curriculum domain adaptation for semantic segmentation of urban scenes. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2017)

49. Zhao, D., Wang, S., Zang, Q., Quan, D., Ye, X., Jiao, L.: Towards better stability and adaptability: Improve online self-training for model adaptation in semantic segmentation. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
50. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Proc. European Conference on Computer Vision (ECCV) (2018)
51. Zou, Y., Yu, Z., Liu, X., Kumar, B., Wang, J.: Confidence regularized self-training. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV) (2019)