Refign: Align and Refine for Adaptation of Semantic Segmentation to Adverse Conditions

Supplementary Material

David Bruggemann Christos Sakaridis Prune Truong Luc Van Gool
ETH Zurich, Switzerland
{brdavid, csakarid, truongp, vangool}@vision.ee.ethz.ch

A. Mathematical Derivations

Derivation of Log-Likelihood Loss We model the likelihood with an uncorrelated, bivariate Gaussian with mean $\hat{\mathbf{F}} = [\hat{F}^u, \hat{F}^v]^\top$ and variance $\hat{\Sigma} = \hat{\Sigma}^u = \hat{\Sigma}^v$ for flow directions u and v.

$$\begin{aligned}
\mathcal{L}^{prob}_{\mathbf{I}'\to\mathbf{I}} &= -\log p(\mathbf{W}|\mathbf{I},\mathbf{I}') \\
&= -\log\Bigg(\frac{1}{\sqrt{2\pi\hat{\Sigma}^u_{\mathbf{I}'\to\mathbf{I}}}} e^{-\frac{1}{2\hat{\Sigma}^u_{\mathbf{I}'\to\mathbf{I}}}\left(\hat{F}^u_{\mathbf{I}'\to\mathbf{I}}-W^u\right)^2}\cdot \\
&\qquad \frac{1}{\sqrt{2\pi\hat{\Sigma}^v_{\mathbf{I}'\to\mathbf{I}}}} e^{-\frac{1}{2\hat{\Sigma}^v_{\mathbf{I}'\to\mathbf{I}}}(\hat{F}^v_{\mathbf{I}'\to\mathbf{I}}-W^v)^2}\Bigg) \\
&= -\log\left(\frac{1}{2\pi\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}}} e^{-\frac{1}{2\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}}}\left\|\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{I}}-\mathbf{W}\right\|^2}\right) \\
&\propto \frac{1}{2\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}}}\left\|\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{I}}-\mathbf{W}\right\|^2 + \log\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}} \\
&= \frac{1}{2\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}}}\mathcal{L}_{\mathbf{I}'\to\mathbf{I}} + \log\hat{\Sigma}_{\mathbf{I}'\to\mathbf{I}}
\end{aligned} \tag{1}$$

Derivation of Confidence Map We integrate the bivariate Gaussian density function over a circle with radius r (subscripts are omitted).

$$\begin{aligned}
P_{\mathcal{R}} &= p(\|\mathbf{F}-\hat{\mathbf{F}}\| \le r) \\
&= \int_0^{2\pi}\int_0^r \frac{1}{2\pi\hat{\Sigma}} e^{-\frac{1}{2\hat{\Sigma}}\rho^2}\rho d\rho d\phi \\
&= 1 - \exp\frac{-r^2}{2\hat{\Sigma}}
\end{aligned} \tag{2}$$

B. Training Details

In this section, we describe training settings and implementation details. Both alignment and segmentation network were trained using Automatic Mixed Precision on a single consumer RTX 2080 Ti GPU.

B.1. Alignment Network

UAWarpC training almost exactly follows the setup of [27]. The training consists of two stages: In the first stage, the network is trained without the visibility mask, as the visibility mask estimate is still inaccurate. In the second stage, the visibility mask is activated and more data augmentation is used.

Data Handling The alignment network is trained using MegaDepth [13], consisting of 196 scenes reconstructed from 1,070,468 internet photos with COLMAP [22]. 150 scenes are used for training, encompassing around 58,000 sampled image pairs. 1800 image pairs sampled from 25 different scenes are used for validation. No ground-truth correspondences from SfM reconstructions are used to train UAWarpC.

During training, the image pairs $\mathbf{I}, \mathbf{J}$ are resized to 750×750 pixels, and a dense flow $\mathbf{W}$ is sampled to create $\mathbf{I}'$. Finally, all three images $\mathbf{I}, \mathbf{J}, \mathbf{I}'$ are center-cropped to resolution 520×520. In the first training stage, $\mathbf{W}$ consists of sampled color jitter, Gaussian blur, homography, TPS, and affine-TPS transformations. In the second stage, local elastic transformations are added, and the strength of the transformations is increased. For the detailed augmentation parameters, we refer to [27].

Architecture and Loss Function Again following [27], a modified GLU-Net [25] is used as a base architecture for flow prediction. GLU-Net is a four-level pyramidal network with a VGG-16 [23] encoder. The encoder is initialized with ImageNet weights and frozen. GLU-Net requires an additional low-resolution input of 256×256 to establish global correlations, followed by repeated levels of upscaling and local feature correlations. As in [27], our flow decoder uses residual connections for efficiency. In addition, we replace all transposed convolutions with bilinear upsampling, and normalize all encoder feature maps, to increase the convergence rate.

The uncertainty estimate is produced using the uncertainty decoder proposed in [26]. However, instead of predicting the parameters of several mixture components, we simply output a single value per pixel—the log-variance.

As in [27], the loss is applied at all four levels of the pyramidal GLU-Net. We simply add the four components. The employed loss functions are explained in Sec. 3.1 of the main paper. To obtain the visibility mask for the second training stage, we use the Cauchy-Schwarz inequality, analogously to [27].

$$V = \mathbf{1}\left[\left\|\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{J}} + \Phi_{\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{J}}}(\hat{\mathbf{F}}_{\mathbf{J}\to\mathbf{I}}) - \mathbf{W}\right\|^2 < \alpha_2 + \alpha_1\left(\left\|\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{J}}\right\|^2 + \left\|\Phi_{\hat{\mathbf{F}}_{\mathbf{I}'\to\mathbf{J}}}(\hat{\mathbf{F}}_{\mathbf{J}\to\mathbf{I}})\right\|^2 + \left\|\mathbf{W}\right\|^2\right)\right] \tag{3}$$

$\mathbf{1}$ denotes the element-wise indicator function. We use $\alpha_1 = 0.03$ and $\alpha_2 = 0.05$.

Optimization Schedule For the first training stage, the alignment network is trained with a batch size of 6 for 400k iterations. We use the Adam optimizer [9] with weight decay $4 \cdot 10^{-4}$. The initial learning rate is 10^{-4}, and is halved after 250k and 325k iterations. For the second training stage, we use 225k training steps with initial learning rate $5 \cdot 10^{-5}$, halved after 100k, 150k, and 200k iterations.

B.2. Segmentation Network

For training the domain adaptive segmentation network, we follow the employed base UDA method, respectively. We summarize here the settings used with DAFormer [7]. For more details, and the DACS [24] settings, we refer to the original papers or the authors' codes[1].

Data Handling Input images are resized to half resolution for Cityscapes [3], ACDC [20], and Dark Zurich [21]. For RobotCar Correspondence [17, 10] and CMU Correspondence [1, 10], we resize to 720×720 and 540×720, respectively. Data augmentation consists of random cropping to 512×512 and random horizontal flipping. For the coarsely labeled extra target images in the semi-supervised domain adaptation for RobotCar and CMU, we additionally apply random rotation with maximum 10° and color jittering.

[1] `https://github.com/lhoyer/DAFormer`, `https://github.com/vikolss/DACS`

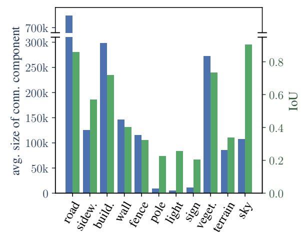

Figure C-1. Correlation between the average size of connected components (on Cityscapes [3]) and mIoU score of warped reference image predictions for static classes. Larger classes benefit heavily from indiscriminate warping.

Optimization Schedule We use the AdamW [15] optimizer with a weight decay of 0.01. The learning rate follows a linear warmup for 1500 steps, followed by linear decay. The peak learning rate is $6 \cdot 10^{-4}$. On ACDC and Dark Zurich, we train for 40k iterations; on RobotCar and CMU, we train for 20k iterations. A batch size of 2 is used throughout.

To mitigate the risk of overfitting, we use the coarsely labeled extra target images in semi-supervised domain adaptation on RobotCar and CMU only in every second training iteration.

C. Small *vs*. Large Static Classes

To motivate the distinction between small and large static classes (as defined in Sec. 3.2), we generate ACDC [20] reference image predictions using a SegFormer [33] trained on Cityscapes [3], and warp them onto the corresponding adverse-image viewpoint. As shown in Fig. C-1, we observe a correlation between the resulting IoU and the average size of the connected class component for static classes (pearson correlation coeff. of 0.70). The classes *pole*, *traffic light*, and *traffic sign* are drastically smaller than the rest, and consequentially have lower accuracy. On the other hand, such indiscriminate warping (*i.e.*, without $P_{\mathcal{R}}$) is surprisingly accurate for the large static classes.

Furthermore, we analyze the mIoU improvement when only considering pixels above a certain $P_{\mathcal{R}}$ threshold for the above mentioned warped SegFormer predictions, see Fig. C-2. While the performance increases monotonically for both dynamic and small static classes, it remains mostly flat for large static classes. This suggests that large static classes are largely insensitive to the warping confidence, while both dynamic and small static classes benefit greatly from confidence guidance.

D. Additional Experimental Results

Due to space restrictions, we present the full class-wise performances of state-of-the-art UDA methods on Dark Zurich-test here in Table D-1. The models reported in Tables 1, 2, and D-1 all use the same image input size at test-time for fairness of comparison. Table D-2 presents models which do not follow that protocol. Using Cityscapes-pretrained weights for initialization, Refign added on top of

Table D-1. State-of-the-art comparison on Dark Zurich-test for Cityscapes→Dark Zurich domain adaptation. Methods above the double line all use a DeepLabv2 [2] model. "Ref.": For each adverse input image a reference image at similar geo-location is used.

Method	Ref.	IoU ↑ road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	**mean**
DeepLabv2 [2]		79.0	21.8	53.0	13.3	11.2	22.5	20.2	22.1	43.5	10.4	18.0	37.4	33.8	64.1	6.4	0.0	52.3	30.4	7.4	28.8
ADVENT [29]		85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
AdaptSegNet [28]		86.1	44.2	55.1	22.2	4.8	21.1	5.6	16.7	37.2	8.4	1.2	35.9	26.7	68.2	45.1	0.0	50.1	33.9	15.6	30.4
BDL [12]		85.3	41.1	61.9	32.7	17.4	20.6	11.4	21.3	29.4	8.9	1.1	37.4	22.1	63.2	28.2	0.0	47.7	39.4	15.7	30.8
DANNet (DeepLabv2) [30]	✓	88.6	53.4	69.8	34.0	20.0	25.0	31.5	35.9	69.5	32.2	82.3	44.2	43.7	54.1	22.0	0.1	40.9	36.0	24.1	42.5
DANIA (DeepLabv2) [31]	✓	89.4	60.6	72.3	34.5	23.7	37.3	32.8	40.0	72.1	33.0	84.1	44.7	48.9	59.0	9.8	0.1	40.1	38.4	30.5	44.8
DACS [24]		83.1	49.1	67.4	33.2	16.6	42.9	20.7	35.6	31.7	5.1	6.5	41.7	18.2	68.8	**76.4**	0.0	61.6	27.7	10.7	36.7
Refign-DACS	✓	89.9	59.7	69.5	28.5	11.6	39.0	17.1	35.0	35.7	18.8	30.4	38.8	43.1	72.3	73.7	0.0	61.6	33.9	24.7	41.2
DMAda (RefineNet) [5]	✓	75.5	29.1	48.6	21.3	14.3	34.3	36.8	29.9	49.4	13.8	0.4	43.3	50.2	69.4	18.4	0.0	27.6	34.9	11.9	32.1
GCMA (RefineNet) [18]	✓	81.7	46.9	58.8	22.0	20.0	41.2	40.5	41.6	64.8	31.0	32.1	53.5	47.5	75.5	39.2	0.0	49.6	30.7	21.0	42.0
MGCDA (RefineNet) [21]	✓	80.3	49.3	66.2	7.8	11.0	41.4	38.9	39.0	64.1	18.0	55.8	52.1	**53.5**	74.7	66.0	0.0	37.5	29.1	22.7	42.5
CDAda (RefineNet) [34]	✓	90.5	60.6	67.9	37.0	19.3	42.9	36.4	35.3	66.9	24.4	79.8	45.4	42.9	70.8	51.7	0.0	29.7	27.7	26.2	45.0
DANNet (PSPNet) [30]	✓	90.4	60.1	71.0	33.6	22.9	30.6	34.3	33.7	70.5	31.8	80.2	45.7	41.6	67.4	16.8	0.0	73.0	31.6	22.9	45.2
CCDistill (RefineNet) [6]	✓	89.6	58.1	70.6	36.6	22.5	33.0	27.0	30.5	68.3	33.0	80.9	42.3	40.1	69.4	58.1	0.1	72.6	47.7	21.3	47.5
DANIA (PSPNet) [31]	✓	91.5	62.7	73.9	**39.9**	25.7	36.5	35.7	36.2	71.4	**35.3**	82.2	48.0	44.9	73.7	11.3	0.1	64.3	36.7	22.7	47.0
DAFormer [7]		**93.5**	**65.5**	73.3	39.4	19.2	53.3	44.1	44.0	59.5	34.5	66.6	53.4	52.7	**82.1**	52.7	**9.5**	89.3	**50.5**	**38.5**	53.8
Refign-DAFormer	✓	91.8	65.0	**80.9**	37.9	**25.8**	**56.2**	**45.2**	**51.0**	**78.7**	31.0	**88.9**	**58.8**	52.9	77.8	51.8	6.1	**90.8**	40.2	37.1	**56.2**

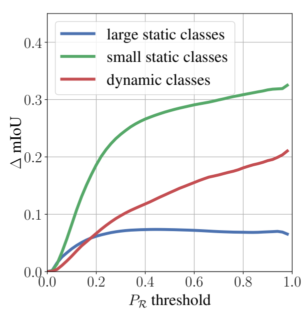

Figure C-2. Performance increase for different class categories as a function of the warp confidence ($P_{\mathcal{R}}$) threshold. Dynamic classes and small static classes (see Sec. 3.2) are more sensitive to the warp confidence, while large static classes do not improve considerably.

Table D-2. State-of-the-art comparison of models which do not follow the common image input resizing protocol. Refign-HRDA currently ranks first on public leaderboards.

Method	Cityscapes→ACDC ACDC [20]	Cityscapes→Dark Zurich Dark Zurich-test [21]	ND [5]	Bn [36, 21]
SePiCo (DAFormer) [32]	-	54.2	57.1	36.9
HRDA [8]	68.0	55.9	55.6	39.1
Refign-HRDA	**72.1**	**63.9**	**57.8**	**40.6**

HRDA [8] achieves 72.1 mIoU and 63.9 mIoU on ACDC and Dark Zurich-test, respectively, ranking first on the public leaderboards of these benchmarks at the time of publication.

In Table D-3, we report the performance of Cityscapes→ACDC Refign-DAFormer on the four different conditions of the ACDC validation set. Refign improves markedly over the baseline for all conditions.

We also compare the Cityscapes→ACDC Refign-DAFormer model with state-of-the-art foggy scene understanding methods in Table D-4. All methods are trained with Cityscapes as source domain, however the foggy scene understanding methods utilize both synthetic foggy data and a larger pool of real foggy data as targets. Surprisingly, our model achieves state-of-the-art performance despite this handicap.

Table D-3. Performance of Cityscapes→ACDC models for different conditions on the validation set.

Method	mIoU ↑ night	snow	rain	fog
DAFormer [7]	34.8	56.3	58.5	67.9
Refign-DAFormer	**48.1**	**65.0**	**65.2**	**73.4**

Table D-4. Performance comparison with specialized foggy scene understanding methods on the Foggy Zurich [4] and Foggy Driving [19] test sets.

Method	Target Domain Training Data: Foggy CS-DBF [4]	Foggy Zurich[4]	ACDC [20]	mIoU ↑ Foggy Zurich [4]	FoggyDriving [19]
CMAda3+ [4]	✓	✓		46.8	49.8
FIFO [11]	✓	✓		48.4	50.7
CuDA-Net+ [16]	✓	✓		49.1	53.5
TDo-Dif [14]	✓	✓		**51.9**	50.7
Refign-DAFormer			✓	51.4	**53.9**

Finally, we conduct experiments substituting the SegFormer [33] based architecture of DAFormer [7] with DeepLabv2 [2]. On both ACDC and Dark Zurich validation sets, this version of Refign improves substantially over the baseline, as reported in Table D-5.

E. Refign at Test-Time

Although designed to refine pseudo-labels during online self-training, Refign can also be applied at test-time to ar-

Table D-5. Performance of Refign *vs*. DAFormer baseline with a DeepLabv2 model on the ACDC and Dark Zurich validation sets.

Method	mIoU↑	
	ACDC [20]	Dark Zurich [21]
DAFormer (DeepLabv2) [7]	46.4	24.8
Refign-DAFormer (DeepLabv2)	**55.6**	**38.7**

Table E-1. Applying Refign only for one refinement iteration at test-time to DAFormer on the ACDC and Dark Zurich validation sets.

Method	mIoU↑	
	ACDC [20]	Dark Zurich [21]
DAFormer [7]	55.6	34.1
DAFormer + Test-Time Refign	**56.8**	**38.0**

bitrary, trained models, if a reference image is available. We report ACDC and Dark Zurich validation set scores in Table E-1. The performance gain is more moderate than if Refign is applied at training-time. This is unsurprising, given that we only conduct a single refinement iteration in that case.

F. Qualitative Results

We show more qualitative results in this section. Fig. F-1 shows the warps and corresponding confidence maps for randomly selected ACDC samples. In Fig. F-2, we show some warp failures. Importantly, the confidence map correctly blends out the inaccurate warps. Finally, Fig. F-3 shows more qualitative segmentation results for randomly selected ACDC validation samples.

G. Potential Negative Societal Impact

We present a method to adapt existing semantic segmentation models to new domains. Even though we restrict ourselves to adverse-condition autonomous driving in this paper, our algorithm could potentially be used in more undesired applications, such as surveillance or military. This risk of potential misuse exists for all semantic segmentation algorithms, and could be mitigated through appropriate legislation.

References

[1] Hernán Badino, Daniel Huber, and Takeo Kanade. Visual topometric localization. In *IEEE Intelligent vehicles symposium (IV)*, 2011.

[2] 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 TPAMI*, 40(4):834–848, 2017.

[3] 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*, 2016.

[4] Dengxin Dai, Christos Sakaridis, Simon Hecker, and Luc Van Gool. Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *IJCV*, 128(5):1182–1204, 2020.

[5] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[6] Huan Gao, Jichang Guo, Guoli Wang, and Qian Zhang. Cross-domain correlation distillation for unsupervised domain adaptation in nighttime semantic segmentation. In *CVPR*, 2022.

[7] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022.

[8] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation, 2022.

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[10] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic segmentation. In *CVPR*, 2019.

[11] Sohyun Lee, Taeyoung Son, and Suha Kwak. FIFO: Learning fog-invariant features for foggy scene segmentation. In *CVPR*, 2022.

[12] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019.

[13] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

[14] Liang Liao, Wenyi Chen, Jing Xiao, Zheng Wang, Chia-Wen Lin, and Shin'ichi Satoh. Unsupervised foggy scene understanding via self spatial-temporal label diffusion. *IEEE Transactions on Image Processing*, 31:3525–3540, 2022.

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[16] Xianzheng Ma, Zhixiang Wang, Yacheng Zhan, Yinqiang Zheng, Zheng Wang, Dengxin Dai, and Chia-Wen Lin. Both style and fog matter: Cumulative domain adaptation for semantic foggy scene understanding. In *CVPR*, 2022.

[17] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[18] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *ICCV*, 2019.

[19] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018.

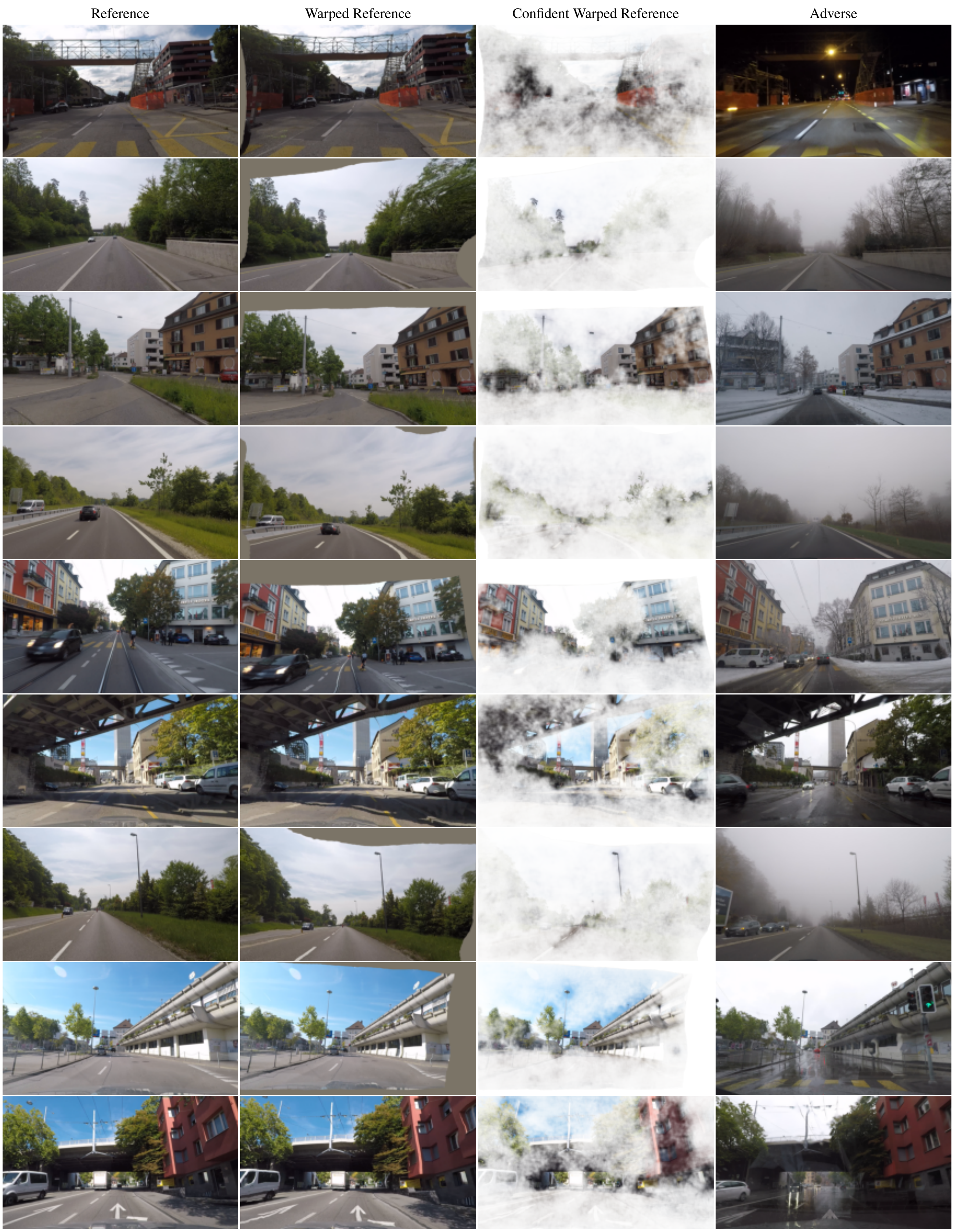

Figure F-1. Example visualizations of warped reference images and the corresponding confidence maps from ACDC.

Figure F-2. Warp failure examples on ACDC.

[20] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The Adverse Conditions Dataset with Correspondences for semantic driving scene understanding. In *ICCV*, 2021.

[21] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *TPAMI*, 44(6):3139–3153, 2022.

[22] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[24] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *WACV*, 2021.

[25] Prune Truong, Martin Danelljan, and Radu Timofte. Glu-net: Global-local universal network for dense flow and correspondences. In *CVPR*, 2020.

[26] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network, 2021.

[27] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. In *ICCV*, 2021.

[28] 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*, 2018.

[29] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

[30] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. DANNet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *CVPR*, 2021.

[31] Xinyi Wu, Zhenyao Wu, Lili Ju, and Song Wang. A one-stage domain adaptation network with image alignment for unsupervised nighttime semantic segmentation. *IEEE TPAMI*, (01):1–1, 2021.

[32] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation, 2022.

[33] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

[34] Qi Xu, Yinan Ma, Jing Wu, Chengnian Long, and Xiaolin Huang. Cdada: A curriculum domain adaptation for nighttime semantic segmentation. In *ICCV*, 2021.

[35] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

[36] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020.

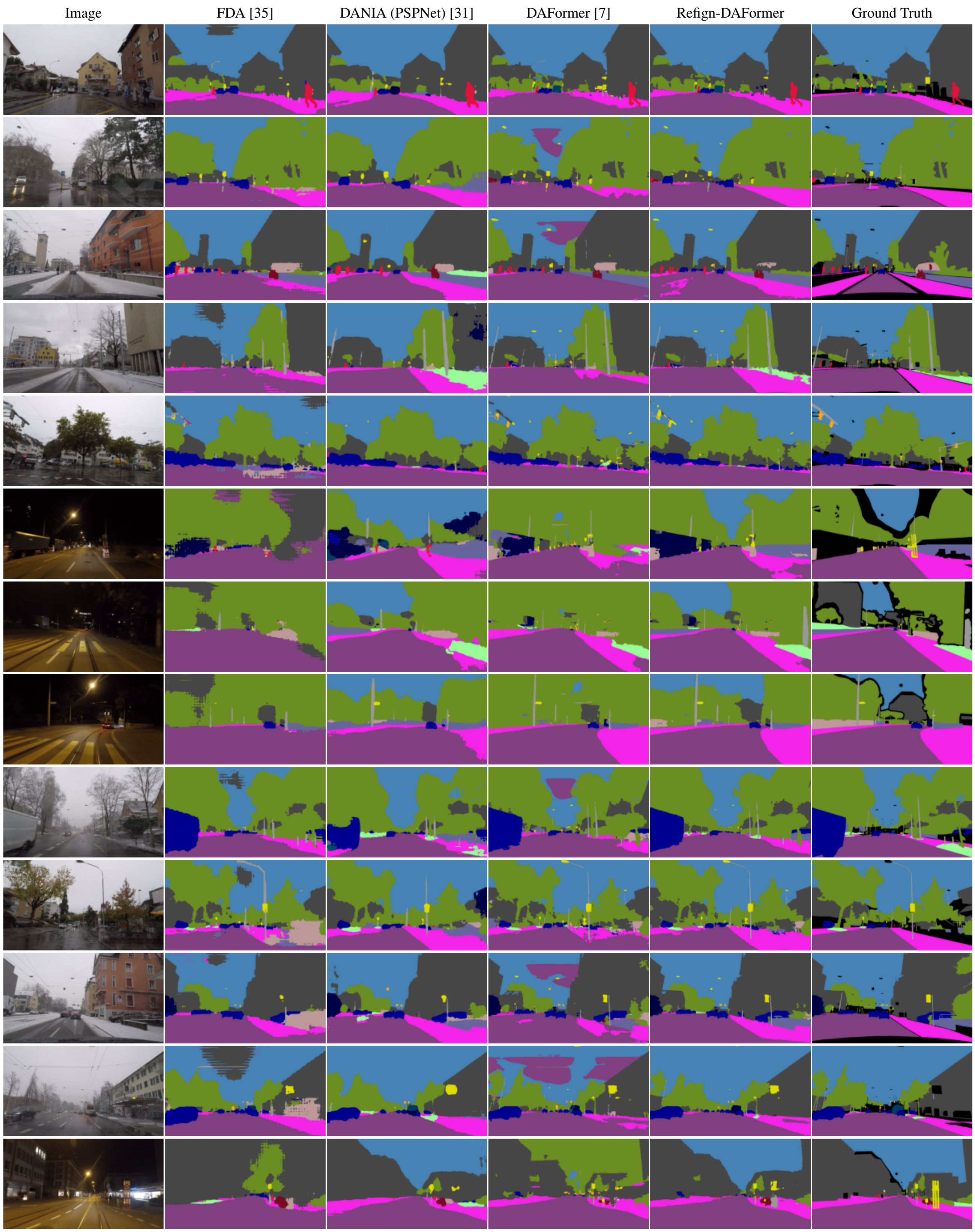

Figure F-3. Prediction samples of the ACDC validation set.