Optical Flow Domain Adaptation via Target Style Transfer
-Supplementary Material-

Jeongbeen Yoon Sanghyun Kim Suha Kwak Minsu Cho

Pohang University of Science and Technology (POSTECH), South Korea
{jeongbeen, sanghyun.kim, suha.kwak, mscho}@postech.ac.kr

(a) The images from VKITTI 2 dataset. It contains 5 scenes (Scene 1, 2, 6, 18, and 20) and each scene has 6 domains (Clone, Fog, Morning, Overcast, Rain and Sunset).

(b) The images from FlyingThings3D, Sintel, and KITTI 2015 dataset.

Figure s.1. Dataset visualization.

A. Appendix

We include implementation details for reproducibility and also provide additional experimental results and analyses in this supplementary material.

A.1. Implementation details

We examine the effectiveness of our proposed method in multiple combinations of experimental settings. This section clearly states the datasets and detailed setups used for the experiments in the main paper.

Datasets. In Figure s.1a, we visualize the examples of VKITTI 2 [2]; the images are arranged to recognize the domain differences easily. The figure shows that the images from a scene share the same flow map, but their domains are significantly different. As can be seen in Table 1 of the

Table s.1. Details for training our model. C + T represents that we bring the model pretrained on FlyingChairs + FlyingThings3D.

Training data		Base network	Weights	Learning rate	Batch size	Weight decay	Crop size
Source	Target						
VKITTI(Clone)	VKITTI	FlowNetC	-	1e-4	8	4e-4	[320, 448]
		FlowNetS	-	1e-4	8	4e-4	[320, 448]
		RAFT	-	4e-4	4	1e-5	[288, 960]
FlyingThings3D	Sintel	RAFT	C + T	1e-4	4	1e-4	[400, 720]
FlyingThings3D	KITTI	RAFT	C + T	1e-4	4	1e-4	[288, 720]

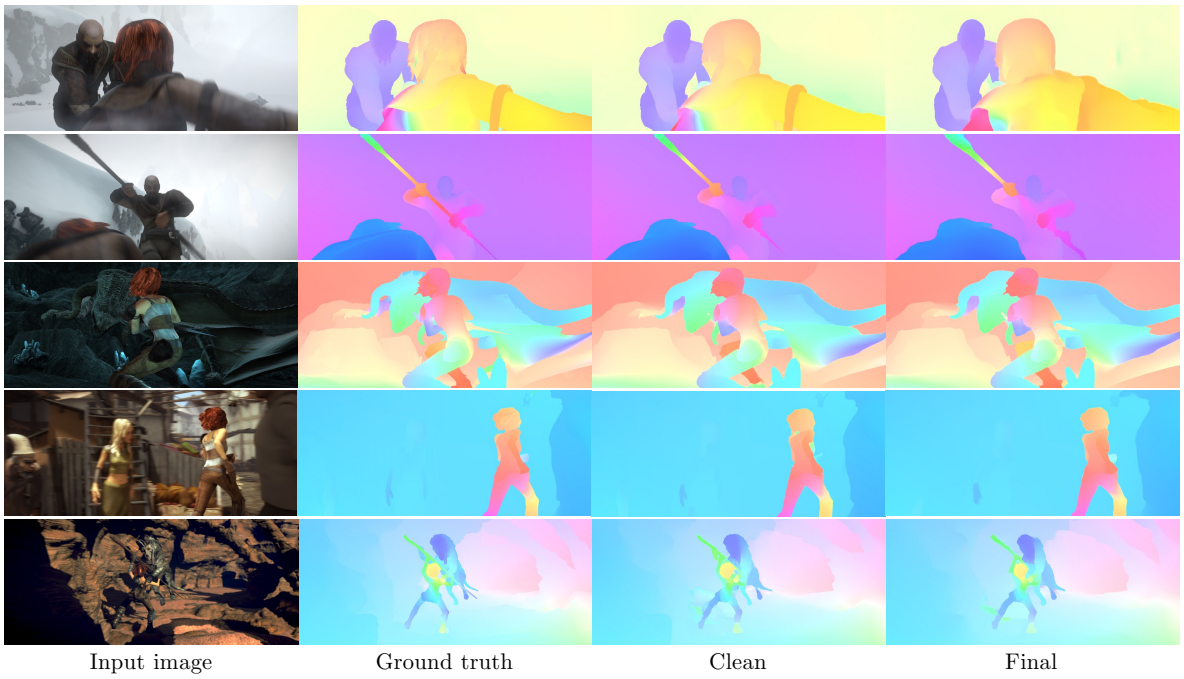


Figure s.2. Qualitative results of Sintel test split using our method with RAFT baseline.

main paper, the performance of ordinary optical flow models trained on one domain of this dataset significantly degrades on another domain. Figure s.1b describes the images from FlyingThings3D [9], Sintel [1], and KITTI 2015 [4] datasets. There also exist apparent differences among the three datasets.

Baselines. We utilize open-source PyTorch [10] implementation of FlowNet [3] [1]. To reproduce RAFT [11], we follow the official PyTorch implementation [2]. In case of FS [5], we bring the official TensorFlow implementation [3]. For all Baseline + AT and Baseline + GST experiments, we use the hyper-parameter values that make the model perform the best. Specifically, in VKITTI 2, the value of λ_{adv} is 0.1 for FlowNet and 0.01 for RAFT. The value of λ_{cons} is 0.01 for FlowNet + GST and 0.1 for RAFT + GST. In Sintel and KITTI 2015 experiments, we set the value of λ_{adv} and λ_{cons} to 0.001 and 0.01, respectively, for RAFT + GST.

Training details. Our training details are summarized in Table s.1. In the VKITTI dataset, we train our model and baselines from scratch. In the case of Sintel and KITTI, we deploy the model trained on FlyingChiars and FlyingThings3D (C + T). As stated in Section 4 of the main paper, we use Adam [6] and AdamW [8] optimizers for FlowNet and RAFT, respectively. The learning rate starts at $1e-4$, except for RAFT in the VKITTI 2 dataset. We set the batch size to 4 for Sintel and KITTI experiments. The specific details of weight decay and crop size are denoted in Table s.1. We follow the same augmentation technique with the baselines.

[1] https://github.com/ClementPinard/FlowNetPytorch

[2] https://github.com/princeton-vl/RAFT

[3] https://github.com/iwbn/flow-supervisor

Table s.2. Average End Point Error (AEPE) results of Fog domain in VKITTI dataset for searching optimal hyper-parameters.

λ_{adv} \ λ_{cons}	0.001	0.01	0.1
0.01	6.28	6.29	6.55
0.1	5.22	5.35	5.25
1	**4.55**	4.66	4.64

Table s.3. Average End Point Error (AEPE) results of Overcast domain in VKITTI dataset for searching optimal hyper-parameters of flow supervisor.

α \ λ_{TU}	0	0.1	0.4	0.7	1
0.4	1.71	2.13	1.98	2.08	2.02
0.7	**1.52**	2.00	2.09	2.03	2.12
1	**1.52**	2.28	2.21	2.02	1.98

Table s.4. The selection of patch sizes according to the dataset and the network.

Dataset	FH	Base net.	$\frac{FH}{2}$	$\frac{FH}{4}$	$\frac{FH}{8}$	$\frac{FH}{16}$	$\frac{FH}{32}$
VKITTI	160	FlowNetC			✓	✓	✓
	160	FlowNetS	✓	✓	✓		
	144	RAFT	✓	✓	✓		
Sintel	200	RAFT			✓	✓	✓
KITTI	144	RAFT		✓	✓	✓	

Hyper-parameters. We diversify the value of hyper-parameters λ_{adv} and λ_{cons} for searching their optimal value. As shown in Equation 12 in the main paper, λ_{adv} is the weighting parameter of the flow adversarial loss ($\mathcal{L}_{adv}$), and λ_{cons} is for both motion consistency losses ($\mathcal{L}^{corr}_{consist}$ and $\mathcal{L}^{pred}_{consist}$). Table s.2 shows experimental results for searching optimal hyper-parameters using FlowNetS in Fog domain of VKITTI dataset. We set λ_{adv} to 1 and λ_{cons} to 0.001 according to the results in Table s.2. We use the same value in all weather domains for FlowNetS. As Table s.2 shows, λ_{adv} is more sensitive than λ_{cons}. This is because, the balance of the generator and the discriminator of adversarial learning is influential in a whole training process. Similarly, we select 0.1 and 0.01 for FlowNetC and 0.001 and 0.001 for RAFT as the value of λ_{adv} and λ_{cons}. In other experiments, we set the value of λ_{adv} and λ_{cons} to 0.01 and 0.1 for Sintel(clean), 0.01 and 0.01 for Sintel(Final), and 0.001 and 0.01 for KITTI. We also search hyper-parameters on VKITTI dataset for flow supervisor. We change the value of α and λ_{TU} for finding their optimal value. We set α to 0.7 and λ_{TU} to 0 according to the results in Table s.3.

Patch size We specify the sets of patch sizes used for training our models in Table s.4. We represent the types of the patch size by dividing the height of the feature map into two to the n-th power. Specifically, we select three kinds of patch sizes among the set $S = \{\frac{FH}{2}, \frac{FH}{4}, \frac{FH}{8}, \frac{FH}{16}, \frac{FH}{32}\}$. FH represents the height of the feature. If the patch size becomes an even number, we add one to the patch size.

B. Additional experimental results.

Qualitative results of the Sintel test split. Figure s.2 shows the qualitative results of the Sintel test split. In this section, we utilize the model trained using Sintel(train) as a target. Images from the Sintel test split are not used for the training. The qualitative results of the third and fourth columns indicate the results of the clean-targeted and final-targeted models, respectively. Although our model does not exploit flow annotations, it successfully predicts fine flow fields.

Comprehensive ablation study. In this supplementary, we report all the domain scenario results of the ablation study in Table 3 in the main paper. As we can see from the average results of Table s.5, our proposed components (TST, MCL, and FAL) show their ability to decrease the domain discrepancy. The effect of our components is more influential in the target domain where the discrepancy is large (*i.e.*, Fog and Rain) than in the one with a small domain gap.

t-SNE visualization. Along with the t-SNE visualization experiment in Section 4.5 of the main paper, we present additional visualizations of embedding spaces for every domain in the VKITTI dataset in Figure s.4. This visualization validates the properties of the synthetic target feature generated by the TST module and its effectiveness for training the model. Before training the model, our TST module generates the synthetic target points (yellow) close to the target points (green). It thus assists the model to be easily adapted to the target domain, as shown in the figure. After training the model with our method, as shown in the second embedding spaces of Figure s.4, the model successfully extracts domain-invariant features in all target domains.

Quantitative results of source domain We also conduct an experiment using VKITTI dataset to evaluate the performance on the source domain after adapting an optical flow model to a target domain. In this experiment, each optical flow model trained on the Clone domain (source) is adapted by our method to the target domain, and its performance is evaluated on the Clone domain. The quantitative results are summarized in Table s.6; note that all the numbers in the table are the AEPE performances on the Clone domain. For example, the third row shows that the performance of RAFT model trained on Clone domain is 0.64 AEPE when it is evaluated on Clone domain without any adaptation. In

Table s.5. Comprehensive ablation study on all domains of the VKITTI 2. We exploit FlowNetC and select Clone as the source domain.

$\mathcal{L}_{\text{epe}}^{\text{ST}}$	$\mathcal{L}_{\text{consist}}^{\text{corr}}$	$\mathcal{L}_{\text{consist}}^{\text{pred}}$	$\mathcal{L}_{\text{adv}}$	Fog	Morning	Overcast	Rain	Sunset	Avg.
				48.53	6.32	9.07	15.18	6.58	17.14
✓				6.06	4.27	4.75	6.91	4.41	5.28
✓	✓			5.36	4.38	4.52	5.53	4.24	4.81
✓		✓		4.95	4.43	4.42	5.60	4.22	4.73
✓	✓	✓		5.20	4.27	4.54	5.43	4.16	4.72
✓	✓	✓	✓	4.91	4.37	4.49	5.39	4.28	4.69

Table s.6. The Average End Point Error (AEPE) performance on the source domain after adaptation.

Method	No adapt	Target domain				
		Fog	Morning	Overcast	Rain	Sunset
FlowNetS [3]	4.04	2.80	2.76	2.76	2.81	2.81
FlowNetC [3]	2.92	2.72	2.70	2.69	2.68	2.75
RAFT [11]	0.64	0.55	0.55	0.55	0.54	0.55

comparison, the model's performance becomes 0.55 AEPE on the Clone domain after adapting to the Fog domain using our method. Interestingly, after domain adaptation to other target domains, all the models improve even on the source domain, which implies that our adaptation process involving additional target data can act as data augmentation and improve generalization.

Without the C + T pretrained model. In the Sintel and KITTI experiments of the main paper, we use the model pretrained on FlyingChairs and FlyingThings3D (C + T in Table s.1). In this experiment, we aim to adapt the model to Sintel (or KITTI) from scratch; *i.e.*, we did not load the weights of the C + T pretrained model at the beginning of the training. We use the same experimental setting (*e.g.*, batch size, learning rate, and weight decay) as our experiment in the main paper. Table s.7 shows that our method outperforms when the pretrained model is not utilized. As stated in the main paper, the reason for this performance gap is that FS trains the model (student) using a supervisor's (teacher's) predictions. As the quality of the supervision primarily depends on the supervisor's performance, the existence of the pretrained model is critical for FS. On the other hand, our method is relatively free from this disadvantage because we use synthetic target features, of which annotations are from ground truth flows, as supervisions rather than a model's predictions.

Multiple source to target domain scenarios. There are six domains in VKITTI dataset: Clone, Fog, Morning, Overcast, Rain, and Sunset. To sufficiently demonstrate the domain adaptability of our model, we conduct experiments of all source-to-target domain scenarios in Table s.8. Among multiple baselines, we compare RAFT, RAFT+FS, and RAFT+ours. The results of the three models in Table s.8 indicate that our model outperforms in all domain scenarios. It also highlights that our method is flexibly adaptable to every source-to-target adaptation scenarios. Note that RAFT+FS is less effective than RAFT when the source domain is Morning or Sunset. We assume this is because pseudo labels extracted from FS distract the model when the source domain is Morning or Sunset.

Table s.7. The experiment without using the C + T pretrained model.

Method	Sintel		KITTI	
	clean	final	EPE	F1
RAFT + FS	3.69	4.77	13.06	38.70
RAFT + ours	**1.67**	**2.90**	**6.41**	**21.36**

Table s.8. Multiple source to target domain scenarios in VKITTI.

RAFT	Clone	Fog	Morning	Overcast	Rain	Sunset	Avg.
Clone	–	2.01	1.35	1.07	2.21	1.02	1.53
Fog	2.63	–	2.48	1.46	2.69	2.46	2.34
Morning	1.21	1.91	–	0.97	1.93	0.95	1.39
Overcast	2.38	2.11	2.03	–	1.23	1.86	1.92
Rain	2.47	1.45	2.48	0.80	–	1.98	1.84
Sunset	1.16	2.12	1.12	1.05	2.27	–	1.54
Avg.	1.97	1.92	1.89	1.07	2.07	1.65	1.76

(a) The Average End Point Error (AEPE) of RAFT model.

RAFT+FS	Clone	Fog	Morning	Overcast	Rain	Sunset	Avg.
Clone	–	1.73	1.51	1.52	1.63	1.44	1.57
Fog	2.30	–	2.83	1.94	1.70	1.96	2.15
Morning	1.70	2.26	–	1.91	1.78	1.76	1.88
Overcast	1.82	1.88	1.83	–	1.72	1.78	1.81
Rain	1.70	1.66	2.01	1.85	–	1.79	1.80
Sunset	1.58	3.11	2.16	1.69	3.51	–	2.41
Avg.	1.82	2.13	2.07	1.78	2.07	1.75	1.94

(b) The Average End Point Error (AEPE) of RAFT+FS model.

RAFT+ours	Clone	Fog	Morning	Overcast	Rain	Sunset	Avg.
Clone	–	1.48	0.99	0.84	1.46	0.82	1.12
Fog	1.98	–	2.04	1.30	1.40	2.02	1.75
Morning	1.03	1.61	–	0.79	1.70	0.80	1.19
Overcast	1.68	1.43	1.40	–	1.01	1.46	1.40
Rain	1.47	1.09	1.51	0.65	–	1.35	1.21
Sunset	0.98	1.63	0.89	0.87	1.96	–	1.27
Avg.	1.43	1.45	1.37	0.89	1.51	1.29	1.32

(c) The Average End Point Error (AEPE) of RAFT+ours model.

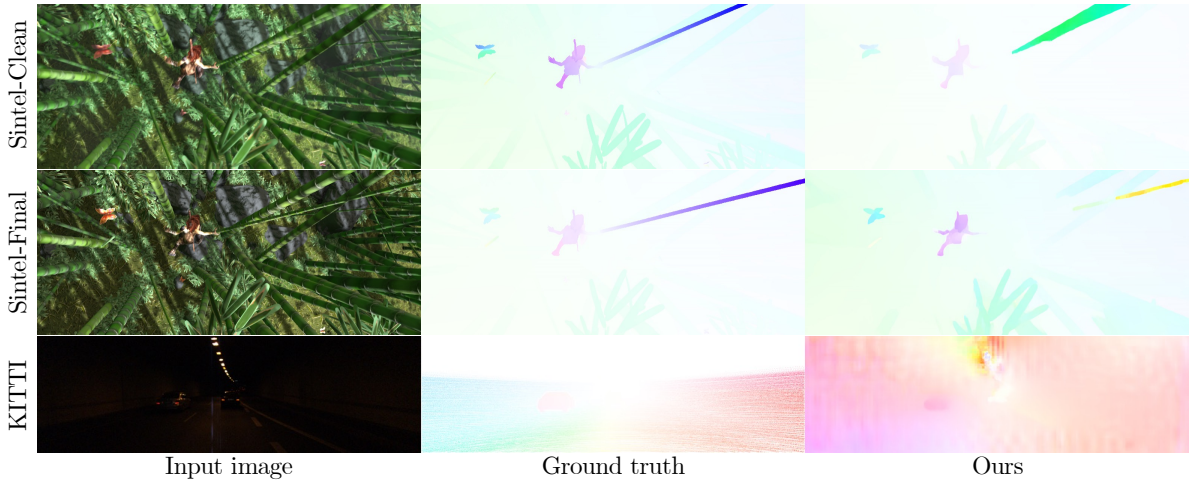


Figure s.3. Qualitative results of failure cases on Sintel and KITTI datasets.

Table s.9. Comparison with more related work in Fog and Rain domains in VKITTI.

Method	Fog	Rain
RainFlow [7]	–	8.27
Yan *et al*. [12]	1.60	–
RAFT	2.01	2.21
RAFT + ours	**1.48**	**1.46**

Comparison with more related work. We also address comparisons to related work [7, 12], which only deals with their specific target domains. According to Table s.9, our method prevails over previous work as well as being capable of covering multiple target domains. In addition, ours can be applied to various optical flow networks (*e.g*., FlowNetS, FlowNetC, and RAFT), thus continuously enabling the model to improve its performance.

Qualitative results of failure cases. Figure s.3 illustrates the qualitative results of failure cases on the Sintel and KITTI datasets. The proposed method encounters problems in estimating optical flows for two bamboo objects due to their similar appearance (*e.g*., texture and color). Similarly, in the second row, the proposed method struggles to capture the motion of a bamboo. In the last row, the method fails to capture optical flows in a low-light scene. The reason is insufficient low-light images to address the domain gap of low-light scenarios within the KITTI dataset.

References

[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 2

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 1

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas fvo. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 2, 4

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[5] Woobin Im, Sebin Lee, and Sung-Eui Yoon. Semi-supervised learning of optical flow by flow supervisor. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 302–318. Springer, 2022. 2

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[7] Ruoteng Li, Robby T Tan, Loong-Fah Cheong, Angelica I Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. Rainflow: Optical flow under rain streaks and rain veiling effect. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7304–7313, 2019. 5

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

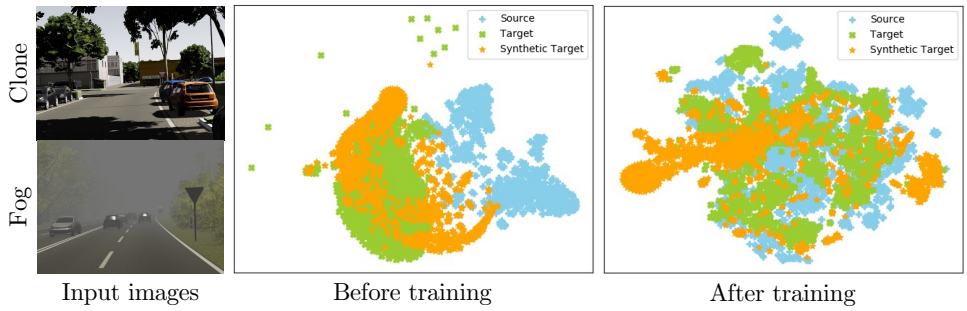
[9] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 2

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Ad-*

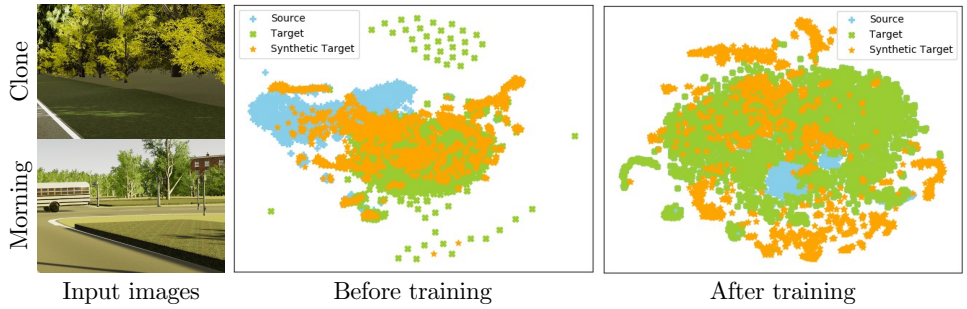
vances in neural information processing systems, 32:8026–8037, 2019. 2

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 2, 4

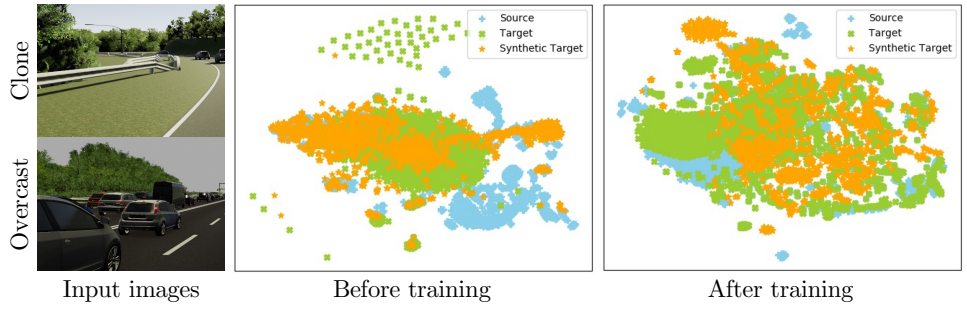
[12] Wending Yan, Aashish Sharma, and Robby T Tan. Optical flow in dense foggy scenes using semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13259–13268, 2020. 5



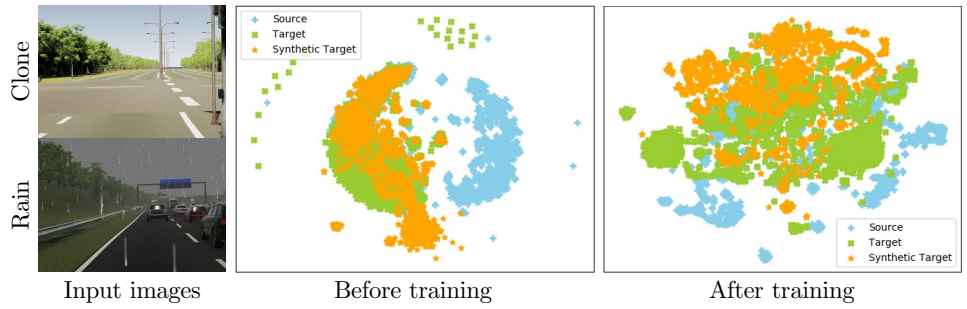
(a) Fog domain



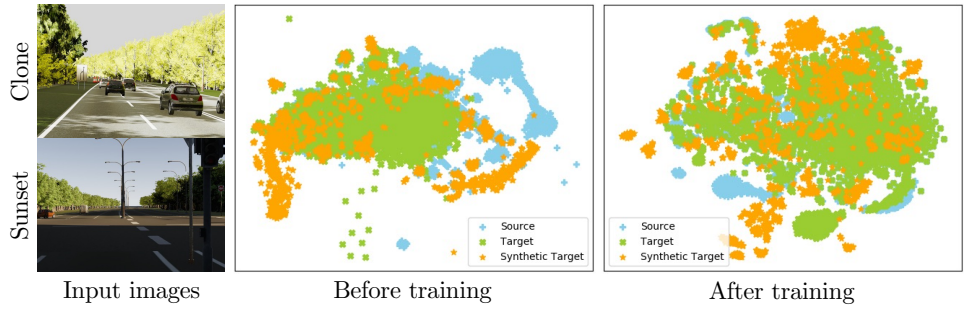
(b) Morning domain



(c) Overcast domain



(d) Rain domain



(e) Sunset domain

Figure s.4. t-SNE visualization of the source, target, and synthetic target features. We deploy VKITTI dataset and use FlowNetC for our base network.