Optical Flow Domain Adaptation via Target Style Transfer

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

Abstract

Optical flows play an integral role for a variety of motion-related tasks such as action recognition, object segmentation, and tracking in videos. While state-of-the-art optical flow methods heavily rely on learning, the learned optical flow methods significantly degrade when applied to different domains, and the training datasets are very limited due to the extreme cost of flow-level annotation. To tackle the issue, we introduce a domain adaptation technique for optical flow estimation. Our method extracts diverse style statistics of the target domain and use them in training to generate synthetic features from the source features, which contain the contents of the source but the style of the target. We also impose motion consistency between the synthetic target and the source and deploy adversarial learning at the flow prediction to encourage domain-invariant features. Experimental results show that the proposed method achieves substantial and consistent improvements in different domain adaptation scenarios on VKITTI 2, Sintel, and KITTI 2015 benchmarks.

1. Introduction

Optical flow estimation is a fundamental computer vision task that aims to estimate per-pixel motion information across two consecutive frames [2, 16, 36]. The estimated flow information is useful for numerous motion-related problems such as action recognition [25, 39], object segmentation [8, 50], and object tracking [10, 34] in videos. While optical flows are supposed to be widely used in different environments, most previous studies [9, 22, 46, 47] have ignored the issue of a domain gap between train and test environments while mainly relying on the consistency of brightness and gradients of corresponding pixels [31, 53]. Most existing optical flow estimators thus suffer from a significant domain difference, *e.g.*, clean to foggy or rainy weather. In a practical sense, this is a critical issue since the existing flow-annotated datasets are limited, and the cost of collecting such a large-scale dataset on a new domain or environment is extremely high.

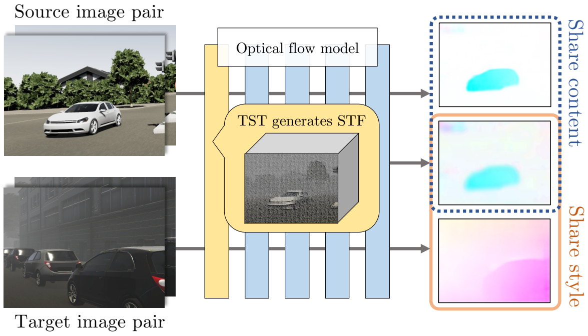

Figure 1. Optical flow domain adaptation. The proposed domain adaptation approach tackles the domain discrepancy issue in optical flow estimation. Our Target Style Transfer (TST) module in a model generates a synthetic target feature (STF), which contains the content of the source but the style of the target. The module enables the flow model to be easily adapted to the target domain by utilizing the properties of the STF. (Best viewed in color.)

There exists some previous work [30,31,53,60] that tackles the issue of domain shifts for optical flow estimation by designing an adaptive module based on the characteristics of a specific domain [30, 31] or training an optical flow model with synthesized data [53, 60]. Nonetheless, the aforementioned methods are only applicable to their fixed target domains. For example, [31] can only utilize their techniques to rainy images because they only consider the properties of rainy scenes, *i.e.*, the presence of rain streaks and veiling effects, being inapplicable to other domains such as fog or dark.

In this paper, we introduce a generic method for optical flow domain adaptation (OFDA), which is applicable to arbitrary domains of optical flow estimation. It aims to adapt an optical flow model to a target domain using a flow-annotated dataset from a source domain and an additional yet unannotated set of samples from the target domain. To achieve the goal, we propose three components: target style transfer (TST), motion consistency learning (MCL), and flow adversarial learning (FAL). The TST module extracts the style statistics of target images and uses them to generate synthetic target features from the source features, which preserve the content of the source but follow the style of the target. In particular, it leverages diverse local style statistics from the target, thus making the synthetic features to contain richer target styles; these synthetic features are used to adapt the given optical flow model to the target domain in training. Since the synthetic features preserve the content of the source, MCL encourages the computed motion from the source images to be close to that of the synthetic images and vice versa. FAL adversarially trains the optical flow model to be unable to distinguish output flows from the source and the target. All the proposed components are architecture-agnostic and thus applicable to different optical flow models. Our cross-domain experiments with Virtual KITTI 2 [5], FlyingThings3D [35], Sintel [4], and KITTI 2015 [13] datasets show that the proposed method provides a substantial gain in optical flow estimation. Our contributions are summarized as follows:

- We investigate the problem of Optical Flow Domain Adaptation (OFDA), which is applicable to arbitrary target domains of optical flows.
- We propose architecture-agnostic techniques for OFDA, which leverage style transfer (TST), motion consistency (MCL), and adversarial training (FAL).
- We demonstrate the substantial gain of the proposed method on cross-domain experiments with different optical flow datasets.

2. Related Work

Optical Flow. Optical flow estimation has been actively explored in computer vision [2, 9, 16, 36, 46, 47]. FlowNet [9] applies deep learning to the optical flow and proposes two distinct FlowNetS and FlowNetC. FlowNet 2 [22] brings both FlowNetS and FlowNetC and then stacks them to obtain more accurate flow maps. RAFT [47] is one of the state-of-the-art optical flow estimation models. Its recurrent update operator retrieves values from the correlation volumes and iteratively updates a flow field. The aforementioned optical flow methods achieve reliable results when input images satisfy the Brightness Constancy Constraint (BCC) and the Gradient Constancy Constraint (GCC) [31, 53]. However, they are not able to handle constraint-broken images such as rainy or foggy scenes.

Optical Flow under Adverse Weather. Recently, several optical flow methods attempt to obtain fine flow estimations under adverse weather [29–31, 38, 53, 54, 60, 61]. RobustNet [30] first suggests estimating flow fields from rainy scenes by utilizing a residue channel which is free from rain. Subsequently, RainFlow [31] generates streak- and veiling-invariant features to alleviate the properties of heavy rainy scenes: the rain streaks and the rain veiling effect. For foggy scenes, [53, 54] proposes to alternate supervised training using synthetic data and unsupervised training using real data with the hazeline loss, which is designed for tackling the fog domain. Also, [61] brings synthetic and real fog images to mitigate both the fog gap and style gap. To handle dark scenes, [60] injects a noise to the clean images, then uses the noisy images and the ground truth for training. These previous methods are all designed for the specific target domain; *e.g.*, RainFlow can train a model that only prevails in rainy scenes. As a result, they cannot fully leverage images from a non-target domain; *e.g.*, foggy scenes are not appropriate for RainFlow. To this end, we develop a generic method to estimate robust optical flows in arbitrary target domains based on DA techniques.

Domain Adaptation. To tackle a domain shift issue, the learner is provided unlabeled data from the target domain for training. DA shows its strength in various computer vision tasks such as classification [6, 11, 32, 52], semantic segmentation [3, 15, 28, 48, 49, 63], detection [7, 43, 45], and recently, action recognition [25, 39]. Although optical flow also suffers from the domain discrepancy issue, DA has not been mainly exploited to solve the problem. Among numerous methods, there are two main approaches to minimize the domain gap. The first well-known approach is to adopt adversarial learning [14]. First proposed by [11], there are several extensions [27, 32, 42, 52]. In semantic segmentation, [49] leverages a gradient reversal layer of [11] to obtain a domain-invariant segmentation map. The second approach is to control normalization. Adaptive Instance Normalization [17], which is one of the effective methods in style transfer, shows that the mean and standard deviation of the feature are related to the style of an image. Recently, other methods [24, 40, 57, 58, 62] stem from these kinds of normalization techniques of style transfer [17, 51] and use them for reducing the domain gap. Our TST module makes additional synthetic target features that contain contents of the source features and follow the style of the target features, thus making the network well-adapted to the target domain. In contrast to previous work, however, our method uses abundant local target statistics to create synthetic target features that properly follows the target distribution for optical flow adaptation.

Optical Flow Domain Adaptation. There are also several attempts to apply DA when estimating OF. In the medical field, [20, 21] attempt to apply DA with teacher-student networks to track patients' tissue motion. Meta-learning, which adapts weights of the pre-trained networks to the test domain, is also used for flow estimation [12, 37]. While recent methods for semi-supervised optical flow [23, 26] can handle various unlabeled data, they do not primarily consider the domain discrepancy issue. In contrast, we aim to handle more realistic and challenging situations; there exists a considerable domain gap between labeled and unlabeled data, *i.e.*, clear weather and adverse weather.

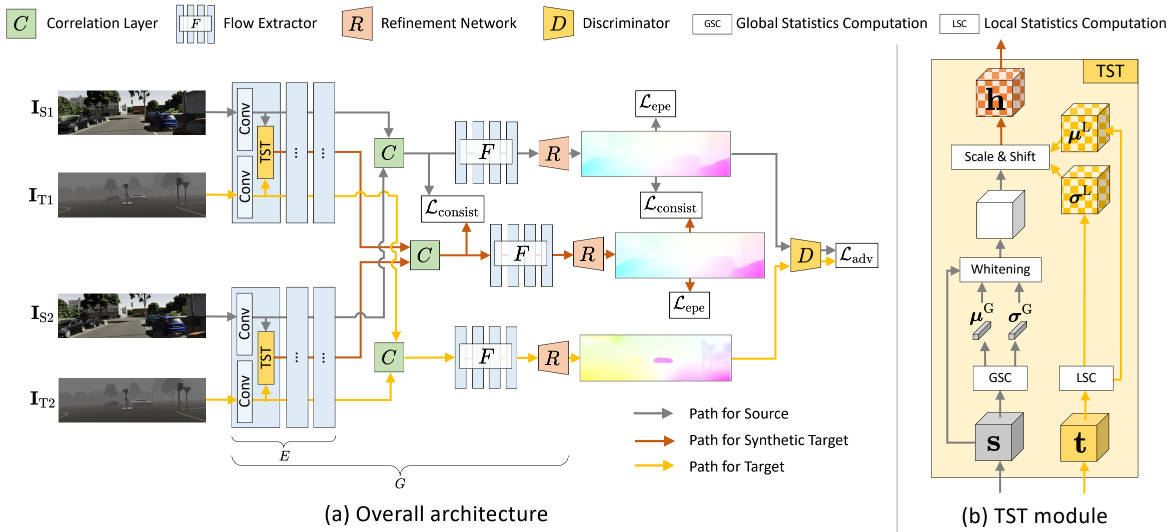
C Correlation Layer
F Flow Extractor
R Refinement Network
D Discriminator
GSC Global Statistics Computation
LSC Local Statistics Computation
$\mathbf{I}_{\mathrm{S}1}$
$\mathbf{I}_{\mathrm{T}1}$
$\mathbf{I}_{\mathrm{S}2}$
$\mathbf{I}_{\mathrm{T}2}$
Conv
TST
C
F
R
$\mathcal{L}_{\mathrm{epe}}$
$\mathcal{L}_{\mathrm{consist}}$
D
$\mathcal{L}_{\mathrm{adv}}$
E
G
Path for Source
Path for Synthetic Target
Path for Target
(a) Overall architecture
TST
h
Scale & Shift
μ^{L}
σ^{L}
Whitening
μ^{G}
σ^{G}
GSC
LSC
s
t
(b) TST module

Figure 2. (a) Overview of our proposed method. A base architecture is based on FlowNetC. $(\mathbf{I}_{\mathrm{S}1}, \mathbf{I}_{\mathrm{S}2})$ and $(\mathbf{I}_{\mathrm{T}1}, \mathbf{I}_{\mathrm{T}2})$ denote the source image pair and the target image pair respectively. TST modules are placed in the early part of the feature extractor E. The module takes both source and target features, thus making the synthetic target features that follow the target style with the contents of the source. The discriminator D is appended at the last part of the optical flow network G. The other letter C, F, and R sequentially represent the correlation layer, the flow extractor, the refinement network. The weights of the each network E, F, and R are shared respectively. (b) Overview of Target style transfer (TST) module

3. Method

Given a pair of consecutive images $\mathbf{I}_1$ and $\mathbf{I}_2$, the task of optical flow estimation aims to predict a displacement (u, v) for all of pixels in $\mathbf{I}_1$, where $\mathbf{I}_2(x', y') = \mathbf{I}_1(x+u, y+v)$. To estimate domain adaptive optical flow, we consider two datasets: a source dataset $\mathcal{D}_{\mathrm{S}} = \{(\mathbf{I}_{\mathrm{S}1}^{(i)}, \mathbf{I}_{\mathrm{S}2}^{(i)}, \mathbf{Y}_{\mathrm{S}}^{(i)})\}_{i=1}^{N_{\mathrm{S}}}$ and a target dataset $\mathcal{D}_{\mathrm{T}} = \{(\mathbf{I}_{\mathrm{T}1}^{(j)}, \mathbf{I}_{\mathrm{T}2}^{(j)})\}_{j=1}^{N_{\mathrm{T}}}$, where $\mathbf{I}$, $\mathbf{Y}$, and N denote an input image, its corresponding ground truth flow, and the number of image pairs, respectively. We suggest three main components to adapt the flow model G to the target domain: a target style transfer (TST) module, motion consistency learning (MCL), and a flow adversarial learning (FAL). The role of TST is to make the source feature pair to be close to the target pair. We place TST modules at the intervals of convolution layers from the early part of the feature extractor E, thus generating synthetic target features. Since the synthetic target feature pair should contain the same content with the source pair, we apply motion consistency loss between the computed motion of the synthetic pair and that of the source pair, specifically, correlation tensors and flow predictions. Moreover, we introduce the flow adversarial training to make two flow predictions from the source and the target domain-indistinguishable. Finally, the discriminator D is appended to the output of the network G.

In this work, we explain our architecture based on FlowNetC [9]. However, it is worth noting that our method is broadly applicable to any other existing optical flow networks such as FlowNetS [9] and RAFT [47].

3.1. Network Architecture

The whole network G consists of the feature extractor E, the correlation layer C, the flow extractor F, the refinement network R. The discriminator D is appended after the network G. Figure 2 illustrates our whole architecture.

Given two source-target pairs of two frames $(\mathbf{I}_{\mathrm{S}1}, \mathbf{I}_{\mathrm{T}1})$ and $(\mathbf{I}_{\mathrm{S}2}, \mathbf{I}_{\mathrm{T}2})$, the feature extractor E takes each $(\mathbf{I}_{\mathrm{S}}, \mathbf{I}_{\mathrm{T}})$ of the two as input and produces a triplet output of source feature $\mathbf{s}^{\mathrm{E}}$, target feature $\mathbf{t}^{\mathrm{E}}$, and synthetic target feature $\mathbf{h}^{\mathrm{E}}$. In the early part of the feature extractor E, we locate the TST module that takes two intermediate features, $\mathbf{s}$ and $\mathbf{t}$, from source and target and generates synthetic target feature $\mathbf{h}$, which resembles the style of the target image while preserving the content of the source image. All the features, $\mathbf{s}$, $\mathbf{t}$, and $\mathbf{h}$, are further updated via the subsequent convolution layers of the feature extractor E, finally resulting in $\mathbf{s}^{\mathrm{E}}$, $\mathbf{t}^{\mathrm{E}}$, and $\mathbf{h}^{\mathrm{E}}$. After obtaining two triplet output for two consecutive frames, the correlation layer C computes a correlation tensor for each of three feature pairs $(\mathbf{s}_1^{\mathrm{E}}, \mathbf{s}_2^{\mathrm{E}})$, $(\mathbf{t}_1^{\mathrm{E}}, \mathbf{t}_2^{\mathrm{E}})$, and $(\mathbf{h}_1^{\mathrm{E}}, \mathbf{h}_2^{\mathrm{E}})$, respectively. The computed correlation tensors are then forwarded through the flow extractor F and the refinement network R. They both gradually transform the correlation tensor to fine flow predictions with high resolution. The flow network G finally predicts the flow maps after the refinement. The flow maps from both the source pair and the target pair are forwarded to the discriminator

D. D is then trained to classify whether the flow map is from the source or from the target. The role of D is to make G generate domain indistinguishable flow map by reversing gradient at the backpropagation. Note that the TST module and the discriminator are only inserted during training.

3.2. Target Style Transfer Module

Our proposed Target Style Transfer (TST) module aims to make a synthetic target feature pair that follows the target style, thus allowing the network to be trained using the target-style source content with its flow labels. While the feature extractor provides two source-target pairs $(\mathbf{s}, \mathbf{t})$ as input, each of the pairs is fed to TST individually.

The TST module takes $\mathbf{s}$ and $\mathbf{t}$ as input and produces a synthetic target feature $\mathbf{h}$ as output; the size of all the features is $H \times W \times c$. Motivated by the fact that the feature statistics are closely related to the style of the image [17], the TST module is designed to make a synthetic target feature by whitening the source feature and then injecting the target statistics to the whitened source feature. To exploit diverse local styles from the target, we use local statistics in style injection, enabling the resultant synthetic features to contain richer target styles. Specifically, the c-dimensional vector of synthetic target feature $\mathbf{h}$ at spatial position (i, j) is computed as

$$\mathbf{h}_{ij} = \boldsymbol{\sigma}^{\mathrm{L}}_{ij}(\mathbf{t}) \frac{\mathbf{s}_{ij} - \boldsymbol{\mu}^{\mathrm{G}}(\mathbf{s})}{\boldsymbol{\sigma}^{\mathrm{G}}(\mathbf{s})} + \boldsymbol{\mu}^{\mathrm{L}}_{ij}(\mathbf{t}), \tag{1}$$

where $\boldsymbol{\mu}^{\mathrm{G}}(\mathbf{z}) \in \mathbb{R}^c$ and $\boldsymbol{\sigma}^{\mathrm{G}}(\mathbf{z}) \in \mathbb{R}^c$ denote the global statistics, *i.e*., the mean and the standard deviation, of feature $\mathbf{z}$ over its spatial dimension while $\boldsymbol{\mu}^{\mathrm{L}}_{ij}(\mathbf{z}) \in \mathbb{R}^c$ and $\boldsymbol{\sigma}^{\mathrm{L}}_{ij}(\mathbf{z}) \in \mathbb{R}^c$ stand for the local statistics of the feature $\mathbf{z}$. Note that all the operations above are done element-wise.

The global statistics, $\boldsymbol{\mu}^{\mathrm{G}}(\mathbf{z})$ and $\boldsymbol{\sigma}^{\mathrm{G}}(\mathbf{z})$, are obtained by

$$\boldsymbol{\mu}^{\mathrm{G}}(\mathbf{z}) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{z}_{hw}, \tag{2}$$

$$\boldsymbol{\sigma}^{\mathrm{G}}(\mathbf{z}) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (\mathbf{z}_{hw} - \boldsymbol{\mu}^{\mathrm{G}}(\mathbf{z}))^2}. \tag{3}$$

The local statistics, $\boldsymbol{\mu}^{\mathrm{L}}_{ij}(\mathbf{z})$ and $\boldsymbol{\sigma}^{\mathrm{L}}_{ij}(\mathbf{z})$, are obtained as follows. First, we compute the statistics for each position (i, j) using its local window:

$$\boldsymbol{\mu}^{\mathrm{L}}_{ij} = \frac{1}{P^2} \sum_{h=i-K}^{i+K} \sum_{w=j-K}^{j+K} \mathbf{z}_{hw}, \tag{4}$$

$$\boldsymbol{\sigma}^{\mathrm{L}}_{ij} = \sqrt{\frac{1}{P^2} \sum_{h=i-K}^{i+K} \sum_{w=j-K}^{j+K} (\mathbf{z}_{hw} - \boldsymbol{\mu}^{\mathrm{L}}_{ij})^2}, \tag{5}$$

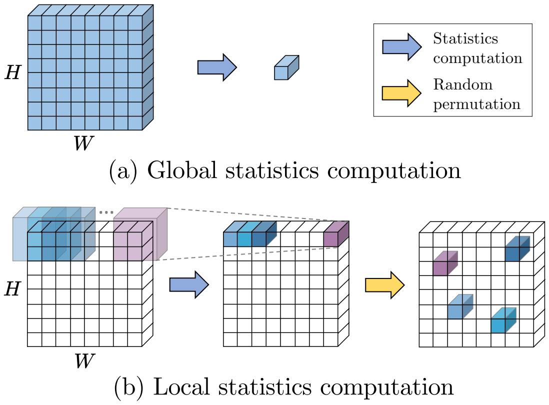
H
W
Statistics computation
Random permutation
(a) Global statistics computation
H
W
(b) Local statistics computation

Figure 3. Style statistics computation. (a) The global statistics of Eqs. 2 and 3 summarize all elements. (b) The local statistics of Eqs. 4 and 5 are computed for each position considering the elements in its local window, thus obtaining richer style information. Note that random permutation is applied to the resultant map of local statistics. All the statistics are computed channel-wise. This illustration uses an example with a single channel for simplicity.

where $P = 2K + 1$ is the size of the local window. Then, the calculated statistics are randomly permuted over spatial positions.

$$\boldsymbol{\mu}^{\mathrm{L}}(\mathbf{z}) \leftarrow \mathrm{PERM}(\boldsymbol{\mu}^{\mathrm{L}}(\mathbf{z})), \tag{6}$$

$$\boldsymbol{\sigma}^{\mathrm{L}}(\mathbf{z}) \leftarrow \mathrm{PERM}(\boldsymbol{\sigma}^{\mathrm{L}}(\mathbf{z})), \tag{7}$$

where PERM($\mathbf{z}$) operation randomly permutes $H \times W$ c-dimensional vectors over spatial positions. The random permutation operation offers implicit data augmentation for synthetic target features, thus encouraging the model to be more robust to the domain discrepancy. These permuted local statistics are used for TST of Eq. 1. Figure 3 illustrates the TST procedure in more detail.

As shown in Figure 2 and explained in Sec. 3.1, the synthetic target feature $\mathbf{h}$ is updated to $\mathbf{h}^{\mathrm{E}}$ through the subsequent convolutional layers of the feature extractor E and used to adapt the given optical flow model to the target domain in training. Note that, during the inference phase, the TST module is removed from the network and the network takes only two target images as input.

3.3. Motion Consistency Learning

The synthetic target features, which are generated by TST, follow the style of the target but retain the content of the source. To leverage this property, we impose the motion consistency on both the correlation tensor and the predicted flows in training so that the motion of the source feature pair becomes similar to that of the synthetic target feature pair, and vice versa. The motion consistency loss is thus formulated as

$$\mathcal{L}_{\text{consist}} = \|\mathbf{M}_{\mathrm{S}} - \mathbf{M}_{\mathrm{H}}\|_2, \tag{8}$$

where $\mathbf{M}_{\mathrm{S}}$ is the motion information (*i.e.*, correlation tensor or flow prediction) from the source pair and $\mathbf{M}_{\mathrm{H}}$ is that from the synthetic target pair. We will denote the consistency loss of the correlation tensor as $\mathcal{L}_{\text{consist}}^{\text{corr}}$ and that of the predicted flow as $\mathcal{L}_{\text{consist}}^{\text{pred}}$.

3.4. Flow Adversarial Learning

Along with the TST module, we append a discriminator at the end of the network to alleviate the domain gap between the source and the target pairs. As mentioned in [49], unlike classification task, optical flow is a task of predicting pixel-level flows. Therefore, we choose flow predictions as inputs of the discriminator. The role of the discriminator is to classify the domain of the flow maps. The loss to train the discriminator is calculated as

$$\mathcal{L}_{\text{dis}} = -\{(1-y)\log(1-D(G(\mathbf{I_1},\mathbf{I_2}))) + y\log D(G(\mathbf{I_1},\mathbf{I_2}))\}, \tag{9}$$

where $y=0$ if the prediction $G(\mathbf{I}_1,\mathbf{I}_2)$ is from the source domain, and $y=1$ if $G(\mathbf{I}_1,\mathbf{I}_2)$ is from the target domain. The discriminator D outputs the probability that the prediction flow is from the target domain. The discriminator D is trained by the loss $\mathcal{L}_{\text{dis}}$ and is only used in the training phase. The flow network G is trained for generating domain-invariant flow maps to fool the discriminator. The loss for G can be written as:

$$\mathcal{L}_{\text{adv}} = -\mathcal{L}_{\text{dis}}. \tag{10}$$

3.5. Overall Loss

Since flow annotations are available for the source dataset $\mathcal{D}_{\mathrm{S}}$, supervised training is conducted for the source. We also deploy the source annotations for the synthetic pair as the synthetic target pair preserves the identical motion with the source pair. We use the end-point-error (EPE) loss, which is the Euclidean distance between the predicted flow map and the ground truth. The EPE loss is computed as:

$$\mathcal{L}_{\text{epe}} = \|G(\mathbf{I}_{\mathrm{S}1},\mathbf{I}_{\mathrm{S}2}) - \mathbf{Y}_{\mathrm{S}}\|_2, \tag{11}$$

where $G(\mathbf{I}_{\mathrm{S}1},\mathbf{I}_{\mathrm{S}2})$ is optical flow prediction of a pair of source images and $\mathbf{Y}_{\mathrm{S}}$ is a corresponding ground-truth flow. Then, the EPE loss for the source will be $\mathcal{L}_{\text{epe}}^{\mathrm{S}}$ and the one for the synthetic will be $\mathcal{L}_{\text{epe}}^{\mathrm{H}}$. As a result, the overall loss for the flow network G is as follows:

$$\mathcal{L}_G = \mathcal{L}_{\text{epe}}^{\mathrm{S}} + \mathcal{L}_{\text{epe}}^{\mathrm{H}} + \lambda_{\text{cons}}\mathcal{L}_{\text{consist}}^{\text{corr}} + \lambda_{\text{cons}}\mathcal{L}_{\text{consist}}^{\text{pred}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}, \tag{12}$$

where λ_{cons} and λ_{adv} are the hyper-parameters for weighting the consistency loss and the flow adversarial loss.

4. Experiments

We introduce three experimental settings to show the transferability of our model. First, we use Virtual KITTI 2 (VKITTI 2) [5] dataset, which contains synthetic driving scenes and consists of six weather conditions: Clone, Fog, Morning, Overcast, Rain, and Sunset. In the experiments, we select Clone as a source domain and the rest of the weather as target domains; thus, we conduct experiments on five domain scenarios. In other experiments, Sintel [4], the popular optical flow benchmark dataset, is selected as the target domain. Lastly, KITTI 2015 [13], which includes 3D scene flows of real-world driving scenes, is utilized for the target domain. In both Sintel and KITTI experiments, we bring the network pretrained on FlyingChairs [1] and FlyingThings3D [35]. Also, FlyingThings3D is selected as the source domain in both experiments. In all settings, our goal is to properly adapt the network to the target domain.

Implementation Details. As we mentioned in Section 1, our proposed method is applicable to existing optical flow methods. In this work, we apply our proposed method to FlowNetS [9], FlowNetC [9] and RAFT [47]. Unless otherwise noted, we bring the identical experimental settings of baselines for training our model. We add the TST module after the first convolution layer that is located in the early part of the network. We set up multiple patch sizes and randomly select one of them at every step in the TST module. The source and synthetic target's motion tensors are extracted after the correlation layer for FlowNetC and RAFT. We consider features before the refinement network as the motion tensors for FlowNetS. The discriminator, which is appended after the optical flow estimation, contains four convolution layers and a fully-connected layer. We set batch sizes to 8 and 4 for FlowNet and RAFT, respectively. We use PyTorch [41] for implementation. We refer the readers to the supplementary material for more details.

Baselines. We choose Instance Normalization (IN) [51], which is known to alleviate domain difference and only extract contents from image features, as a baseline. Furthermore, we investigate the effect of the adversarial learning by appending a discriminator at the end of the baseline network. FlowNetS + IN and FlowNetC + IN are the FlowNetS and FlowNetC model, which contain IN operation at the early three convolution layers. Note that RAFT+IN model does not exist because RAFT already contains IN in the feature encoder. The model with + AT represents that it is trained with adversarial learning. GST stands for global synthetic target, and the statistics computation in our TST module is replaced with the global statistics computation. In Sintel and KITTI experiments, we bring several prior work [18, 19, 22, 46, 47, 55, 56, 59].

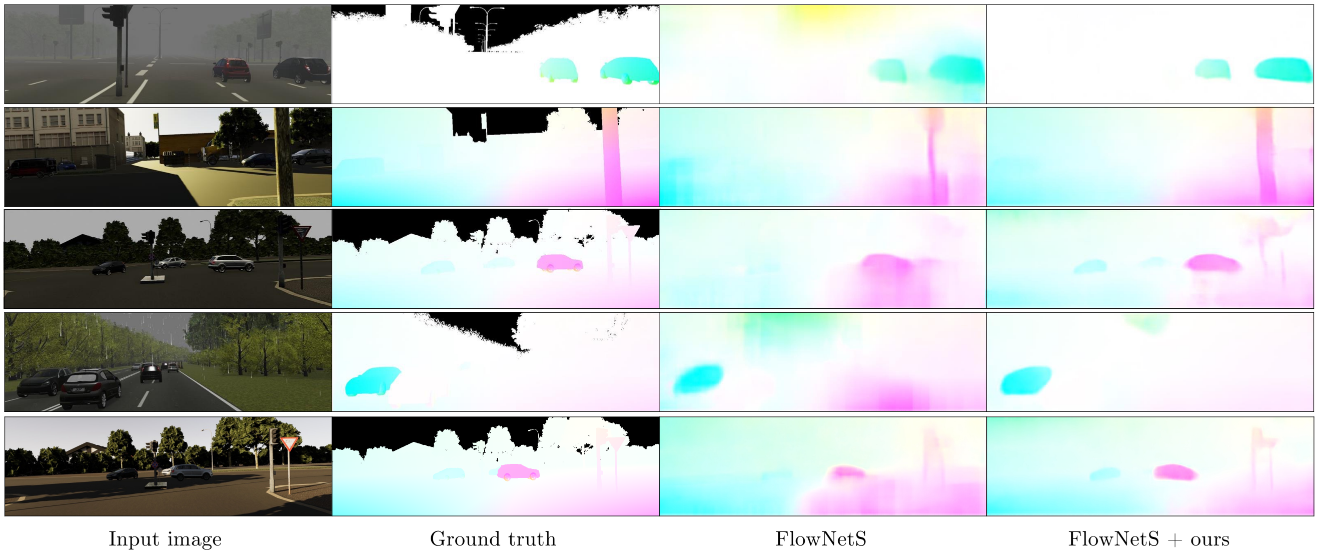

Figure 4. Qualitative results of VKITTI dataset.

Table 1. Average End Point Error (AEPE) results of domain scenarios in VKITTI dataset. We set Clone as the source domain for all experiments (see the first row of the table). Target domains are denoted on the second row of the table. + ours denote the full implementation of our proposed model.

Method	Clone					Average
	Fog	Morning	Overcast	Rain	Sunset	
FlowNetS [9]	14.85	6.43	6.80	12.46	6.19	9.34
FlowNetS + IN	7.91	5.40	5.15	9.16	5.15	6.55
FlowNetS + AT	5.42	5.09	4.52	5.65	4.75	5.09
FlowNetS + GST	5.46	**4.40**	**4.10**	5.00	**4.16**	4.62
FlowNetS + ours	**4.55**	4.44	4.30	**4.96**	4.41	**4.53**
FlowNetC [9]	48.53	6.32	9.07	15.18	6.58	17.14
FlowNetC + IN	14.23	6.47	6.88	8.67	7.05	8.64
FlowNetC + AT	11.95	5.51	6.25	9.25	5.83	7.76
FlowNetC + GST	6.13	**4.32**	4.67	6.14	4.50	5.15
FlowNetC + ours	**4.91**	4.37	**4.49**	**5.39**	**4.28**	**4.69**
RAFT [47]	2.01	1.35	1.07	2.21	1.02	1.53
RAFT + FS [23]	1.73	1.51	1.52	1.63	1.44	1.57
RAFT + AT	1.91	1.35	1.07	2.24	1.01	1.51
RAFT + GST	1.64	1.26	1.10	1.82	0.99	1.36
RAFT + ours	**1.48**	**0.99**	**0.84**	**1.46**	**0.82**	**1.12**

Table 2. Average End Point Error (AEPE) and F1 results on Sintel and KITTI-15 datasets. We only measure AEPE for the results of Sintel column following RAFT. (†FlowNet2 reported results on the disparity split of Sintel, 3.54 is the EPE when their model is evaluated on the standard data [18].)

Method	FlyingThings3D			
	Sintel		KITTI-15	
	Clean	Final	AEPE	Fl-all
HD3 [56]	3.84	8.77	13.17	24.0
LiteFlowNet [18]	2.48	4.04	10.39	28.5
PWC-Net [46]	2.55	3.93	10.35	33.7
LiteFlowNet2 [19]	2.24	3.78	8.97	25.9
VCN [55]	2.21	3.68	8.36	25.1
MaskFlowNet [59]	2.25	3.61	-	23.1
FlowNet2 [22]	2.02	$3.54^\dagger$	10.08	30.0
RAFT [47]	1.43	2.71	5.04	17.4
RAFT + FS[1] [23]	**1.30**	**2.46**	4.69	**14.5**
RAFT + GST	1.36	2.58	4.67	17.2
RAFT + ours	1.32	2.57	**4.45**	16.4

4.1. VKITTI 2

As shown in Table 1, existing methods, especially, FlowNetS and FlowNetC, are incapable of dealing with domain changes. The performance of FlowNetS and FlowNetC severely drops, especially when Fog and Rain are the target domains. The two domains are very different from Clone domain as they are both low-light weather. The result indicates that the bigger the gap between the source and the target is, the more vulnerable the networks without domain adaptive techniques. As we can see in the table, + IN and + AT model help to make domain-invariant feature, and thus improving the model performance in the target domain. Moreover, + GST model assists the network to be more adapted to the target domain by generating global synthetic target features. When we apply our method (TST, MCL, and FAL), the network outperforms other baselines. Especially the method shows its efficacy even when the discrepancy between the source and target is severe. Although + FS model is effective in Fog and Rain domains, it degrades the performance of RAFT in all the other domains, indicating that the benefit of the mutual supervision strategy of FS is sensitive to target domains.

Figure 4 shows the qualitative results of the VKITTI 2 dataset. The domain of the input target images are Fog, Morning, Overcast, Rain, and Sunset from top to bottom. The results indicate that the FlowNetS fails to get a reliable optical flow map in the target domains, which have a large domain gap with the source domain (Clone). On the other hand, the flow map results of our method show clear improvements.

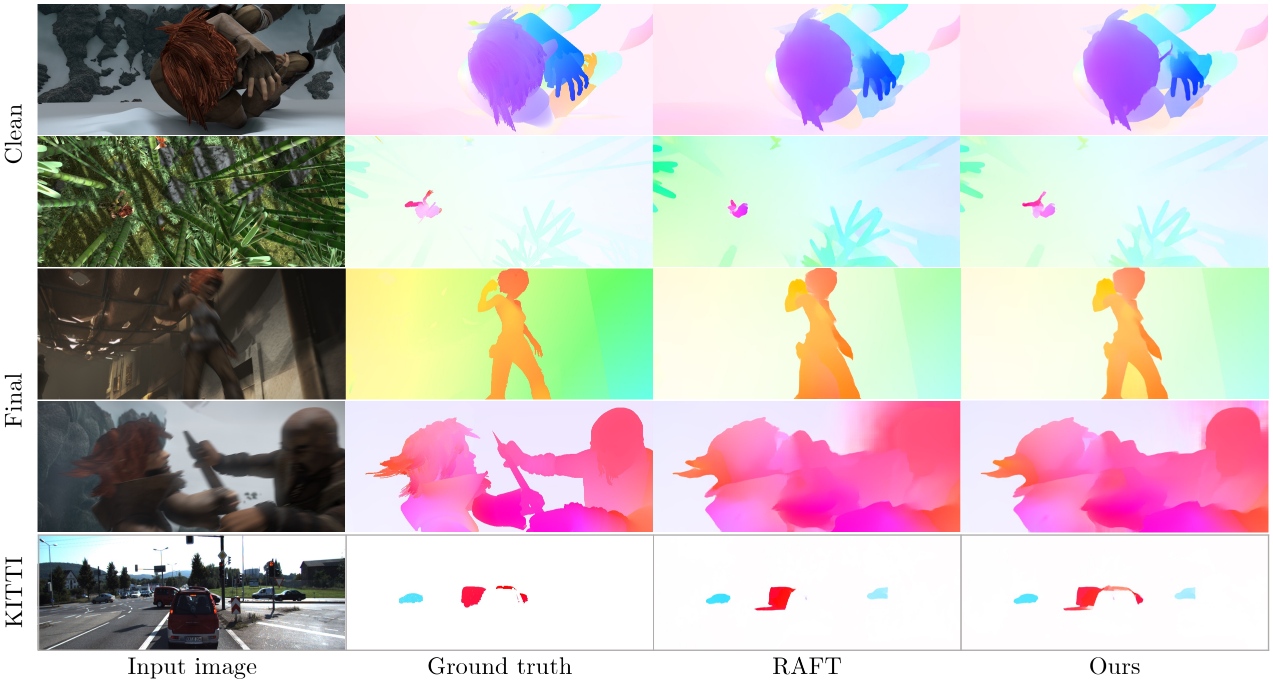

Figure 5. Qualitative results of Sintel and KITTI dataset. The images in the top two rows are from Sintel (clean) and the following two rows are from Sintel (final). The images at the bottom row are from KITTI.

4.2. Sintel

To evaluate the transferability of our model between two datasets, we select FlyingThings3D as a source dataset and Sintel as a target dataset. We use the official RAFT model for our training, which is pretrained on FlyingThings3D. Our quantitative results are denoted in Table 2. It shows that our method successfully adapts the model to the target dataset. Our method achieves 1.32 AEPE on Clean split and 2.57 AEPE on Final split of Sintel (train). The results describe an interesting fact that our method not only can handle the domain difference between the source and the target but also can deal with the different motion distributions between them compared to VKITTI experiments. However, FS shows marginally better results and we assume the reason is that it fully exploits its teacher network, which generates guidance for the student network, based on their pretrained model. Although FS demonstrates its effectiveness, it heavily relies on the prediction of the optical flow model, which is pretrained, as supervision in learning; FS shows unstable performances in the VKITTI experiment, where the model is trained from scratch. In contrast, the supervisory signal in our method mainly comes from source annotations, thus being robust to the prediction of the optical flow model. We demonstrate our model's capability without pretrained model by experiments in supplementary. As shown in Figure 5, our method is capable of predicting fine flow fields. In the second row, our method successfully catches the movement of a left arm and a left leg of a person compared with RAFT. In the third row, our method clearly distinguishes the movement of the legs. Additionally, in the fourth row, the method clearly detects the movement of a right side man's head.

4.3. KITTI 2015

Additionally, we choose FlyingThings3D as a source dataset and KITTI 2015 as a target dataset. Similar to the Sintel experiments, we utilize FlyingThings3D pretrained model. The difference with the Sintel experiments is that KITTI is a real-world dataset. Therefore, this experiment shows our models' transferability from synthetic to real domain. The results of Table 2 show that our proposed model is capable of dealing with the difference between synthetic and real domains. As shown in the bottom row of Figure 5, the result indicates that our method is also capable of dealing with occluded objects.

4.4. Ablation Study

We conduct multiple ablation studies on the VKITTI dataset to validate the effectiveness of the proposed method. We use FlowNetC [9] and set Clone as the source domain.

Proposed Components. In Table 3, we conduct an extensive ablation study to validate the effectiveness of each component by incrementally adding a component to the baseline. The first row of the table is the result of FlowNetC. By only attaching the TST module, and thus generating syn-

[1] We report the performance of FS without SMURF [44], which is an additional training strategy using unlabeled images, for a fair comparison.

Table 3. Comprehensive ablation study.

$\mathcal{L}_{\text{epe}}^{\text{ST}}$	$\mathcal{L}_{\text{consist}}^{\text{corr}}$	$\mathcal{L}_{\text{consist}}^{\text{pred}}$	$\mathcal{L}_{\text{adv}}$	Fog	Rain	Sunset	Avg.
				48.53	15.18	6.58	23.43
✓				6.06	6.91	4.41	5.79
✓	✓			5.36	5.53	4.24	5.04
✓		✓		4.95	5.60	4.22	4.93
✓	✓	✓		5.20	5.43	4.16	4.93
✓	✓	✓	✓	4.91	5.39	4.28	4.86

Table 4. The effect of the permutation.

Permutation	Fog	Rain	Average
	5.95	5.41	5.68
✓	**4.91**	**5.39**	**5.15**

Table 5. Quantitative results of multiple patch sizes.

Patch size	Fog	Morning	Average
5	5.32	4.52	4.92
11	5.18	4.52	4.85
21	5.08	4.52	4.80
$\{5, 11, 21\}$	**4.91**	**4.37**	**4.64**

thetic target feature for training, the model obtains significant performance improvement. Also, each motion consistency loss ($\mathcal{L}_{\text{consist}}^{\text{corr}}$ and $\mathcal{L}_{\text{consist}}^{\text{pred}}$) also contribute to the model's performance. Adversarial loss ($\mathcal{L}_{\text{adv}}$) shows its efficacy as well. Finally, the last row of the table is the full implementation of our proposed model. The result shows that our components complement each other and effectively train the model to adapt to the target domain. The results of other domains are reported in the supplementary.

Effects of random permutations. The random permutation, after obtaining local target style statistics, has an effect of implicit data augmentation. To demonstrate its efficacy, we conduct an ablation study in Table 4. The results indicate that random permutation meaningfully benefits the training of the model.

Multiple Patch Size. We select three kinds of patch sizes from $\frac{1}{32}$, $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, and $\frac{1}{2}$ of the feature height. In VKITTI 2 experiments, the feature height is 160. For the ablation study, we select the patch size of 5, 11, and 21, *i.e.*, $\frac{1}{32}$, $\frac{1}{16}$, and $\frac{1}{8}$ of the feature height respectively. Table 5 indicates that utilizing multiple patch sizes are more beneficial than using only one size. Other patches' size settings are in the supplementary materials.

4.5. Visualization of Synthetic Target

Feature visualization. To demonstrate that our synthetic target feature follows the content of the source, we visualize the source and synthetic target feature in Figure 6. For visualization, we average the feature across the channel dimension. The results show that the created feature properly contains source contents.

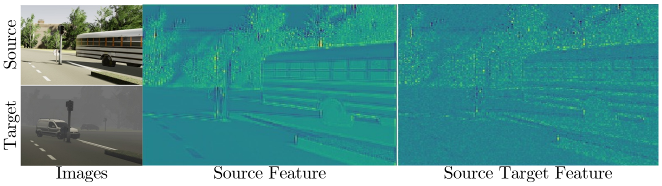

Figure 6. Feature visualizations of the source and the synthetic target.

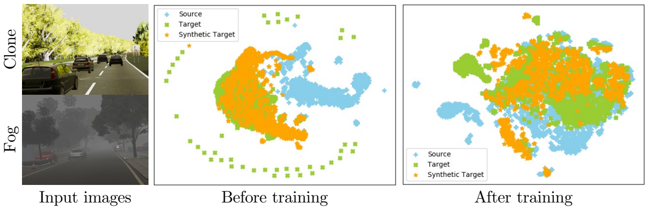

Figure 7. t-SNE visualization of the source (Clone), target (Fog), and synthetic target features.

***t*-SNE visualization.** To investigate that synthetic target features follow the style of the target and contribute to reducing domain gap, we visualize the embedding space of the source, target and synthetic target features using t-SNE [33] in Figure 7. The points in the embedding space represent 256-dim pixel embeddings from $40 \times 56 \times 256$ feature maps before the correlation layer. Before training, the embeddings of the synthetic target (yellow) are closer to the points of the target (green), which represents that the synthetic feature successfully capture the target style. After training, the model efficiently extracts domain-invariant features after being trained with our method.

5. Conclusion

Optical flow estimation on a novel domain is a challenging and critical issue. We have proposed the target style transfer (TST) module for creating synthetic target feature, which effectively assists to reduce domain discrepancy. Motion consistency loss (MCL) enforces the computed motion from the source feature to be close to that of the synthetic feature and vice versa. We also deploy adversarial training for flow adversarial learning (FAL). Moreover, our components can be applied to numerous other optical flow models. We conduct extensive experiments on various datasets and show that our model predicts fine flow maps from the target domain. We believe that our method can be broadly deployed to other computer vision tasks.

Acknowledgements. This work was supported by Samsung Electronics Co., Ltd. (IO201208-07822-01) and the IITP grants (2022-0-00290: Visual Intelligence for Space-time Understanding and Generation based on Multi-Layered Visual Common Sense (50%), 2022-0-00113: Sustainable Collaborative Multi-modal Lifelong Learning (50%)) funded by Ministry of Science and ICT, Korea.

References

[1] Mathieu Aubry, Daniel Maturana, Alexei A Efros, Bryan C Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3762–3769, 2014. 5

[2] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 1, 2

[3] David Brüggemann, Christos Sakaridis, Prune Truong, and Luc Van Gool. Refign: Align and refine for adaptation of semantic segmentation to adverse conditions. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3174–3184, 2023. 2

[4] 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, 5

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 2, 5

[6] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7354–7362, 2019. 2

[7] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2

[8] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017. 1

[9] 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. 1, 2, 3, 5, 6, 7

[10] Bo Du, Shihan Cai, and Chen Wu. Object tracking in satellite videos based on a multiframe optical flow tracker. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):3043–3055, 2019. 1

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[12] Zhiyi Gao, Yonghong Hou, Yan Liu, and Xiangyu Li. Metaflow: a meta-learning-based network for optical flow estimation. *Journal of Electronic Imaging*, 30(3):033029–033029, 2021. 2

[13] 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

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998, 2018. 2

[16] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1, 2

[17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017. 2, 4

[18] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 5, 6

[19] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn–revisiting data fidelity and regularization. *arXiv preprint arXiv:1903.07414*, 2019. 5, 6

[20] Sontje Ihler, Felix Kuhnke, Max-Heinrich Laves, and Tobias Ortmaier. Self-supervised domain adaptation for patient-specific, real-time tissue tracking. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part III 23*, pages 54–64. Springer, 2020. 2

[21] Sontje Ihler, Max-Heinrich Laves, and Tobias Ortmaier. Patient-specific domain adaptation for fast optical flow based on teacher-student knowledge transfer. *arXiv preprint arXiv:2007.04928*, 2020. 2

[22] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1, 2, 5, 6

[23] 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

[24] Juwon Kang, Sohyun Lee, Namyup Kim, and Suha Kwak. Style neophile: Constantly seeking novel styles for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7130–7140, 2022. 2

[25] Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. Learning cross-modal contrastive features for video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13618–13627, 2021. 1, 2

[26] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. *Advances in neural information processing systems*, 30, 2017. 2

[27] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019. 2

[28] Sohyun Lee, Taeyoung Son, and Suha Kwak. Fifo: Learning fog-invariant features for foggy scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18911–18921, 2022. 2

[29] Haipeng Li, Kunming Luo, and Shuaicheng Liu. Gyroflow: gyroscope-guided unsupervised optical flow learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12869–12878, 2021. 2

[30] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. Robust optical flow in rainy scenes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 288–304, 2018. 1, 2

[31] 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. 1, 2

[32] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018. 2

[33] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[34] Yasushi Mae, Yoshiaki Shirai, Jun Miura, and Yoshinori Kuno. Object tracking in cluttered background based on optical flow and edges. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 1, pages 196–200. IEEE, 1996. 1

[35] 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, 5

[36] Etienne Mémin and Patrick Pérez. Dense estimation and object-based segmentation of the optical flow with robust techniques. *IEEE Transactions on Image Processing*, 7(5):703–719, 1998. 1, 2

[37] Chaerin Min, Taehyun Kim, and Jongwoo Lim. Meta-learning for adaptation of deep optical flow networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2145–2154, 2023. 2

[38] József Molnár, Dmitry Chetverikov, and Sándor Fazekas. Illumination-robust variational optical flow using cross-correlation. *Computer Vision and Image Understanding*, 114(10):1104–1114, 2010. 2

[39] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 122–132, 2020. 1, 2

[40] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8690–8699, 2021. 2

[41] 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. *Advances in neural information processing systems*, 32:8026–8037, 2019. 5

[42] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. *arXiv preprint arXiv:1711.01575*, 2017. 2

[43] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 2

[44] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2021. 7

[45] Peng Su, Kun Wang, Xingyu Zeng, Shixiang Tang, Dapeng Chen, Di Qiu, and Xiaogang Wang. Adapting object detectors with conditional domain normalization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 403–419. Springer, 2020. 2

[46] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 2, 5, 6

[47] 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. 1, 2, 3, 5, 6

[48] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 129:341–360, 2021. 2

[49] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7472–7481, 2018. 2, 5

[50] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3899–3908, 2016. 1

[51] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 2, 5

[52] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. *arXiv preprint arXiv:1912.01805*, 2019. 2

[53] 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. 1, 2

[54] Wending Yan, Aashish Sharma, and Robby T Tan. Optical flow estimation in dense foggy scenes with domain-adaptive networks. *IEEE Transactions on Artificial Intelligence*, 2022. 2

[55] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Advances in Neural Information Processing Systems*, pages 793–803, 2019. 5, 6

[56] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6044–6053, 2019. 5, 6

[57] Jeongbeen Yoon, Dahyun Kang, and Minsu Cho. Semi-supervised domain adaptation via sample-to-sample self-distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1978–1987, 2022. 2

[58] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact feature distribution matching for arbitrary style transfer and domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8035–8045, 2022. 2

[59] Shengyu Zhao, Yilun Sheng, Yue Dong, Eric I Chang, Yan Xu, et al. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6278–6287, 2020. 5, 6

[60] Yinqiang Zheng, Mingfang Zhang, and Feng Lu. Optical flow in the dark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6749–6757, 2020. 1, 2

[61] Hanyu Zhou, Yi Chang, Wending Yan, and Luxin Yan. Unsupervised cumulative domain adaptation for foggy scene optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9569–9578, 2023. 2

[62] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 2

[63] Qianyu Zhou, Qiqi Gu, Jiangmiao Pang, Xuequan Lu, and Lizhuang Ma. Self-adversarial disentangling for specific domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2