DistractFlow: Improving Optical Flow Estimation via Realistic Distractions and Pseudo-Labeling

Jisoo Jeong Hong Cai Risheek Garrepalli Fatih Porikli
Qualcomm AI Research†
{jisojeon, hongcai, rgarrepa, fporikli}@qti.qualcomm.com

Abstract

We propose a novel data augmentation approach, DistractFlow, for training optical flow estimation models by introducing realistic distractions to the input frames. Based on a mixing ratio, we combine one of the frames in the pair with a distractor image depicting a similar domain, which allows for inducing visual perturbations congruent with natural objects and scenes. We refer to such pairs as distracted pairs. Our intuition is that using semantically meaningful distractors enables the model to learn related variations and attain robustness against challenging deviations, compared to conventional augmentation schemes focusing only on low-level aspects and modifications. More specifically, in addition to the supervised loss computed between the estimated flow for the original pair and its ground-truth flow, we include a second supervised loss defined between the distracted pair's flow and the original pair's ground-truth flow, weighted with the same mixing ratio. Furthermore, when unlabeled data is available, we extend our augmentation approach to self-supervised settings through pseudo-labeling and cross-consistency regularization. Given an original pair and its distracted version, we enforce the estimated flow on the distracted pair to agree with the flow of the original pair. Our approach allows increasing the number of available training pairs significantly without requiring additional annotations. It is agnostic to the model architecture and can be applied to training any optical flow estimation models. Our extensive evaluations on multiple benchmarks, including Sintel, KITTI, and SlowFlow, show that DistractFlow improves existing models consistently, outperforming the latest state of the art.

1. Introduction

Recent years have seen significant progress in optical flow estimation thanks to the development of deep learning, e.g., [4,7,8,23]. Among the latest works, many focus on developing novel neural network architectures, such as PWC-Net [29], RAFT [30], and FlowFormer [6]. Other studies investigate how to improve different aspects of supervised training [27], e.g., gradient clipping, learning rate, and training compute load. More related to our paper are those incorporating data augmentation during training (e.g., [30]), including color jittering, random occlusion, cropping, and flipping. While these image manipulations can effectively expand the training data and enhance the robustness of the neural models, they fixate on the low-level aspects of the images.

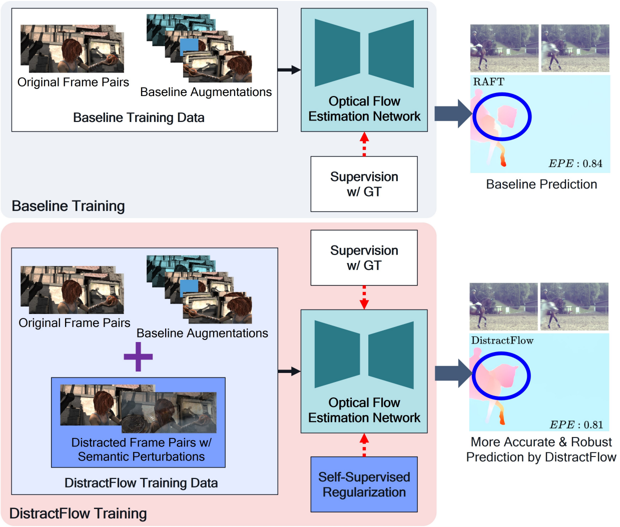

Figure 1. Existing augmentation schemes apply low-level visual modifications, such as color jittering, block-wise random occlusion, and flipping, to augment the training data (top), while DistractFlow introduces high-level semantic perturbations to the frames (bottom). DistractFlow can further leverage unlabeled data to generate a self-supervised regularization. Our training leads to more accurate and robust optical flow estimation models, especially in challenging real-world settings.

Since obtaining ground truth optical flow on real data is very challenging, another line of work investigates how to leverage unlabeled data. To this end, semi-supervised methods [9, 12] that utilize frame pairs with ground-truth flow annotations in conjunction with unlabeled data in training have been proposed. For instance, FlowSupervisor [9] adopts a teacher-student distillation approach to exploit unlabeled data. This method, however, does not consider lo-

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

calized uncertainty but computes the loss for the entire image between the teacher and student network.

In this paper, we present a novel approach, DistractFlow, which performs semantically meaningful data augmentations by introducing images of real objects and natural scenes as distractors or perturbations to training frame pairs. More specifically, given a pair of consecutive frames, we combine the second frame with a random image depicting similar scenarios based on a mixing ratio. In this way, related objects and scenes are overlaid on top of the original second frame; see Figure 1 for an example. As a result, we obtain challenging yet appropriate distractions for the optical flow estimation model that seeks dense correspondences from the first frame to the second frame (and in reverse too). The original first frame and the composite second frame constitute a *distracted* pair of frames, which we use as an additional data sample in both supervised and self-supervised training settings. Unlike our approach, existing data augmentation schemes for optical flow training apply only low-level variations such as contrast changes, geometric manipulations, random blocks, haze, motion blur, and simple noise and shapes insertions [28, 30]. While such augmentations can still lead to performance improvements, they are disconnected from natural variations, scene context, and semantics. As we shall see in our experimental validation, the use of realistic distractions in training can provide a bigger boost to performance.

Figure 1 provides a high-level outline of DistractFlow. We apply DistractFlow in supervised learning settings using the ground-truth flow of the original pair. Distracted pairs contribute to the backpropagated loss proportional to the mixing ratios used in their construction. Additionally, when unlabeled frame pairs are available, DistractFlow allows us to impose a self-supervised regularization by further leveraging pseudo-labeling. Given an unlabeled pair of frames, we create a distracted version. Then, we enforce the estimated flow on the distracted pair to match that on the original pair. In other words, the prediction of the original pair is treated as a pseudo ground truth flow for the distracted pair. Since the estimation on the original pair can be erroneous, we further derive and impose a confidence map to employ only highly confident pixel-wise flow estimations as the pseudo ground truth. This prevents the model from reinforcing incorrect predictions, leading to a more stable training process.

In summary, our main contributions are as follows:

- We introduce DistractFlow, a novel data augmentation approach that improves optical flow estimation by utilizing distractions from natural images. Our method provides augmentations with realistic semantic contents compared to existing augmentation schemes.
- We present a semi-supervised learning scheme for optical flow estimation that adopts the proposed distracted pairs to leverage unlabeled data. We compute a confidence map to generate uncertainty-aware pseudo labels and to enhance training stability and overall performance.
- We demonstrate the effectiveness of DistractFlow in supervised [6, 14, 30] and semi-supervised settings, showing that DistractFlow outperforms the very recent FlowSupervisor [9] that require additional in-domain unlabeled data.

2. Related Work

Optical Flow Estimation: Several deep architectures have been proposed for optical flow [4, 8, 23, 29, 30, 38]. Among these, Recurrent All Pairs Field Transforms (RAFT) [30] have shown significant performance improvement over previous methods, inspiring many subsequent works [6, 14, 26, 27, 35]. Following the structure of RAFT architecture, complementary studies [12, 14, 33, 35, 39] proposed advancements on feature extraction, 4D correlation volume, recurrent update blocks, and more recently, transformer extensions [6,39]. In DistractFlow, we introduce a new model-agnostic training method that can help any model.

Data Augmentation: Data augmentation is a widely used technique to better train deep learning models. Common augmentations include color and contrast jittering, flipping, geometric manipulation, and random noise. While they can improve the robustness of the model, these operations mainly focus on the low-level visual aspects and do not account for variations in the semantic contents.

Recently, several data augmentation schemes, such as adversarial perturbation and regularization methods, have been proposed for classification tasks. Adversarial perturbation [32] is one of the well-known augmentation methods for classification tasks, but recent work [24] shows that it does not work well for optical flow estimation. Interpolation-based Regularization (IR) [34, 36], which mixes a couple of images and trains the model with mixed label, improves the performance in classification tasks and is employed in other fields such as object detection [13, 37] and segmentation [10]. In a regression problem, however, mixed ground truth does not correspond with mixed input. Fixmatch [25] as a data augmentation scheme has demonstrated state-of-the-art performance in classification tasks. It combines pseudo-labeling and consistency regularization by applying two different augmentations (weak, strong) to the same image and generates pseudo-labels with weak augmentation output. For classification tasks, it is possible to create pseudo-labels since the output and ground-truth annotations represent class probabilities. However, there are no such class probabilities for optical flow. Thus, these methods cannot be readily applicable.

AutoFlow [28] proposed a new dataset for optical flow, taking a versatile approach to data rendering, where motion,

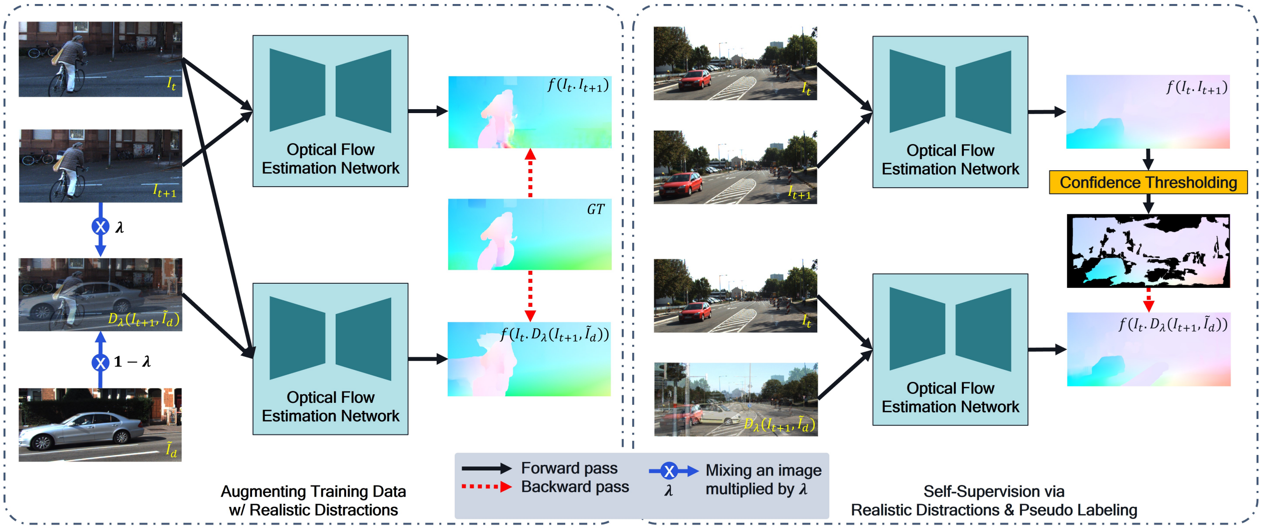
I_t
I_{t+1}
λ
$D_\lambda(I_{t+1}, \tilde{I}_d)$
$1-\lambda$
$\tilde{I}_d$
Optical Flow Estimation Network
Optical Flow Estimation Network
$f(I_t, I_{t+1})$
GT
$f(I_t, D_\lambda(I_{t+1}, \tilde{I}_d))$
Augmenting Training Data w/ Realistic Distractions
Forward pass
Backward pass
λ
Mixing an image multiplied by λ
I_t
I_{t+1}
I_t
$D_\lambda(I_{t+1}, \tilde{I}_d)$
Optical Flow Estimation Network
Optical Flow Estimation Network
$f(I_t, I_{t+1})$
Confidence Thresholding
$f(I_t, D_\lambda(I_{t+1}, \tilde{I}_d))$
Self-Supervision via Realistic Distractions & Pseudo Labeling

Figure 2. Left: Introducing realistic distractions in the supervised learning setting. Right: Semi-supervised learning leveraging distractions and pseudo-labeling.

shape, and appearance are controlled via learnable hyperparameters. Though its performance gain is notable, AutoFlow employs synthetic augmentations. The work in [27] utilizes AutoFlow and argues that it is important to disentangle architecture and training pipeline. [27] also points out that some of the performance improvements of the recent methods are due to hyperparameters, dataset extensions, and training optimizations. Our work focuses on more capable augmentations for model agnostic training, not architecture novelties.

Semi-Supervised Learning: Semi-supervised optical flow learning methods [12, 17] aim to make the best use of unlabeled data as there is only a limited amount of optical flow ground truth data for real and natural scenes. And even in those datasets, the optical flow ground truths are computed. RAFT-OCTC [12] proposed transformation consistency for semi-supervised learning, which applies spatial transformations to image pairs and enforces flow equivariance between the original and transformed pairs. FlowSupervisor [9] introduced a teacher network for stable semi-supervised fine-tuning. Its student model is trained for all pixels using teacher network output. In DistractFlow, we propose uncertainty-aware pseudo-labeling, which uses two different image pairs instead of different networks. We further employ forward-backward consistency to derive dense confidence scores, which steer the training process to impose loss only within high-consistency image regions to prevent feedback from incorrect flow estimates.

Since acquiring optical flow ground truth for real videos is problematic (not possible for most cases), unsupervised training methods [15, 21, 26] seek out training models without ground truth flows. Even though they report promising results comparable to the earlier deep learning approaches, unsupervised training methods still entail limitations.

3. DistractFlow

Our approach incorporates augmentation and supervision techniques to enhance the training of optical flow estimation models. In Section 3.1, we describe how we construct realistic distractions for optical flow training and how we employ them in supervised settings. Next, in Section 3.2, we extend our approach to semi-supervised learning with additional unlabeled data. We derive a self-supervised regularization objective by utilizing the distracted samples and pseudo-labeling.

3.1. Realistic Distractions as Augmentation

Consider a pair of video frames during training: $(I_t,\, I_{t+1})$. The distracted version of them is denoted as $(I_t,\, D_\lambda(I_{t+1}, \tilde{I}_d)$, where $D_\lambda(I_{t+1}, \tilde{I}_d)$ is the perturbed second frame obtained by combing with another image $\tilde{I}_d$ based on a mixing ratio of $\lambda \in (0,\, 1)$. Specifically, $D_\lambda(I_{t+1}, \tilde{I}_d)$ is calculated as $\lambda \cdot I_{t+1} + (1-\lambda) \cdot \tilde{I}_d$, where λ is sampled from a Beta(α, α) distribution, same as defined in [34, 36].

Figure 2 shows an example of a distracted pair of video frames. It can be seen that the actual objects and the real scene from one image are overlaid onto the second one. Such perturbations can reflect challenging real-world scenarios, e.g., foreground/background objects that only start to appear in the second frame, drastic motion blur, out-of-focus artifacts, reflections on specular surfaces, partial occlusions, etc. Furthermore, since the distractions are from the same dataset, the visual context of the original pair and distractor image are similar. For instance, the original pair and distractor depict similar classes (road, car, building, etc.). The spatial arrangement of the class and object regions in those images are similar, e.g., roads are within the lower part of the images, and so do the sky, buildings, and

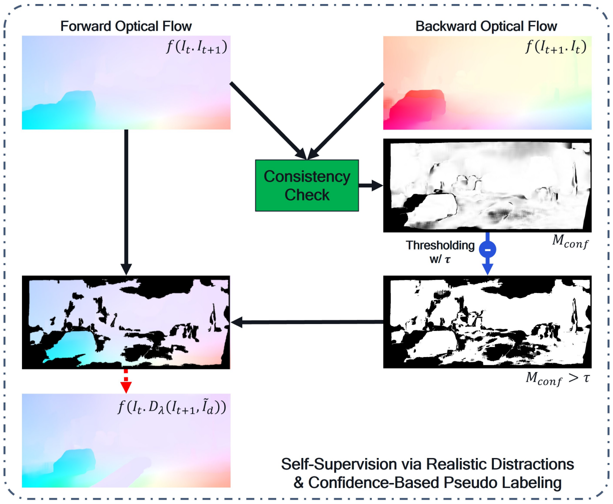

Figure 3. Semi-supervised learning that leverages distractions and confidence-aware pseudo-labeling.

vehicles. While one may attempt to render such scenarios synthetically, our DistractFlow provides a convenient and automatic way that can still capture such natural and semantically related variations. Compared to conventional augmentation methods that use noise or simple shapes, our method allows the model to be more robust to perturbations caused by real-world image contents.

Additionally, we note that applying realistic distractions to the first or both frames is possible. As we shall see in the experiments, all of these options will result in improved accuracy.

To provide supervision on the distracted pair of frames, we use the ground-truth flow of the original pair. The loss is computed as follows:

$$\mathcal{L}_{\text{dist}} = \|V^f_{(I_t,\, I_{t+1})} - f(I_t, D_\lambda(I_{t+1}, \tilde{I}_d))\|_1, \qquad (1)$$

where $V^f_{(I_t,\, I_{t+1})}$ is the ground-truth forward flow for the original pair $(I_t,\, I_{t+1})$ and $f(\cdot,\, \cdot)$ denotes the predicted flow based on model f.

In the supervised learning setting, where all training samples are labeled, the total training loss is then given as follows:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{base}} + w_{\text{dist}} \mathcal{L}_{\text{dist}}, \qquad (2)$$

where L_{base} is the conventional supervised loss and $w_{\text{dist}} > 0$ weights $\mathcal{L}_{\text{dist}}$. For iterative models like RAFT, we compute and apply this loss at each recurrent iteration.

3.2. Semi-Supervised Learning via Realistic Distraction and Pseudo-Labeling

During training, when unlabeled frame pairs are available, we can further leverage our distracted pairs of frames to derive additional self-supervised regularization. More specifically, given a distracted pair of frames, $(I_t,\, D_\lambda(I_{t+1},\, \tilde{I}_d))$, and the original pair, (I_t, I_{t+1}), we enforce the model's prediction on the distracted pair to match that on the original pair. In other words, the prediction on the original pair, $f(I_t,\, I_{t+1})$, is treated as the pseudo label. By doing this, the model learns to produce optical flow estimation on the distracted pair that is consistent with that of the original pair, despite the distractions. Such regularization promotes the model's robustness when processing real-world data, as we show in our experiments.

We note that, however, using all of $f(I_t,\, I_{t+1})$ as the training target for the distracted pair is problematic. This is because the model's prediction can be erroneous and noisy during the training process, even on the original frame pairs. The low-quality pseudo labels can be detrimental to the overall training, even leading to instability.

To address this issue, we adopt uncertainty-aware pseudo labels by calculating a confidence map based on forward backward consistency, and only using highly confident pixels' predictions as pseudo ground truth, as shown in Figure 3. On a frame pair $(I_t,\, I_{t+1})$, let $\widehat{V}^f(x)$ and $\widehat{V}^b(x)$ denote the predicted forward and backward flows at the pixel location x. When they satisfy the following constraint [21], we can assume that the prediction is accurate.

$$\begin{aligned} &|\widehat{V}^f(x) + \widehat{V}^b(x + \widehat{V}^f(x))|^2 \\ &\quad < \gamma_1 \Big(|\widehat{V}^f|^2 + |\widehat{V}^b(x + \widehat{V}^f(x))|^2 \Big) + \gamma_2, \end{aligned} \qquad (3)$$

where γ_1 = 0.01 and γ_2 = 0.5 from [21].

As such, we derive the confidence map as follows:

$$M_{\text{conf}} = \exp\left(- \frac{|\widehat{V}^f(x) + \widehat{V}^b(x + \widehat{V}^f(x))|^2}{\gamma_1 \Big(|\widehat{V}^f|^2 + |\widehat{V}^b(x + \widehat{V}^f(x))|^2 \Big) + \gamma_2} \right). \qquad (4)$$

Our confidence map provides a measure of reliability of the predicted optical flow. Specifically, in Eq. 4, if the numerator and the denominator are equal, the confidence value is then approximately 0.37 (e^{-1}). In this paper, we use a very high threshold and it could provide a more accurate optical flow pseudo ground truth.

By incorporating the confidence map, our self-supervised regularization is then given as follows:

$$\mathcal{L}_{\text{self}} = \|[M_{\text{conf}} \geq \tau]\, (f(I_t,\, I_{t+1}) - f(I_t, D_\lambda(I_{t+1},\, \tilde{I}_d)))\|_1, \qquad (5)$$

where $[\cdot]$ is the Iverson bracket and τ is the confidence threshold.[1]

In summary, in the semi-supervised learning setting, where labeled and unlabeled data are both used in training, the total loss is as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + w_{\text{self}} \mathcal{L}_{\text{self}}, \qquad (6)$$

where $\mathcal{L}_{\text{sup}}$ is the supervised loss of Eq. 2, $\mathcal{L}_{\text{self}}$ is the self-supervised loss of Eq. 5, and $w_{\text{self}} > 0$ weights $\mathcal{L}_{\text{self}}$.

[1] By $[M_{\text{conf}} \geq \tau]$, we note that the Iverson bracket is applied pixel-wise to produce a binary confidence map.

Table 1. Optical flow estimation results on SlowFlow, Sintel (train), and KITTI (train) datasets. We train the models on FlyingChairs (C) and FlyingThings (T) in the supervised setting. In the semi-supervised setting, we finetune the model on FlyingThings (T) as labeled data and Sintel (test) and KITTI (test) (S/K) as unlabeled data. * indicates test results of existing models generated by us.

	Method	Model	Labeled data	Unlabeled data	SlowFlow (100px)				Sintel (train)	KITTI (train)	
					1	3	5	7	(Final-epe)	(Fl-epe)	(Fl-all)
Supervised	Supervised	RAFT [30]	C+T		3.73	7.98	6.72	9.96	2.73*	4.94*	16.9*
	DistractFlow (Our)				**3.37**	**3.93**	**6.21**	**8.18**	**2.61**	**4.57**	**16.4**
	Supervise	GMA [14]			**2.49**	4.99	5.95	9.15	2.85*	4.88*	17.1*
	DistractFlow (Our)				2.53	**3.49**	**5.43**	**8.24**	**2.66**	**4.76**	**16.9**
	Supervised	FlowFormer [6]			2.72	3.73	5.24	6.78	2.39*	4.10*	14.7*
	DistractFlow (Our)				**2.51**	**2.77**	**4.39**	**6.30**	**2.31**	**4.00**	**13.9**
Semi-Supervised	Supervised	RAFT [30]	C + T		3.73	7.98	6.72	9.96	2.73*	4.94*	16.9*
	RAFT-A [28]		AutoFlow [28]		-	-	-	-	2.57	4.23	-
	RAFT-OCTC [12]		C + T	T (subsampled)	-	-	-	-	2.67	4.72	16.3
	Fixed Teacher [9]			*S*/*K*	-	-	-	-	2.58	4.91	15.9
	FlowSupervisor [9]			*S*/*K*	-	-	-	-	2.46	3.35	**11.1**
	DistractFlow (Our)			*S*/*K*	**2.46**	**3.60**	**5.15**	**6.95**	**2.35**	**3.01**	11.7
	Supervised	GMA [14]			2.49	4.99	5.95	9.15	2.85*	4.88*	17.1*
	DistractFlow (Our)			*S*/*K*	**2.44**	**2.79**	**4.38**	**6.57**	**2.31**	**3.21**	**11.0**
	Supervised	FlowFormer [6]			2.72	3.73	5.24	6.78	2.39*	4.29*	15.4*
	DistractFlow (Our)			*S*/*K*	**2.48**	**2.69**	**4.31**	**6.29**	**2.33**	**3.03**	**11.8**

4. Experiments

In this section (and in the supplementary), we present a comprehensive evaluation of DistractFlow on several benchmark datasets, compare it with baselines and the latest state-of-the-art (SOTA) methods, and conduct extensive ablation studies. We focus our evaluations on realistic or real-world data, including SlowFlow, Sintel (final), and KITTI; we provide results on Sintel (clean) in the Supplementary.

4.1. Experimental Setup

Evaluation Settings: We consider two settings for running the experiments. In the first setting, we consider a supervised learning setting where the network is trained on fully labeled data. The second setting considers semi-supervised learning, where unlabeled data can be used during training, in addition to labeled training data.

Evaluation Metrics: We use the common evaluation metrics for optical flow estimation, including End-Point Error (EPE) and Fl-all, which is the percentage of optical flow with EPE larger than 3 pixels or over 5% of the ground truth. The goal is to lower both of these metrics.

Datasets: Following commonly adopted evaluation protocols in the literature [6, 14, 30, 35], we train our model on FlyingChairs (C) [4] and FlyingThings3D (T) [20] for supervised training when evaluating on SlowFlow [11] (100px flow magnitude) dataset and the training splits of Sintel (S) [2] and KITTI (K) [5, 22]. In particular, on SlowFlow, we use 4 blur durations with a larger duration having larger motion blurs. When we evaluate on Sintel test set, we use FlyingThings3D, Sintel training set, KITTI training set, and HD1K (H) [16] for supervised training. And, we finetune on KITTI training dataset for evaluation on KITTI test set.

For the additional unlabeled data used during training, we followed the same setting from [9]. We use FlyingThings as labeled dataset and Sintel test set as unlabeled dataset for evaluating on Sintel and SlowFlow, and the video frames from KITTI test raw sequences for evaluating on KITTI training dataset. When we evaluate on Sintel and KITTI test sets, we use the labeled dataset from supervised training settings, and use additional KITTI train raw sequences, Sintel training data (only using every other frame), Monkaa (M) [20], and Driving (D) [20] as the unlabeled datasets. Note that we do not use Sintel and KITTI test sets as unlabeled datasets for Sintel and KITTI test evaluation. We further experiment with utilizing unlabeled open-source Blender videos as additional unsupervised training data, such as Sintel and Big Buck Bunny.

Networks and Training: We use RAFT [30], GMA [14], FlowFormer [6] as our baselines, and utilize their official codes.[2] For objectiveness, we train all the baselines in the same framework and reported the results we obtained. We set (w_{dist}, w_{self}) and τ to (λ (mixing ratio), 1) and 0.95, respectively. For supervised training, we follow RAFT and GMA learning parameters such as optimizer, number of GRU iterations, training iterations, and so on. For semi-supervised training, we use labeled and unlabeled data with a 1:1 ratio. For FlowFormer, we reduce the total batch size and only finetune it due to GPU memory overflow. All settings and training details are provided in the Supplementary.

4.2. Evaluations on SlowFlow, Sintel (train), and KITTI (train)

Supervised Setting: Table 1 shows the performance evaluation of the supervised and semi-supervised training results on SlowFlow, Sintel (train), and KITTI (train). In the top section of Table 1, we provide results of existing supervised models and our models trained using DistractFlow

[2] RAFT: https://github.com/princeton-vl/RAFT, GMA: https://github.com/zacjiang/GMA, FlowFormer: https://github.com/drinkingcoder/FlowFormer-Official

Table 2. Optical flow estimation results on Sintel (test) and KITTI (test) datasets. We train the models on FlyingChairs (C), FlyingThings (T), Sintel (S), KITTI (K), and HD1K (H) in the supervised setting. When using the semi-supervised setting/using additional data for training, RAFT-A trains on A+S+K+H+T datasets where A stands for AutoFlow, FlowSupervisor trains on C+T+S+K+H (labeled) and uses Sintel, KITTI, and Spring as unlabeled datasets. In our case, we use C+T+S+K+H as labeled data and Sintel, KITTI, and Sceneflow (Monkaa, Driving) as unlabeled data. * indicates results obtained using warm-start [30].

Method	Model	Sintel (test)	KITTI (test)
		(Final-epe)	(Fl-all)
Supervised			
Supervised DistractFlow (Our)	RAFT [30]	2.86^* $\mathbf{2.77}^*$	5.10 **4.82**
Semi-Supervised / Additional dataset			
Supervised RAFT-A [28] RAFT-OCTC [12] FlowSupervisor [9] DistractFlow (Our)	RAFT [30]	2.86^* 3.14 3.09 2.79^* $\mathbf{2.71}^*$	5.10 4.78 4.72 4.85 **4.71**

supervised augmentation. Our supervised models trained on FlyingChairs and FlyingThings3D show significant improvements across the test datasets for all the architectures. For SlowFlow with a 3-frame blur duration, the existing RAFT model shows very larges error (over 100 EPE) on a few samples. DistractFlow, on the other hand, leads to more robust results and a much smaller EPE.

Semi-Supervised Setting: In this part, we evaluate our proposed approach when additional unlabeled data becomes available. In the bottom section of Table 1, we show results of the previous semi-supervised learning methods and our proposed method. Following FlowSupervisor, we finetune each model on FlyingThings (labeled) and Sintel test (unlabeled), and evaluate on the SlowFlow and Sintel train dataset. For evaluation on KITTI train dataset, we finetune the pre-trained models (from Sintel unlabeled) on FlyingThings (labeled) and raw KITTI test (unlabeled). We can see that DistractFlow (RAFT) shows better performance as compared to RAFT-OCTC, RAFT-A, FixTeacher, and FlowSupervisor. Furthermore, we apply our semi-supervised approach to GMA and FlowFormer, and DistractFlow improves their performance.

4.3. Evaluations on Sintel (test) and KITTI (test)

Supervised Setting: The top section of Table 2 summarizes the supervised and semi-supervised training results on Sintel (test) and KITTI (test) datasets. From the first part of Table 2, we can see that our proposed DistractFlow permits significant improvements over the baseline RAFT model. It is noteworthy that without using additional unlabeled data, our model trained under the supervised setting already outperforms the semi-supervised FlowSupervisor, which leverages additional data.

Semi-Supervised Setting: In the second part of Table 2, we provide results for semi-supervised methods or methods that train with additional datasets. RAFT-A is trained with an additional AutoFlow dataset but still underperforms the RAFT trained with our DistractFlow approach. RAFT-OCTC applies a semi-supervised method as well as changes the architecture for occlusion prediction. Although RAFT-OCTC uses a slightly more complex architecture, our DistractFlow-trained RAFT still shows better performance. In addition, our method also considerably outperforms FlowSupervisor.

Table 3. Effects of different types of perturbations applied to the frames, as data augmentation during training.

Method	Perturbation	Sintel (train)	KITTI (train)	
		(Final-epe)	(Fl-epe)	(Fl-all)
RAFT [30]		2.73	4.94	16.9
Our	Gaussian noise Random shapes Realistic Distractions	2.68 2.66 **2.61**	4.86 4.82 **4.57**	17.6 16.8 **16.4**

Table 4. Distracting I_t and/or I_{t+1} on Sintel (train) and KITTI (train) datasets. We train RAFT with distractions to I_1 or I_2 or both. α is the coefficient in the Beta distribution for sampling λ. α_1 and α_2 are for applying distractions to I_1 and I_2, respectively.

Method	Distraction	α_1	α_2	Sintel (train)	KITTI (train)	
				(Final-epe)	(Fl-epe)	(Fl-all)
RAFT [30]				2.73	4.94	16.9
DistractFlow	On I_t	0.1		2.69	**4.81**	**16.4**
		1		2.64	5.25	17.6
		10		**2.55**	5.32	17.8
	On I_{t+1}		0.1	2.65	4.66	**16.3**
			1	**2.61**	**4.57**	16.4
			10	2.70	4.82	17.2
	On I_t & I_{t+1} (same)	0.1	1	2.70	5.33	18.3
	On I_t & I_{t+1} (diff)	0.1	1	**2.64**	**4.92**	**16.7**

4.4. Ablation Studies

We conduct extensive ablation studies on various aspects of our method, by using RAFT as the base model. In the supervised setting, we train on FlyingChairs (C) and FlyingThings3D (T). In the semi-supervised setting, we take the RAFT model pretrained from the supervised setting and then finetune it using FlyingThings3D (labeled) and Sintel (test)/KITTI (test) (unlabeled). The evaluation is done on Sintel (train) and KITTI (train).

Type of Perturbations: When applying visual perturbations, we compare using realistic distractions in DistractFlow with using synthetic noise such as Gaussian noise and random shapes. For generating random shapes, we make random background colors and add 5 to 10 shapes (e.g., circles, triangles, and rectangles) of random colors and sizes. As shown in Table 3, while introducing perturbations with Gaussian noise or random shapes can result in performance gains, the improvements are not as significant as compared to the case of using our proposed augmentation strategy.

Distracting I_t or I_{t+1} or both: We compare applying distractions to I_t or I_{t+1} or both in Table. 4. At the top of Table. 4, we generate distracted I_t using the α_1 values

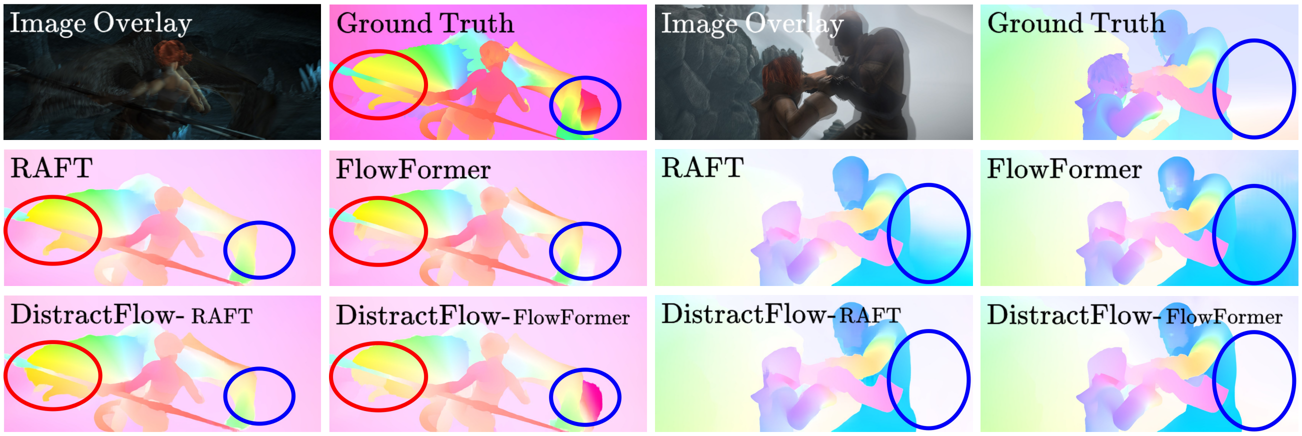

Figure 4. Qualitative results on Sintel (train) using RAFT, FlowFormer, DistractFlow-RAFT, and DistractFlow-FlowFormer (the last two are empowered with our proposed method). All models are trained on FlyingChairs and FlyingThings3D. It can be seen that DistractFlow can generate more accurate predictions, with better spatial consistency and finer details, as highlighted by circles.

Table 5. Effectiveness of L_{dist}

Method	$\mathcal{L}_{base}$	$\mathcal{L}_{dist}$	Sintel (train)	KITTI (train)	
			(Final-epe)	(Fl-epe)	(Fl-all)
RAFT [30]	✓		2.73	4.94	16.9
DistractFlow		✓	2.82	5.43	19.0
	✓	✓	**2.61**	**4.57**	**16.4**

Table 6. Effectiveness of confidence map and λ weight for $\mathcal{L}_{\text{self}}$ in semi-supervised setting. τ is the confidence threshold in Eq. 5 and in w_{self} is the weight for the self-supervised loss. Note that 0.37 corresponds to having equal nominator and denominator in Eq. 3.

Method	τ	Sintel (train)	KITTI (train)	
		(Final-epe)	(Fl-epe)	(Fl-all)
RAFT [30]		2.73	4.94	16.9
DistractFlow		diverged		
	✓(0.37)	**2.35**	3.37	12.42
	✓(0.95)	**2.35**	**3.01**	**11.71**

Table 7. Effects of using different unlabeled datasets in the semi-supervised training. Our DistractFlow enables consistent performance improvements when using any of the unlabeled datasets.

Method	Unlabeled data (# of pairs)	Sintel (train)
		(Final-epe)
RAFT [30]		2.73
DistractFlow	Sintel-test (1.1k)	**2.35**
	Monkaa (17k) & Driving (9k)	2.42
	Big Buck Bunny (14k)	2.58

for sampling λ and train the model accordingly.[3] All three variants show improvements on Sintel. However, when α_1s are 1 or 10, the performance drops on KITTI compared to RAFT. We suspect that the Sintel (final) dataset has a heavy visual effect even in I_t, and using strongly distracted I_t could help the training.[4] On the other hand, since KITTI has relatively cleaner images, strongly distracted I_t does not improve the performance.

In the middle of Table 4, we distract I_{t+1} according to the different α_2 values and train the model. In most of these cases DistractFlow improves upon the baseline RAFT, and it shows the best performance at $\alpha_2 = 1$. This result shows that various mixing of the images is helpful.

At the bottom of Table 4, we distract I_t and I_{t+1} simultaneously. When we apply the same distraction to both frames, two consecutive videos, it degrades the performance. This is because applying the same distraction to both frame introduces new correspondences which are not part of the ground truth. On the other hand, when we apply distractions from two different images to I_t and I_{t+1}, the trained model shows better performance compared to baseline RAFT. Overall, distracting I_{t+1} shows the best performance and we carry out experiments with this setting.

Effectiveness of $\mathcal{L}_{\text{dist}}$: Table 5 shows the effectiveness of $\mathcal{L}_{\text{dist}}$. Without the supervised loss on the original pairs, a model trained only with $\mathcal{L}_{\text{dist}}$ still provides reasonable estimation performance, but shows worse accuracy as compared to the original model. Combining $\mathcal{L}_{\text{base}}$ with $\mathcal{L}_{\text{dist}}$ shows a significant improvement.

Effectiveness of Confidence Map: Table 6 shows our study on the impacts of confidence-based thresholding in the semi-supervised training. When training the model without any confidence maps, erroneous predictions are used for backpropagation and training the network, causing the training to diverge. On the other hand, our confidence map allows the network to only train on highly accurate predictions and enables stability when training the model.

When we set $\tau = 0.37$ (a common choice for forward-backward consistency), we can stably train the model and it shows improvement. Higher τ (e.g., 0.95) shows further accuracy improvement on KITTI. This may be due to the larger displacements in KITTI, which can make the model

[3] When $\alpha < 1$, sampled λ is close to 0 or 1. When $\alpha > 1$, sampled λ is close to 0.5. When $\alpha = 1$, λ is sampled from a uniform distribution.

[4] Since we set $w_{dist} = \lambda$, when λ is close to zero, the distracted pair only impacts the training minimally.

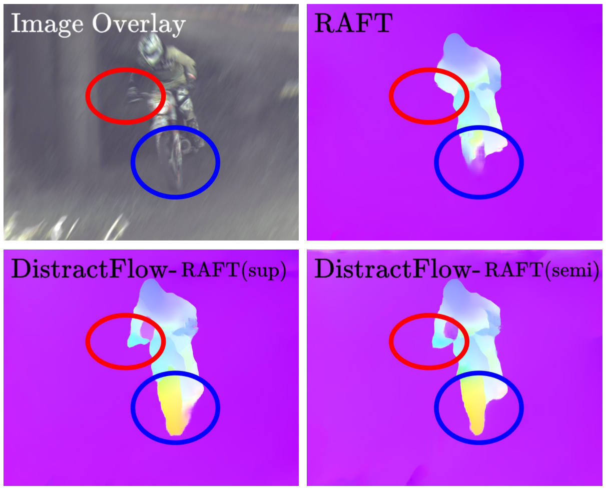

Figure 5. Qualitative results on SlowFlow using the original RAFT and DistractFlow-trained RAFT models in supervised and semi-supervised model settings. We see that DistractFlow training enables the network to produce more accurate results, despite the severe motion blur.

more prone to in accurate predictions. As such, creating pseudo labels with a higher confidence threshold is beneficial in this case.

4.5. Qualitative Results

Figure 4 shows qualitative results on the Sintel (train, final) using the original RAFT and FlowFormer, as well as our DistractFlow-trained RAFT and FlowFormer. These models are trained on FlyingChairs and FlyingThings3D. Because Sintel (final) contains visual effects such as fog and blur, the original models generate erroneous estimations. On the other hand, our DistractFlow-trained models show more accurate and robust flow estimation results. Especially, it shows accurate predictions at the object boundary without any edge- or segmentation-aware training [1,3].

Figure 5 shows qualitative results on SlowFlow using the original RAFT, as well as DistractFlow-trained RAFT models (supervised and semi-supervised). Our supervised training allows the model to generate more accurate and robust flows compared to the baseline RAFT. With our semi-supervised setting, our model shows further improvements.

5. Discussion

Unlabeled Dataset: Table 7 shows the results on Sintel (train) when using different unlabeled data in semi-supervised training. Sintel (test) shows significant improvement compared to supervised training since it is in the same domain as Sintel (train). Although Monkaa & Driving or Bunny dataset have more unlabeled pairs and can still improve upon the original RAFT, they exhibit worse performance than using Sintel (test) pairs. This indicates that for semi-supervised setting, it is important to use unlabeled data with scenes and distributions resembling the target use case. We leave data distribution robustness (e.g., addressing out-of-distribution samples [18, 19, 31] as part of future work. Nevertheless, our semi-supervised method improves the performance when using any of the unlabeled datasets.

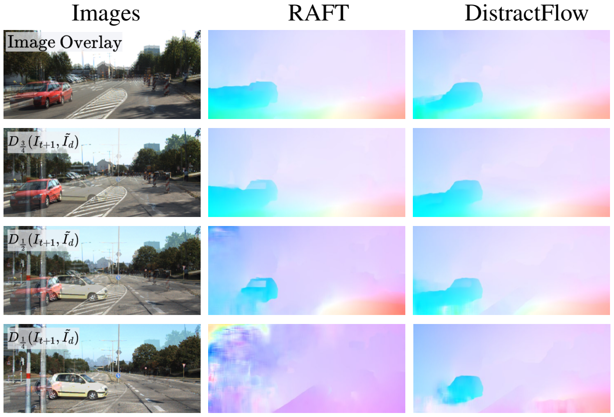
Images
RAFT
DistractFlow
Image Overlay
$D_{\frac{3}{4}}(I_{t+1}, \tilde{I}_d)$
$D_{\frac{1}{2}}(I_{t+1}, \tilde{I}_d)$
$D_{\frac{1}{4}}(I_{t+1}, \tilde{I}_d)$

Figure 6. Predictions on distracted video frame pairs. In the first column, the first row shows the overlaid original pair and the second to fourth rows show the distracted pairs. The mixing weight for the distractor increases from top to bottom. The RAFT model trained with DistractFlow performs robustly, while the original RAFT model completely fails when the distraction becomes large.

Robust Prediction: Figure 6 shows the predictions of RAFT and DistractFlow-RAFT (trained with FlyingChairs and FlyingThings3D) on distracted frames pairs, using mixing ratios of 1, 0.75, 0.5, and 0.25. When the original I_{t+1} has a small portion of a distracted image , the original RAFT has degraded performance. It completely fails to find the correspondence for the red car when $\lambda = 0.25$. In contrast, the model trained using DistractFlow still robustly finds the correspondence in the distracted image.

6. Conclusion

We proposed a novel method, DistractFlow, to augment optical flow training. We introduced realistic distractions to the video frame pairs which provided consistent improvements to optical flow estimation models. When unlabeled data was available, based on the original and distracted pairs, we devised a semi-supervised learning scheme using pseudo labels. We also incorporated forward-backward consistency through confidence maps that provided training stability and enhanced the performance further. Through extensive experiments on several optical flow estimation benchmarks: SlowFlow, Sintel, and KITTI, we showed that our method achieved significant improvements over the previous state of the art without inducing additional complexity during inference. In particular, models trained using our DistractFlow strategy are more robust in practical, challenging scenarios (e.g., consistent error reductions on SlowFlow despite strong motion blurs).

References

[1] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5911, 2021. 8

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 611–625. Springer, 2012. 5

[3] Hong Cai, Janarbek Matai, Shubhankar Borse, Yizhe Zhang, Amin Ansari, and Fatih Porikli. X-distill: Improving self-supervised monocular depth via cross-task distillation. In *British Machine Vision Conference*, 2021. 8

[4] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2758–2766, 2015. 1, 2, 5

[5] 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. 5

[6] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2, 5

[7] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2019. 1

[8] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017. 1, 2

[9] Woobin Im, Sebin Lee, and Sung-Eui Yoon. Semi-supervised learning of optical flow by flow supervisor. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 2, 3, 5, 6

[10] Md Amirul Islam, Matthew Kowal, Konstantinos G Derpanis, and Neil DB Bruce. Segmix: Co-occurrence driven mixup for semantic segmentation and adversarial robustness. *International Journal of Computer Vision*, pages 1–16, 2022. 2

[11] Joel Janai, Fatma Güney, Jonas Wulff, Michael Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017. 5

[12] Jisoo Jeong, Jamie Menjay Lin, Fatih Porikli, and Nojun Kwak. Imposing consistency for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3181–3191, 2022. 1, 2, 3, 5, 6

[13] Jisoo Jeong, Vikas Verma, Minsung Hyun, Juho Kannala, and Nojun Kwak. Interpolation-based semi-supervised learning for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11602–11611, 2021. 2

[14] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021. 2, 5

[15] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 557–572. Springer, 2020. 3

[16] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016. 5

[17] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *Advances in neural information processing systems*, pages 354–364, 2017. 3

[18] Si Liu, Risheek Garrepalli, Thomas Dietterich, Alan Fern, and Dan Hendrycks. Open category detection with pac guarantees. In *International Conference on Machine Learning*, pages 3169–3178. PMLR, 2018. 8

[19] Si Liu, Risheek Garrepalli, Dan Hendrycks, Alan Fern, Debashis Mondal, and Thomas G Dietterich. Pac guarantees and effective algorithms for detecting novel categories. *Journal of Machine Learning Research*, 23:44–1, 2022. 8

[20] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 5

[21] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018. 3, 4

[22] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 5

[23] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 1, 2

[24] Simon Schrodi, Tonmoy Saikia, and Thomas Brox. Towards understanding adversarial robustness of optical flow networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8916–8924, 2022. 2

[25] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 2

[26] 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. 2, 3

[27] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentangling architecture and training for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 165–182. Springer, 2022. 1, 2, 3

[28] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10093–10102, 2021. 2, 5, 6

[29] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1, 2

[30] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020. 1, 2, 5, 6, 7

[31] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022. 8

[32] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020. 2

[33] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1d attention and correlation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10498–10507, 2021. 2

[34] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 2, 3

[35] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10807–10817, 2021. 2, 5

[36] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2, 3

[37] Zhi Zhang, Tong He, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of freebies for training object detection neural networks. *arXiv preprint arXiv:1902.04103*, 2019. 2

[38] 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. 2

[39] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17592–17601, 2022. 2