Benchmarking the Robustness of Optical Flow Estimation to Corruptions

Zhonghua Yi[1], Hao Shi[1], Qi Jiang[1], Yao Gao[1], Ze Wang[1], Yufan Zhang[1],
Kailun Yang[2],*, Kaiwei Wang[1],*
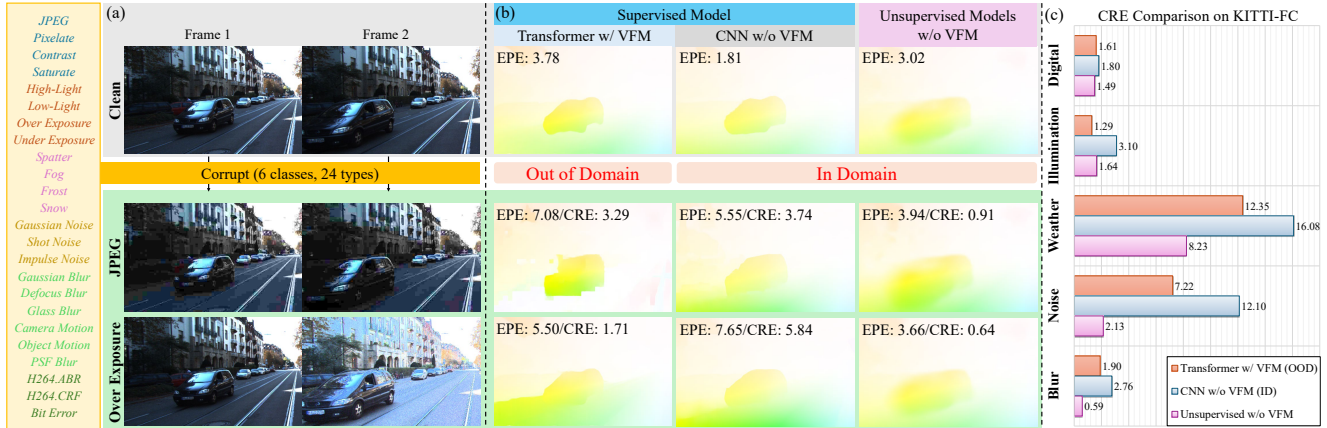
[1]Zhejiang University [2]Hunan University

Figure 1. **Benchmarks overview.** (a) Clean image pairs are corrupted to construct optical flow robustness benchmarks **KITTI-FC** and **GoPro-FC**. (b) 29 model variants from 15 mainstream methods are evaluated through proposed robustness metrics in **Out-Of-Domain** (OOD) and **In-Domain** (ID) settings. SAMFlow-H [89] with Vision Foundation Model (VFM) [23], RAFT [70], and ARFlow [31] are taken as examples. (c) Comprehensive results and analyses of multiple models of a range of corruptions are presented.

Abstract

Optical flow estimation is extensively used in autonomous driving and video editing. While existing models demonstrate state-of-the-art performance across various benchmarks, the robustness of these methods has been infrequently investigated. Despite some research focusing on the robustness of optical flow models against adversarial attacks, there has been a lack of studies investigating their robustness to common corruptions. Taking into account the unique temporal characteristics of optical flow, we introduce 7 temporal corruptions specifically designed for benchmarking the robustness of optical flow models, in addition to 17 classical single-image corruptions, in which advanced PSF Blur simulation method is performed. Two robustness benchmarks, KITTI-FC and GoPro-FC, are subsequently established as the first corruption robustness benchmark for optical flow estimation, with Out-Of-Domain (OOD) and In-Domain (ID) settings to facilitate comprehensive studies. Robustness metrics, Corruption Robustness Error (CRE), Corruption Robustness Error ratio (CREr), and Relative Corruption Robustness Error (RCRE)
are further introduced to quantify the optical flow estimation robustness. 29 model variants from 15 optical flow methods are evaluated, yielding 10 intriguing observations, such as 1) the absolute robustness of the model is heavily dependent on the estimation performance; 2) the corruptions that diminish local information are more serious than that reduce visual effects. We also give suggestions for the design and application of optical flow models. We anticipate that our benchmark will serve as a foundational resource for advancing research in robust optical flow estimation. The benchmarks and source code will be released at https://github.com/ZhonghuaYi/optical_flow_robustness_benchmark.

1. Introduction

Optical flow estimation [17, 35, 67, 70, 89] is a long-standing problem in computer vision, which estimates the pixel-level 2D motion of two continuous frames. Benefiting from its provided spatial-temporal correspondence, optical flow models are widely used in tasks such as autonomous driving [5, 71] and video editing [7, 77]. Early optical flow estimation methods [26, 35, 53] are knowledge-driven, while in recent years more and more deep learning mod-

---

*Corresponding authors (e-mail: kailun.yang@hnu.edu.cn, wangkaiwei@zju.edu.cn).

els [9, 59, 79] are developed with promising performance.

Research on the robustness of optical flow models is of paramount importance due to their widespread application, particularly in safety-critical domains. In recent years, though robustness benchmarking research has been studied for various vision tasks like object detection [8, 90] and segmentation [19, 76], there is relatively little literature on the robustness of optical flow models, which are mainly in the field of adversarial attacks [1, 51, 57]. The impact of domain perturbation, which commonly occurs in real-world applications, especially autonomous driving and video editing, however, has not been well studied. Furthermore, as the optical flow contains object motion information, the former corruption robustness benchmarking methods [14, 50, 78] designed for single-image perception tasks lack simulation of temporal corruptions, making it impossible to simply borrow them completely. As a result, building a thorough benchmark with carefully considered corruptions for optical flow robustness research is necessary and urgent.

To our knowledge, we are the first to build such a corruption robustness benchmark. The overview is shown in Fig. 1. We propose 7 temporal corruptions that fully consider all the possible scenarios for optical flow applications. In addition, 17 common corruptions designed for single images are also included in our benchmarks for a comprehensive study. We upgrade the *PSF Blur* simulation method by introducing an automatic lens design approach. For the two most concerned applications, autonomous driving and video editing, we build two benchmarks, *KITTI-FC* and *GoPro-FC* for them respectively. We focus on large displacement in KITTI-FC and small displacement in GoPro-FC by performing different interframe intervals. Furthermore, we construct *In-Domain (ID)* test in KITTI-FC and *Out-Of-Domain (OOD)* test in both benchmarks, for real-world deployment with or without optical flow ground-truth, to research domain-related robustness.

To quantify the robustness under data perturbations, we further propose robustness metrics *Corruption Robustness Error (CRE)* and *Relative Corruption Robustness Error (RCRE)*, which can effectively represent the absolute robustness of the optical flow model when it receives perturbations. We also propose *Corruption Robustness Error ratio (CREr)* to evaluate the relative robustness of the model, by eliminating the impact of model estimation performance.

Based on the constructed benchmarks and evaluation metrics, we test 29 model variants from 15 optical flow estimation methods including traditional methods, supervised methods, and unsupervised models. Through comprehensive experiments, we found 7 interesting observations. For example, the absolute robustness of the model is heavily dependent on the estimation performance, and the corruptions that diminish local information have more serious impacts than those that reduce visual effects. The observations pro-vide new views for the future optical flow model design, by suggesting to: 1) improve estimation performance of the optical flow model; 2) introduce Transformer-like architecture and semantic information; 3) use unsupervised models or introduce unsupervised training approaches; 4) reduce the possibility of unreliable local information.

2. Related Work

Optical flow estimation. Early optical flow estimation methods [3, 10, 16, 26, 35, 53] are knowledge-driven, focusing on modeling spatial-temporal correspondences through energy minimization. In recent years, data-driven methods have gained traction, with CNN-based models [18, 59, 60, 67, 79] trained under supervised settings using ground truth flows. RAFT [70] introduced iterative refinement, significantly improving upon previous direct flow prediction approaches. Following works [20, 36, 69, 73, 84, 86] focus on large displacement and occlusion challenges. Transformer-based architectures have also been explored, yielding state-of-the-art performance [17, 34, 63, 66]. SAMFlow [89] fuses features from pre-trained SAM [23] with an optical flow context encoder, surpassing FlowFormer [17] in accuracy. More recently, diffusion models [15, 64] have shown promise in optical flow estimation, with DDVM [54] and FlowDiffuser [37] demonstrating their effectiveness. Given the lack of ground truth optical flow labels in real-world videos, unsupervised learning approaches [31–33, 38, 40, 44, 52, 65, 82] have been investigated to overcome this limitation. For challenging environmental conditions, such as rain [28, 29], fog [75, 88], and low-light [87], tailored optical flow models have been developed. When target domains are known, domain adaptation methods [47, 81] are also effective. While these methods offer advancements, a comprehensive benchmark on the robustness of optical flow models across various real-world corruptions remains critical to fully assess their reliability in diverse conditions.

Robustness of networks. The robustness of deep learning models remains a significant challenge, with adversarial attacks, common corruptions, and distribution shifts all impacting model performance. Robustness benchmarks have been established across tasks such as classification [14, 48, 78], segmentation [19, 22, 50, 76, 80, 85], action detection [83], pose estimation [39], and object detection [8, 46, 90]. Research on adversarial robustness [2, 27, 74] explores how small, optimized perturbations can significantly degrade model performance, effectively misleading models into incorrect outputs. Domain robustness also poses ongoing challenges [41, 62], as training data domains rarely align with real-world deployment scenarios, where varying conditions and settings create discrepancies. Optical flow model robustness is especially crucial for applications like autonomous driving and robotics. Previous studies [1, 51, 56] have primarily focused on adversarial ro-
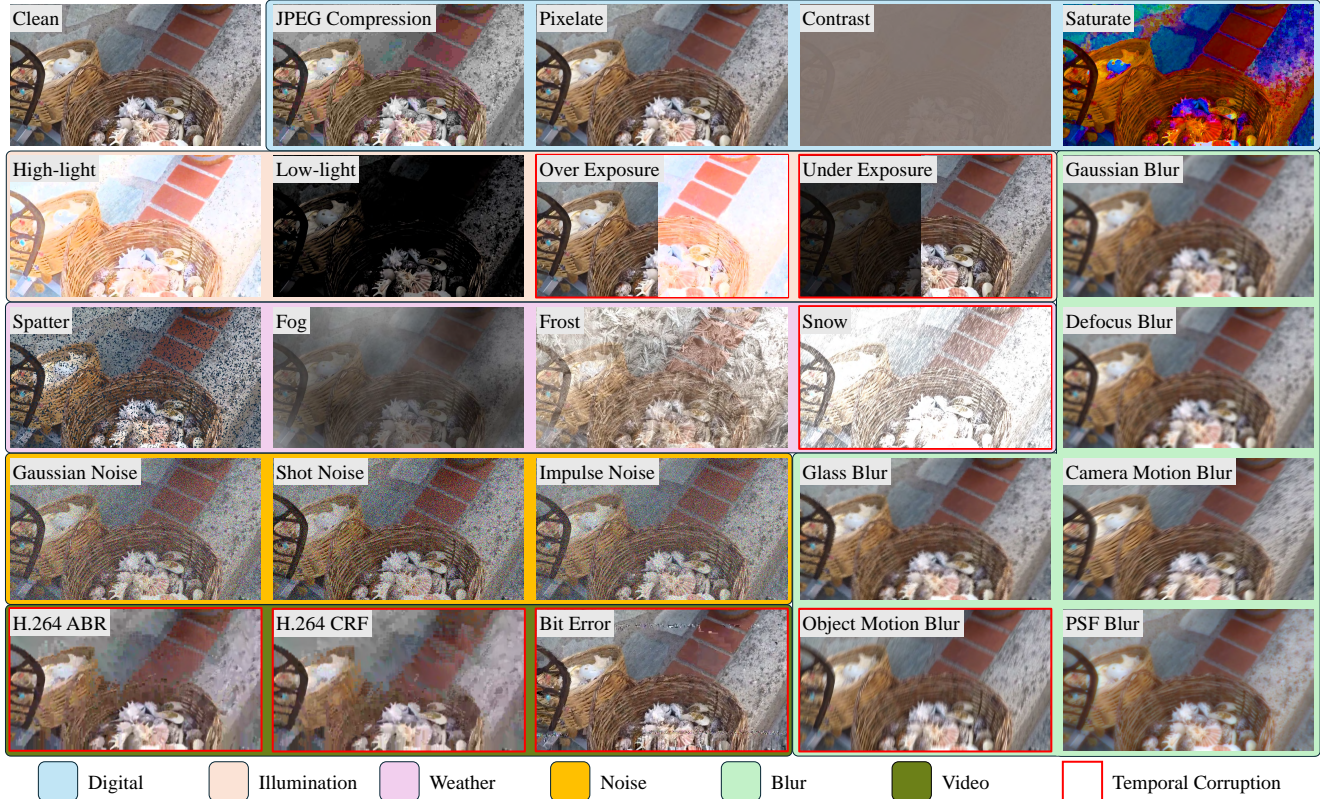
Figure 2. **Effects of all the** 24 **corruptions under severity of** 5. Corruptions are split into 6 classes. 7 temporal corruptions are in red boxes. The previous and next frames are displayed on the left and right sides of the image respectively for *Over Exposure* and *Under Exposure* for better visualization. Examples are from GoPro-FC.

bustness, manipulating pixel [25] or patch information [58] to disrupt flow estimation. Recently, Schmalfuss *et al*. [57] introduced a differentiable particle rendering framework to simulate adversarial weather attacks on optical flow models. To the best of our knowledge, there is a lack of systematic benchmarking on the robustness of optical flow estimation under common corruptions. Our study addresses this gap, providing a foundational analysis of optical flow robustness in corrupted real-world scenarios to support the development of more resilient models.

3. Corruptions for Optical Flow

There are numerous successful corruption methods [8, 14, 22] proposed for benchmarking single-image vision tasks, which arise from diverse real-world scenarios. Since optical flows are derived from image pairs with temporal associations, it is essential to account for additional **temporal corruptions** that occur in real-world applications. We therefore propose temporal corruptions including *Object Motion Blur*, *Over Exposure*, *Under Exposure* and *Snow*. Referring to video-oriented benchmarking works [55, 72, 78], we further introduce *H.264 CRF Compression*, *H.264 ABR Compression*, and *Bit Error* for a more comprehensive temporal robustness benchmark. We also introduce a more general

Point-Spread-Function (PSF) simulation method for accurate and extensive *PSF Blur* corruption generation. All the 24 corruptions are visualized in Fig. 2 and summarized into 6 classes below. In experiments, every corruption is adopted with 5 severity levels. The simulation details can be found in the supplementary.

Digital corruptions. Digital change occurs during the digital distribution process on the web, resulting in changes to image pixels, which are image-independent. These corruptions are well explored by image understanding works [22]. We consider 4 digital corruptions: *JPEG Compression*, *Pixelate*, *Contrast*, and *Saturate*, as they are commonly observed in web content delivery.

Illumination corruptions. In real-world situations, the same scene will appear under different ambient light intensities. Whether visual tasks can be robust to such changes needs to be studied. In addition, cameras sometimes face illumination problems, resulting in unsatisfied exposure and histogram, especially in high dynamic range scenes. We therefore use *High-light* and *Low-light* to simulate the various ambient light intensities, by increasing or decreasing the same intensity for the whole image. Because the camera's metering module always has a delay when the scene brightness changes rapidly, resulting in different exposure

levels of the front and back images, we further propose two temporal corruptions for optical flow benchmarking: *Over Exposure* and *Under Exposure*, in which one of the image pair is overexposed or underexposed while the other is well-exposed. We change Exposure values (EVs) to simulate these two exposure-related corruptions.

Weather corruptions. Weather corruptions are always considered by benchmarking methods [90], as they are common in real applications. Referring to the previous work [46], we introduce four types of weather corruptions: *Spatter*, *Fog*, *Frost*, and *Snow*. We note that the first three disturbances generally change slowly in the real world, so the two images captured by the camera are affected by them in approximately the same way. For *Snow*, the changes in the rendering results over time can usually be captured by the camera. Thus the rendering content is modified to be different for the two continuous images in our practical implementation. Due to the characteristics of texture rendering, *Spatter*, *Frost* and *Snow* tend to occlude objects, while *Fog* tends to reduce the visible area of the image.

Noise corruptions. The noises are time-independent in most situations, as they are usually caused by camera defects. Here we consider three corruptions that sample *Gaussian*, *Shot*, and *Impulse* noises for images.

Blur corruptions. Static blurry effects like *Gaussian*, *Defocus*, and *Glass* blur have been investigated. However, the influence of a physically realizable optical system is rarely studied, in which a blurring effect results from the Point-Spread-Function (PSF) of the optical system. Some works consider optical systems designed by *Zemax* [22] or Zernike polynomials [48], while the blurring effect is not satisfiable, and the lens chosen can not represent a wide range of low-quality optical systems, since the aberration of some minimalist optical systems is more serious. Given the recent success of the end-to-end optical system generation approaches [12], we generate and sample 5 low-quality lenses, with the Root-Mean-Square (RMS) radius varying from 0.02 to 0.19. The enhanced *PSF Blur* expands the scope of research on optical systems and provides a valuable framework for studying the robustness of visual methods on low-quality optical systems [11]. For motion blur, we propose two temporal corruptions with different motion patterns, in which *Camera Motion Blur* is caused by camera vibration and *Object Motion Blur* by scene moving.

Video corruptions. In modern vision tasks, video tasks like editing [7, 77] heavily depend on optical flow. In such applications, digital video processing also affects the final image quality and information. Referring to the video task benchmarking works, we introduce three corruptions: *H.264 CRF Compression*, *H.264 ABR Compression* and *Bit Error*, which all appear when applying H.264 encoding.

Comparison Fairness. Although a total of 24 corruptions have been investigated, the simulation approaches and pa-

rameters vary, making it challenging to determine the appropriate simulation parameters for different corruptions to achieve a consistent real-world probability of occurrence. Consequently, for the majority of corruptions, we are unable to equally compare their impact. Currently, *Hight-Light* and *Low-Light* can be compared with each other, as they are performed with the same approach. Similarly, *Over Exposure* and *Under Exposure* are also comparable.

4. Robustness Benchmarks for Optical Flow

4.1. Robustness Metrics

The performances of optical flow estimation are often evaluated by End-Point-Error (EPE), defined as the Euclidean distance between predicted flow and ground truth. It reflects the absolute performance of the model, but cannot reflect the performance degradation of the model on perturbed data (*i.e.* robustness). Therefore, we propose three evaluation metrics for optical flow robustness benchmarks.

Given the ground-truth flow $\mathbf{f}_{gt}$, Corruption Robustness Error (CRE) measures the EPE difference between predicted flow $\mathbf{f}_{clean}$ on clean data and $\mathbf{f}_{c,s}$ on corrupted data for each corruption c at each severity s:

$$\begin{aligned} \mathrm{CRE}_{c,s} &= \mathrm{EPE}_{c,s} - \mathrm{EPE}_{clean} \\ &= \|\mathbf{f}_{c,s} - \mathbf{f}_{gt}\| - \|\mathbf{f}_{clean} - \mathbf{f}_{gt}\|. \end{aligned} \quad (1)$$

The CRE_c of corruption c is then computed as the average $\mathrm{CRE}_{c,s}$ across all severity s, and CRE is averaged from all corruptions of CRE_c to measure the *absolute robustness*.

Notice that the CRE may correspond to the EPE_{clean}, we propose Corruption Robustness Error ratio (CREr) to investigate the relative stability of the model:

$$\mathrm{CREr} = \frac{\mathrm{CRE}}{\mathrm{EPE}_{clean}}. \quad (2)$$

It removes the impact of the estimation performance of the model on CRE, thus only reflecting the *relative robustness*.

For the situation $\mathbf{f}_{gt}$ is not known, the Relative Corruption Robustness Error (RCRE) is performed. It straightly measures the end-point-error between $\mathbf{f}_{clean}$ and $\mathbf{f}_{c,s}$, without the reference of $\mathbf{f}_{gt}$:

$$\mathrm{RCRE}_{c,s} = \|\mathbf{f}_{c,s} - \mathbf{f}_{clean}\|. \quad (3)$$

RCRE_c and RCRE are then calculated the same way as CRE_c and CRE. The calculating detail is illustrated in Fig 3.

4.2. Robustness Benchmarks

The most popular downstream applications of optical flow are autonomous driving [5, 71] and video editing [6, 7, 30, 61]. To enhance the completeness of our robustness test scenarios for optical flow estimation, we create benchmarks for each application using real-world datasets.
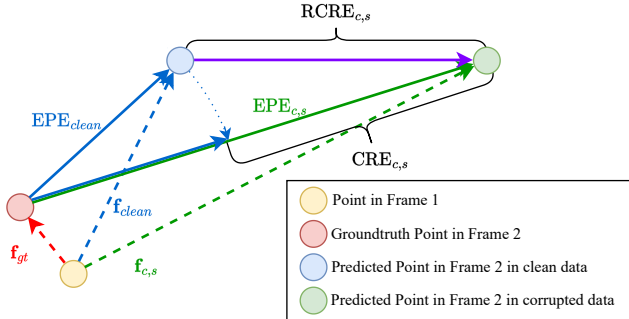
Figure 3. **Calculating procedure of CRE$_{c,s}$ and RCRE$_{c,s}$.** RCRE$_{c,s}$ is computed without using ground-truth optical flow.

Model	Type	Encoder	Decoder	Year
Farnebäck [10]	Knowledge	-	-	2003
DIS [26]	Knowledge	-	-	2016
RAFT [70]	Supervised	CNN	GRU	2020
GMA [20]	Supervised	CNN	GRU	2021
CSFlow [59]	Supervised	CNN	GRU	2022
SKFlow [69]	Supervised	CNN	GRU	2022
GMFlowNet [86]	Supervised	CNN	GRU	2022
CRAFT [66]	Supervised	CNN&Transformer	GRU	2022
FlowFormer [17]	Supervised	Transformer	GRU	2022
FlowFormer++ [63]	Supervised	Transformer	GRU	2023
SAMFlow [89]	Supervised	Transformer&SAM	GRU	2024
FlowDiffuser [37]	Supervised	Transformer	Diffusion	2024
ARFlow [31]	Unsupervised	CNN	Pyramid	2020
UPFlow [38]	Unsupervised	CNN	Pyramid	2021
BrightFlow [40]	Unsupervised	CNN	GRU	2023

Table 1. **All of the** 15 **evaluated methods.** Traditional methods, supervised models with different architectures, and unsupervised models with different designs are included.

There are various proposed optical flow datasets, but most of them are synthetic [4, 9, 42, 43]. Real datasets [13, 24, 45] need large and heavy devices to record data and compute the optical flow ground-truth, thus all of them are capturing the scenes in city roads. For autonomous driving, we construct an optical flow corruption benchmark KITTI-FC based on the current KITTI-2015 optical flow benchmark [45], which has 200 image pairs with optical flow ground truth, recorded from a car driving through streets. Another considered scenario is video editing, in which optical flow is widely used to provide spatial-temporal consistency. To enable the benchmark for this situation, Go-Pro [49], a dataset commonly used in video tasks like video interpolation [21] and deblurring [68], is performed to build the proposed GoPro-FC corruption benchmark.

Supervised optical flow models [17, 20, 66, 70] always face Out-Of-Domain (OOD) problems during real-world deployment, as the ground-truth optical flow is hard to obtain. Sometimes the ground-truth optical flow is known, and OOD models could be fine-tuned to be In-Domain (ID), leading to more satisfying performance. We thus research both the OOD and ID robustness of supervised models to provide a systematic evaluation framework.

KITTI-FC. The KITTI-FC is constructed upon 20 corruptions, excluding *Object Motion Blur* and 3 video corruptions, since 1) original KITTI data are recorded in 10FPS, thus *Object Motion Blur* which performs video interpolation and accumulation may cause artifacts, 2) optical flow data provided by KITTI do not contain enough frames to implement the proposed 3 video corruptions. Since the KITTI optical flow dataset provides ground truth for 200 image pairs, we construct both OOD and ID benchmarks based on them, by splitting 80 image pairs for robustness evaluation and 120 for ID model training. It focuses on the large displacement due to its 10FPS frame rate.

GoPro-FC. Based on GoPro, we construct GoPro-FC, which consists of 5 challenging dynamic video sequences under all the 24 corruptions we proposed in Sec. 3. The Go-Pro dataset is recorded at $240Hz$, and we sample images in 30FPS to construct the final test image pairs in GoPro-FC. For *object motion blur*, we perform a video interpolation

framework FLAVR [21] for $4\times$ interpolation first, obtaining a 960FPS high frame rate video. Then we sample the same 30FPS test corrupted image pairs as the other corruptions, while each corrupted image is blurred by accumulating the previous and next frames. Note that the consistent frames in GoPro-FC are in 30FPS, the benchmark reflects the robustness of models in small displacement scenarios.

5. Benchmarking Results

5.1. Implementation Details

We evaluate 29 model variants of 15 commonly used methods for comprehensive research of optical flow estimation robustness, as categorized in Tab. 1.

Knowledge-driven methods. We use the OpenCV implementation for *Farnebäck* [10] method and official implementation for DIS [26] with default settings.

Supervised methods. Supervised models are firstly trained on mixed data combining FlyingThings [42], Sintel [4], and HD1K [24], utilizing the officially provided pre-trained C+T checkpoints to obtain Out-Of-Domain (OOD) models. These OOD models are evaluated on both the KITTI-FC and GoPro-FC benchmarks. Subsequently, the OOD models are fine-tuned on the training split of KITTI-FC to establish the in-domain (ID) benchmark within KITTI-FC. For SAMFlow, "-T" indicates the SAM-tiny [23] model, "-B" indicates the SAM-B model, and "-H" indicates the SAM-H model. Note that all the GRU decoders are trained and tested with 12 iterations to promise fair comparison. The training procedures adhere to the official setup outlined in the supplementary materials to ensure optimal performance.

Unsupervised methods. Benefiting from their capability to be trained on unlabeled data, the unsupervised methods are evaluated exclusively on the ID benchmark of KITTI-FC, utilizing models trained on the KITTI raw dataset.

Model	Clean	Digital				Illumination				Weather				Noise			Blur					AVG EPE
		JPEG	Pixelate	Contrast	Saturate	HL*	LL*	OE*	UE*	Spatter	Fog	Frost	Snow	Gaussian	Shot	Impulse	Gaussian	Defocus	Glass	Camera	PSF	
Farnebäck	25.60	26.02	26.71	32.82	25.33	25.76	27.54	27.56	28.60	27.84	31.19	29.68	27.62	24.35	24.33	24.16	31.00	31.39	28.75	28.78	30.60	28.00
DIS	20.56	21.02	20.44	20.82	20.38	20.09	21.97	24.35	27.23	24.33	24.59	27.23	23.20	21.40	21.05	21.41	20.13	20.14	20.37	20.33	20.13	22.03
out-of-domain (OOD)																						
RAFT	4.29	10.28	4.53	4.77	5.73	5.20	7.25	6.74	4.34	21.49	6.34	27.75	14.01	13.77	12.22	14.20	6.15	6.31	9.02	5.22	5.41	9.54
GMA	4.19	10.01	4.69	4.63	5.70	4.97	7.36	6.21	4.19	23.40	6.04	28.98	14.15	15.02	13.55	15.99	6.41	6.96	10.38	5.26	5.50	9.97
CSFlow	4.11	9.24	4.39	4.79	**5.13**	4.88	7.47	6.53	4.15	21.52	6.23	27.54	13.98	10.82	**9.75**	10.95	5.56	5.93	8.55	5.10	5.06	8.88
SKFlow	3.97	8.86	4.21	**4.54**	5.24	**4.53**	6.68	5.63	3.97	**20.75**	5.55	**26.54**	13.65	14.67	13.52	15.54	6.04	6.45	9.17	5.07	5.23	9.29
GMFlowNet	4.08	9.41	4.52	4.85	5.31	4.93	8.05	5.98	4.28	21.52	6.33	27.03	11.57	12.47	11.32	13.04	5.55	5.91	**8.19**	5.06	5.01	9.02
CRAFT	4.66	10.49	5.12	5.54	6.56	5.51	7.99	7.09	4.80	23.17	6.91	30.13	18.59	17.14	15.69	17.90	7.01	7.33	12.22	6.12	5.99	11.07
FlowFormer	4.78	9.86	5.31	7.16	6.33	5.70	8.51	6.73	4.71	23.25	7.06	28.98	15.10	13.32	11.95	14.20	6.84	6.82	10.51	5.87	6.00	10.21
FlowFormer++	4.70	10.66	5.88	6.48	6.68	6.05	8.80	6.57	4.75	23.35	6.77	29.37	14.45	14.01	13.07	15.04	6.74	6.83	11.48	6.12	6.05	10.46
SAMFlow-T	4.22	9.23	4.76	5.67	5.56	5.11	6.82	5.94	4.12	21.59	5.69	27.66	13.18	11.87	12.51	15.04	6.16	6.18	9.03	4.86	5.39	9.32
SAMFlow-B	4.06	8.09	4.47	5.03	5.25	4.94	6.65	5.78	4.07	21.50	5.45	27.44	11.13	**10.69**	9.85	**10.28**	5.63	5.91	8.93	4.62	4.84	8.53
SAMFlow-H	**3.78**	**7.08**	**4.14**	4.98	5.37	4.73	**6.27**	**5.50**	**3.77**	21.16	**5.30**	27.32	**10.73**	11.81	10.61	10.58	**5.42**	**5.60**	8.49	**4.18**	**4.75**	**8.39**
FlowDiffuser	6.16	14.12	6.60	7.80	8.43	8.12	9.38	9.10	6.10	22.37	8.06	29.07	22.67	19.30	17.49	19.75	9.22	9.43	11.64	8.25	7.83	12.74
in-domain (ID)																						
RAFT	1.81	5.55	2.06	4.26	2.55	3.31	6.81	7.65	1.87	12.00	3.34	20.45	35.76	15.96	10.93	14.84	4.53	4.30	4.47	5.37	4.17	8.51
GMA	2.43	5.69	3.24	3.56	3.00	6.31	4.65	7.56	2.28	14.98	4.31	23.00	51.76	22.51	15.21	21.33	4.20	4.31	4.89	6.17	3.82	10.64
CSFlow	1.96	3.76	2.21	3.12	2.63	2.56	5.04	4.07	2.05	11.18	3.56	19.33	17.32	11.31	8.51	12.96	3.23	3.28	3.47	2.99	2.77	6.27
SKFlow	2.78	3.93	2.61	3.31	3.44	5.18	3.90	7.18	2.81	13.27	3.56	20.41	39.65	23.73	17.09	25.47	3.14	3.28	4.12	4.01	2.81	9.65
GMFlowNet	1.81	3.39	2.00	2.83	2.15	2.70	4.12	8.70	1.94	12.24	3.06	20.17	22.50	15.01	12.67	16.28	3.29	3.33	3.61	3.11	2.89	7.30
CRAFT	3.67	6.45	4.59	4.33	5.32	4.85	4.74	5.77	4.22	13.85	3.54	23.12	31.93	14.72	11.88	14.80	5.44	5.62	7.68	5.00	4.69	9.13
FlowFormer	1.72	3.69	2.32	4.58	2.21	2.67	4.45	3.81	1.96	16.35	3.94	22.63	21.92	8.81	6.78	9.54	4.39	4.40	4.06	3.58	3.56	6.78
FlowFormer++	1.63	3.83	2.41	3.46	2.16	2.19	4.18	2.96	1.60	17.05	3.12	23.74	12.63	9.65	7.56	9.17	3.58	3.76	4.22	3.04	3.15	6.17
SAMFlow-T	1.54	3.19	2.00	3.77	1.86	1.97	3.29	2.15	1.66	16.89	2.86	22.16	11.24	5.14	4.48	5.36	2.97	2.95	3.47	2.77	2.60	5.14
SAMFlow-B	1.51	3.32	2.00	3.07	**1.76**	2.25	3.25	2.12	1.53	17.50	2.56	22.54	**9.68**	5.82	4.62	5.46	2.58	2.65	3.90	2.68	2.33	5.08
SAMFlow-H	**1.45**	**2.82**	**1.91**	**2.52**	1.78	**1.84**	**2.94**	**2.01**	**1.52**	16.82	**2.38**	21.71	10.56	**4.68**	**3.29**	**4.70**	**2.40**	**2.51**	**3.43**	**2.37**	**2.04**	**4.71**
FlowDiffuser	4.71	8.81	4.92	7.94	7.01	7.21	10.85	9.75	5.66	16.87	6.74	24.36	32.40	28.20	24.81	28.15	6.10	6.16	8.50	5.48	5.55	12.77
BrightFlow	3.17	4.78	3.40	3.37	3.36	3.88	4.87	4.23	3.15	17.26	6.04	23.29	14.66	6.56	5.08	6.24	4.03	4.19	4.53	3.85	3.86	6.53
UPFlow	3.52	5.20	3.79	7.22	4.07	4.55	6.31	5.21	4.02	12.64	7.81	22.62	14.87	8.87	7.07	8.26	4.78	4.91	5.15	4.52	4.74	7.33
ARFlow	3.02	3.94	3.06	7.69	3.36	4.66	6.88	3.66	3.44	**8.76**	7.12	**18.52**	10.59	5.30	4.79	5.36	3.29	3.32	3.83	3.86	3.75	5.76

* HL: *High-light*, LL: *Low-light*, OE: *Over Exposure*, UE: *Under Exposure*.

Table 2. **EPE results on KITTI-FC.** 29 model variants on 20 corruptions from 5 corruption classes are reported. Supervised models are evaluated in both OOD and ID settings, whereas unsupervised models are tested solely in ID, given their capability to be trained on unlabeled data. The best EPE results on each corruption are highlighted for OOD and ID respectively.

5.2. Results on KITTI-FC

We first report the EPE of all 29 model variants to give the optical flow estimation performance in Tab. 2, then discuss the absolute robustness CRE and relative robustness CREr in Fig. 4 for deep research on optical flow robustness. Relations between corruptions and representative models are illustrated in Fig. 5. Relative robustness along with computing costs is discussed in Tab. 3.

Since the estimation performance of knowledge-based models and FlowDiffuser on both clean data and corrupted data are not satisfactory, we do not give deep discussion on them. The poor results of FlowDiffuser are mainly from its unstable training process, which lies as a common problem during training diffusion models.

Observation 1: *Absolute robustness of the model depends heavily on estimation performance.* The optical flow estimation performance of the model on corrupted data in Tab. 2 is positively correlated with the estimation performance on clean data. The same pattern can be observed in Fig. 4, where the CRE is positively correlated with the estimation performance on clean data.

Observation 2: *SAMFlow and CSFlow show remarkable estimation performance and absolute robustness among supervised models.* Benefit from the Vision Foundation

Method	Param. (M)	FLOPs↓ (G)	CREr↓ & Rank		Average Rank	
			OOD	ID	OOD	ID
RAFT	5.25	245.6 (4)	1.22 (6)	3.70 (14)	2	7
GMA	5.79	275.4 (6)	1.38 (11)	3.38 (13)	8	8
CSFlow	5.60	251.7 (5)	1.16 (3)	2.19 (5)	**1**	4
SKFlow	6.19	297.7 (8)	1.34 (9)	2.46 (9)	8	6
GMFlowNet	9.30	321.9 (9)	1.21 (5)	3.02 (12)	4	12
CRAFT	6.37	280.9 (7)	1.37 (10)	1.49 (4)	8	5
FlowFormer	16.08	418.5 (11)	1.13 (2)	2.93 (11)	3	14
FlowFormer++	16.01	417.6 (10)	1.22 (6)	2.77 (10)	6	10
SAMFlow-T	28.00	444.3 (12)	1.17 (4)	2.33 (7)	6	8
SAMFlow-B	108.69	545.0 (13)	1.09 (1)	2.36 (8)	4	12
SAMFlow-H	653.84	1026.0 (14)	1.22 (6)	2.25 (6)	11	10
UPFlow	3.49	134.1 (2)	-	1.08 (3)	-	2
BrightFlow	5.25	245.5 (3)	-	1.06 (2)	-	2
ARFlow	2.23	22.6 (1)	-	0.91 (1)	-	**1**

Table 3. **Average ranking of FLOPs and CREr.** The number in the bracket is the ranking of the model for the corresponding item.

Model (VFM), SAMFlow outperforms other supervised models as shown in both Tab. 2 and Fig. 4, building upon the same architecture of FlowFormer. Larger VFM provides better performance and robustness. CSFlow shows close robustness to SAMFlow at the same time, ahead of the other CNN-based models by a large margin.

Observation 3: *Unsupervised methods are more stable to corruptions.* Although unsupervised methods do not perform well in clean data, they achieve better estimation results on corrupted data in Tab. 2 than most supervised models. The CRE results and CREr results in Fig. 4 also demonstrate it. Among unsupervised models, ARFlow shows the
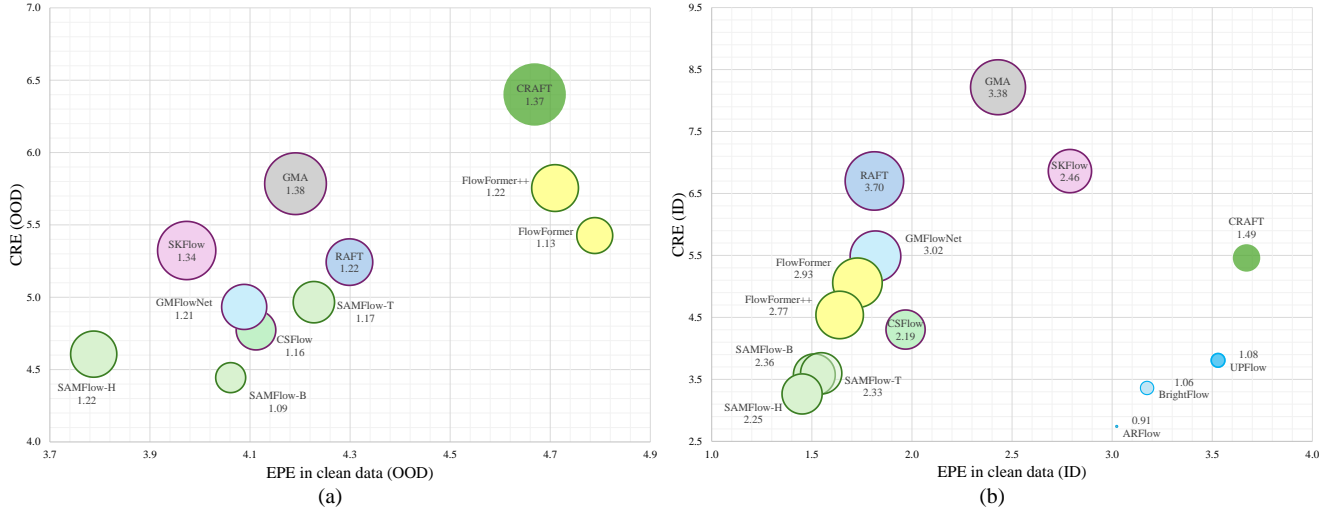
Figure 4. **The EPE, CRE, and CREr results of** 12 **optical flow models on OOD and ID benchmarks of KITTI-FC.** CREr is represented by the size of the bubble and its value is indicated below the model name. Purple circles represent CNN-based models, green circles represent Transformer-based models, and blue circles represent unsupervised models.
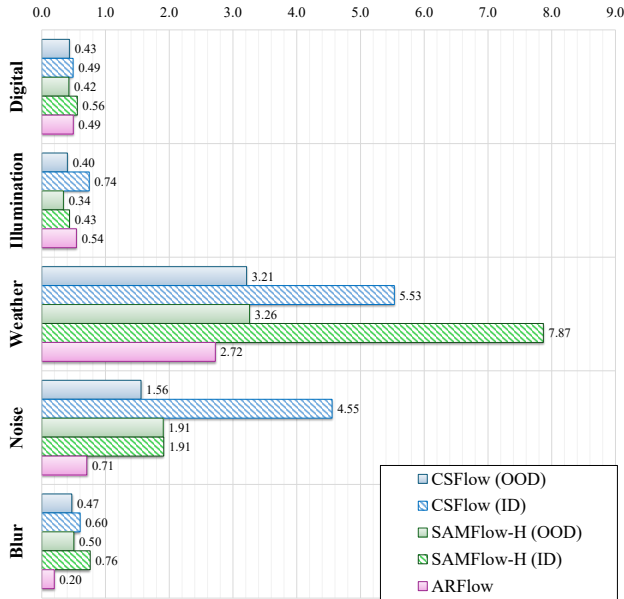

Figure 5. **CREr of representative models on different corruption classes.** Weather corruptions heavily influence the supervised models from OOD to ID.

best absolute robustness and relative robustness.

Observation 4: *The relative robustness of supervised methods drop in ID scenario.* As illustrated in Fig. 4, the CREr of all supervised methods increase from OOD to ID setup, meaning they all experience a robustness drop. Although the CRE of Transformer-based models have decreased, the better absolute robustness is mainly from the estimation performance improvement. This phenomenon shows the supervised models are more likely to trust the clean ID data when fine-tuning, and thus more sensitive to data perturbation.

Observation 5: *Relative robustness drop from OOD to ID is mainly from the instability in Weather conditions.* As

shown in Fig. 5, both supervised methods face a relative robustness drop in all 5 types of corruption. However, the influences of corruptions from Weather for two methods are more serious. Furthermore, CSFlow also experiences serious degradation of robustness in Illumination and Noise conditions, with an observed increase of up to 291% in CREr. On the other hand, ARFlow shows great robustness in Weather and Noise conditions in ID setup, pointing out where the main advantages of unsupervised methods are.

Observation 6: *After considering computing cost, unsupervised methods show the highest priority.* As shown in Tab. 3, ARFlow achieves the best robustness by relying on the smallest computing cost. The other two unsupervised methods also outperform the supervised methods in rankings. Additionally, CSFlow ranks highest among all supervised models in both OOD and ID scenarios.

Observation 7: *Low-Light and Over Exposure are more serious than High-Light and Under Exposure respectively.* As described in Sec. 3, the corruptions are hard to compare each other due to different implementation approaches. However, it is possible to analyze Illumination corruptions, as the simulation of HL is similar to LL, and so is OE to UE. As shown in Tab. 2, All the models perform worse in the Low-Light condition than in *High-Light*, suggesting the low-light condition is more serious for optical flow models. As for the exposure problem, *Under Exposure* does not provide enough difficulty as *Over Exposure* because increasing the EV value will lose more information.

Observation 8: *Corruptions that lose local information are more influential than corruptions that lose visual effects.* We found that the corruptions that make local information totally lost, such as noises, *JPEG compression*, and occlusions (*Spatter*, *Frost*, *Snow*), are much more serious than corruptions that only reduce visual effects (*e.g. Contrast*,

Corruption		RAFT	GMA	CSFlow	SKFlow	GMFlowNet	CRAFT	FlowFormer	FlowFormer++	SAMFlow-T	SAMFlow-B	SAMFlow-H
Digital	JPEG	1.30	1.10	1.38	**0.81**	1.38	0.82	1.07	1.10	1.85	1.76	1.52
	Pixelate	0.19	0.18	0.19	**0.17**	0.19	0.19	0.24	0.29	0.35	0.28	0.26
	Contrast	0.28	0.27	0.26	**0.23**	**0.23**	0.32	**0.23**	0.26	0.34	0.28	0.27
	Saturate	0.45	0.27	0.50	0.28	0.37	0.29	**0.24**	**0.24**	0.99	0.44	0.72
Illumination	HL*	0.25	0.21	0.25	0.20	0.22	0.22	**0.16**	0.17	0.21	0.21	0.19
	LL*	0.40	0.33	0.41	0.31	0.51	0.35	**0.24**	**0.24**	0.39	0.35	0.32
	OE*	0.66	0.44	0.72	0.39	0.62	0.43	0.47	**0.34**	1.12	0.58	1.10
	UE*	0.07	**0.06**	0.07	**0.06**	0.08	0.07	0.08	0.09	0.08	0.07	0.07
Weather	Spatter	7.62	7.63	7.57	7.44	7.55	7.37	**6.99**	7.51	7.90	7.77	7.93
	Fog	0.73	0.53	0.75	**0.31**	0.80	0.48	0.41	0.38	0.70	0.68	0.57
	Frost	11.13	11.19	10.92	**10.09**	10.76	10.88	10.30	10.73	11.05	11.11	11.13
	Snow	3.61	1.21	2.40	1.18	2.04	1.24	1.20	**0.99**	2.61	2.11	3.56
Noise	Gaussian	1.36	1.38	1.59	1.35	1.32	1.30	**0.59**	0.80	0.92	1.04	0.99
	Shot	1.29	1.33	1.44	1.33	1.27	1.27	**0.61**	0.79	0.91	1.03	0.91
	Impulse	1.34	1.33	1.43	1.35	1.27	1.27	**0.63**	0.83	0.95	0.97	0.86
Blur	Gaussian	0.39	0.36	0.36	0.45	0.36	0.40	**0.27**	0.29	0.37	0.36	0.36
	Defocus	0.40	0.37	0.38	0.43	0.36	0.38	**0.27**	0.29	0.38	0.38	0.37
	Glass	0.74	0.68	0.72	0.70	0.64	0.73	**0.63**	0.67	0.70	0.71	0.72
	Camera	0.38	0.35	0.37	0.38	0.36	0.36	**0.28**	0.32	0.39	0.36	0.35
	Object	1.23	1.08	1.19	1.01	1.11	1.14	0.94	**0.93**	1.04	1.04	1.09
	PSF	0.35	0.34	0.34	0.37	0.33	0.35	**0.24**	0.26	0.34	0.35	0.32
Video	ABR	1.64	1.49	1.68	**1.42**	1.61	1.46	1.73	1.73	1.80	1.77	1.80
	CRF	2.57	2.36	2.67	**2.31**	2.57	2.32	2.72	2.70	2.86	2.76	2.83
	BitError	0.53	0.40	0.56	**0.37**	0.78	0.40	0.52	0.56	0.80	0.62	0.75
AVG RCRE		1.62	1.45	1.59	1.37	1.53	1.42	**1.29**	1.35	1.63	1.54	1.62

* HL: *High-light*, LL: *Low-light*, OE: *Over Exposure*, UE: *Under Exposure*.

Table 4. **RCRE results on GoPro-FC.** 11 supervised OOD models are evaluated on all 24 corruptions from 6 classes.
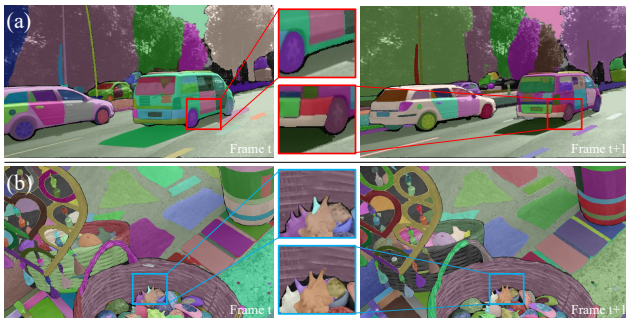

Figure 6. **SAM Segmentation results on (a) KITTI-FC and (b) GoPro-FC.** Small motions in GoPro-FC result the pixel-level segmentation misalignment not helpful for optical flow estimation.

Saturate, *Fog*). For example, as shown in Fig. 2, the corrupted image under *JPEG Compression* looks closer to the clean image than that under *Saturate*, but the estimation performance in Tab. 2 on *Saturate* images is much better than *JPEG Compression* images for all tested models.

5.3. Results on GoPro-FC

As mentioned in Sec. 4, only OOD models are evaluated in GoPro-FC, which consists of small displacement scenarios. Since no optical flow ground truth is provided in GoPro, only RCRE results are reported in Tab. 4, which reflects the absolute robustness of the model.

Observation 9: *FlowFormer shows as the best robust model in small displacement condition.* As shown in Tab. 4, FlowFormer outperforms other supervised models in most corruptions and achieves the best overall RCRE result. In such small displacement conditions, a few corruptions could change neighborhood information heavily, while the Transformer encoder could avoid this high-frequency perturbation. In addition, SKFlow as a CNN-based method,

achieves great results in some corruption conditions and performs as the best CNN method.

Observation 10: *VFM does not help in small displacement condition.* While SAMFlow shines on KITTI-FC, it gets as bad as most CNN methods on GoPro-FC. As shown in Fig. 6, the motion in GoPro-FC is only a few pixels, much smaller than KITTI-FC. In such cases, sufficient fine local features can offer good local correlation search results. However, the segmentation results from SAM may introduce pixel-level misalignment, which is detrimental to the extraction of fine local features, although it can facilitate searching long-range relationships in KITTI-FC.

6. Discussion and Conclusion

This paper presents two optical flow robustness benchmarks for real-world applications including autonomous driving and video editing, along with 24 corruptions which consist of 7 temporal corruptions and 17 single-image corruptions. An advanced *PSF Blur* simulation method is included. Considering the common domain gap problem in practical applications, we established an OOD setting for both benchmarks and an ID setting for KITTI-FC. Through comprehensive evaluation, the results of 29 models from 15 optical flow methods provide 10 meaningful observations. Based on those findings, we also give some suggestions for robust optical flow model development:

- improve estimation performance of the optical flow model for better absolute robustness;
- introduce Transformer-like architecture and semantic information into the model;
- rely on unsupervised models or introduce unsupervised training approach into supervised models;
- reduce the possibility of unreliable local information

in actual application.

We anticipate that our comprehensive benchmarks on corruptions, along with our in-depth findings and analyses, will enhance the understanding of optical flow estimation robustness to corruptions and contribute to the development of more resilient methods for real-world applications.

References

[1] Shashank Agnihotri, Steffen Jung, and Margret Keuper. CosPGD: an efficient white-box adversarial attack for pixel-wise prediction tasks. In *ICML*, 2024. 2

[2] Anurag Arnab, Ondrej Miksik, and Philip H. S. Torr. On the robustness of semantic segmentation models to adversarial attacks. In *CVPR*, 2018. 2

[3] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *ICCV*, 2015. 2

[4] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 5, 14

[5] Linda Capito, Umit Ozguner, and Keith Redmill. Optical flow based visual potential field for autonomous driving. In *IV*, 2020. 1, 4

[6] Ernie Chu, Shuo-Yen Lin, and Jun-Cheng Chen. Video controlnet: Towards temporally consistent synthetic-to-real video translation using conditional image diffusion models. *arXiv preprint arXiv:2305.19193*, 2023. 4

[7] Ernie Chu, Tzuhsuan Huang, Shuo-Yen Lin, and Jun-Cheng Chen. MeDM: Mediating image diffusion models for video-to-video translation with temporal correspondence guidance. In *AAAI*, 2024. 1, 4

[8] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking robustness of 3D object detection to common corruptions in autonomous driving. In *CVPR*, 2023. 2, 3

[9] 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 *ICCV*, 2015. 2, 5, 14

[10] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003. 2, 5, 13

[11] Shaohua Gao, Kailun Yang, Hao Shi, Kaiwei Wang, and Jian Bai. Review on panoramic imaging and its applications in scene understanding. *IEEE Transactions on Instrumentation and Measurement*, 2022. 4

[12] Yao Gao, Qi Jiang, Shaohua Gao, Lei Sun, Kailun Yang, and Kaiwei Wang. Global search optics: Automatically exploring optimal solutions to compact computational imaging systems. *arXiv preprint arXiv:2404.19201*, 2024. 4, 13

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012. 5

[14] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 2, 3

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2

[16] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 1981. 2

[17] 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 *ECCV*, 2022. 1, 2, 5, 14

[18] 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 *CVPR*, 2017. 2

[19] Qi Jiang, Hao Shi, Shaohua Gao, Jiaming Zhang, Kailun Yang, Lei Sun, Huajian Ni, and Kaiwei Wang. Computational imaging for machine perception: Transferring semantic segmentation beyond aberrations. *IEEE Transactions on Computational Imaging*, 2024. 2

[20] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 2, 5, 14

[21] Tarun Kalluri, Deepak Pathak, Manmohan Chandraker, and Du Tran. FLAVR: Flow-agnostic video representations for fast frame interpolation. In *WACV*, 2023. 5, 13, 14

[22] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. In *CVPR*, 2020. 2, 3, 4

[23] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloé Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. In *ICCV*, 2023. 1, 2, 5, 14

[24] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Güssefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, and Bernd Jähne. The HCI benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPRW*, 2016. 5, 14

[25] Tom Koren, Lior Talker, Michael Dinerstein, and Ran Vitek. Consistent semantic attacks on optical flow. In *ACCV*, 2022. 3

[26] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *ECCV*, 2016. 1, 2, 5, 13

[27] Ping Li, Yu Zhang, Li Yuan, Jian Zhao, Xianghua Xu, and Xiaoqin Zhang. Adversarial attacks on video object segmentation with hard region discovery. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 2

[28] Ruoteng Li, Robby T. Tan, and Loong-Fah Cheong. Robust optical flow in rainy scenes. In *ECCV*, 2018. 2

[29] Ruoteng Li, Robby T. Tan, Loong Fah Cheong, Angelica I. Avilés-Rivero, Qingnan Fan, and Carola Schönlieb. Rain-Flow: Optical flow under rain streaks and rain veiling effect. In *ICCV*, 2019. 2

[30] Feng Liang, Bichen Wu, Jialiang Wang, Licheng Yu, Kunpeng Li, Yinan Zhao, Ishan Misra, Jia-Bin Huang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. FlowVid: Taming imperfect optical flows for consistent video-to-video synthesis. In *CVPR*, 2024. 4

[31] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, 2020. 1, 2, 5, 14

[32] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. SelFlow: Self-Supervised learning of optical flow. In *CVPR*, 2019.

[33] Shuaicheng Liu, Kunming Luo, Nianjin Ye, Chuan Wang, Jue Wang, and Bing Zeng. OIFlow: Occlusion-inpainting optical flow estimation by unsupervised learning. *IEEE Transactions on Image Processing*, 2021. 2

[34] Yawen Lu, Qifan Wang, Siqi Ma, Tong Geng, Yingjie Victor Chen, Huaijin G. Chen, and Dongfang Liu. TransFlow: Transformer as flow learner. In *CVPR*, 2023. 2

[35] Bruce D. Lucas. An iterative technique of image registration and its application to stereo. In *IJCAI*, 1981. 1, 2

[36] Ao Luo, Fan Yang, Xin Li, Lang Nie, Chunyu Lin, Haoqiang Fan, and Shuaicheng Liu. GAFlow: Incorporating gaussian attention into optical flow. In *ICCV*, 2023. 2

[37] Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. FlowDiffuser: Advancing optical flow estimation with diffusion models. In *CVPR*, 2024. 2, 5, 14

[38] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. UPFlow: Upsampling pyramid for unsupervised optical flow learning. In *CVPR*, 2021. 2, 5, 14

[39] Sihan Ma, Jing Zhang, Qiong Cao, and Dacheng Tao. PoseBench: Benchmarking the robustness of pose estimation models under corruptions. *arXiv preprint arXiv:2406.14367*, 2024. 2

[40] Rémi Marsal, Florian Chabot, Angélique Loesch, and Hichem Sahbi. BrightFlow: Brightness-change-aware unsupervised learning of optical flow. In *WACV*, 2023. 2, 5, 14

[41] Alexander Mathis, Thomas Biasi, Steffen Schneider, Mert Yüksekgönül, Byron Rogers, Matthias Bethge, and Mackenzie W. Mathis. Pretraining boosts out-of-domain robustness for pose estimation. In *WACV*, 2021. 2

[42] 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 *CVPR*, 2016. 5, 14

[43] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023. 5

[44] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 2

[45] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 5, 14

[46] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 2, 4, 12

[47] Chaerin Min, Taehyun Kim, and Jongwoo Lim. Meta-learning for adaptation of deep optical flow networks. In *WACV*, 2023. 2

[48] Patrick Müller, Alexander Braun, and Margret Keuper. Classification robustness to common optical aberrations. In *ICCVW*, 2023. 2, 4

[49] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, 2017. 5, 14

[50] Yu Qiao, Chaoning Zhang, Taegoo Kang, Donghun Kim, Chenshuang Zhang, and Choong Seon Hong. Robustness of SAM: Segment anything under corruptions and beyond. *arXiv preprint arXiv:2306.07713*, 2023. 2

[51] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. Attacking optical flow. In *ICCV*, 2019. 2

[52] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, 2017. 2

[53] Jérôme Revaud, Philippe Weinzaepfel, Zaïd Harchaoui, and Cordelia Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *CVPR*, 2015. 1, 2

[54] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J. Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *NeurIPS*, 2023. 2

[55] Madeline Schiappa, Shruti Vyas, Hamid Palangi, Yogesh Rawat, and Vibhav Vineet. Robustness analysis of video-language models against visual and language perturbations. *NeurIPS*, 2022. 3

[56] Jenny Schmalfuss, Philipp Scholze, and Andrés Bruhn. A perturbation-constrained adversarial attack for evaluating the robustness of optical flow. In *ECCV*, 2022. 2

[57] Jenny Schmalfuss, Lukas Mehl, and Andrés Bruhn. Distracting downpour: Adversarial weather attacks for motion estimation. In *ICCV*, 2023. 2, 3

[58] Simon Schrodi, Tonmoy Saikia, and Thomas Brox. Towards understanding adversarial robustness of optical flow networks. In *CVPR*, 2022. 3

[59] Hao Shi, Yifan Zhou, Kailun Yang, Xiaoting Yin, and Kaiwei Wang. CSFlow: Learning optical flow via cross strip correlation for autonomous driving. In *IV*, 2022. 2, 5, 14

[60] Hao Shi, Yifan Zhou, Kailun Yang, Xiaoting Yin, Ze Wang, Yaozu Ye, Zhe Yin, Shi Meng, Peng Li, and Kaiwei Wang. PanoFlow: Learning 360° optical flow for surrounding temporal understanding. *IEEE Transactions on Intelligent Transportation Systems*, 2023. 2

[61] Hao Shi, Qi Jiang, Kailun Yang, Xiaoting Yin, Huajian Ni, and Kaiwei Wang. Beyond the field-of-view: Enhancing scene visibility and perception with clip-recurrent transformer. *IEEE Transactions on Intelligent Vehicles*, 2024. 4, 15

[62] Xiangxi Shi and Stefan Lee. Benchmarking out-of-distribution detection in visual question answering. In *WACV*, 2024. 2

[63] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and

Hongsheng Li. FlowFormer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *CVPR*, 2023. 2, 5, 14

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[65] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. SMURF: Self-teaching multi-frame unsupervised RAFT with full-image warping. In *CVPR*, 2021. 2

[66] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. CRAFT: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022. 2, 5, 14

[67] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 1, 2

[68] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *ECCV*, 2022. 5

[69] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. SKFlow: Learning optical flow with super kernels. In *NeurIPS*, 2022. 2, 5, 14

[70] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 1, 2, 5, 13, 14

[71] Hengli Wang, Peide Cai, Rui Fan, Yuxiang Sun, and Ming Liu. End-to-end interactive prediction and planning with optical flow distillation for autonomous driving. In *CVPRW*, 2021. 1, 4

[72] ZiDong Wang, Zeyu Lu, Di Huang, Tong He, Xihui Liu, Wanli Ouyang, and Lei Bai. Predbench: Benchmarking spatio-temporal prediction across diverse disciplines. *arXiv preprint arXiv:2407.08418*, 2024. 3

[73] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. GMFlow: Learning optical flow via global matching. In *CVPR*, 2022. 2

[74] Xing Xu, Jingran Zhang, Yujie Li, Yichuan Wang, Yang Yang, and Heng Tao Shen. Adversarial attack against urban scene segmentation for autonomous vehicles. *IEEE Transactions on Industrial Informatics*, 2021. 2

[75] Wending Yan, Aashish Sharma, and Robby T. Tan. Optical flow estimation in dense foggy scenes with domain-adaptive networks. *IEEE Transactions on Artificial Intelligence*, 2023. 2

[76] Xu Yan, Chaoda Zheng, Ying Xue, Zhen Li, Shuguang Cui, and Dengxin Dai. Benchmarking the robustness of LiDAR semantic segmentation models. *International Journal of Computer Vision*, 2024. 2

[77] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Asia*, 2023. 1, 4

[78] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. In *NeurIPS*, 2021. 2, 3, 13

[79] Zhonghua Yi, Hao Shi, Kailun Yang, Qi Jiang, Yaozu Ye, Ze Wang, Huajian Ni, and Kaiwei Wang. FocusFlow: Boosting key-points optical flow estimation for autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2023. 2

[80] Zijin Yin, Kongming Liang, Bing Li, Zhanyu Ma, and Jun Guo. Benchmarking segmentation models with mask-preserved attribute editing. In *CVPR*, 2024. 2

[81] Jeongbeen Yoon, Sanghyun Kim, Suha Kwak, and Minsu Cho. Optical flow domain adaptation via target style transfer. In *WACV*, 2024. 2

[82] Shuai Yuan, Lei Luo, Zhuo Hui, Can Pu, Xiaoyu Xiang, Rakesh Ranjan, and Denis Demandolx. UnSAMFlow: Unsupervised optical flow guided by segment anything model. In *CVPR*, 2024. 2

[83] Runhao Zeng, Xiaoyong Chen, Jiaming Liang, Huisi Wu, Guangzhong Cao, and Yong Guo. Benchmarking the robustness of temporal action detection models against temporal corruptions. In *CVPR*, 2024. 2

[84] Feihu Zhang, Oliver J. Woodford, Victor Prisacariu, and Philip H. S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 2

[85] Jiaming Zhang, Ruiping Liu, Hao Shi, Kailun Yang, Simon Reiß, Kunyu Peng, Haodong Fu, Kaiwei Wang, and Rainer Stiefelhagen. Delivering arbitrary-modal semantic segmentation. In *CVPR*, 2023. 2

[86] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris N. Metaxas. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022. 2, 5, 14

[87] Yinqiang Zheng, Mingfang Zhang, and Feng Lu. Optical flow in the dark. In *CVPR*, 2020. 2

[88] Hanyu Zhou, Yi Chang, Wending Yan, and Luxin Yan. Unsupervised cumulative domain adaptation for foggy scene optical flow. In *CVPR*, 2023. 2

[89] Shili Zhou, Ruian He, Weimin Tan, and Bo Yan. SAMFlow: Eliminating any fragmentation in optical flow with segment anything model. In *AAAI*, 2024. 1, 2, 5, 14

[90] Zijian Zhu, Yichi Zhang, Hai Chen, Yinpeng Dong, Shu Zhao, Wenbo Ding, Jiachen Zhong, and Shibao Zheng. Understanding the robustness of 3D object detection with bird's-eye-view representations in autonomous driving. In *CVPR*, 2023. 2, 4

A. Corruption Details

A.1. Digital corruptions

JPEG compression. We corrupt every image independently by performing different compression qualities with the JPEG algorithm. The qualities are set to be $[25, 18, 15, 10, 7]$ with the severity ranging from 1 to 5.

Pixelation. The image is first downsampled using a box filter and then resized to the original resolution. With severity ranging from 1 to 5, the downsampled resolution is set to $[0.6, 0.5, 0.4, 0.3, 0.25]$ times the original resolution.

Elastic Transform. A random pixel offset is sampled for every pixel. The sampling rule of the offset is as follows:

$$
\begin{aligned}
dx &\sim \alpha N\left(U\left(-0.05H, 0.05H\right), (0.01W)^2\right), \\
dy &\sim \alpha N\left(U\left(-0.05H, 0.05H\right), (0.01H)^2\right),
\end{aligned}
\quad (4)
$$

where H, W are the height and width of the image, and α is set to $[12.5, 16.25, 21.25, 25, 30]$ with the severity ranging from 1 to 5. Once the dx and dy are sampled, a mesh grid is then computed. The image is remapped into a transformed image according to the mesh grid.

Contrast. The contrast corruption is operated as follows:

$$
\hat{I} = (I - mean\,(I)) \times c + mean\,(I), \quad (5)
$$

where the c is set to be $[0.4, 0.3, 0.2, 0.1, 0.05]$ with the severity ranging from 1 to 5.

Saturate. The RGB image is first transformed into HSV space, and then the saturation value is modified:

$$
\hat{S} = S \times \alpha + \beta, \quad (6)
$$

where (α, β) are in $[(0.1, 0), (0.3, 0), (2, 0), (5, 0.1), (20, 0.2)]$ with the severity ranging from 1 to 5.

A.2. Illumination corruptions

High-Light. The image is first transformed into HSV space, and the brightness value is changed according to the following equation:

$$
\hat{V} = V + c, \quad (7)
$$

where c is in $[0.1, 0.2, 0.3, 0.4, 0.5]$ with the severity ranging from 1 to 5.

Low-Light. Similar to *High-light* corruption, the *Low-light* is as follows:

$$
\hat{V} = V - c, \quad (8)
$$

where c is in $[0.1, 0.2, 0.3, 0.4, 0.5]$ with the severity ranging from 1 to 5.

Over Exposure. The *Over Exposure* corruption considers the two images are in different exposure environments, to simulate the real-world situation that the camera cannot adjust the exposure parameters quickly when the ambient light

changes rapidly, resulting in one image being overexposed. The over-exposure image is simulated by changing the EV offset during image capture:

$$
\hat{V} = V \times 2^{ev}, \quad (9)
$$

where the ev is in $[0.4, 0.8, 1.2, 1.6, 2.0]$ with the severity ranging from 1 to 5.

Under Exposure. The *Under Exposure* operation is similar to *Over Exposure*, except the ev is in $[-0.4, -0.8, -1.2, -1.6, -2.0]$.

A.3. Weather corruptions

Weather corruptions overlay a particle rendering image on the frame to simulate the weather effects. We refer [46] to utilize these corruptions. Because the image pairs used for optical flow estimation are continuous in time, only corruption that changes rapidly over time has time-dependent changes for image pairs. As a result, we only modify Snow as temporal corruptions. While for the other three corruptions, we threat it unchanged for the image pair. For specific parameters, please refer to the official repository. Below we introduce their respective characteristics.

Spatter. The Spatter aims to simulate an environment in which the camera suffers from spatter occlusion, which usually happens when the camera is not cleaned thoroughly or is stained. For different severity, different occlusion ratios are performed.

Fog. Foggy scenarios are a long-standing problem for vision problems, and it is particularly vital for autonomous driving. For different severity, different contrasts of the fog are deployed.

Frost. The Frost happens when the camera is covered with a translucent layer of frost. For different severity, we cover different predefined frost layers in the image.

Snow. The Snow is performed by rendering moving particles. Different severity levels are implemented via different amounts of particles and motion blur. To have it be time-dependent, we maintain the motion direction for particles while re-generating the particles to have the different snow context for the two images in one pair.

A.4. Noise corruptions

Gaussian Noise. The corruption is as follows:

$$
\hat{I} = I + c \times n, n \sim N(0, 1), \quad (10)
$$

where c is set to be in $[0.08, 0.12, 0.18, 0.26, 0.38]$ with the severity ranging from 1 to 5.

Shot Noise. The corruption is as follows:

$$
\hat{I} = \frac{n}{c}, n \sim Pois(I \times c), \quad (11)
$$

where c is set to be in $[60, 25, 12, 5, 3]$ with the severity ranging from 1 to 5.

Impulse Noise. The *Impulse Noise* corruption replaces the values of pixels into 0 or 255. The ratio of the replaced pixels is $[0.03, 0.06, 0.09, 0.17, 0.27]$ with the severity ranging from 1 to 5.

A.5. Blur corruptions

Gaussian blur. We use the Gaussian filter to blur the image. The standard deviation is changed according to the severity. The utilized standard deviation is in $[1, 2, 3, 4, 6]$.

Defocus Blur. A circular mean filter is created and used to convolve the image. The radius of the filter is set to be $[3, 4, 6, 8, 10]$ with the severity ranging from 1 to 5.

Glass Blur. The image is first filtered using a Gaussian filter with standard deviation σ. Then the pixels are shuffled in their neighborhoods, with neighborhood range α and shuffle iterations β. (σ, α, β) are set to be $[(0.7, 1, 2), (0.9, 2, 1), (1, 2, 3), (1.1, 3, 2), (1.5, 4, 2)]$ with the severity ranging from 1 to 5.

Camera Motion Blur. The *Camera Motion Blur* simulates the camera shaking situation. In this case, the whole frame is displaced in one direction and becomes blurred. We assume the shaking direction is the same for both frames, as the shaking is not likely to be rapidly changing in the real world. We first determine the motion radius α and standard deviation σ for the Gaussian filter. Then we sample a motion direction and then blur the two images in that motion direction. The (α, σ) are set to be $[(10, 3), (15, 5), (15, 8), (15, 12), (20, 15)]$ with the severity ranging from 1 to 5.

Object Motion Blur. We only perform *Object Motion Blur* for the GoPro-FC dataset, as it has native 240FPS frames which avoids artifacts caused by interpolation and accumulation. We first perform FLAVR [21], a high-quality video interpolation model for $4\times$ interpolation. Then we select one frame every 31 frames to be the keyframes and accumulate all frames within $[i - c, i + c]$, where i is the key frame index and c is the corruption parameter, ranging in $[3, 6, 9, 12, 15]$.

PSF Blur. We use PSF fields from 5 real existing lenses designed by GSO [12] for each severity. For those PSF blur, the blurry effects are varied from different fields of view. The RMS radius of the used lens is $0.0296mm$, $0.0832mm$, $0.1102mm$, $0.1588mm$, and $0.1939mm$. The pixel width is set to $4\mu m$. The blurred image is obtained by convolving the image with the PSF field. The PSFs of the 5 lens are illustrated in Fig. 7.

A.6. Video corruptions

H.264 CRF. We perform different Constant Rate Factor (CRF) values to be in $[23, 30, 37, 44, 51]$ for the 5 severity following [78].

H.264 ABR. We perform different video bit rate values in $[25M, 12.5M, 6.25M, 3.125M, 1.5625M]$ for the interpo-

Figure 7. **PSF illustration from severity** 1 **to** 5**.** The PSFs of 4 different fields of view are shown.

lated videos in GoPro-FC.

Bit Error. We apply the Bit Stream Filter (BSF) in FFM-PEG with a noise option for the interpolated videos in GoPro-FC, in which the noise parameter is set to be in $[50M, 25M, 15M, 10M, 1M]$.

B. Model Implementation

For all tested models, we use the official repository to create an unbiased benchmark. In this section, we describe the implementations of different categories of the evaluated optical flow estimation models.

B.1. Knowledge-driven methods

Gunnar Farnebäck's algorithm. We use the OpenCV implementation with the default parameters for Gunnar Farnebäck's algorithm [10].

DIS. The DIS [26] is implemented with its RGB version following the official repository.

B.2. Supervised methods

We constrain the same training pipeline for all supervised models, making sure they are trained on the same dataset combination. All tested models perform the same training pipeline as RAFT [70] in their original paper. They

Model	Training Step	Learning Rate	Batch Size	Image Size	Weight Decay	Gamma	Decoder Iteration	Mix Precision
RAFT [70]	100k	1.25e-4	6	368×768	1e-5	0.85	12	✗
GMA [20]	120k	1.25e-4	6	368×768	1e-5	0.85	12	✓
CSFlow [59]	120k	1.25e-4	6	368×768	1e-5	0.85	12	✗
SKFlow [69]	180k	1.75e-4	6	368×768	1e-5	0.85	12	✓
GMFlowNet [86]	160k	1.75e-4	6	368×768	1e-5	0.85	12	✗
CRAFT [66]	120k	1.25e-4	6	368×768	1e-5	0.85	12	✓
FlowFormer [17]	120k	1.25e-4	6	432×960	1e-5	0.85	12	✗
FlowFormer++ [63]	120k	1.25e-4	6	432×960	1e-5	0.75	12	✗
SAMFlow [89]	240k	1.25e-4	3	432×960	1e-5	0.85	12	✗
FlowDiffuser [37]	120k	8.75e-5	6	432×960	1e-5	0.85	12	✗

Table 5. **Implementation of supervised OOD models.** Training parameters are listed.

first pretrain the model on FlyingChairs [9] (C) and then on FlyingThings [42] (C+T). After that, the model is trained on the mixed data consisting of FlyingThings, Sintel [4], KITTI [45], and HD1K [24] (C+T+S+K+H) for Sintel validation. Finally, the model is further fine-tuned on KITTI for KITTI validation. Regarding this, we establish our Out-Of-Domain (OOD) benchmark and In-Domain (ID) benchmark on KITTI-FC. For the OOD benchmark, we build on the pre-trained model from the C+T stage and train on mixed data combined with FlyingThings, Sintel, and HD1K, without KITTI-FC training data. For the ID benchmark, we use the KITTI-FC training data for in-domain information to fine-tune the OOD model.

We have implemented 10 supervised methods, in which the SAMFlow [89] consists of 3 model variants, SAMFlow-T, SAMFlow-B, and SAMFlow-H, with different SAM [23] encoders. The training implementation details are listed in Tab. 5 and Tab. 6 for OOD and ID respectively. All models are trained with the official recommended hyper-parameters, except FlowDiffuser [37] for which we fine-tuned the learning rate to avoid training collapse. Notice that for supervised methods that utilize iterative flow estimation, the official iteration number is 12, thus we also perform 12 estimation iterations during inference in OOD and ID benchmarks for a fair comparison. In addition, the SAMFlow uses gradient accumulation during training due to the large memory requirement of SAM, so training with 3 batch size is equivalent to training with a batch size of 6.

B.3. Unsupervised methods

Since the unsupervised methods can be easily trained on unlabeled data, thus we only test them on the ID benchmark of KITTI-FC. Notice that for our evaluated 3 unsupervised methods, ARFlow [31], UPFlow [38], and BrightFlow [40], official checkpoints on KITTI data are available, we directly test these checkpoints on our ID benchmark of KITTI-FC. For these methods, official configurations are performed.

C. Benchmark Implementation

C.1. KITTI-FC

The 200 image pairs in the original KITTI [45] training set are split into two parts, in which 120 pairs are used for ID model training and the remaining 80 pairs are used for benchmarking.

During data corruption, we corrupt the image pairs from the validation set of KITTI-FC, with 20 corruption types. The 3 Video corruptions and Object-motion-blur corruption are not implemented, as the frames from KITTI optical flow data are not sufficient.

C.2. GoPro-FC

We choose 5 different scenarios in GoPro [49] to construct GoPro-FC robustness benchmark: *GOPR0374_11_02*, *GOPR0379_11_00*, *GOPR0384_11_02*, *GOPR0385_11_00* and *GOPR0386_11_00*. Due to the utilization of Video corruptions, we first convert the image sequences into videos through FFMPEG, then apply $4\times$ interpolation with FLAVR [21] for each video. All the corruptions and image pairs are built upon those $4\times$ videos, to ensure consistent information sources across clean image pairs, non-video corrupted image pairs, and Video corrupted image pairs. Since the original GoPro data are recorded as $240Hz$, the $4\times$ videos are in 960FPS. We sample 30FPS images from $4\times$ videos, *i.e.*, sample final images every 32 frames in the $4\times$ video. The image pairs are constructed from two adjacent frames of the video. Through the above construction approach, each sequence could generate 135 image pairs.

During data corruption, the Video corruptions are performed in those $4\times$ videos, while non-video corruptions are performed directly on the clean image pairs just like the operations in KITTI-FC.

C.3. Comparison of KITTI-FC and GoPro-FC

Since the KITTI-FC data is recorded in $10Hz$, whereas images in GoPro-FC are sampled in 30FPS, the pixel displacement in KITTI-FC is much larger than in GoPro-FC, resulting in slight differences in the test results of the two benchmarks. We counted the displacement distribution of the two

Model	Training Step	Learning Rate	Batch Size	Image Size	Weight Decay	Gamma	Decoder Iteration	Mix Precision
RAFT	50k	1e-4	6	288×960	1e-5	0.85	12	✗
GMA	50k	1.25e-4	6	288×960	1e-5	0.85	12	✓
CSFlow	50k	1e-4	6	288×960	1e-5	0.85	12	✗
SKFlow	50k	1.75e-4	6	288×960	1e-5	0.85	12	✓
GMFlowNet	50k	1.75e-4	6	288×960	1e-5	0.85	12	✗
CRAFT	50k	1.25e-4	6	288×960	1e-5	0.85	12	✓
FlowFormer	50k	1.25e-4	6	432×960	1e-5	0.85	12	✗
FlowFormer++	50k	1.25e-4	6	432×960	1e-5	0.85	12	✗
SAMFlow	50k	1.25e-4	3	432×960	1e-5	0.85	12	✗
FlowDiffuser	50k	1e-4	6	288×960	1e-5	0.85	12	✗

Table 6. **Implementation of supervised ID models.** Training parameters are listed.
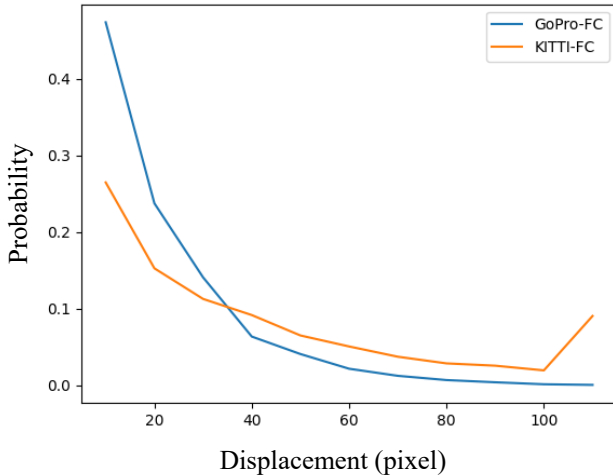
Figure 8. **Optical flow amplitude statistic of KITTI-FC and GoPro-FC.** GoPro-FC contains much more percent of small displacement.

benchmarks, as shown in Fig. 8, where the displacement in GoPro-FC is estimated by RAFT following [61]. The displacement distribution of KITTI-FC is more biased towards large displacements, while the displacement of GoPro-FC is generally smaller. Different benchmark properties may cause the model to perform slightly differently on the two benchmarks.

D. More Results

CRE results on KITTI-FC. We report the complete CRE results in Tab. 7. Notice that although SAMFlow does not perform the best in all items, SAMFlow-B and ARFlow achieve the best robustness in OOD and ID respectively. In addition, due to the computation procedure of CRE, negative CRE values have appeared, demonstrating that the model performs slightly better in corrupted data than in clean data. Given that all observed negative values are small, this suggests that the model is not affected by this corruption in any significant manner.

RCRE results on KITTI-FC. Although the ground truth

is provided in KITTI-FC, we could also compute RCRE without using it. The RCRE of KITTI-FC is reported in Tab. 8. According to RCRE, SAMFlow-B and ARFlow also achieve the best robustness in OOD and ID respectively. It reaches the same conclusion as CRE results, as the CRE and RCRE all represent absolute robustness.

Relationship between EPE, CRE, and RCRE. In our benchmark results, several metrics are used for comparison and analysis. We plot the relationship of them in Fig. 9. EPE in corrupted data, CRE, and RCRE are all positively correlated with the EPE in clean data in OOD or ID conditions. It shows that the absolute performance in corrupted data and absolute robustness are all related to the model performance in clean data, suggesting that models with better performance are generally more robust.

In addition, we investigate the relationship between CRE and RCRE. The computation procedures of these two metrics have a slight difference, as CRE measures the EPE difference whereas RCRE measures the difference of the predicted flow on clean and corrupted data directly. However, as shown in Fig. 9 (d), the CRE and RCRE have almost the same linear correspondence for models. It demonstrates that the CRE and RCRE could both reflect the same absolute robustness of the model.

E. More Discussion

We give more discussion and analyses of the observations in the paper.

Absolute robustness of the model depends heavily on estimation performance. As shown in Fig. 9 (a), (b), and (c), the absolute robustness of the model is positively correlated with the estimation performance on the clean data. However, it is not a linear relationship, as the relative robustness of the model varies.

Low-Light and Over Exposure are more serious than High-Light and Under Exposure respectively. The corruption details described in Sec. A show the difference of corruption in the Illumination class. For High-Light and Low-Light, the images are added or subtracted from the same brightness. However, for KITTI-FC, an autonomous driving dataset with sparse ground truth, we only compute all

Model	Digital				Illumination				Weather				Noise			Blur					AVG CRE
	JPEG	Pixelate	Contrast	Saturate	HL*	LL*	OE*	UE*	Spatter	Fog	Frost	Snow	Gaussian	Shot	Impulse	Gaussian	Defocus	Glass	Camera	PSF	
Farnebäck	0.42	1.11	7.21	-0.27	0.16	1.94	1.96	3.00	2.24	5.59	4.08	2.02	-1.25	-1.27	-1.44	5.40	5.79	3.15	3.18	5.00	2.40
DIS	0.45	-0.12	0.26	-0.19	-0.48	1.40	3.78	6.66	3.77	4.02	6.66	2.64	0.83	0.48	0.85	-0.44	-0.42	-0.20	-0.23	-0.43	1.47
Out-Of-Domain (OOD)																					
RAFT	5.99	**0.24**	0.48	1.44	0.90	2.95	2.45	0.05	17.20	2.05	23.46	9.71	9.48	7.92	9.91	1.85	2.02	4.72	0.93	1.11	5.24
GMA	5.82	0.51	**0.45**	1.51	0.79	3.17	2.02	0.00	19.22	1.86	24.79	9.97	10.83	9.37	11.81	2.23	2.78	6.20	1.07	1.32	5.78
CSFlow	5.14	0.28	0.69	**1.03**	**0.78**	3.37	2.42	0.04	17.41	2.13	23.43	9.87	**6.71**	**5.64**	6.85	**1.45**	1.83	4.45	0.99	0.96	4.77
SKFlow	4.89	**0.24**	0.57	1.28	0.56	2.71	**1.66**	0.00	**16.79**	1.58	**22.57**	9.68	10.70	9.55	11.57	2.08	2.49	5.20	1.10	1.26	5.32
GMFlowNet	5.33	0.44	0.77	1.23	0.85	3.96	1.89	0.20	17.43	2.25	22.95	7.49	8.39	7.24	8.96	1.46	1.83	**4.11**	0.97	0.93	4.93
CRAFT	5.83	0.46	0.88	1.90	0.85	3.33	2.43	0.14	18.50	2.24	25.46	13.92	12.48	11.03	13.24	2.34	2.66	7.55	1.46	1.33	6.40
FlowFormer	5.07	0.53	2.38	1.55	0.92	3.73	1.94	-0.07	18.46	2.27	24.19	10.32	8.54	7.16	9.42	2.06	2.03	5.73	1.09	1.21	5.43
FlowFormer++	5.96	1.18	1.78	1.98	1.34	4.10	1.87	0.05	18.65	2.07	24.67	9.75	9.31	8.36	10.34	2.04	2.13	6.78	1.41	1.35	5.75
SAMFlow-T	5.01	0.53	1.45	1.33	0.88	2.43	1.72	**-0.10**	17.36	1.47	23.44	8.55	8.96	7.65	8.28	1.94	1.96	4.80	0.64	1.00	4.97
SAMFlow-B	4.03	0.42	0.97	1.19	0.88	**2.33**	1.72	0.02	17.44	**1.39**	23.38	7.07	6.64	5.79	**6.23**	1.58	1.85	4.88	0.56	**0.52**	4.44
SAMFlow-H	**3.30**	0.36	1.20	1.59	0.95	2.49	1.72	-0.01	17.38	1.52	23.54	**6.95**	8.03	6.83	6.79	1.63	**1.82**	4.71	**0.39**	0.97	4.61
FlowDiffuser	7.98	0.47	1.68	2.34	2.06	3.28	3.06	0.02	16.27	1.90	22.95	16.61	13.17	11.40	13.67	3.15	3.32	5.60	2.17	1.69	6.64
In-Domain (ID)																					
RAFT	3.75	0.25	2.46	0.74	1.51	5.00	5.84	0.06	10.19	1.53	18.64	33.96	14.15	9.13	13.03	2.72	2.49	2.66	3.57	2.36	6.70
GMA	3.26	0.81	1.13	0.57	3.89	2.23	5.14	**-0.15**	12.56	1.89	20.57	49.34	20.08	12.78	18.91	1.78	1.89	2.46	3.74	1.40	8.21
CSFlow	1.80	0.25	1.16	0.66	0.60	3.08	2.10	0.08	9.22	1.60	17.36	15.36	9.34	6.54	11.00	1.27	1.31	1.51	1.03	0.81	4.30
SKFlow	1.15	**-0.18**	**0.53**	0.65	2.39	**1.12**	4.40	0.02	10.49	0.78	17.63	36.87	20.94	14.30	22.69	0.35	0.50	1.34	1.23	**0.03**	6.86
GMFlowNet	1.58	0.19	1.01	0.34	0.88	2.30	6.89	0.13	10.43	1.25	18.35	20.69	13.20	10.86	14.47	1.48	1.52	1.79	1.30	1.08	5.49
CRAFT	2.78	0.92	0.67	1.65	1.19	1.08	2.10	0.55	10.18	**-0.12**	19.45	28.26	11.05	8.21	11.13	1.77	1.95	4.01	1.33	1.02	5.46
FlowFormer	1.97	0.59	2.86	0.49	0.95	2.73	2.09	0.24	14.63	2.21	20.91	20.20	7.09	5.06	7.81	2.67	2.68	2.34	1.85	1.84	5.06
FlowFormer++	2.20	0.78	1.83	0.53	0.56	2.55	1.32	-0.03	15.41	1.48	22.10	11.00	8.02	5.93	7.54	1.95	2.13	2.58	1.41	1.52	4.54
SAMFlow-T	1.65	0.45	2.23	0.32	0.43	1.75	0.61	0.11	15.34	1.32	20.62	9.70	3.60	2.94	3.82	1.43	1.41	1.93	1.23	1.06	3.60
SAMFlow-B	1.81	0.49	1.56	0.25	0.75	1.74	0.61	0.02	15.99	1.06	21.03	8.18	4.31	3.11	3.96	1.07	1.14	2.39	1.17	0.83	3.57
SAMFlow-H	1.37	0.46	1.07	**0.33**	**0.40**	1.49	**0.56**	0.07	15.38	0.93	20.27	9.11	3.23	1.85	3.26	0.95	1.07	1.98	0.92	0.59	3.26
FlowDiffuser	4.12	0.22	3.26	2.32	2.45	6.11	5.02	0.94	12.14	2.03	19.56	27.60	23.48	20.17	23.38	1.32	1.43	3.90	0.77	0.80	8.05
BrightFlow	1.68	0.26	3.70	0.55	1.02	2.79	1.69	0.49	9.12	4.29	19.10	11.34	5.35	3.55	4.73	1.25	1.39	1.63	1.00	1.22	3.81
UPFlow	1.61	0.23	0.20	0.19	0.71	1.70	1.06	-0.01	14.09	2.87	20.12	11.49	3.39	1.91	3.07	0.86	1.02	1.36	0.68	0.69	3.36
ARFlow	**0.92**	0.05	4.67	0.34	1.64	3.86	0.64	0.42	**5.74**	4.10	**15.51**	7.58	**2.28**	**1.77**	**2.35**	**0.28**	**0.30**	**0.81**	**0.84**	0.73	**2.74**

* HL: *High-light*, LL: *Low-light*, OE: *Over Exposure*, UE: *Under Exposure*.
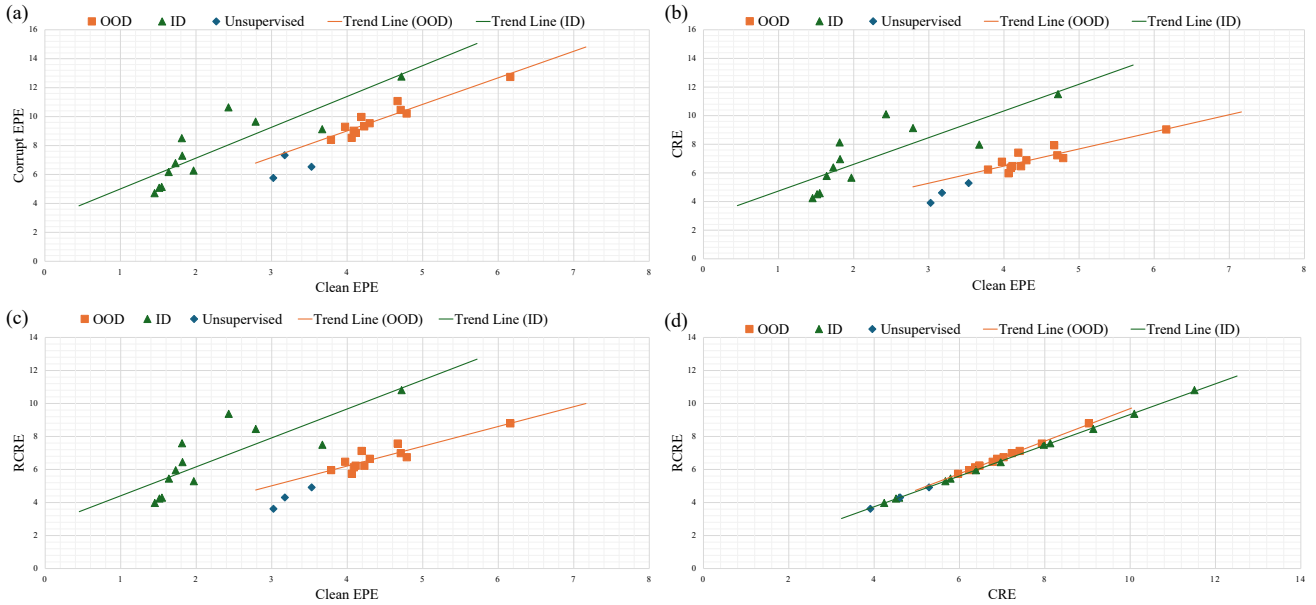
Table 7. **CRE results on KITTI-FC.**

Figure 9. **Relationship between clean EPE, corrupt EPE, CRE, and RCRE.**

Model	Digital				Illumination				Weather				Noise			Blur					AVG RCRE
	JPEG	Pixelate	Contrast	Saturate	HL*	LL*	OE*	UE*	Spatter	Fog	Frost	Snow	Gaussian	Shot	Impulse	Gaussian	Defocus	Glass	Camera	PSF	
Farnebäck	2.34	2.43	10.18	3.30	4.05	4.47	7.46	6.02	6.22	7.99	8.06	9.70	5.54	5.46	5.95	8.10	8.71	5.80	6.61	7.84	6.31
DIS	1.89	0.95	1.89	2.29	2.71	3.74	8.05	10.75	5.41	5.80	8.85	6.48	2.54	2.29	2.66	1.59	1.78	1.99	2.34	1.95	3.80
Out-Of-Domain (OOD)																					
RAFT	7.58	1.47	1.54	2.41	2.02	4.17	3.30	0.44	18.59	3.15	25.23	12.20	10.96	9.26	11.50	3.40	3.78	6.40	2.82	2.72	6.65
GMA	7.46	1.52	**1.31**	2.51	1.78	4.48	2.91	0.40	20.75	2.94	26.57	11.96	12.58	10.82	13.78	3.58	4.22	7.59	2.54	2.68	7.12
CSFlow	6.95	1.21	1.66	**1.99**	2.00	4.77	3.30	0.47	18.99	3.18	25.36	11.95	8.59	**7.15**	8.97	3.07	3.61	6.11	2.76	2.47	6.23
SKFlow	6.32	**1.03**	1.34	2.20	1.46	3.75	2.55	**0.32**	**18.12**	2.44	**24.14**	11.07	12.11	10.81	13.10	3.35	3.80	6.35	2.44	2.67	6.47
GMFlowNet	6.66	1.35	1.57	2.11	1.88	4.99	2.83	0.59	18.66	3.08	24.46	9.06	9.88	8.58	10.48	2.86	**3.29**	5.50	2.40	2.39	6.13
CRAFT	7.23	1.40	1.77	2.88	2.00	4.47	3.32	0.50	19.81	3.25	26.89	15.28	13.81	12.25	14.63	3.60	4.00	8.73	2.82	2.75	7.57
FlowFormer	6.54	1.64	3.54	2.89	2.07	5.01	3.19	0.96	19.77	3.37	25.52	11.96	9.70	8.35	10.62	3.58	3.65	7.04	2.69	2.70	6.74
FlowFormer++	7.36	2.27	2.91	2.97	2.34	5.38	2.84	0.80	20.01	3.18	26.17	11.15	10.64	9.68	11.72	3.47	3.61	7.95	2.72	2.56	6.99
SAMFlow-T	6.70	1.47	2.70	2.70	1.80	3.68	2.61	0.43	18.55	2.44	24.89	10.07	10.17	8.93	9.77	3.14	3.32	**6.06**	2.56	2.50	6.22
SAMFlow-B	5.92	1.46	2.16	2.30	**1.74**	**3.55**	**2.42**	0.39	18.65	**2.42**	24.82	8.69	**8.10**	7.32	**7.92**	3.02	3.31	6.23	**2.20**	**2.20**	**5.74**
SAMFlow-H	**5.16**	1.79	2.41	2.74	2.02	3.83	2.70	0.44	18.55	2.45	24.90	**8.58**	9.45	8.36	8.53	**3.01**	3.33	6.10	2.28	2.54	5.96
FlowDiffuser	10.43	2.81	3.74	4.02	4.18	5.65	4.62	1.52	18.36	4.05	25.20	18.84	15.61	13.60	16.30	5.38	5.44	7.56	4.43	4.14	8.79
In-Domain (ID)																					
RAFT	4.65	0.83	3.22	1.38	2.33	5.77	6.52	0.29	11.32	2.47	19.99	35.16	15.20	10.09	14.06	3.64	3.49	3.59	4.50	3.29	7.59
GMA	4.22	1.52	2.49	1.47	4.55	3.42	6.31	0.44	14.19	2.99	22.39	51.06	21.14	13.84	19.96	3.06	3.31	3.91	4.64	2.60	9.37
CSFlow	2.87	0.87	1.89	1.38	1.83	3.97	2.83	0.32	10.00	2.31	18.31	16.75	10.74	7.66	12.36	2.37	2.47	2.49	2.35	2.02	5.29
SKFlow	3.46	1.19	1.84	1.22	3.17	3.25	5.00	0.33	12.72	2.25	20.40	38.37	22.05	15.26	23.82	2.84	3.12	3.45	2.97	2.36	8.45
GMFlowNet	2.82	0.75	1.95	0.95	1.56	3.43	7.64	0.37	11.68	2.41	19.84	21.78	14.10	11.71	15.39	2.57	2.70	2.76	2.25	2.25	6.45
CRAFT	4.71	1.68	3.40	2.22	2.42	4.77	2.89	0.76	14.50	3.95	23.97	31.99	13.04	9.76	12.84	3.27	3.51	4.79	2.97	2.52	7.50
FlowFormer	2.84	1.34	3.73	1.21	1.69	3.60	2.89	0.82	15.66	3.10	22.05	21.39	8.02	5.94	8.80	3.59	3.64	3.29	2.72	2.77	5.96
FlowFormer++	3.10	1.62	2.64	1.31	1.32	3.40	2.10	0.55	16.33	2.27	23.14	12.13	9.01	6.91	8.54	2.95	3.18	3.62	2.32	2.50	5.45
SAMFlow-T	2.40	0.95	2.90	0.89	0.96	2.40	1.16	0.39	16.16	1.92	21.59	10.64	4.39	3.65	4.64	2.13	2.18	2.73	1.98	1.78	4.29
SAMFlow-B	2.55	0.98	2.19	**0.75**	1.29	2.36	1.06	0.24	16.80	1.65	21.99	**9.06**	5.08	3.83	4.74	1.74	1.86	3.16	1.91	1.52	4.24
SAMFlow-H	2.11	1.06	**1.76**	0.83	**0.95**	**2.15**	1.09	0.33	16.24	**1.55**	21.26	9.98	4.00	**2.55**	**4.04**	1.71	1.89	2.85	1.69	**1.37**	3.97
FlowDiffuser	6.45	2.63	5.82	4.45	4.73	8.80	7.70	3.00	15.59	4.74	23.64	31.01	26.20	22.59	26.34	4.25	4.46	6.04	3.82	3.91	10.81
BrightFlow	2.43	0.74	4.92	1.30	1.93	3.96	2.90	1.14	10.24	5.43	20.65	13.37	6.94	4.88	6.32	2.14	2.36	2.55	1.98	2.16	4.92
UPFlow	2.49	0.64	0.68	0.78	1.54	2.70	1.79	**0.20**	15.29	3.58	21.70	13.03	4.57	2.86	4.32	1.85	2.18	2.61	**1.49**	1.78	4.30
ARFlow	**1.62**	**0.33**	6.05	0.97	2.41	5.15	1.27	0.78	**6.73**	5.23	**16.92**	9.18	**3.30**	2.62	3.35	**0.85**	**0.93**	**1.52**	1.70	1.41	**3.62**

* HL: *High-light*, LL: *Low-light*, OE: *Over Exposure*, UE: *Under Exposure*.

Table 8. **RCRE results on KITTI-FC.**

the metrics in pixels with ground truth. In this case, the estimated results of the bright sky area will be ignored, while the estimated results of the relatively dim road, vehicles, buildings, and trees will be adopted. As a result, Low-Light will lead to a more serious result, as the dim parts are more likely to suffer from losing intensity information. For Over Exposure and Under Exposure, the situation is a little different. The brightness change of each pixel is independent and only depends on its initial brightness, and this brightness change is exponential. In this case, Under Exposure will retain more information than Over Exposure, especially for 8-bit images.

Corruptions that lose local information are more influential than corruptions that lose visual effects. As shown in the paper, the corruptions like Saturate that are quite visually different from the original image can be less influential than that with little visual difference like JPEG. This is relevant to the task of the optical flow model as a correlation estimation. Although some corruptions reduce visual effects for humans, they contain the basic intensity and context information for establishing the correlation between the two images. However, corruptions like JPEG lose neighborhood information, leading to ambiguous local correlation and low-quality estimation results. the same reason is for

occlusion corruptions like Snow, Frost, and Spatter. However, Pixelate should be noticed. It also reduces the local information in principle but does not have much impact on the estimation results. The main reason is that all optical flow estimation methods compute optical flow in the downsampled images to reduce computation cost, and the full-resolution optical flow is obtained by upsampling the low-resolution optical flow. As a result, the optical flow models are robust to the Pixelate corruption.

F. Impacts and Limitations

F.1. Social impacts.

In real-world applications, the robustness of the model is always a serious problem, especially for dense prediction tasks like optical flow estimation, in which the degradation of each pixel has a direct impact on the result. Since there is no existing corruption robustness benchmark established for optical flow estimation, this work provides a preliminary attempt to systematically research the robustness of the optical flow model. The two benchmarks are from two popular datasets, KITTI and GoPro, which have been used for autonomous driving and video editing applications for a long time. So our proposed two benchmarks could help

researchers deeply understand the robustness of the optical flow model in these two mainstream areas, and further improve the effects of optical flow estimation in a wide of downstream applications.

F.2. Limitations.

Although two comprehensive benchmarks are constructed, along with the evaluation of 29 model variants, this work still faces some limitations and there is space for further improvement in the future.

First, while our benchmark considers a wide range of corruptions, it does not account for scenarios in which the model is subjected to multiple corruptions simultaneously.

In addition, our benchmark focuses on the two-frame optical flow model, while some multi-frame optical flow estimation methods have been proposed recently, with great estimation performance. As a result, investigating the robustness of them is also interesting and important.

Furthermore, this study is proposed towards image-based optical flow methods and does not evaluate the robustness of optical flow methods designed for other types of sensors. We intend to further incorporate other modalities such as event sensors and panoramic cameras.