Distracting Downpour: Adversarial Weather Attacks for Motion Estimation

Jenny Schmalfuss Lukas Mehl Andrés Bruhn
Institute for Visualization and Interactive Systems, University of Stuttgart
firstname.lastname@vis.uni-stuttgart.de

Abstract

Current adversarial attacks on motion estimation, or optical flow, optimize small per-pixel perturbations, which are unlikely to appear in the real world. In contrast, adverse weather conditions constitute a much more realistic threat scenario. Hence, in this work, we present a novel attack on motion estimation that exploits adversarially optimized particles to mimic weather effects like snowflakes, rain streaks or fog clouds. At the core of our attack framework is a differentiable particle rendering system that integrates particles (i) consistently over multiple time steps (ii) into the 3D space (iii) with a photo-realistic appearance. Through optimization, we obtain adversarial weather that significantly impacts the motion estimation. Surprisingly, methods that previously showed good robustness towards small per-pixel perturbations are particularly vulnerable to adversarial weather. At the same time, augmenting the training with non-optimized weather increases a method's robustness towards weather effects and improves generalizability at almost no additional cost. Our code is available at https://github.com/cv-stuttgart/DistractingDownpour.

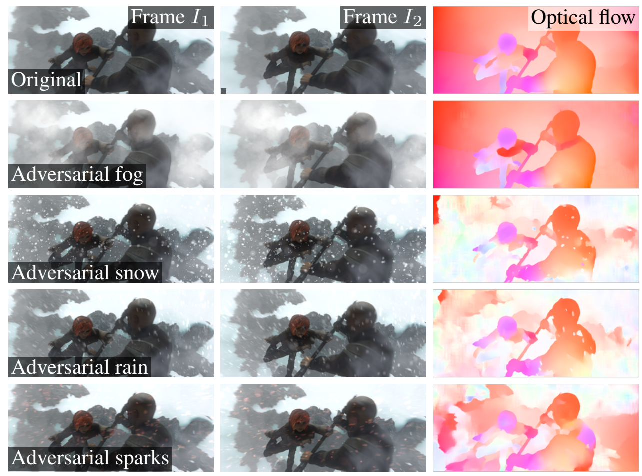

Figure 1. Weather attacks with *adversarial fog, snow, rain* and *sparks* to perturb optical flow estimation with GMA [15]. Our weather attacks obey the 3D geometry and camera motion, which is visible in the dynamic motion blur.

1. Introduction

Adversarial attacks that pose a severe threat to neural networks have recently been introduced in the context of optical flow. There, the goal is to compute the pixel-wise 2D motion f between two consecutive frames I_1 and I_2 of an image sequence over time. Current attacks on optical flow [1, 17, 31, 36, 37] modify these two frames in the 2D space and consequently ignore the actual 3D geometry of the scene as well as the objects moving within. Moreover, when modifying pixels, they impose bounds on the perturbation's L_p norm rather than imposing visual constraints, which yields attacked images that lack naturalism. Therefore, robustness analyses with these attacks might not necessarily reflect the robustness of optical flow methods in the real world – where perturbations are more likely to appear in the form of weather phenomena.

This work investigates whether naturally occurring weather effects like snow, rain or fog can be manipulated to serve as adversarial samples for motion estimation. However, simulating weather in this context requires special care: First, the motion of weather elements should be consistent with the 3D geometry of the scene. Snowflakes should disappear behind objects and their falling distance should appear larger when closer to the camera. Second, their motion should be coherent in time. A raindrop should fall from top to bottom over the first and second frame, and a fog cloud between two objects should remain there – even if the camera moved or rotated.

Taking all this into account, we propose an adversarial attack framework that augments images with particle-based weather effects that feature a high degree of realism: We create weather particles with a view-consistent 3D motion over time, insert them into the 3D scene in a depth-aware manner, and ensure photo-realism through visual effects. This enables us to generate adversarially manipulated weather that significantly deteriorates optical flow predictions, while still satisfying the spatiotemporal and visual constraints of naturalistic weather. Our proposed augmentation and attack procedure can generate a wide range of particle effects, where single particles or super-particles move independently of the remaining scene content. Fig. 1 shows examples of adversarial snowflakes, rain streaks, fire sparks and fog clouds, that differ in size, speed or motion blur, color and transparency.

Contributions. Our contributions are three-fold:

(i) We present a differentiable particle-to-scene rendering framework that generates realistically moving particles in the 3D scene over multiple time steps. It supports a multitude of particle effects ranging from rain and snow over sparks to mist and fog.

(ii) Based on this differentiable rendering framework, we devise the first adversarial weather attacks for optical flow. They optimize 3D spatial positions and color properties of particles in the scene rather than 2D per-pixel perturbations, resulting in highly realistic images with regard to particle motion and appearance.

(iii) While being visually indistinguishable from benign weather augmentations, our adversarial weather achieves significant degradations of optical flow predictions. Interestingly, this particularly holds for methods with high robustness towards small L_p perturbations.

2. Related work

Tab. 1 provides an overview of weather attacks or spatiotemporal weather augmentations, without direct links to motion estimation and optical flow. Before we discuss these methods in more detail, we review attacks and robustness towards weather for motion estimation with optical flow.

Optical flow attacks and robustness to weather. Current optical flow methods based on neural networks are susceptible to adversarially modified input images, which dramatically alter the attacked flow prediction. Existing adversarial attacks on optical flow methods generate either perturbations with small L_p norms [1, 17, 36, 37] or adversarial patches [31]. Koren *et al.* [17] add a constraint to modify semantically coherent pixels only, but none of the attacks introduces geometrical constraints for plausible motion in the 3D space or over time. Regarding the robustness of optical flow towards weather conditions, few methods explicitly consider rain [18, 19], snow [34] or fog [49, 50]. However, adversarial attacks have not yet been used to assess the robustness of optical flow methods towards weather effects.

Adversarial weather attacks. In contrast, adversarial attacks that imitate weather effects have been investigated for classification [7,8,16,23,53], object detection [8,35,52], instance segmentation [8], human pose estimation [44] or autonomous steering [21]. They range from rain [8,23,35,52] over snow [8, 16, 23] to fog [7, 16, 21] and shadows [53]. As these weather attacks have only been applied to single images rather than sequences, they do not consider temporal consistency. With exception of [7], they also neglect the 3D scene geometry. Both shortcomings prevent their application to realistic motion estimation scenarios. Moreover, the visual results of weather attacks are often only moderately convincing [16, 23] compared to conventional, non-differentiable rendering of weather effects [9, 40, 42]. Weather effects and attack capabilities are summarized in Tab. 1.

Method	Weather	Realism	Attack	3D	Tempor.
Adversarial weather attacks					
Sava *et al.* [35]	🌧	●●○	✓	-	-
Zhai *et al.* [52]	🌧	●●●	✓	-	-
Marchisio *et al.* [23]	🌧 ❄	○○○	✓	-	-
Gao *et al.* [8]	🌧 ❄	●○○	✓	-	-
Wang *et al.* [44]	❄	●●○	✓	-	-
Kang *et al.* [16]	🌫❄	●●○	✓	-	-
Machiraju *et al.* [21]	🌫	●●○	✓	-	-
Gao *et al.* [7]	🌫	●●●	✓	✓	-
Realistic weather augmentations					
Rousseau *et al.* [33]	🌧	●○○	-	✓	-
Starik & Werman [38]	🌧	●●○	-	✓	-
Volk *et al.* [41]	🌧	●●●	-	✓	✓
Garg & Nayar [9]	🌧	●●●	-	✓	✓
Halder *et al.* [10]	🌧🌫	●●○	-	✓	✓
Tremblay *et al.* [40]	🌧🌫	●●●	-	✓	✓
von Bernuth *et al.* [42]	🌫❄	●●●	-	✓	✓
Wiesemann & Jiang [46]	🌫	●●●	-	✓	-
Ours	🌧🌫❄	●●●	✓	✓	✓

Table 1. Generating rain 🌧, fog 🌫 and snow ❄ in images. The *methods* may support adversarial *attacks*, respect the scene's *3D geometry* or ensure *temporal* consistency over frames.

Realistic weather augmentations. Before applying any vision-based method in the real world, testing its performance under non-perfect weather conditions is crucial. As a result, there are numerous augmentations to transform clean images into their bad-weather counterparts, *e.g.* via modeled distributions [11, 28], generative networks [20, 29, 32, 43, 45, 51], or classical rendering techniques [40, 42].

However, only a few augmentations respect the 3D geometry of the scene and, ideally, create time-consistent effects for realistic motion of weather across multiple frames and camera perspectives, see Tab. 1. All such augmentations use classical rendering because generative mod-

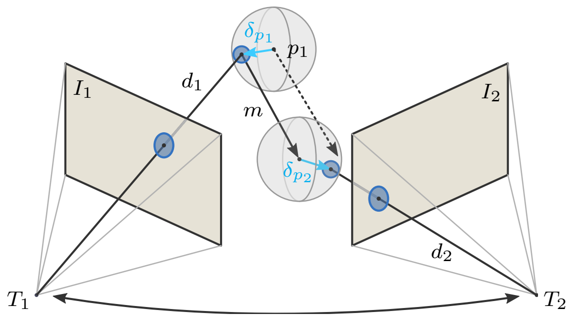

Figure 2. Model for particle motion in the 3D space.

els [20,32,45] cannot ensure the spatiotemporal consistency of their generated effects. Augmentations that respect both, 3D geometry and temporal consistency were proposed for rain [9, 41], rain & fog [10, 40], or fog & snow [42]. Augmentations that respect only the 3D geometry but not the temporal consistency exist for rain [33, 38] and fog [46]. To ensure a realistic 3D motion in time, our attack explicitly models the trajectory of weather particles, which is close in spirit to the augmentation of Halder *et al*. [10], and its extension by Tremblay *et al*. [40]. However, unlike all discussed rendering approaches, our augmentation is differentiable and thus can readily be used for adversarial attacks.

3. Adversarial weather for motion estimation

To study the robustness of optical flow methods towards weather effects, we design an adversarial attack framework that augments image sequences with particle-based weather. There, we augment an image sequence with parametrized particles to simulate snowflakes, rain streaks or fog clouds of realistic appearance and motion. Then, we optimize the particle parameters to cause wrong flow predictions with these snowy, rainy or foggy images.

3.1. Particle-based weather augmentation

The generation of spatiotemporally consistent and visually appealing weather imposes several constraints on the particles: Because motion estimation detects moving objects in a 3D scene, a simple 2D animation of the weather particles in the image plane is not realistic enough. Instead, we model their 3D motion, which also respects object depth and camera motion. Moreover, expanding our pursuit of realism to the appearance of the weather effects, the particles are integrated with appropriate visual effects. These include an occlusion-aware depth placement as well as out-of-focus and motion blur. Finally, the parametrized particles need to be rendered in a differentiable manner to allow their adversarial optimization.

To create weather-augmented 2D images I_1, I_2, we initialize 3D particles and then render them into the images. During the initialization, we generate a fixed set of particles

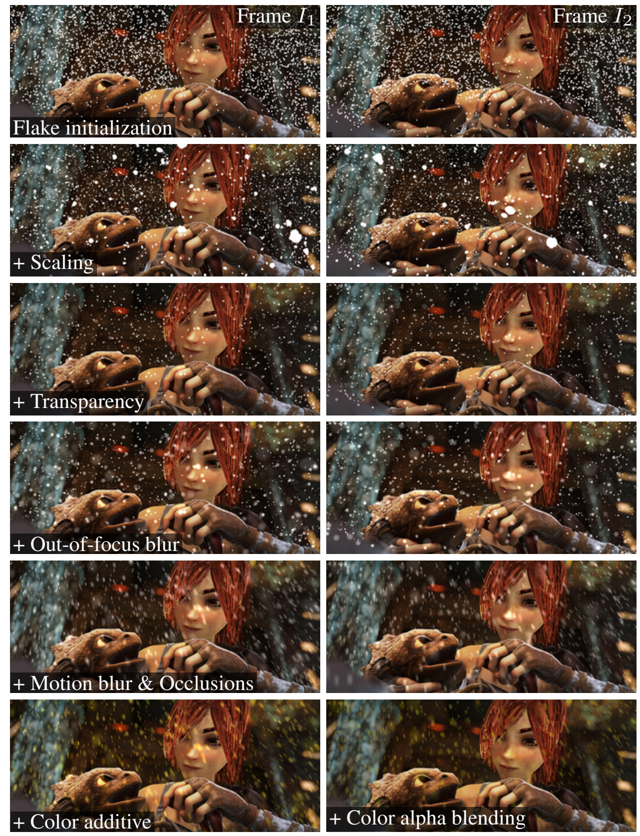

Figure 3. Breakdown of our realistic snow rendering.

$\mathcal{P}$ in the 3D scene and equip them with properties: initial 3D positions p_1, 3D motion m, 3D offsets δ_{p_1} before and δ_{p_2} after the motion, shapes, scaling, color γ and transparencies θ (see Fig. 2 for the motion model). Here, p_1, m, δ_{p_1}, $\delta_{p_2} \in \mathbb{R}^3$ are vectors and $\gamma, \theta \in \mathbb{R}$ scalars. For the differentiable rendering of particles in both frames, we make use of the 3D scene information and assume that a depth map of the scene $D \in \mathbb{R}^{H \times W}$, camera poses $T_1, T_2 \in SE(3)$ and a camera projection matrix P are given. Below, we describe initialization and rendering in more detail.

Weather particle initialization. To initialize the particle positions, we uniformly sample a fixed number of points p_1 from the 3D scene that is visible in the first frame I_1 or – after adding the 3D motion m – in the second frame I_2. Every particle is assigned a 2D gray-scale particle template $B \in \mathbb{R}^{h \times w}$ (billboard), randomly sampled from a template library and rotated by a random angle (Fig. 3, row 1). Then, each particle template is scaled by its particle's inverse depth (row 2), and the particle transparency θ is set to a depth-dependent value (rows 3). Finally, we generate realistic out-of-focus blur by convolving the particle template with a disk-shaped point spread function (row 4).

Weather particle rendering. We render the particles with their associated motion blur, 3D positions, colors and transparencies in the given input frames in four steps, detailed below: We initially add motion blur particles, then project all particle templates onto the image plane, subsequently handle occlusions and finally update the pixels with the colored particle template.

First, if motion blur is added, each initial particle is replaced by K particles. These are evenly spaced along the 3D motion vector and their transparency is reduced to $\frac{1}{K}$; Otherwise, the rendering proceeds as described below. In contrast to simple 2D approximations, this true motion blur respects 3D motion and camera motion (Fig. 3, row 5).

Second, for each particle, we compute the 2D points in both frames from its position in 3D. This yields the center positions p_1^I, $p_2^I \in \mathbb{R}^2$ of the 2D particle templates in the 2D images I_1, $I_2 \in \mathbb{R}^{H\times W\times 3}$. Using the camera projection matrix P and the relative transformation matrix $T_{\text{rel}} = T_2 T_1^{-1}$, we project the 3D points and their motion-displaced positions into the first and second frame, respectively:

$$p_1^I = P\big(p_1 + \delta_{p_1}\big), \tag{1}$$

$$p_2^I = P\big(T_{\text{rel}}(p_1 + \delta_{p_1}) + (m + \delta_{p_2})\big). \tag{2}$$

Because this maps the template to subpixel locations, interpolating the 2D particle templates at the true pixel locations becomes necessary. Using bilinear interpolation enables differentiation w.r.t. the 3D particle positions.

Third, we handle occlusions by multiplying the particle template with a visibility map. The visibility map $V \in \mathbb{R}^{h\times w}$ uses a scene depth map $D \in \mathbb{R}^{h\times w}$, cropped to the location of B, and the particle depth $d \in \mathbb{R}$ per camera:

$$V_t = (1 + e^{\beta(d_t - D_t)})^{-1} \quad \text{for } t = 1, 2. \tag{3}$$

This sigmoid function is $\approx$1 (full visibility) for particles whose depth is smaller than the scene depth, and $\approx$0 (full occlusion) otherwise. When the depths are similar, it creates a smooth transition to allow differentiation. We use sharper transitions $\beta = 250$ for pure rendering and smoother ones $\beta = 30$ for differentiation. Overall, this yields a realistic, occlusion-aware scene integration (Fig. 3, row 5).

Fourth and last, the particle color templates can be applied to the previously computed pixel positions. Our rendering framework supports two color modes for this: additive color blending and alpha blending. Additive blending creates a brightening effect (Fig. 3, last row left), similar to colored light sources, by updating the pixel color as

$$I_c = I_c + \sum_{j\in\mathcal{P}} \gamma_c^j \theta^j B^j \quad \text{for } c = \text{R,G,B}. \tag{4}$$

For each particle, γ_c is the color per channel, θ the transparency scaling and B the template, which itself is a transparency map. In contrast, alpha blending creates more "solid" particles (Fig. 3, last row right) by weighting background and particle color according to particle transparency. We use Meshkin's method [27] for an order-independent alpha blending that can process all particles in parallel:

$$I_c = I_c\Big(1 - \sum_{j\in\mathcal{P}} \theta^j B^j\Big) + \sum_{j\in\mathcal{P}} \gamma_c^j \theta^j B^j \;\text{ for } c = \text{R,G,B}. \tag{5}$$

3.2. Adversarial weather optimization

After the particles $\mathcal{P}$ are initialized and rendered, we adversarially optimize certain weather parameters to change the output $\check{f}$ of optical flow networks towards a desired target flow f^T. In this context, we consider the particle motion offsets δ_{p_1} before and δ_{p_2} after the motion as well as transparency δ_θ and color δ_γ offsets. Other parameters like initial 3D positions, 3D motion and 2D template are fixed. To ensure a valid range of color γ and transparency θ values after the optimization, we transform these bounded variables to unbounded ones η_γ, η_θ via an atanh-transformation [5]

$$\eta_\xi = \operatorname{atanh}(2\xi - 1), \quad \xi = \theta, \gamma \tag{6}$$

and optimize $\eta_\gamma + \delta_\gamma$ and $\eta_\theta + \delta_\theta$ in this domain. Then, our loss function measures the difference between initial and attacked flow via the average endpoint error (AEE) [36]:

$$\mathcal{L}(\check{f}, f^T, \mathcal{P}) = \text{AEE}(\check{f}, f^T) + \sum_{t\in 1,2} \frac{\alpha_t}{|\mathcal{P}|} \sum_{j\in\mathcal{P}} \frac{\|\delta_{p_t}^j\|_2^2}{d_t^j}. \tag{7}$$

Additionally, this loss restricts the magnitude of the motion offset via an α-balanced MSE-like term, where $|\mathcal{P}|$ is the number of particles. It allows larger offsets δ_{p_1}, δ_{p_2} for distant snowflakes, as the same 3D motion in the background yields smaller 2D offsets than in the foreground. Hence, we encourage similar motion offsets in the rendered 2D images by scaling the offsets with the inverse particle depth d.

4. Experiments

In several experiments, (i) we demonstrate our augmentation framework and identify weather that strongly impacts the optical flow estimation, (ii) we attack current optical flow methods with adversarially optimized particles to evaluate their sensitivity and (iii) we augment training data with snow to improve quality and robustness towards weather. A full list of parameters for the experiments is given in the supplement. Our PyTorch framework is available at https://github.com/cv-stuttgart/DistractingDownpour.

In the experiments, we augment frames from Sintel [4], a standard optical flow dataset providing depth and camera information. We calculate the adversarial robustness $\text{AEE}(f, \check{f})$ [36], which measures how the benign optical flow f on unchanged images differs from the optical flow on weather-augmented images $\check{f}$. For robust methods, the output should only change proportional to the input. This is

	Weather	FN2	FNCR	SpyNet	RAFT	GMA	FF
Particles	1000	3.94	5.28	3.55	1.39	1.16	0.83
	2000	7.58	7.94	5.33	2.97	2.51	1.86
	3000	11.95	10.29	6.75	5.03	4.14	3.27
	4000	17.01	12.35	7.75	7.40	5.91	4.42
	5000	**23.42**	**14.62**	**8.67**	**9.81**	**7.91**	**5.53**
Motion blur	0.0	11.95	10.29	**6.75**	**5.03**	**4.14**	**3.27**
	0.0375	**15.60**	12.95	6.44	4.04	3.22	3.17
	0.075	15.01	**13.35**	5.78	3.90	3.04	3.22
	0.1125	13.27	12.97	5.30	3.78	2.76	2.73
	0.15	10.86	11.52	4.64	3.50	2.49	2.05
Color α-bld.	white	**10.03**	**10.70**	**4.95**	**3.88**	**3.74**	**2.93**
	red	9.41	9.03	3.14	2.72	2.56	2.74
	green	6.81	8.46	2.84	2.44	2.35	2.11
	blue	6.32	8.18	2.67	2.69	2.76	1.85
	color	8.20	8.14	3.22	2.91	3.17	2.39
Color additive	white	**14.05**	**14.68**	**6.47**	**5.49**	**4.63**	**4.68**
	red	12.57	12.07	4.21	3.74	2.95	3.03
	green	9.02	9.64	3.68	3.16	2.76	2.56
	blue	7.84	10.47	3.37	3.52	3.31	2.30
	color	11.17	11.50	4.36	4.11	3.75	3.18
Size	small	**5.48**	**5.52**	4.41	**4.58**	**4.41**	**4.47**
	medium	4.45	4.47	5.63	3.04	3.03	2.50
	large	2.23	2.91	3.51	1.17	1.16	0.92
	fog	4.72	5.25	**5.87**	3.59	3.66	3.24

Table 2. Robustness AEE$(f, \check{f}) \downarrow$ [36] of particle-based weather augmentations for *optical flow methods* on Sintel train [4], worst robustness is bold. "α-bld." is "alpha blending". The main augmentations, highlighted in grey, are from top to bottom: snow, rain, sparks and fog. They are visualized in Fig. 4.

formalized by the Lipschitz constant (the concept underlying adversarial robustness), which allows robustness comparisons for input changes of similar magnitude. Note that this robustness definition is independent of the ground truth optical flow, which would be ambiguous for blurred, semi-transparent particles using the classical definition of optical flow. Following [36], we select RAFT [39] & GMA [15], FlowNet2 (FN2) [14] and SpyNet [30] as optical flow methods with either high quality & low robustness, medium quality and robustness or low quality & high robustness, respectively. Additionally, we consider FlowFormer (FF) [13] for its transformer architecture and its top results, and FlowNetCRobust (FNCR) [37] for its robustified design.

4.1. Weather augmentations

To observe how *random augmentations* change the predictions of optical flow methods, we first investigate the impact of various particle effects on Sintel [4] data and select default configurations for snow, rain, sparks and fog.

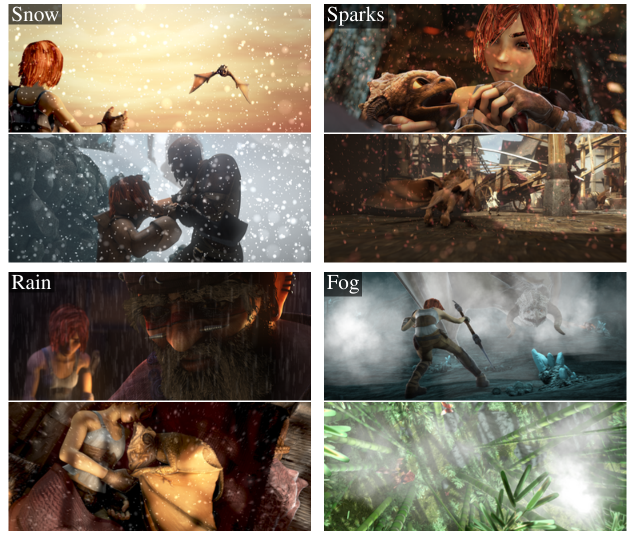

Figure 4. Visual examples for *snow, rain, sparks* and *fog* augmentations on a single frame for highlighted effects from Tab. 2. Note the realistic motion blur (birds-eye view) in column 1 row 4.

Then, we illustrate our flexible rendering on further datasets and finally discuss how augmenting realistic datasets with weather differs from augmenting the synthetic Sintel data.

Particle parameters for weather creation. We create diverse weather effects through complex hyper-parameter combinations in our rendering framework to test their effect on flow predictions. The hyperparameters listed in Tab. 2 are the most prominent ones that were altered, *i.e.* the number of particles, the motion blur length, the color (with different blending modes) and the size. A full list of altered parameters is provided in the supplement.

Tab. 2 summarizes the robustness of optical flow methods on particle-augmented Sintel training data *without* adversarial optimization. Visualizations for all parameter setups are in the supplement. Being most sensitive to the number of particles, all methods change their prediction strongest when many particles are present. The sensitivity also increases for non-transparent effects, *e.g.* for *motion blur: 0.0* or *particle size: small*. Also, large color offsets on multiple channels are strongly perturbing, *i.e.* most for white or random colors. Additive blending perturbs more than alpha blending, but the ranking across colors is the same for both color-blending methods. To summarize, optical flow methods change their predictions significantly in the presence of many small, bright particles, which do not exist in the standard training datasets [4,6,24,26]. However, we find that accurate methods like FlowFormer, RAFT or GMA are more robust, already hinting at an improved particle recognition that is discussed in the next subsection.

Because the most effective configurations, *i.e. motion blur: 0.0*, *color additive: white* and *size: small*, all basically represent snow, we also select setups that represent

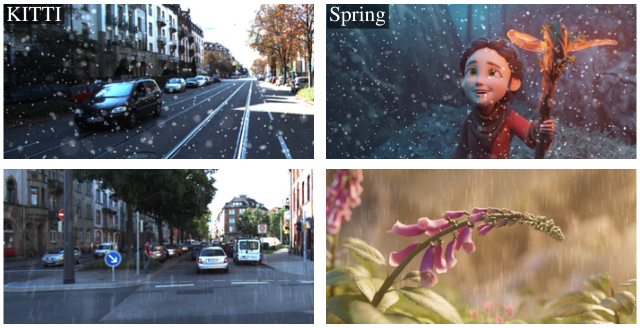

Figure 5. Example augmentations for *KITTI* [26] and *Spring* [25] datasets, with snow (top) and rain (bottom).

Augment.	FN2	FNCR	SpyNet	RAFT	GMA	FF
snow	**8.32**	**7.71**	5.49	**3.08**	**3.27**	4.21
rain	3.70	4.99	3.59	1.51	1.91	2.67
sparks	4.16	4.28	3.15	1.43	1.91	2.73
fog	7.30	7.20	**11.48**	2.98	3.25	**4.83**

Table 3. Robustness AEE$(f, \check{f})$ [36] of random particle-based weather augmentations for *optical flow methods* on KITTI train [26], worst robustness is bold. The results correspond to the highlighted weather augmentations in Tab. 2 on the Sintel dataset.

other weather effects for further experiments. Hence, we choose **snow** (*particles: 3000*), **rain** (*motion blur: 0.15*), **sparks** (*color additive: red*) and **fog** (*size: fog*) as representative effects, which are highlighted gray in Tab. 2 and illustrated in Fig. 4. Note that for snow, *particles: 3000* is computationally more efficient than the most perturbing configuration *particles: 5000*.

Augmenting different datasets. Even though we focus on Sintel, our rendering approach also permits the augmentation of other datasets. Augmented samples from KITTI [26] and Spring [25] are shown in Fig. 5. For KITTI, we use interpolated depth maps and estimate camera poses from the 3D motion in rigid parts of the scene [2].

Weather augmentations on real-world data. To test how well our experiments on synthetic Sintel data transfer to real-world data, we evaluate the robustness values [36] AEE$(f, \check{f})$ on weather-augmented realistic data from the KITTI train dataset [26]. Tab. 3 summarizes the robustness values for all optical flow methods on KITTI data with random augmentations using snow, rain, sparks and fog. Compared to the Sintel augmentations in Tab. 2, snow, rain and sparks (which are based on additive color-rendering) behave similarly, *i.e.* snow is the most effective, sparks and rain have comparable strength.

In contrast, fog has a larger effectiveness on KITTI as it obfuscates more objects because the dataset has fewer foreground objects and more scene depth than Sintel. Another potential cause for its larger effectiveness is its additive color blending in combination with the lighter colors in KITTI scenes. The scenes are often captured in bright or sunny conditions, where further brightening through alpha-blending would cause less color change. However, real situations with snow or rain in bright sunshine are not very common. Therefore, we focus our subsequent evaluation on Sintel data due to its dense depth fields but have seen that results on Sintel data can generally be transferred to real-world scenarios.

Parameters	FN2	FNCR	SpyNet	RAFT	GMA	FF
Initial	10.23	10.68	4.42	3.80	3.77	2.56
δ_{p_1}	13.54	15.65	7.08	7.39	8.64	5.33
δ_{p_2}	11.99	14.21	5.64	5.83	6.69	4.04
δ_{γ}	12.86	15.95	7.52	6.00	7.58	4.74
δ_{θ}	11.70	14.45	6.75	5.29	6.24	3.56
δ_{p_1,p_2}	14.08	15.87	7.71	8.27	9.42	5.49
$\delta_{\gamma,\theta}$	14.06	**16.71**	**8.94**	7.39	8.99	**5.84**
$\delta_{p_1,p_2,\gamma,\theta}$	**14.23**	16.01	7.78	**8.32**	**9.50**	5.71

Table 4. Adversarial robustness AEE$(f, \check{f}) \downarrow$ [36] of adversarial particles, optimized for combinations of *particle parameters* δ_{p_1}, δ_{p_2}, δ_{γ} and δ_{θ} on Sintel-tr115. *Initial* measures the robustness of randomly initialized particles. The most vulnerable setup is bold.

4.2. Adversarial weather attacks

With our framework to generate natural weather effects, we now evaluate the *attack capabilities* of this differentiable weather. First, we investigate the sensitivity of optical flow methods toward optimizing different particle parameters. Second, we attack them with snow, rain, sparks and fog from the previous section. Third and last, we compare the effectiveness of a non-L_p snow attack to previous L_p attacks on optical flow. All attacks use $\alpha_1 = \alpha_2 = 1000$ in the loss, Adam with learning rate 1e-5 and, following [36], a zero-flow target $f^T = 0$ which yields a white flow visualization.

Investigation of weather attack parameters. To understand the impact of adversarial particles on optical flow, we consider artificial weather, initialized with 3000 gray particles that fall down without motion blur. Each particle starts with own initial values $(p_1, p_2, \gamma, \theta)$ *without* offsets $\delta_{p_1} = \delta_{p_2} = \delta_{\gamma} = \delta_{\theta} = 0$. Then, we adversarially optimize the offsets per particle, *i.e.* the positions before the motion δ_{p_1}, the positions after the motion δ_{p_2}, the colors δ_{γ} and the transparencies δ_{θ}. For optimizing δ_{γ}, δ_{θ} and $\delta_{\gamma,\theta}$, we set the learning rate to 1e-3 and use a subset "Sintel-tr115" with 115 frame pairs (the first five per scene) of Sintel-train.

Tab. 4 summarizes the adversarial robustness for the dif-

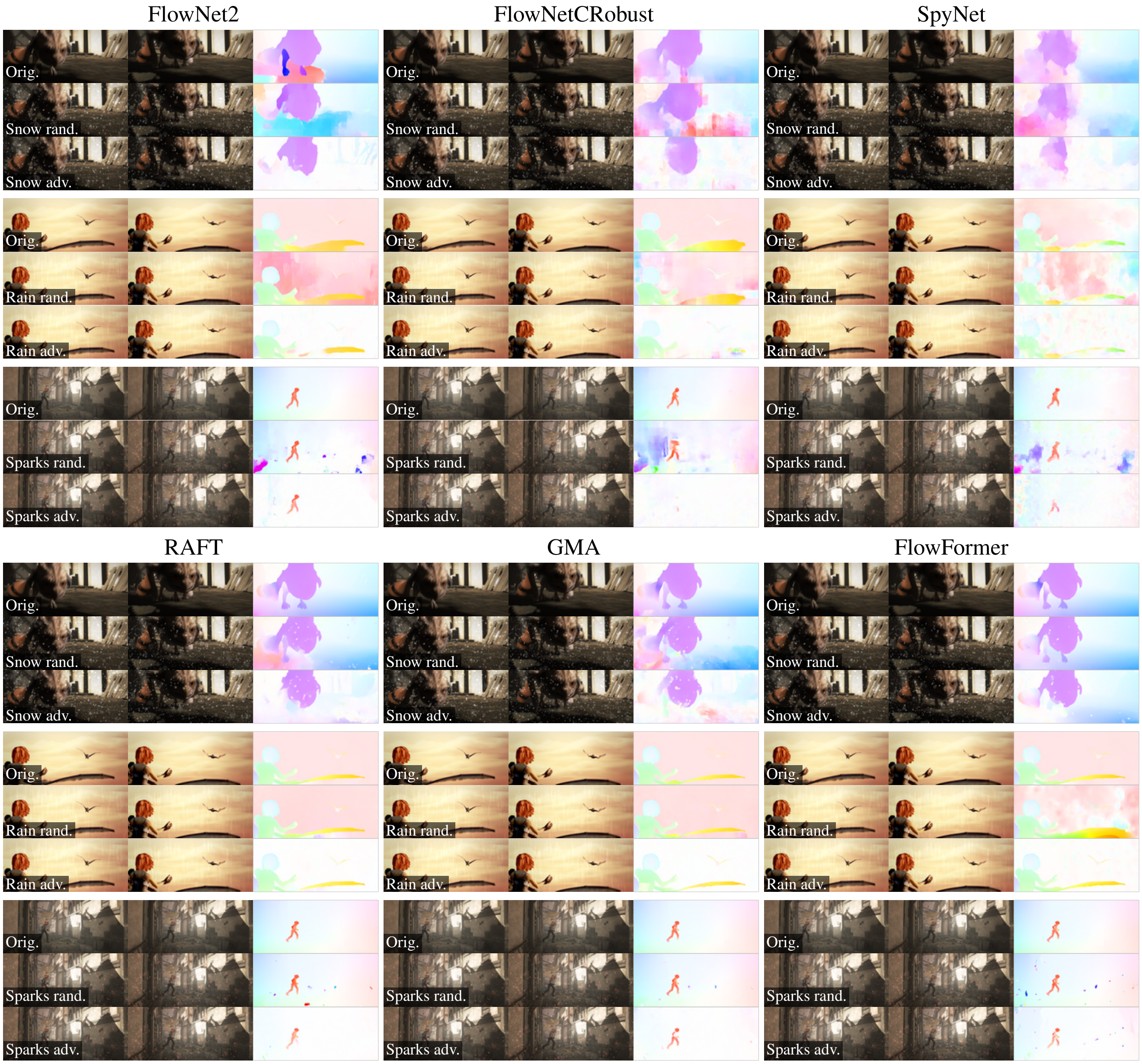

Figure 6. Qualitative results for weather attacks on optical flow predictions for FlowNet2 [14], FlowNetCRobust [37], SpyNet [30], RAFT [39], GMA [15] and FlowFormer [13] (*top left to bottom right*). Images from the Sintel final dataset, with *random* initialization and after *adversarial* weather optimization towards zero-target (white flow). See the supplement for more visualizations.

ferent optimization parameters on all tested optical flow methods. Considering single parameters, the particle offset δ_{p_1} before the motion has the strongest influence. That motion offsets have the strongest influence on motion estimation is intuitively plausible. However, our motion model favors the first motion offset over the second, as δ_{p_1} affects both frames while δ_{p_2} affects only the second one. Jointly optimizing all parameters generally leads to the worst degradation of optical flow estimates. Yet, focusing on motion parameters δ_{p_1,p_2} or hue parameters $\delta_{\gamma,\theta}$ alone also strongly degrades performance. Interestingly, the tested flow methods show either a high sensitivity towards motion, for RAFT and GMA, or a high sensitivity towards hues, for FlowNetCRobust, SpyNet and FlowFormer. For the latter, optimizing hues even yields the strongest degradation overall. This insight is valuable for color-reduced environments, *e.g.* night scenes, where greater independence of the color representation may be wanted.

Robustness against snow, rain, sparks and fog. Next, we transition to more natural attacks with snow, rain, sparks

Attack	FN2	FNCR	SpyNet	RAFT	GMA	FF
snow	21.37	18.23	**9.99**	**11.20**	**10.90**	**7.22**
rain	21.95	**19.85**	8.37	9.53	8.22	5.82
sparks	**22.76**	19.54	8.25	8.72	9.39	6.41
fog	2.32	3.37	2.28	0.92	0.97	0.73

Table 5. Adversarial robustness AEE$(f, \check{f})\downarrow$ [36] for *adversarial snow, rain, sparks* and *fog* on Sintel-tr115. Worst robustness bold.

Attack	FN2	FNCR	SpyNet	RAFT	GMA	FF
PCFA [36]	11.77	13.82	7.83	**12.96**	**12.83**	**14.68**
I-FGSM [37]	7.58	13.69	5.07	11.07	11.40	12.35
Snow (ours)	**16.83**	**16.28**	**9.94**	10.32	9.85	7.10

Table 6. Adversarial robustness AEE$(f, \check{f})\downarrow$ [36] for *different attacks* on Sintel train, the worst robustness per method is bold.

and fog. We optimize all parameters for snow, rain and sparks, but do not optimize δ_{p_2} for fog, keeping it static in the scene. Tab. 5 summarizes the optical flow robustness against adversarial weather, again on Sintel-tr115. For adversarial weather, the methods rank similar to pure augmentation, *cf*. Tab. 2, but the optimization amplifies optical flow changes. For every weather, lower-quality methods, *e.g*. FlowNet2, are very vulnerable while high-quality methods, *e.g*. FlowFormer, are comparatively robust against any weather. For GMA, Fig. 1 visualizes the attacked weather and resulting flows. Remarkably, moving particles eradicate the estimated motion despite their constant movement due to falling and camera motion. When we compare randomly initialized particles to their adversarial counterparts in Fig. 6 their positions hardly differ, making the adversarial sample indistinguishable from random weather to human observers. As adversarial snow greatly affects all optical flow methods, we select it for further analysis.

Comparison to L_p attacks. To conclude our attack evaluation, we compare our adversarial snow attack to previous attacks on optical flow and analyze the performance of optical flow methods in detail. Tab. 6 compares the robustness of optical flow methods under two L_p attacks to our non-L_p attack with adversarial snow on the full Sintel training set. The L_2 attack PCFA [36] is the strongest adversarial attack in the literature, while I-FGSM [37] is a weaker L_∞ attack. Despite being much more constrained by its physically plausible motion, our adversarial snow can compete with PCFA in terms of induced flow perturbation.

Surprisingly, high-quality methods like RAFT, GMA or FlowFormer that suffer most from L_p attacks [36] offer the best robustness towards adversarial snow. Instead, lower-quality methods like FlowNet2 and SpyNet that are most robust towards L_p attacks alter their predictions disproportionately to the added snow particles – or any other particle-based weather (*cf*. Tab. 5). We ascribe the better weather robustness to the more detailed flow estimations of high-quality methods, which detect the localized motion of single particles (*cf*. Fig. 6, snow and sparks on RAFT and FlowFormer, where circular particles are visible). The less accurate methods FlowNet2, FlowNetCRobust and SpyNet instead propagate the detected particle motion over larger areas, rather than attributing it to small moving objects (*cf*. Fig. 6, rows 2/3, where flow predictions have few details). Notably, the robustness of FlowNetCRobust against patch attacks as reported in [37] does not transfer, making it one of the most vulnerable methods irrespective of the attack.

4.3. Training with weather

As all optical flow methods change their predictions significantly in the presence of weather, we end our experiments by presenting a robustifying training strategy. Here, we choose RAFT [39], which is the baseline architecture for GMA and FlowFormer. We retrain RAFT from the author-provided C+T checkpoint according to their training protocol [39] but augment 0%, 50% or 100% of the Sintel final training data with *random* snow. We evaluate the quality, and the robustness towards random augmentations as well as optimized weather attacks, *cf*. Tab. 2 and Tab. 5.

Tab. 7 summarizes the results. Compared to standard training, augmenting any percentage of Sintel-final frames with snow clearly improves the robustness. Furthermore, augmenting half of Sintel clean improves the quality on all datasets and shows a better generalization. It is remarkable that training with random snow has such a positive effect on robustness and quality [39], because training with L_p perturbations does not generally improve the robustness towards adversarial perturbations. For example, FlowFormer [13] augments its training with random noise, but is highly vulnerable against L_p attacks, *cf*. Tab. 6. Therefore, adversarial training [22] is commonly used to improve the robustness against L_p attacks. However, adversarial training (i) significantly increases the training time because adversarial samples are continuously included, leading to a slowly-converging training and (ii) often lowers the quality of non-attacked samples. Both drawbacks are not observed for training with snow augmentations. This makes it particularly noteworthy that a simple augmentation with 50% non-L_p snow improves robustness, quality and generalization at the same time. It also clearly shows that simulated weather has merits for training methods that shall be exploited in the real world, as we find that augmenting 50% of the synthetic Sintel data with snow during training improves the accuracy on real-world sequences from the KITTI dataset.

Snow	Sintel EPE ↓ (te.) clean	Sintel EPE ↓ (te.) final	KITTI ↓ (tr.) F1-all	Augmentation robustness ↓ snow	rain	sparks	fog	Attack robustness ↓ snow	rain	sparks	fog
0%	1.642	3.167	5.65	4.19	3.60	3.64	3.54	9.93	8.02	8.47	**0.87**
50%	1.589	**3.155**	**5.54**	0.91	1.66	**1.29**	**3.52**	3.76	5.96	5.68	0.93
100%	**1.551**	3.384	5.69	**0.83**	**1.37**	1.32	3.57	**3.48**	**5.61**	**5.49**	1.04

Table 7. Training RAFT [39] with *0, 50* or *100% snowy* Sintel-final frames during the Sintel/KITTI (S/K) training phase [39] The *quality* is measured on Sintel test and KITTI train, robustness values for *weather augmentations* on Sintel test and *weather attacks* on Sintel-tr115.

5. Limitations

Although we focus on realism, our attack does not aim at threatening optical flow methods in the real world, where manipulating weather is clearly impossible. While most optical flow attacks [1, 17, 36, 37] are not designed be directly applicable in the real world, our adversarial weather assesses methods under worst-case weather conditions, which is a more realistic attack scenario that even allows significant alterations without being noticeably adversarial. Furthermore, optimizing snow on Sintel-test may take several days on an Nvidia A100 GPU. However these higher computational costs are tolerable in such an offline benchmarking setting.

6. Conclusion

In this paper, we developed a novel framework for adversarial attacks on motion estimation with realistic weather. We proposed a differentiable particle renderer that can be used to generate adversarial weather with a strong impact on optical flow methods. With its realistic appearance, our adversarial weather is hard to notice; yet it lets optical flow networks predict zero-flow although the particles undergo both individual and camera motion. Surprisingly, accurate methods that are very vulnerable to L_p attacks appear to be more robust towards adversarially optimized weather, as they detect the motion of single particles rather than propagating it into the wider image. Finally, we find that augmenting a network's training with unoptimized weather not only improves the robustness towards weather augmentations and attacks but also increases generalization across datasets at a much lower cost than adversarial training.

Also, our weather attacks could easily be extended to problems that also require 3D-awareness or temporal motion consistency. Such attacks would target domains like monocular depth estimation [12, 48], stereo reconstruction [3, 47] or scene flow computation.

Acknowledgments. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (B04). Jenny Schmalfuss is supported by the International Max Planck Research School for Intelligent Systems (IMPRS-IS).

References

[1] Shashank Agnihotri and Margret Keuper. CosPGD: A unified white-box adversarial attack for pixel-wise prediction tasks. In *arXiv preprint 2302.02213*. arXiv, 2023.

[2] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 9(5):698–700, 1987.

[3] Zachary Berger, Parth Agrawal, Tyan Yu Liu, Stefano Soatto, and Alex Wong. Stereoscopic universal perturbations across different architectures and datasets. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2022.

[4] Daniel Butler, Jonas Wulff, Garrett Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 611–625, 2012.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

[6] 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 *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[7] Ruijun Gao, Qing Guo, Felix Juefei-Xu, Hongkai Yu, and Wei Feng. AdvHaze: Adversarial haze attack. In *arXiv preprint 2104.13673*. arXiv, 2021.

[8] Xiangbo Gao, Cheng Luo, Qinliang Lin, Weicheng Xie, Minmin Liu, Linlin Shen, Keerthy Kusumam, and Siyang Song. Scale-free and task-agnostic attack: Generating photo-realistic adversarial patterns with patch quilting generator. In *arXiv preprint 2208.06222*. arXiv, 2022.

[9] Kshitiz Garg and Shree K. Nayar. Photorealistic rendering of rain streaks. *ACM Transactions on Graphics (TOG)*, 25(3):996–1002, 2006.

[10] Shirsendu Halder, Jean-Francois Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10202–10211, 2019.

[11] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proc. International Conference on Learning Representations (ICLR)*, pages 1–16, 2019.

[12] Junjie Hu and Takayuki Okatani. Analysis of deep networks for monocular depth estimation through adversarial attacks with proposal of a defense method. In *arXiv preprint 1911.08790*. arXiv, 2019.

[13] 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 *Proc. European Conference on Computer Vision (ECCV)*, pages 668–685, 2022.

[14] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017.

[15] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9772–9781, 2021.

[16] Daniel Kang, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. Testing robustness against unforeseen adversaries. In *arXiv preprint 1908.08016*. arXiv, 2019.

[17] Tom Koren, Lior Talker, Michael Dinerstein, and Ran Vitek. Consistent semantic attacks on optical flow. In *Proc. Asian Conference on Computer Vision (ACCV)*, pages 1658–1674, 2022.

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

[19] Ruoteng Li, Robby T. Tan, Loong-Fah Cheong, Angelica I. Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. RainFlow: Optical flow under rain streaks and rain veiling effect. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7304–7313, 2019.

[20] Xuelong Li, Kai Kou, and Bin Zhao. Weather GAN: Multi-domain weather translation using generative adversarial networks. In *arXiv preprint 2103.05422*. arXiv, 2021.

[21] Harshitha Machiraju and Vineeth N Balasubramanian. A little fog for a large turn. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2891–2900, 2020.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. International Conference on Learning Representations (ICML)*, pages 1–10, 2018.

[23] Alberto Marchisio, Giovanni Caramia, Maurizio Martina, and Muhammad Shafique. fakeWeather: Adversarial attacks for deep neural networks emulating weather conditions on the camera lens of autonomous systems. In *Proc. International Joint Conference on Neural Networks (IJCNN)*, pages 1–9, 2022.

[24] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.

[25] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4981–4991, 2023.

[26] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015.

[27] Houman Meshkin. Sort-independent alpha blending. In *Game Developers Conference*, 2007.

[28] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Akexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *Proc. Conference on Neural Information Processing Systems Workshops (NeurIPSW)*, 2019.

[29] Siqi Ni, Xueyun Cao, Tao Yue, and Xuemei Hu. Controlling the rain: From removal to rendering. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6328–6337, 2021.

[30] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4161–4170, 2017.

[31] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. Attacking optical flow. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2004–2013, 2019.

[32] Christopher X. Ren, Amanda Ziemann, James Theiler, and Alice M. S. Durieux. Deep snow: Synthesizing remote sensing imagery with generative adversarial nets. In *Proc. SPIE Defense + Commercial Sensing*, pages 196–205, 2020.

[33] Pierre Rousseau, Vincent Jolivet, and Djamchid Ghazanfarpour. Realistic real-time rain rendering. *Computers & Graphics*, pages 507–518, 2006.

[34] Hidetomo Sakaino, Yang Shen, Yuanhang Pang, and Lizhuang Ma. Falling snow motion estimation based on a semi-transparent and particle trajectory model. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1609–1612, 2009.

[35] Paul Andrei Sava, Jan-Philipp Schulze, Philip Sperl, and Konstantin Böttinger. Assessing the impact of transformations on physical adversarial attacks. In *Proc. ACM Workshop on Artificial Intelligence and Security (AiSec)*, pages 79–90, 2022.

[36] Jenny Schmalfuss, Philipp Scholze, and Andrés Bruhn. A perturbation-constrained adversarial attack for evaluating the robustness of optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, pages 183–200, 2022.

[37] Simon Schrodi, Tonmoy Saikia, and Thomas Brox. Towards understanding adversarial robustness of optical flow networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8916–8924, 2022.

[38] Sonia Starik and Michael Werman. Simulation of rain in videos. In *Proc. IEEE/CVF International Conference*

on Computer Vision Workshops (ICCVW), pages 406–409, 2003.

[39] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.

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

[41] Georg Volk, Stefan Müller, Alexander von Bernuth, Dennis Hospach, and Oliver Bringmann. Towards robust CNN-based object detection through augmentation with synthetic rain variations. In *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 285–292, 2019.

[42] Alexander von Bernuth, Georg Volk, and Oliver Bringmann. Simulating photo-realistic snow and fog on existing images for enhanced CNN training and evaluation. In *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 41–46, 2019.

[43] Hong Wang, Zongsheng Yue, Qi Xie, Qian Zhao, Yefeng Zheng, and Deyu Meng. From rain generation to rain removal. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14791–14801, 2021.

[44] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11855–11864, 2021.

[45] Yanyan Wei, Zhao Zhang, Yang Wang, Mingliang Xu, Yi Yang, Shuicheng Yan, and Meng Wang. DerainCycleGAN: Rain attentive CycleGAN for single image deraining and rainmaking. *IEEE Transaction on Image Processing (TIP)*, 30:4788–4801, 2021.

[46] Thomas Wiesemann and Xiaoyi Jiang. Fog augmentation of road images for performance analysis of traffic sign detection algorithms. In *Proc. International Conference on Advanced Concepts for Intelligent Vision Systems (ACVIS)*, pages 685–697, 2016.

[47] Alex Wong, Mukund Mundhra, and Stefano Soatto. Stereopagnosia: Fooling stereo networks with adversarial perturbations. *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 2879–2888, 2021.

[48] Koichiro Yamanaka, Keita Takahashi, Toshiaki Fujii, and Ryuraroh Matsumoto. Simultaneous attack on CNN-based monocular depth estimation and optical flow estimation. *IEICE Transactions on Information and Systems*, pages 785–788, 2021.

[49] Wending Yan, Aashish Sharma, and Robby T. Tan. Optical flow in dense foggy scenes using semi-supervised learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13259–13268, 2020.

[50] Wending Yan, Aashish Sharma, and Robby T. Tan. Optical flow estimation in dense foggy scenes with domain-adaptive networks. *IEEE Transactions on Artificial Intelligence (AI)*, pages 1–12, 2022.

[51] Yuntong Ye, Yi Chang, Hanyu Zhou, and Luxin Yan. Closing the loop: Joint rain generation and removal via disentangled image translation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2053–2062, 2021.

[52] Liming Zhai, Felix Juefei-Xu, Qing Guo, Xiaofei Xie, Lei Ma, Wei Feng, Shengchao Qin, and Yang Liu. Adversarial rain attack and defensive deraining for DNN perception. In *arXiv preprint 2009.09205*. arXiv, 2020.

[53] Yiqi Zhong, Xianming Liu, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15345–15354, 2022.