# Attacking Motion Estimation with Adversarial Snow

Jenny Schmalfuss Lukas Mehl Andrés Bruhn
University of Stuttgart
firstname.lastname@vis.uni-stuttgart.de

## Abstract

*Current adversarial attacks for motion estimation (optical flow) optimize small per-pixel perturbations, which are unlikely to appear in the real world. In contrast, we exploit a real-world weather phenomenon for a novel attack with adversarially optimized snow. At the core of our attack is a differentiable renderer that consistently integrates photorealistic snowflakes with realistic motion into the 3D scene. Through optimization we obtain adversarial snow that significantly impacts the optical flow while being indistinguishable from ordinary snow. Surprisingly, the impact of our novel attack is largest on methods that previously showed a high robustness to small $L_p$ perturbations.*

## 1. Introduction

Adversarial attacks, which are 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 of an image sequence at times $t$ and $t{+}1$. Current attacks [11, 14] modify these two frames in the 2D space and consequently ignore the actual 3D geometry of the scene and the objects moving within. Moreover, when modifying pixels, they do not impose visual constraints, yielding attacked images that lack naturalism. Therefore, the conclusions drawn from 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 aims to answer the question whether a naturally occurring weather effect like snow can be manipulated to serve as an adversarial sample for motion estimation. To this end, we propose an adversarial attack that augments images with falling snowflakes featuring a high degree of realism: We create snowflakes 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 (see Fig. 1). This enables us to generate adversarially manipulated snow that significantly deteriorates optical flow predictions, while still satisfying the spatio-temporal and visual constraints of naturalistic snow. We consider snow as representative weather effect where single particles move independent of the remaining scene content, but note that the proposed attack procedure could also be used to model rain or sleet.

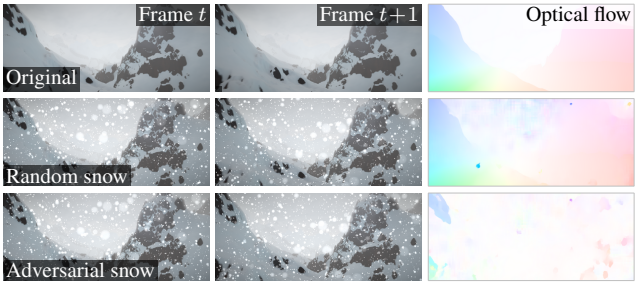


Figure 1. Snow attack with 3000 snowflakes that are first placed randomly in the 3D scene (*Random snow*) and then optimized (*Adversarial snow*) to perturb optical flow estimation with GMA [6].

**Related work.** Current optical flow methods based on neural networks [5, 6, 10] were recently shown to be susceptible to adversarially modified input images, which alter the resulting attacked flow $\check{f}$ to resemble a specified target flow $f^T$. The few existing adversarial attacks on optical flow methods generate either perturbations with small $L_p$ norms [14, 15] or adversarial patches [11], while adversarial weather attacks are completely unexplored. In contrast, adversarial perturbations that imitate snow effects have been investigated in the context of classification [7, 8] or human pose estimation [16]. However, for these applications, weather attacks [7, 8] or snow augmentations [4, 9] only have to be applied to single images rather than sequences. For optical flow estimation, a realistic motion of the weather effect over multiple frames and camera perspectives is required, imposing certain geometric constraints in time, which prevents the direct application of existing single-image adversarial weather generation schemes. Also, learning models for variations in images from data rather than modeling them explicitly has been explored for adversar-

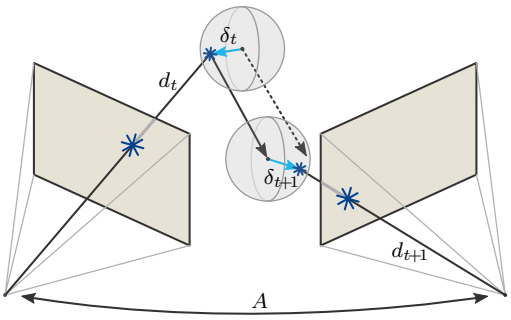


Figure 2. Model for snowflake motion in the 3D space.

ial training of robust classification methods [3, 13, 17] or to synthesize snowy versions of satellite images [12]. To ensure a realistic 3D motion of the weather effect in time, our attack explicitly models the motion of snow particles. This enables the computation of a ground-truth optical flow field for scenes with generated snow as the motion of each snowflake is known, which would not directly be possible with learned weather models. In terms of visual quality, the results of previous snow attacks were so far moderately convincing [7, 8] compared to conventional, non-differentiable rendering of snow effects [1].

**Contributions.** (i) In our paper, we present a differentiable snow-to-scene rendering framework that generates visually appealing snow, which moves realistically over multiple times steps. (ii) Based on this rendering framework, we devise the first adversarial snow attack for optical flow. It optimizes 3D spatial positions of snowflakes in the scene rather than 2D per-pixel perturbations, resulting in attacked images that retain high realism in snow movement and appearance. (iii) And finally, our snow attack not only leads to a significant degradation of optical flow results, but also illustrates that methods with little sensitivity to small $L_p$ perturbations are particularly affected when the snow-parameters are optimized.

## 2. Adversarial snow

To study the robustness of optical flow algorithms towards weather effects, we design an adversarial attack that augments image sequences with snowfall. To this end, we first equip an image sequence with parametrized snowflakes of realistic appearance and motion. Second, we optimize the snowflake parameters such that the resulting snowy image sequence causes a specified (wrong) flow prediction.

This approach imposes three constraints on the creation of the snowflakes: (i) Because motion estimation is designed to cope with moving objects in a 3D scene, a simple 2D animation of the snowflakes in the image plane is not sufficient. Instead, we have to model a realistic 3D motion, which also respects camera motion and object depth.

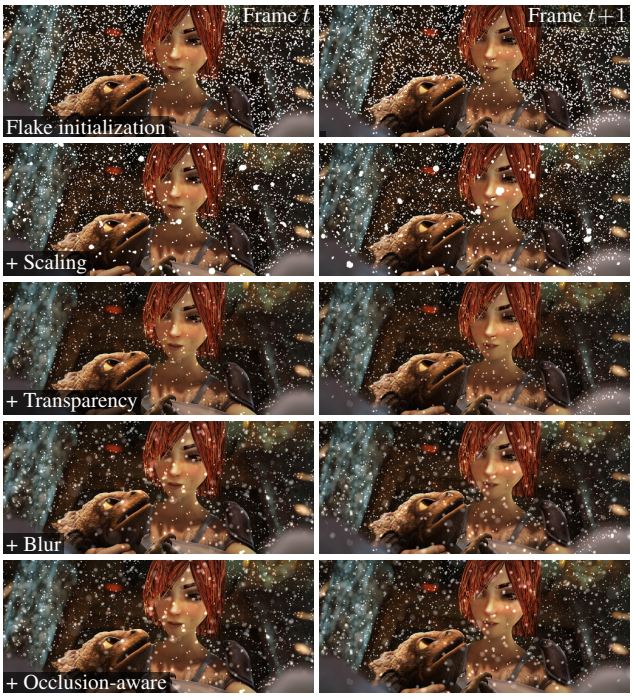


Figure 3. Breakdown of our realistic snow rendering process.

(ii) Moreover, expanding our pursuit of realism also to the appearance of the snow, the snowflakes should be integrated with appropriate visual effects. Such effects include an occlusion-aware depth placement as well as an out-of-focus blur. (iii) Finally, to enable the adversarial optimization of the snowflake, the whole rendering of the parametrized snowflakes needs to be differentiable.

### 2.1. Snow generation and rendering

To create 2D images of spatio-temporally consistent and visually appealing snowflakes we proceed in two steps. First, we initialize a fixed set of snowflakes in the 3D scene and equip them with properties: Initial 3D positions, 3D motion, 3D offsets before and after the motion ($\delta_t$, $\delta_{t+1}$), shapes, scaling and transparencies $\theta$ (see Fig. 2 for the motion model). Second, we make use of the 3D scene information to differentiably render the snowflakes in both frames, assuming that depth and camera information is given.

**Snowflake initialization.** To initialize the snowflake positions, we uniformly sample a fixed number of points in the 3D scene that are visible in the first frame or – after adding the 3D motion – in the second frame. Every snowflake is associated with a 2D flake image, which is randomly sampled from a set of snowflake templates and rotated by a random angle (Fig. 3, row 1). Then, we scale the flake image according to the snowflake's inverse depth, and initialize the transparency with a depth-dependent value (rows 2, 3). Finally, we add realistic out-of-focus blur by convolving the

flake image with a point spread function (row 4).

**Snowflake rendering.** We render the snowflakes with their associated 3D positions and transparencies in the given input frames as follows: First, we compute the corresponding 2D points in both frames, which yields center positions for the 2D flake images. Using the camera projection matrix and the relative transformation matrix $A$, we project the 3D points and their motion-displaced positions into the first and second frame, respectively. Next, we determine the visibility for each pixel of the flake-image by interpolating a visibility map computed from frame depth and the flake depth $d$ per camera, which allows a realistic, occlusion-aware scene integration (Fig. 3, last row). Lastly, we add the 2D flake images at the correct subpixel locations to the frames through bilinear interpolation, which enables a differentiation w.r.t. the snowflake parameters.

### 2.2. Adversarial optimization

After the snowflakes $\mathcal{S}$ are initialized and rendered, we modify certain snowflake parameters to change the output $\check{f}$ of optical flow networks towards a desired target flow $f^T$. We optimize transparency $\theta$ and offsets $\delta_t$, $\delta_{t+1}$ before and after the motion. Other parameters like initial 3D positions, 3D motion and 2D flake image are fixed. To ensure a valid range of transparency values, we transform the bounded variable to a continuous one before optimization. Our loss function measures flow differences with the average endpoint error (AEE) [14] and allows larger offsets $\delta_t$, $\delta_{t+1}$ for distant snowflakes via an $\alpha$-balanced MSE-like term, weighted with the inverse of the snowflake depth $d$:

$$\mathcal{L}(\check{f}, f^T, \mathcal{S}) = \text{AEE}(\check{f}, f^T) + \sum_{i \in t, t+1} \frac{\alpha_i}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{\|\delta_i(s)\|_2^2}{d_i(s)} \quad (1)$$

## 3. Experiments

We implement the snow attack in PyTorch and generate snowy versions for all sequences of the Sintel dataset [2], as it is the de-facto standard for optical flow benchmarking and provides depth and camera information. Attacked are the optical flow methods GMA [6], FlowNet2 [5], SpyNet [10] with Sintel checkpoints, which were identified in [14] to represent approaches with either high quality / low robustness, medium quality and robustness or low quality / high robustness, respectively. We choose a zero-flow target $f^T$ (white flow visualization) [14], offset penalty weights $\alpha_t = \alpha_{t+1} = 1000$ and snow falling down:right (ratio 5:2). In several experiments, we investigate the attack strength $\text{AEE}(\check{f}, f^T)$ (small value when attacked flow and target coincide), when (i) the number of snowflakes increases, (ii) different snow parameters are optimized, and (iii) optimized snow for one method is transferred to another.

**Snow density.** First, we optimize $\delta_t$, $\delta_{t+1}$ and $\theta$, and study the attack strength when the total number of snowflakes in both sequences is varied from 1000 to 5000. In Table 1, the attack strength (distance from attacked flow to target) of adversarial snow increases with the snowflake number, independent of the attacked method. On average, the distance is more than halved from 1000 to 5000 snowflakes. This demonstrates the realistic behavior of our adversarial snow, which increases its influence with the covered image area. We use 3000 snowflakes for further experiments because they balance attack strength and visual snow density.

| Method | 1000 | 2000 | 3000 | 4000 | 5000 |
|---|---|---|---|---|---|
| SpyNet | 7.04 | 5.44 | 4.54 | 4.14 | **3.62** |
| FlowNet2 | 6.33 | 4.33 | 3.19 | 2.38 | **2.28** |
| GMA | 8.56 | 6.38 | 4.42 | 3.38 | **2.29** |

Table 1. Attack strength $\text{AEE}(\check{f}, f^T)$ of adversarial snow with an increasing *number of snowflakes* per frame-pair (1000–5000) on different optical *flow methods*, best attack strength is bold.

| Parameters | SpyNet | FlowNet2 | GMA |
|---|---|---|---|
| Initial snow | 13.29 | 21.93 | 12.25 |
| $\delta_t$ | 5.26 | 4.51 | 5.76 |
| $\delta_{t+1}$ | 6.53 | 4.72 | 6.88 |
| $\theta$ | 12.74 | 19.72 | 11.98 |
| $\delta_t$, $\delta_{t+1}$ | <u>4.68</u> | <u>3.73</u> | **4.41** |
| $\delta_t$, $\theta$ | 5.15 | 4.11 | 5.95 |
| $\delta_{t+1}$, $\theta$ | 6.21 | 4.84 | 6.89 |
| $\delta_t$, $\delta_{t+1}$, $\theta$ | **4.54** | **3.19** | <u>4.42</u> |

Table 2. Attack strength $\text{AEE}(\check{f}, f^T)$ of adversarial snow, optimized for combinations of *snow parameters* $\delta_t$, $\delta_{t+1}$ and $\theta$. *Initial snow* measures the attack strength of randomly initialized snow.

**Optimizing snow attack strength.** Next, we investigate how optimizing different snow parameters influences the attack strength. Tab. 2 summarizes the attack strength for snow attacks with 3000 snowflakes, optimized for all parameter combinations, and compared to a random snow initialization. In all combinations, optimizing the position offset in the first frame $\delta_t$ yields the strongest attacks, while varying the snowflake transparency $\theta$ has the smallest impact. Nonetheless, all parameters $\delta_t$, $\delta_{t+1}$, $\theta$ must be optimized to reach the strongest attack - only for GMA it suffices to optimize $\delta_t$ and $\delta_{t+1}$ alone. Fig. 4 visualizes the attacked frames and predicted flows for all methods when optimizing $\delta_t$, $\delta_{t+1}$ and $\theta$. There, we make several interesting observations: When comparing randomly initialized and adversarially optimized snowflakes, their positions differ only slightly and the adversarial sample is indistinguishable

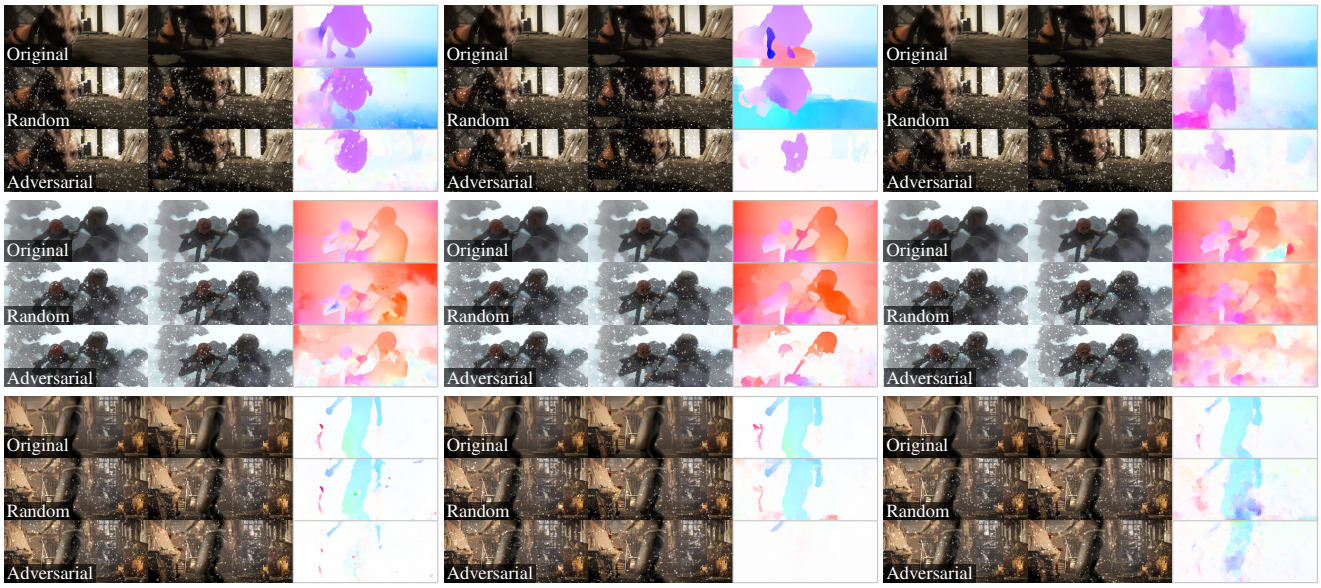


Figure 4. Qualitative results for 3000 snowflakes on images from the Sintel final dataset with *random* initialization and *adversarial* optimization with optical flow predictions for GMA [6], FlowNet2 [5] and SpyNet [10] (*left to right*).

from random snow to a human observer. Also, it is surprising that snowflakes eradicate the estimated motion despite their inability to stand still due to falling and camera motion. Moreover, when comparing the initial snow to the optimized results, FlowNet2 and SpyNet that were previously identified to be relatively robust [14], alter their predictions significantly in the presence of adversarial snow. We ascribe this phenomenon to the more detailed flow estimations of GMA, which detects the localized motion of single snowflakes (*cf*. Fig. 4, col. 3, where circular snowflakes are visible). The less accurate methods FlowNet2 and SpyNet instead propagate the detected motion from snowflakes over larger areas, rather than attributing it to small moving objects (*cf*. Fig. 4, col. 6,9, where optical flow predictions have few details).

**Snow transferability.** Finally, we investigate if snowflakes optimized for one method can change the flow predictions of another. We use the adversarial snow attack with 3000 snowflakes and optimize the snow for $\delta_t$, $\delta_{t+1}$ and $\theta$, before evaluating all networks on the resulting adversarial images. Tab 3 summarizes the results. While snowflakes are most effective on the method they were optimized for, transferred snow has a measurable negative impact on FlowNet2 and GMA compared to random snow (*cf*. Tab. 2, Init). For even more transferable configurations, snowflakes could be optimized over several images and methods.

| Test \ Train | SpyNet | FlowNet2 | GMA |
|---|---|---|---|
| SpyNet | **4.54** | 13.80 | 13.44 |
| FlowNet2 | 16.54 | **3.19** | 14.21 |
| GMA | 10.33 | 10.67 | **4.42** |

Table 3. Transferability of adversarial snow strength AEE($\check{f}, f^T$) of snowflake positions that were optimized for one *optical flow method* to another, best attack in bold.

## 4. Conclusion

In this paper we proposed a novel adversarial attack on motion estimation algorithms with realistic snow. To this end, we developed a differentiable snowflake renderer that can be used to generate adversarial samples with a strong impact on optical flow methods. Interestingly, our attack demonstrates the ability to let networks predict zero-flow although the snowflakes undergo both individual and camera motion. At the same time, the resulting attacked images are visually indistinguishable from random snow images, making our attack unnoticeable for a human observer. Finally, more accurate methods appear to be more robust towards adversarially optimized snow than towards small $L_p$ perturbations, as they detect the motion of single snowflakes rather than propagating the motion into the wider image.

**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] A. v. Bernuth, G. Volk, and O. Bringmann. Simulating photo-realistic snow and fog on existing images for enhanced CNN training and evaluation. In *ITSC*, 2019. 2

[2] D. Butler, J. Wulff, G. Stanley, and M. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 3

[3] Sven Gowal, Chongli Qin, Po-Sen Huang, Taylan Cemgil, Krishnamurthy Dvijotham, Timothy Mann, and Pushmeet Kohli. Achieving robustness in the wild via adversarial mixing with disentangled representations. In *CVPR*, 2020. 2

[4] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2018. 1

[5] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 1, 3, 4

[6] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021. 1, 3, 4

[7] D. Kang, Y. Sun, D. Hendrycks, T. Brown, and J. Steinhardt. Testing robustness against unforeseen adversaries. arXiv:1908.08016, 2019. 1, 2

[8] A. Marchisio, G. Caramia, M. Martina, and M. Shafique. fakeWeather: Adversarial attacks for deep neural networks emulating weather conditions on the camera lens of autonomous systems. arXiv:2205.13807, 2022. 1, 2

[9] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. In *NeurIPS-W*, 2019. 1

[10] A. Ranjan and M. Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 1, 3, 4

[11] A. Ranjan, J. Janai, A. Geiger, and M. Black. Attacking optical flow. In *ICCV*, 2019. 1

[12] C. Ren, A. Ziemann, J. Theiler, and A. Durieux. Deep snow: Synthesizing remote sensing imagery with generative adversarial nets. In *Defense + Commercial Sensing*, 2020. 2

[13] Alexander Robey, Hamed Hassani, and George J Pappas. Model-based robust deep learning: Generalizing to natural, out-of-distribution data. arXiv:2005.10247, 2020. 2

[14] J. Schmalfuss, P. Scholze, and A. Bruhn. A perturbation-constrained adversarial attack for evaluating the robustness of optical flow. arXiv:2203.13214, 2022. 1, 3, 4

[15] S. Schrodi, T. Saikia, and T. Brox. Towards understanding adversarial robustness of optical flow networks. In *CVPR*, 2022. 1

[16] Jiahang Wang, Sheng Jin, Wentao Liu, Weizhong Liu, Chen Qian, and Ping Luo. When human pose estimation meets robustness: Adversarial algorithms and benchmarks. In *CVPR*, pages 11855–11864, 2021. 1

[17] Eric Wong and J. Zico Kolter. Learning perturbation sets for robust machine learning. arXiv:2007.08450, 2020. 2