Towards Understanding Adversarial Robustness of Optical Flow Networks

Simon Schrodi Tonmoy Saikia Thomas Brox
University of Freiburg, Germany
{schrodi, saikiat, brox}@cs.uni-freiburg.de

Abstract

Recent work demonstrated the lack of robustness of optical flow networks to physical patch-based adversarial attacks. The possibility to physically attack a basic component of automotive systems is a reason for serious concerns. In this paper, we analyze the cause of the problem and show that the lack of robustness is rooted in the classical aperture problem of optical flow estimation in combination with bad choices in the details of the network architecture. We show how these mistakes can be rectified in order to make optical flow networks robust to physical patch-based attacks. Additionally, we take a look at global white-box attacks in the scope of optical flow. We find that targeted white-box attacks can be crafted to bias flow estimation models towards any desired output, but this requires access to the input images and model weights. However, in the case of universal attacks, we find that optical flow networks are robust. Code is available at https://github.com/lmb-freiburg/understanding_flow_robustness.

Figure 1. **Overview.** Physical patch-based adversarial attacks on optical flow can be avoided by minor architectural changes. First row: attacked first frame. Second row: ground truth optical flow. Third and fourth rows: the resulting optical flow estimates of FlowNetC [8] and our proposed Robust FlowNetC. FlowNetC is strongly affected by the adversarial patch, whereas Robust FlowNetC is barely affected. For the robust version we make simple design changes based on causes of the attack; see Section 6.

1. Introduction

While deep learning has been conquering many new application domains, it has become increasingly evident that deep networks are vulnerable to distribution shifts. Adversarial attacks are a particular way to showcase this vulnerability, where one finds the minimal input perturbation that is sufficient to corrupt the network output. As the small perturbation moves the sample out of the training distribution, the network is detached from its learned patterns and follows the suggestive pattern of the attack. Although many methods have been proposed to improve robustness [34], they only alleviate the problem but do not solve it [1].

While most white-box adversarial attacks are mainly of academic relevance as they reveal the weaknesses of deep networks w.r.t. out-of-distribution data, physical adversarial attacks have serious consequences for safe deployment. In physical attacks, the input is not perturbed artificially, but a confounding pattern is placed in the real world to derail the machine learning approach.

Most work on adversarial attacks has been concerned with recognition problems, and it looked for a while as if correspondence problems are not a good target for adversarial attacks. However, Ranjan *et al*. [25] showed that they can successfully perform physical adversarial patch attacks on optical flow networks. They optimized an adversarial local patch that they can paste into both images, such that large errors appear in the estimated optical flow field even far away from the affected image location. They also showed that the same adversarial patch worked on all vulnerable architectures, and even demonstrated physical attacks, where the printed patch is physically added to a scene and derails the optical flow estimation. Ranjan *et al*. found that different network architectures show different levels of vulnerability, whereas conventional optical flow methods are not vulnerable at all. They hypothesized the cause for the vulnerability to be in the common encoder-decoder architecture of FlowNet [8] and its derivatives but did not provide a conclusive analysis.

In this paper, we continue their work by a deeper analysis of the actual reason behind the vulnerability. In particular, we answer the following questions.
(1) What is the true cause of adversarial patch attacks?
(2) Knowing the cause, can the patch-based attack also be built without optimizing it for the particular network (zero query black-box attack)?
(3) Can the severe vulnerability be avoided by a specific design of the network architecture or by avoiding mistakes in such design? For an overview see Figure 1.

After answering these questions positively, we turn towards (global) adversarial perturbation attacks, *i.e.*, attacks that modify the whole image. We demonstrate that any target optical flow field can be generated; see Figure 11. On the other hand, we show that this attack strategy does not apply to universal (input-agnostic) attacks, *i.e.*, global attacks on optical flow networks must exploit the structure of the input images. This is different from unprotected recognition networks, which are vulnerable to imperceptible universal attacks [14, 22].

2. Related Work

Optical flow. For many decades, optical flow was estimated with approaches that minimize an energy function consisting of a matching cost and a term that penalizes deviation from smoothness [4, 6, 7, 16].

Inspired by the success of CNNs on recognition tasks, Dosovitskiy *et al*. [8] introduced estimation of optical flow with a deep network, by training it end-to-end. They proposed two network architectures – FlowNetS and FlowNetC – of which the first is a regular encoder-decoder architecture, whereas the second includes an additional correlation layer that explicitly computes a cost volume for feature correspondences between the two images – like the correlation approaches from the very early days of optical flow estimation, but integrated into the surrounding of a deep network for feature learning and interpretation of the correlation output. The concept of these architectures has been picked up by many follow-up works that introduced, for instance, coarse-to-fine estimation [24, 28], stacking [17], and multi-scale 4D all-pairs correlation volumes combined with the separate use of a context encoder as well as a recurrent unit for iterative refinement [31]. Most of the architectures have an explicit correlation layer like the original FlowNetC.

Adversarial attacks. The first works that brought up the issue of vulnerability of deep networks to adversarial examples were in the context of image classification and generated the examples by solving a box-constrained optimization problem [30] or by perturbing the input images with the gradient w.r.t. the input [11]. Su *et al*. [27] showed that neural networks can be even attacked by just changing a single pixel. Kurakin *et al*. [18] showed that adversarial attacks also work in the physical world by printing out adversarial examples. Several follow-up works have confirmed this behavior in other contexts [3, 5, 9]. Hendrycks *et al*. [15] showed that adversarial examples can even exist in natural, real-world images, which relates adversarial attacks to the more general issue of out-of-distribution samples.

Works on adversarial attacks concentrated on various sorts of recognition tasks, *i.e.*, tasks where the output depends directly on some feature representation of the input image, such as classification, semantic segmentation, single-view depth estimation, or image retrieval. Recently, Ranjan *et al*. [25] showed that optical flow networks are also vulnerable to adversarial patch attacks and can also attack flow networks in a real-world setting. From their experimental evidence, they hypothesized that the encoder-decoder architecture is the main cause for the adversarial vulnerability, whereas spatial pyramid architectures, as well as classical optical flow approaches, are robust to patch-based attacks. Further, they showed that flow networks are not spatially invariant and the deconvolutional layers lead to an amplification of activations as well as checkerboard artifacts [23]. Recently, Wong *et al*. [33] showed that imperceptible perturbations added to each pixel individually can significantly deteriorate the output of stereo networks. They used adversarial data augmentation to make stereo networks more robust. While stereo networks are vulnerable to image-specific attacks, they showed that perturbations do not transfer well to the next time step.

3. Adversarial Patch Attacks

Adversarial patch. Ranjan *et al*. [25] proposed attacking flow networks by pasting a patch p of resolution $h \times w$ onto the image frames $(I_t, I_{t+1}) \in \mathcal{I}$ of resolution $H \times W$ at the same location, rotation, and scaling. To craft an adversarial patch for flow network F, they minimized the cosine similarity between the unattacked flow (u, v) and the attacked one $(\tilde{u}, \tilde{v})$. More formally, they optimized

$$\hat{p} = \underset{p}{\operatorname{argmin}}\, \mathbb{E}_{(I_t, I_{t+1}) \sim \mathcal{I}, l \sim \mathcal{L}, \delta \sim \mathcal{T}} \frac{(u, v) \cdot (\tilde{u}, \tilde{v})}{||(u, v)|| \cdot ||(\tilde{u}, \tilde{v})||}, \tag{1}$$

where they randomly sample the location $l \in \mathcal{L}$ and affine transformations $\delta \in \mathcal{T}$, *i.e.*, rotation and scaling, to generalize better to a real-world setting.

Vulnerability of existing optical flow methods. Ranjan *et al*. [25] found that different flow network architectures show different degrees of vulnerability. Table 1 shows the performance degradation of different architectures w.r.t. patch-based attacks. FlowNetC is the only truly vulnerable flow network, whereas the others are much more robust.

Ranjan *et al*. [25] attributed the vulnerability to the encoder-decoder architecture and the higher robustness to the spatial pyramid of PWC-Net and SPyNet. However, there is a counterexample that proves this hypothesis wrong:

Table 1. **Patch attacks on different flow networks.** We show average unattacked and worst-case attacked End-Point-Error (EPE) on the KITTI 2015 training dataset (for details see Section 4). We only show results for larger patch sizes (102×102 and 153×153), since smaller patches show simply a weaker effect [25].

Network	Un-attacked EPE	Attacked 102x102 (2.1%)	Attacked 153x153 (5.8%)
FlowNetC [8]	11.50	52.66	51.99
FlowNetS [8]	14.33	17.35	17.92
FlowNet2 [17]	10.07	12.40	13.36
SPyNet [24]	24.26	27.47	25.84
PWC-Net [28]	12.55	18.08	17.70
RAFT [31]	5.86	8.48	9.01

FlowNetS – the direct counterpart of FlowNetC *without* correlation layer – is a plain encoder-decoder architecture without a spatial pyramid and, as Table 1 reveals, is quite robust to the attack. Thus, the encoder-decoder architecture cannot be the root cause for the vulnerability, even though the decoder can be responsible for amplifying the effect.[1]

4. What Causes a Successful Patch Attack?

We build on the attack procedure of Ranjan *et al*. [25], *i.e*., we also use KITTI 2012 [10] for patch optimization and their white-box evaluation procedure on KITTI 2015 [21]. We show the importance of the spatial location and analyze the flow networks' feature embeddings. Through this analysis, we can trace the adversarial patch attacks back to the classical aperture problem in optical flow. For sake of brevity, we focus on FlowNetC, since it is the most vulnerable flow network (Table 1), as well as PWC-Net and RAFT. See Supplement Section 1 for all implementation details.

4.1. Spatial Location Heat Map

We analyze the impact of the spatial location of the adversarial patch by computing the attacked End-Point-Error (EPE) for each location over a coarse grid on the image. For visualizations of the resulting heat map, we linearly interpolate between values and clip them. This allows us to identify three potential attacking scenarios: best case, median case, and worst case. For example, in the worst-case scenario, we paste the patch at the image location with the highest attacked EPE. Figure 2 shows that the sensitivity to patch-based attacks depends much on the image and the location of the patch. The sensitivity of PWC-Net and RAFT can also sometimes be high. In particular, image regions with large flow (*e.g*., fast-moving objects) can lead to a severe deterioration of flow estimations.

[1]Like strong rain is the root cause for flooding but a dam (*i.e*., spatial pyramid) can avoid the flooding despite strong rain to a certain degree.

Table 2. **Replacing attacked features.** Average EPE of the attacked FlowNetC (left) and the average EPE when the respective features are replaced by those from the unattacked FlowNetC (right) on the KITTI 2015 training dataset using adversarial and uniform noise 102×102 patches, respectively. See Figure 6 left for the encoder before the correlation layer in the original FlowNetC [8].

Replace features of	Without replacement	With replacement
conv3⟨a,b⟩	25.95	11.31
conv_redir	25.95	28.36
corr	25.95	12.67

4.2. Correlations and Correlation Layer

To analyze the features of flow networks during the attack, we visualize the unattacked and attacked features' distributions using t-SNE [32] and compute the Maximum Mean Discrepancy (MMD) [12] between the two distributions. Comparing FlowNetC's feature embeddings with and without the attack reveals a large separation of the unattacked and attacked feature distributions after the correlation layer (Figure 3), while the distributions were quite close before that layer. This is also indicated by the rapid increase of MMD from 0.246 to 3.331. On the other hand, the unattacked and attacked feature distributions of PWC-Net and RAFT are close to each other before and after applying the correlation layer, and also the MMDs stay similar. Hence, we hypothesize that the feature correlation of FlowNetC causes the vulnerability to patch-based attacks.

We validate this hypothesis by replacing attacked features with unattacked features to simulate what happens if an architectural component of FlowNetC would be robust w.r.t. patch-based attacks. We used a 102×102 patch with uniform noise, pasted it at a random location, and saved the feature maps. Afterward, we attacked FlowNetC with an adversarial patch of the same size at the same location and replaced the attacked feature maps with the previously saved unattacked feature maps for different architectural components. Table 2 shows that a robust correlation layer (corr) could remove the effect of the attack. Trivially, a robust encoder before the correlation layer (conv3⟨a,b⟩) can do the same. In contrast, if the convolution that bypasses the correlation layer (conv_redir) is made robust, the attack still remains fully effective. This shows that the feature correlation is the root cause, and also explains why FlowNetC's sibling FlowNetS is much more robust, as it has no correlation layer.

4.3. Relationship to the Aperture Problem

While we have identified the correlation layer as the cause on the network side, we do not yet know what is caus-

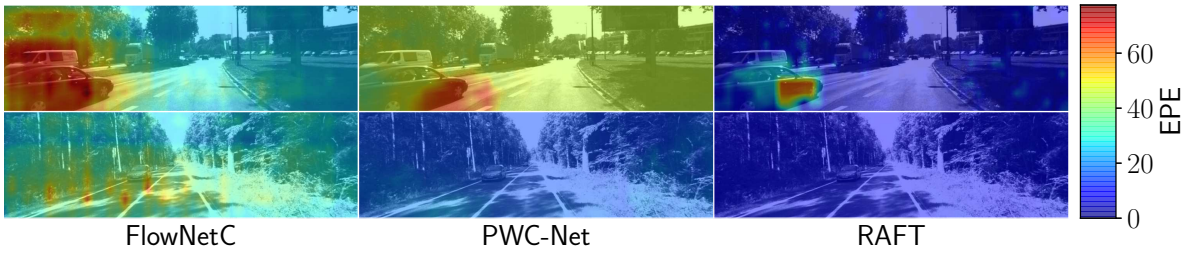
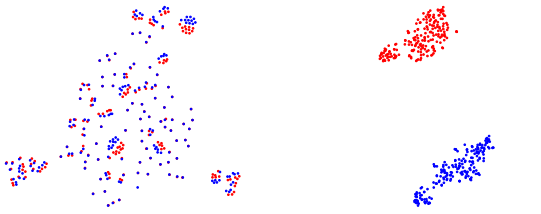
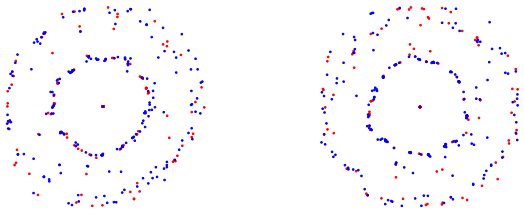


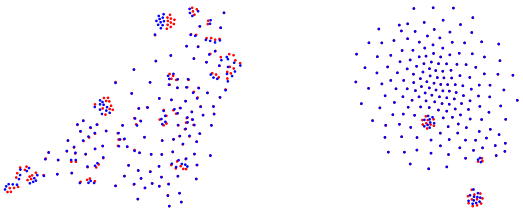
Figure 2. **Impact of spatial locations.** Effect of the spatial location of an adversarial 102×102 patch. Best viewed in color.



(a) FlowNetC [8], MMD before: 0.246, MMD after: 3.331.



(b) PWC-Net [28], MMD before: 0.0, MMD after: 0.0.



(c) RAFT [31], MMD before: 0.158, MMD after: 0.003.

Figure 3. **t-SNE embeddings of features from FlowNetC, PWC-Net and RAFT.** Left: t-SNE embeddings of features before correlation layer. Right: t-SNE embeddings of features after correlation layer. We use our best found adversarial 102×102 patch (2.1% of the image size). Blue and red points correspond to unattacked or attacked features, respectively. We visualize the t-SNE embeddings of features of PWC-Net and RAFT before or after applying the correlation layer for the first time. Note that a larger MMD indicates that the unattacked and attacked features are more separable. Best viewed in color and with zoom.

ing it in the images. There is good reason to suspect that the attack builds on *self-similar patterns* within the adversarial patch; and indeed, they contain multiple self-similar patterns (Figure 1). This suggests that patches trigger matching ambiguities that show as a large active area in the correlation output. Successive layers, that are supposed to interpret

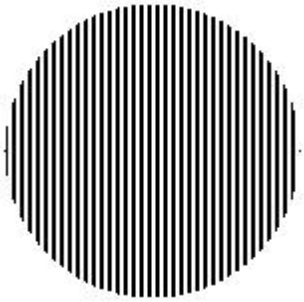


Figure 4. **Handcrafted patch.** Patch is enlarged for visualization.

this output, successively spread the dominating ambiguous signals into the wider neighborhood, whereas the true correlation is outnumbered. This is related to the well-known aperture problem in optical flow, where repetitive patterns lead to an ambiguity in the optical flow and the receptive field (the aperture) determines the perceived motion.

However, why are other flow networks, *e.g.*, PWC-Net or RAFT, which also have a correlation layer, much more robust to the attack? We hypothesize that higher vulnerability is due to the smaller size of FlowNetC's aperture, *i.e.*, a smaller receptive field before the correlation layer (*i.e.*, 31×31). More specifically, the larger receptive field size at the (first) correlation layer in PWC-Net and RAFT (*i.e.*, 631×631 and 106×106) sees also areas of the image that are not affected by the attack. In addition, RAFT uses all-pairs correlation and correlation pooling, which further increases its effective receptive field size. We hypothesize that this helps their correlation layers to keep the correlation peaked.

5. Can We Attack Without Optimization?

To show that self-similar patterns within patches cause the vulnerability, we handcraft a circular high-frequency black and white vertically striped patch; see Figure 4. Note that there is no need for optimization. We ablate the ingredients of our handcrafted patch in Supplement Section 4.

Table 3 shows that FlowNetC is also vulnerable to our handcrafted patch, providing further evidence that high correlation within the patch causes matching ambiguities in the correlation layer. The median performance of the other flow

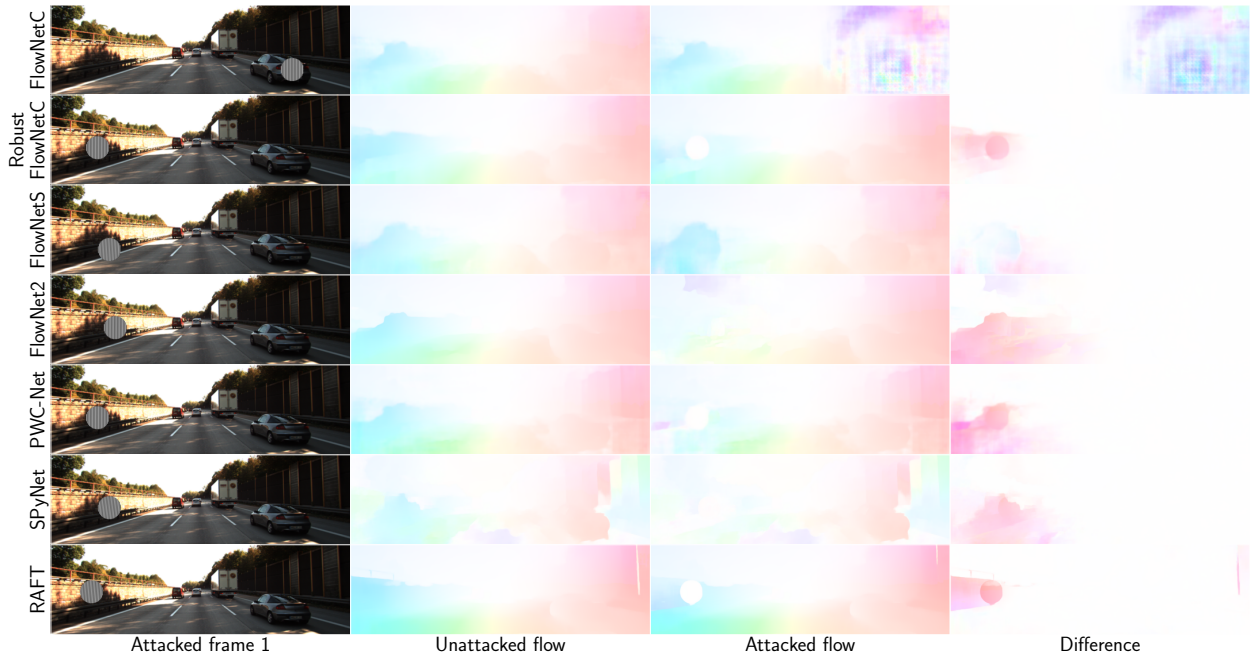


Figure 5. **Handcrafted patch attack.** Handcrafted 102×102 patch attack on all flow networks. We show the patch at the worst possible spatial location for each flow network. Robust flow networks predict zero flow (white color) at the patch location (third column). See Supplement Section 3 for additional examples.

Table 3. **Handcrafted patch attack.** Effect of the handcrafted patch attack (in pixel and percent of image size) on different flow networks. We show average median and worst-case attacked EPE over a coarse grid of spatial locations of the patch on the KITTI 2015 training dataset for each flow network.

Flow Network	Unattacked EPE	25x25 (0.1%)		51x51 (0.5%)		102x102 (2.1%)		153x153 (4.8%)	
		Median	Worst	Median	Worst	Median	Worst	Median	Worst
FlowNetC [8]	11.50	11.66	16.66	15.81	29.08	23.41	46.12	30.97	52.27
Robust FlowNetC	9.95	9.95	11.14	9.86	11.74	9.60	13.08	9.27	13.64
FlowNetS [8]	14.33	14.35	15.66	14.50	17.00	14.64	20.10	14.55	22.32
FlowNet2 [17]	10.07	10.11	13.80	10.56	19.10	12.08	21.63	13.84	24.35
SPyNet [24]	24.26	24.22	26.24	24.06	27.41	23.28	27.46	22.20	26.62
PWC-Net [28]	12.55	12.54	14.82	12.45	16.10	12.02	16.87	11.42	16.26
RAFT [31]	5.86	5.80	7.08	5.74	7.44	5.49	8.69	5.17	8.96

networks, also the proposed Robust FlowNetC (Section 6), is not affected, as they limit the effect of the ambiguous correlation signal to its local region or can even estimate the correct zero flow motion in this region. However, all flow networks are affected by the patch to some degree in the worst-case scenario, *i.e.*, when we place the patch at the worst possible location in the image frames (Table 3 and Figure 5). This is hard to exploit in a physical attack and is similar to other optical flow estimation errors that naturally appear locally in some difficult image frames.

6. Can the Vulnerability be Controlled?

Based on the previous analysis, we add corresponding architectural (and training improvements) to FlowNetC and show that this intervention also makes it robust to patch-based attacks. The components we add to FlowNetC are already included in most modern architectures and can be regarded as the important ingredients that make an optical flow network robust to self-similar patterns as exploited by adversarial patch attacks. Complementary, we also show that we can create a more vulnerable RAFT variant.

6.1. Architecture

We increase the receptive field before the correlation layer by adding (spatial resolution preserving) convolutional layers in each resolution level before the correlation layer. Moreover, we replace 5×5 convolutional layers in FlowNetC by 3×3 convolutional layers. This allows us

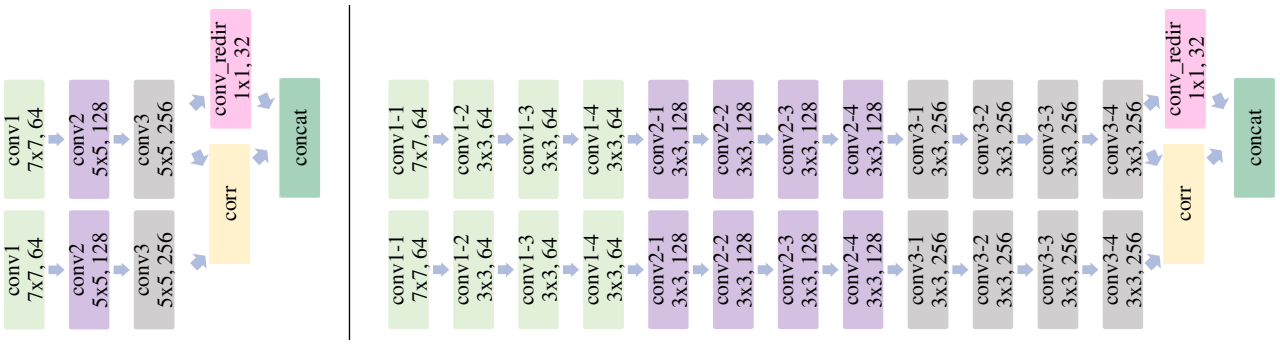


Figure 6. **Modified encoder before the correlation layer for Robust FlowNetC.** Left: original FlowNetC encoder [8]. Right: our Robust FlowNetC encoder. Blocks show the name, kernel size, and number of filters. For Robust FlowNetC, the layers conv1-1, conv2-1, and conv3-1 are used for downsampling and hence have a stride of 2.

Table 4. **Overview over FlowNetC encoder variants.** The very first layer is always a 7×7 convolutional layer. See Figure 6 for visualizations of the original FlowNetC [8] (second row) and our Robust FlowNetC encoder (last row).

Kernel size	Convs per resolution level	Receptive field
3	1	19
5	1	31
3	2	47
3	3	75
5	2	87
3	4	103

to use deeper encoders with a larger receptive field before the correlation layer. Alternatively, we can use larger dilation rates for larger receptive fields (Supplement Section 5). We call the FlowNetC variant with kernel size 3 and 4 convolutional layers per resolution level *Robust FlowNetC*, illustrated in Figure 6 right. For an ablation, we also created other variants of FlowNetC; see Table 4.

6.2. Training Procedure

It has been shown that the training procedure is also an important factor for good optical flow performance [17, 29]. Since we showed in the previous section that the patch-based attack is not a classical adversarial attack but simply makes the local estimation problem harder, stronger performance should also yield better robustness w.r.t. patch-based attacks. Hence, for Robust FlowNetC we used the training pipeline of RAFT, *i.e.*, we use the AdamW optimizer [19], one cycle scheduler [26], gradient clipping, same augmentation pipeline, and also initialized the weights of the models with Kaiming initialization [13]. Different from RAFT's training procedure, we used a multiscale l_2 loss, pre-train on FlyingChairs [8] for $600k$ iterations with an initial learning rate of 10^{-4} and then trained on FlyingThings3D [20] for $300k$ iterations with an initial learning rate of 10^{-5}.

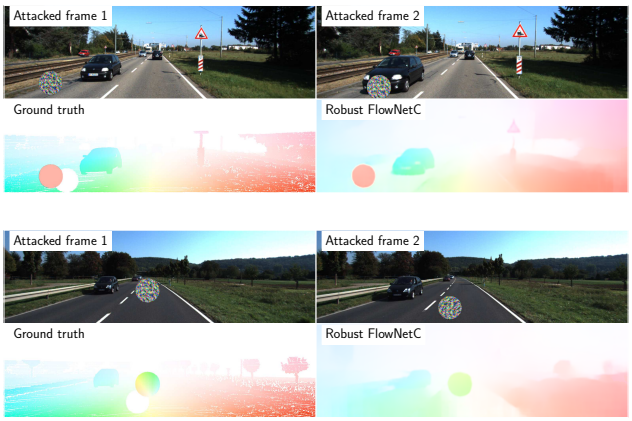


Figure 7. **Moving patch between image frames.** For each example block; top row shows attacked first and second image frames. Bottom row show ground truth and the predicted optical flow of Robust FlowNetC. We apply random affine transformations, *i.e.*, translation, rotation, and scaling, to the patch between the two images frames. Note that the patch can also move in the opposite direction w.r.t. its neighborhood, making it even more adversarial. Robust FlowNetC correctly estimates the optical flow. Note, however, that rotations of the patch are not estimated correctly and can lead to slight estimation errors of the motion of the patch.

6.3. Evaluation

Figure 5 and Table 3 clearly show the effect of above changes: Robust FlowNetC is as robust to adversarial patch attacks as PWC-Net or RAFT. The handcrafted patch attack rules out that this robustness is due to obfuscated gradients [2]. See Supplement Section 6 for examples using optimized patches. In Figure 7, we show a scenario where the patch is allowed to (freely) move between image frames. See Supplement Section 7 for results for a static patch. Figures 5 and 7 show that Robust FlowNetC correctly predicts the flow whether the patch moves or not between the image frames. The patch has only a negligible impact on the sur-

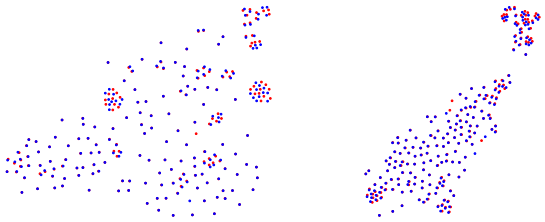


Figure 8. **t-SNE embeddings of features from Robust FlowNetC.** Left: t-SNE embeddings of features before the correlation layer (MMD: 0.012). Right: t-SNE embeddings of features after the correlation layer (MMD: 0.007). Blue and red points correspond to unattacked or attacked features, respectively. Best viewed in color. In contrast to the original FlowNetC (Figure 3a), the attacked and unattacked t-SNE embeddings stay well-aligned.

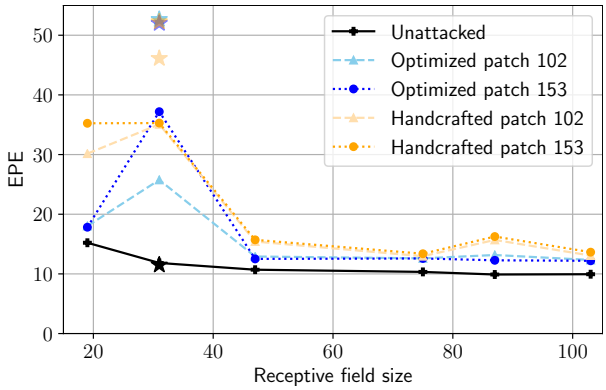


Figure 9. **Performance of FlowNetC variants with different receptive field sizes.** We show both unattacked and attacked worst-case EPE. Stars show results for the original FlowNetC. For optimized patches, we show results using the patch with the highest attacked EPE after optimization over ten runs. Larger receptive fields reduce the attacked worst-case EPE. We report the worst-case attack w.r.t. location, *i.e.*, there remains a small gap between the attacked and the unattacked result, as for all networks; see Table 3. The two local peaks correspond to the variants with 5×5 kernel sizes; see Table 4.

rounding image region, even if we move the patch between image frames. We also tested the l_2 loss for patch optimization against Robust FlowNetC, and it also did not lead to any (significant) degradation in flow performance, *i.e.*, we report worst-case EPE of 12.54 for a 102×102 patch. Figure 8 shows that the embeddings between the attacked and unattacked features are well-aligned – in contrast to the original FlowNetC. Figure 9 shows that the improved robustness stems from larger receptive field sizes.

6.4. Pushing Vulnerability

In the previous subsections, we showed that we can make FlowNetC robust by increasing its depth and, thus, its receptive field. In this section, we show the other direction by

Figure 10. **Results for our vulnerable RAFT variant.** For each block; first column shows image overlays where we place the patch at the worst location. Second column shows the predicted optical flows of RAFT and its vulnerable variant. The vulnerable RAFT variant is vulnerable to patch-based attacks.

making a previously robust flow network (*i.e.*, RAFT) vulnerable to patch-based attacks by replacing its encoder with FlowNetC's original encoder before the correlation layer (and removing the separate context encoder). Note that with these changes, the architectural part before the cost volume is the same as in FlowNetC. We followed RAFT's training strategy [31]. Table 5 and Figure 10 show that even with robust parts after the correlation layer, *i.e.*, iterative refinement, there can be severe adversarial noise in the flow estimates during an attack with our handcrafted patch.

7. Adversarial Perturbation Attacks

Recently, Wong *et al.* [33] showed that they could attack stereo networks using commonly used (global) untargeted adversarial perturbation attacks for recognition networks. Their approach is also effective against flow networks (Supplement Section 8). In the following, we propose how we can make flow networks predict any desired flow estimate by adding imperceptible adversarial perturbations, and also investigate universal perturbation attacks. See Supplement Section 1 for implementation details. Furthermore, in Supplement Section 11, we show that we can make flow networks robust through adversarial data augmentation.

Targeted adversarial attacks. While Wong *et al.* showed that they can disturb stereo networks' estimations, we show that we can make flow networks predict any desired flow by adding only small additive perturbations (*e.g.*, L_∞ norm $\epsilon = 0.02$). To craft perturbations, we used the Iterative - Fast Gradient Sign Method (I-FGSM) [18] with learning rate $\alpha = 0.002$, l_2 loss, and minimized toward a target flow. Figure 11 shows that we can make flow networks predict an arbitrary target flow from the same or even a completely different domain.

Table 5. **Results for our vulnerable RAFT variant.** We show average median and worst-case EPE over a coarse grid of spatial locations of our handcrafted patch on the KITTI 2015 training dataset for different RAFT variants. Even though RAFT has robust architectural ingredients, *e.g.*, iterative refinement after the cost volume, we can substantially increase vulnerability by simple architectural changes before the cost volume.

FlowNetC Encoder	Without Context Encoder	Unattacked EPE	25x25 (0.1%) Median	25x25 (0.1%) Worst	51x51 (0.5%) Median	51x51 (0.5%) Worst	102x102 (2.1%) Median	102x102 (2.1%) Worst	153x153 (4.8%) Median	153x153 (4.8%) Worst
-	-	5.86	5.80	7.08	5.74	7.44	5.49	8.69	5.17	8.96
-	✓	6.88	6.78	9.09	6.65	9.35	6.43	10.09	6.10	11.31
✓	-	5.84	5.85	7.47	5.84	9.31	5.78	10.50	5.92	11.60
✓	✓	6.33	6.38	13.61	6.43	16.61	7.03	19.12	9.37	20.99

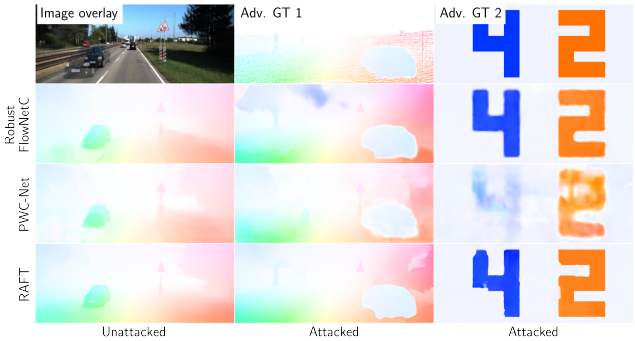


Figure 11. **Targeted adversarial attacks.** We can add adversarial perturbations that make flow networks predict arbitrary flows - in this case, a different flow from another scene from the KITTI 2015 training dataset or an arbitrary flow corresponding to an image with the number 42. Note that the perturbations become more effective as the number of steps of adversarial optimization increases. For additional examples see Supplement Section 9.

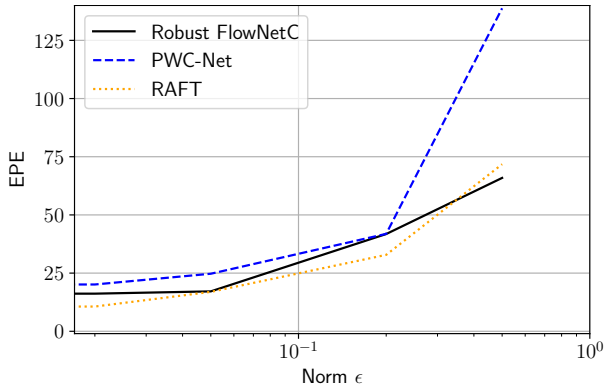


Figure 12. **Universal adversarial attacks.** We show EPE for universal adversarial attacks on KITTI 2015 training dataset for different L_∞ norms (*i.e.*, $\epsilon = \{0.0, 0.02, 0.05, 0.2, 0.5\}$) and flow networks. See exemplary images in Supplement Section 10.

Universal adversarial attacks. We adapted the adversarial optimization of Ranjan *et al*. [25] to craft universal adversarial perturbations. We used the I-FGSM attack with five steps, learning rate $\alpha = 0.002$, and l_2 loss; all other parts remain the same as for adversarial patch optimization. Figure 12 shows that there is no severe drop in flow performance for smaller L_∞ norms; only for larger L_∞ norms does the flow performance drop significantly. We find that (imperceptible) universal adversarial perturbations do not retain the severe effect of white-box adversarial attacks.

8. Discussion

We have shown that self-similar patterns in conjunction with the correlation layer explain the vulnerability of flow networks to adversarial patch attacks. Self-similar patterns are a well-known problem for optical flow estimation and can be related to the aperture problem. In fact, we showed that a simple handcrafted self-similar patch has almost the same effect as an optimized adversarial patch.

As we understand the cause of the problem, there is a reliable way to prevent it: increasing the depth, and thereby increasing the receptive field size, such that the ambiguity caused by the self-similar pattern gets resolved. Many modern networks already have a deep encoder before the correlation layer with a large enough receptive field, and, hence are robust to patch-based attacks via self-similar patterns. Thanks to our analysis, this is not simply a coincidence but can be explained.

We also showed that with targeted adversarial perturbations, an attacker can produce virtually every desired flow. We also find that universal adversarial perturbations do not retain the effect of white-box adversarial attacks. This leads to an interesting interpretation: well-designed flow networks are not vulnerable to adversarial perturbations themselves but to the superposition of image pairs and a corresponding adversarial perturbation. In practice, this means that flow networks are robust to adversarial attacks as long as attackers do not have access to the image stream.

Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG) – BR 3815/10-1, INST 39/1108-1, and the German Federal Ministry for Economic Affairs and Climate Action" within the project KI Delta Learning – 19A19013N.

References

[1] Anish Athalye and Nicholas Carlini. On the robustness of the cvpr 2018 white-box adversarial example defenses. *arXiv*, 2018. 1

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 6

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *ICML*, 2018. 2

[4] Michael J Black and Padmanabhan Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *CVIU*, 1996. 2

[5] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv*, 2017. 2

[6] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004. 2

[7] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. PAMI*, 2010. 2

[8] 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. 1, 2, 3, 4, 5, 6

[9] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018. 2

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 3

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[12] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *NeurIPS*, 2006. 3

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[14] Jan Hendrik Metzen, Mummadi Chaithanya Kumar, Thomas Brox, and Volker Fischer. Universal adversarial perturbations against semantic image segmentation. In *ICCV*, 2017. 2

[15] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 2

[16] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 1981. 2

[17] 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, 3, 5, 6

[18] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ICLR*, 2017. 2, 7

[19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[20] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 6

[21] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 3

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017. 2

[23] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 2

[24] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 2, 3, 5

[25] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J Black. Attacking optical flow. In *ICCV*, 2019. 1, 2, 3, 8

[26] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, 2019. 6

[27] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 2019. 2

[28] 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. 2, 3, 4, 5

[29] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE Trans. PAMI*, 2019. 6

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 2

[31] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2, 3, 4, 5, 7

[32] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008. 3

[33] Alex Wong, Mukund Mundhra, and Stefano Soatto. Stereopagnosia: Fooling stereo networks with adversarial perturbations. In *AAAI*, 2021. 2, 7

[34] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 2020. 1