Adversarial Vulnerability of Temporal Feature Networks for Object Detection

Svetlana Pavlitskaya[1], Nikolai Polley[2], Michael Weber[1], and J. Marius Zöllner[1,2]

[1] FZI Research Center for Information Technology, 76131 Karlsruhe, Germany
`pavlitskaya@fzi.de`
[2] Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

Abstract. Taking into account information across the temporal domain helps to improve environment perception in autonomous driving. However, it has not been studied so far whether temporally fused neural networks are vulnerable to deliberately generated perturbations, i.e. adversarial attacks, or whether temporal history is an inherent defense against them. In this work, we study whether temporal feature networks for object detection are vulnerable to universal adversarial attacks. We evaluate attacks of two types: imperceptible noise for the whole image and locally-bound adversarial patch. In both cases, perturbations are generated in a white-box manner using PGD. Our experiments confirm, that attacking even a portion of a temporal input suffices to fool the network. We visually assess generated perturbations to gain insights into the functioning of attacks. To enhance the robustness, we apply adversarial training using 5-PGD. Our experiments on KITTI and nuScenes datasets demonstrate, that a model robustified via K-PGD is able to withstand the studied attacks while keeping the mAP-based performance comparable to that of an unattacked model.

Keywords: adversarial attacks, temporal fusion, object detection

1 Introduction

Deep neural networks (DNNs) have become an indispensable component of environment perception in autonomous driving systems. The inherent vulnerability of DNNs to adversarial attacks [25] makes adversarial robustness one of the crucial requirements before their wide adoption in autonomous vehicles is possible. Recent studies [3,6,11] demonstrate that adversarial attacks can be performed in the real world and thus present a significant threat to self-driving cars.

Previous works have already investigated adversarial vulnerability of DNNs for specific tasks like object detection [11] or semantic segmentation [14,17], and also of DNNs with specific architectures like sensor fusion [29] or multi-task learning [13]. In this work, we focus on temporal feature networks as a model under attack. Although the emphasis of recent studies was mostly on LiDAR data used in conjunction with camera images, the temporal fusion is a further
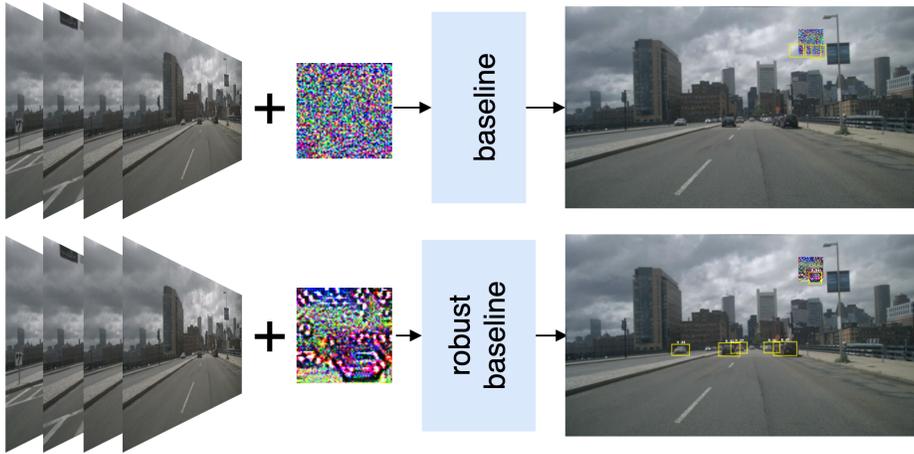
Fig. 1: We evaluate universal attacks on temporal feature networks for object detection (here – with a universal patch). Adversarial patch suppresses all detections of an undefended baseline and creates fake detections instead (top). Detection of ground truth objects by a robust model is no longer restrained by an adversarial patch (bottom). Note, that the patch against robust models has a more complex structure

approach to increase the object detection accuracy, which deserves attention. There has been very little research into fusing multiple images in the temporal dimension for object detection. Object detectors with temporal fusion, however, receive more context data from previous images and outperform single-image object detectors [30]. In particular, the prediction of obstructed objects that are visible in previous images might be a possible advantage.

To the best of our knowledge, we are the first to explore attacks and defenses for this type of DNNs. As an exemplary model under attack, we consider the late slow fusion architecture for object detection proposed by Weber et al. [30], which is in turn inspired by DSOD [23] and expanded to incorporate several input images. The model proposed by Weber et al. outperformed its non-temporal counterpart, thus demonstrating the importance of the temporal history for the object detection accuracy. The main goal of our work is to study, whether CNNs with temporal fusion are prone to adversarial attacks and what is the impact of the temporal history on the adversarial vulnerability of these models.

For this, we evaluate two variants of universal white-box attacks against this model: with an adversarial patch (see Figure 1) and with adversarial noise. In our setting, a single instance of malicious input is generated to attack all possible data. This way, if printed out, the generated patch can be used for an attack in the real world. The latter was already demonstrated for state-of-the-art object detectors in previous works [3,24,26]. To increase the adversarial robustness of the model under attack, we consider adversarial training using K-PGD [12].

2   Related Work

2.1   Adversarial Attacks

Since the discovery of adversarial attacks by Szegedy et al. [25], a number of algorithms to attack DNNs have been proposed, the most prominent being the Fast Gradient Sign Method (FGSM) [9] and the Projected Gradient Descent (PGD) [12].

While the mentioned attacks are traditionally performed in a per-instance manner, i.e. an adversarial perturbation is applied to a single image in order to fool a model, universal perturbations, that are able to attack multiple instances, are also possible [15]. Universal adversarial inputs pose a special threat, because they are also able to attack images beyond the training data.

Another line of research aims at developing physical adversarial attacks. For this, visible adversarial perturbations are generated within a certain image area. The resulting *adversarial patch* can then be printed out to perform an attack in the real world. After their introduction by Brown et al. in [3], adversarial patches have been shown to successfully fool various deep learning models, including object detectors [11,18], semantic segmentation networks [17] and end-to-end driving models [19]. A combination of adversarial patches with universal attacks is especially interesting. Taking into consideration the transferability of adversarial examples across DNNs, such attacks might also be performed in a black-box manner in the real world [26,32].

Although no previous work on adversarial vulnerability of temporal fusion networks is known, a certain effort was already made in the community to develop adversarial attacks against sensor fusion models [33,29]. In particular, the work by Yu et al. has revealed, that late fusion is more robust against attacks than early fusion [33]. The evaluated dataset, however, is relatively small with only 306 training and 132 validation samples from the KITTI dataset.

2.2   Adversarial Training

Adversarial training (AT) is currently one of the few defenses that are able to combat even strong attacks [2]. It consists in training a DNN while adding adversarial inputs to each minibatch of training data. It has recently been shown, that adversarial training not only increases the robustness of neural networks to adversarial attacks but also leads to better interpretability [28].

While the idea originates from [9], the first strong defense was demonstrated with a multi-step PGD algorithm (the K-PGD adversarial training) [12]. For each minibatch, a forward/backward step is first executed k times to generate an adversarial input and then a single forward/backward step follows, which aims to update the model parameters. The PGD loop thus drastically increases the overall training time. For this reason, K-PGD adversarial training is intractable for large datasets.

One of the recently proposed strategies to speed up adversarial training is the so-called AT for free [21], which reuses gradient information during training. Instead of performing separate gradient calculations to generate adversarial

examples during training, adversarial perturbations and model parameters are updated simultaneously in a single backward pass. This way, multiple FGSM steps are performed on a single minibatch to simulate the PGD algorithm while concurrently training the model. The authors were the first to apply adversarial training to the large-scale ImageNet classification task. The robustified models demonstrate resistance to attacks, comparable to that of K-PGD, while being 7 to 30 times faster. The approach, however, is still more time-consuming than standard training.

A similar method to accelerate adversarial training named YOPO (You Only Propagate Once) is proposed by Zhang et al. [34]. In YOPO, the gradients of the early network layers are frozen and reused to generate an adversarial input. YOPO is four to five times faster than K-PGD, although the results are only provided for relatively small datasets. YOPO reaches a performance similar to the free adversarial training but is less computationally expensive.

Most recently, a further approach to accelerate adversarial training was proposed by Wong et al. [31]. Starting with the assumption, that iterative attacks like K-PGD do not necessarily lead to more robust defenses, the authors propose to use R-FGSM AT instead. R-FGSM applies FGSM after a random initialization. This adversarial training method is claimed to be as effective as K-PGD.

Enhancing adversarial training to increase robustness against universal attacks was first addressed by Shafahi et al. [22]. Each training step uses FGSM to update a universal adversarial perturbation, which is then simultaneously used to update the model parameters. The proposed extension to adversarial training introduces almost no additional computational cost, which makes adversarial training on large datasets possible.

A further approach to harden DNNs against universal attacks is the shared adversarial training, proposed in [16]. To generate a shared perturbation, each batch is split into heaps, which are then attacked with single perturbations. These perturbations are aggregated and shared across heaps and further used for standard adversarial training.

3   Adversarial Attacks

3.1   Threat Model

We consider two types of white-box attacks in this work: adversarial patch and adversarial noise. All attacks are performed in a universal manner, i.e. a single perturbation is used to to attack all images [15]. We use a slightly modified version of the PGD algorithm [12] for the attack. In order to apply PGD in a universal manner, an empty mask is introduced, which is added to each input image. We then only update this mask in contrast to the original PGD, which updates the input images directly.

We use Adam [10] for faster PGD convergence as suggested in [5]. We also do not take a sign of the gradients but use actual gradient values instead. In an unsigned case, pixels, that strongly affect the prediction, can be modified to a

much larger extent than less important pixels. Noise initialization strategy is a further setting, influencing training speed. For the FGSM attack, the advantage of initialization with random values has already been shown [27]. We perform patch-based attacks with randomly initialized patch values and noise-based attacks initialized with Xavier [8].

3.2 Adversarial Training

Adversarial training expands the training dataset with adversarial examples, created on the fly during training. Training on this dataset should enable the model to predict the correct label even in the presence of an attack.

We consider the established K-PGD AT with patch/noise generated for each input sequence. We create a single adversarial example (with either patch or noise) for all images in an input sequence. Thus, the generated adversarial perturbation is not universal and is only intended to fool the data from the current batch. The usage of this approach is motivated by a recent observation, that defending against non-universal attacks also protects against universal attacks [16].

Although Free [21] and YOPO [34] approaches offer a considerable speed up in training, they are not applicable in our case. Both algorithms require the same loss function to train a model and create adversarial examples. This is only possible for untargeted attacks, where the goal is to maximize the loss for the correct class. Instead, we want to perform object vanishing attacks, which require different loss functions than training the model. This way, reusing the gradients, already computed for the parameter update step, as foreseen in these AT algorithms, cannot be performed in our case.

4 Experimental Setup

4.1 Dataset

For the evaluation, we consider the late slow fusion architecture for object detection, proposed by Weber et al. [30]. While the original work by Weber et al. has focused on the KITTI Object Detection dataset [7], we additionally run our experiments on the nuScenes data [4]. We focus on the models with four input frames with equal temporal distance.

KITTI Images from the object detection benchmark are resized to 1224×370. We only select images, that have the corresponding three temporally preceding frames, delivered in the dataset, resulting in 3689 sequences of length four for training and 3754 sequences of length four for validation.

nuScenes The dataset contains 850 annotated scenes, whereas each scene contains about 40 keyframes per camera, taken at a frequency of 2Hz. We generate input sequences with a length of four from the keyframes as follows: if keyframes
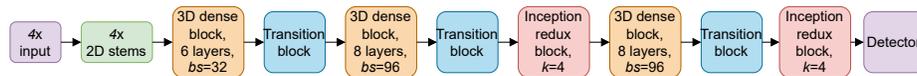
Fig. 2: Architecture of the temporal feature network used for baselines. *bs* denotes the bottleneck size. Inception block uses $k{\times}1{\times}1$ convolutions for depth reduction

{*a, b, c, d, e*} belong to the same scene, the two sequences {*a, b, c, d*} and {*b, c, d, e*} are created. For the training dataset, we use front-facing and backward-facing images. For the latter, we have inverted the order of the sequence. Additionally, we augment the training dataset by horizontal mirroring of images. Since the nuScenes dataset contains images from a left-driving country, the introduced images remain within the domain. We use the 700/150 scene split for training and validation, as recommended by nuScenes, so that the training subset contains 52060 and the validation subset – 5569 sequences of length four. Images are resized from an initial 1600×900 to 1024×576 pixels and normalized to a range of [0,1]. For attack, we do not apply horizontally mirrored images, so that the training dataset contains 26030 images.

Since we focus on the detection of cars and pedestrians, the nuScenes classes *adult, child, construction worker, police officer, stroller* and *wheelchair* are consolidated into one new class *pedestrian*. As a second class we define *car*, which corresponds to the nuScenes class *vehicle.car*. We do not include other vehicles available in the nuScenes labels (e.g. busses, motorcycles, trucks, and trailers) to minimize intra-class variance.

4.2   Baselines

The baselines follow the late slow fusion architecture (see Figure 2) proposed by Weber et al. [30]. The input is a sequence of images with temporal distance $\Delta t = 500ms$, whereas prediction is learned for the last input frame. Each input image is processed by a separate 2D convolutional stem, being identical to the 2D DSOD detector [23]. Stems are followed by a series of 3D dense blocks, intervened with transition blocks. The subsequent Inception redux blocks aim at reducing the temporal depth by half. Finally, a detector subnet similar to that of YOLOv2 [20] follows.

For each dataset, we deliberately defined new anchor boxes to have high IoU scores with the bounding boxes in the training data. For this, we analyzed all objects in the training dataset and clustered them using k-means with IoU as a distance metric. As a result, we obtained eight anchor boxes, which comprise various box forms to detect both pedestrians (upright vertical rectangles) and cars (horizontal rectangles).

We train the baselines on an NVIDIA RTX 2080 Ti GPU. KITTI models are trained for 50 epochs, whereas nuScenes models are trained for 100 epochs. For validation, the PASCAL VOC implementation of mAP is used. The mAP is always calculated for a confidence threshold of 0.01 and nms threshold of 0.5.

Model / Attack	No attack	Universal patch	Universal noise $\epsilon = 5/255$	Universal noise, $\epsilon = 10/255$
Baseline	74.82	13.62	34.73	2.08
5-PGD AT with patch	66.47	52.18	24.46	23.07
5-PGD AT with noise, $\epsilon = 5/255$	62.77	42.82	61.52	61.75
5-PGD AT with noise, $\epsilon = 10/255$	53.24	32.15	53.24	53.07

Table 1: AP_{car} in % of the 1-class KITTI baseline and AT models

Model / Attack	No attack	Universal patch	Universal noise $\epsilon = 5/255$	Universal noise, $\epsilon = 10/255$
Baseline	50.93	4.36	34.49	4.43
5-PGD AT with patch	50.86	44.18	23.88	14.83
5-PGD AT with noise, $\epsilon = 5/255$	41.95	27.90	41.95	41.35
5-PGD AT with noise, $\epsilon = 10/255$	38.50	25.76	38.49	38.49

Table 2: mAP in % of the 2-class KITTI baseline and AT models

We train models to detect either only objects of the class *car* (1-class models) or objects of the classes *car* and *pedestrian* (2-class model). Tables 1 and 2 show performance of the 1-class and 2-class KITTI baselines. Whereas average precision for the class *car* is comparable for both models (AP_{car} reaches 74.82% for the 1-class model vs. 73.21% for the 2-class), the worse performance for the underrepresented class *pedestrian* (28.65%) explains the overall worse mAP of the 2-class model.

Tables 3 and 4 summarize results on the nuScenes baselines. Due to class imbalance (108K annotated cars vs. 48K annotated pedestrians), the results for the 2-class baseline for the underrepresented class *pedestrian* are again significantly worse.

The results achieved on the KITTI and nuScenes baselines, described below, are comparable if the opposite is not stated.

4.3   Adversarial Noise Attack

For a universal noise attack, we have initially applied the proposed universal PGD with the changes motivated above. However, it turned out, that this attack leads to the detection of nonexistent new objects (see Figure 3a). The loss maximization goal apparently favors creating new features that resemble objects rather than preventing the detection of existing objects. We have therefore adapted the attack algorithm by replacing the gradient ascent with the gradient descent on an empty label. Figure 3b shows, that adversarial noise generated using targeted PGD on an empty label can successfully suppress all objects

Model / Attack	No attack	Universal patch	Universal noise $\epsilon = 5/255$	Universal noise, $\epsilon = 10/255$
Baseline	73.20	6.20	9.03	0.15
5-PGD AT with patch	74.04	61.89	27.71	27.65
5-PGD AT with noise, $\epsilon = 5/255$	72.24	3.51	71.85	68.19

Table 3: AP_{car} in % of the 1-class nuScenes baseline and AT models

Model / Attack	No attack	Universal patch	Universal noise $\epsilon = 5/255$	Universal noise, $\epsilon = 10/255$
Baseline	27.55	0.98	0.74	0.16
5-PGD AT with patch	28.69	17.95	7.02	0.53
5-PGD AT with noise, $\epsilon = 5/255$	27.10	12.83	25.78	20.69
AT with reused patches	27.24	9.13	6.01	0.48
R-FGSM AT with noise, $\epsilon = 5/255$	27.51	2.72	0.24	0.02

Table 4: mAP in % of the 2-class nuScenes baseline and AT models. For this model, AT with reused patches and R-FGSM AT were additionally evaluated

present in the input. All perturbations are trained using the Adam optimizer for 100 epochs.

4.4   Adversarial Patch Attack

To generate a universal patch, we apply PGD with unsigned gradients using the Adam optimizer for 100 epochs. We evaluated patches of size 71×71, 51×51 and 31×31. As expected, the largest patch led to a stronger attack and was used in the following experiments. Note, that a 71×71 patch still takes only about 1% of the image area both in the case of KITTI and nuScenes images.

To evaluate the impact of patch position, we evaluated a total of 33 patch positions. Average precision for different positions fluctuated only within few percentage points, so patch position apparently has only a minor impact on its attack strength.

4.5   Adversarial Training

We apply adversarial training as a method to improve the robustness of the studied models. We consider K-PGD AT and two additional AT strategies: (1) reusing the patches, already generated for the baseline, during the training and (2) R-FGSM.

In the case of K-PGD AT, creating an adversarial example iteratively with k steps increases the number of forward and backward propagations by a factor of

(a) Untargeted attack with gradient as-
cent

(b) Targeted attack for an empty label
with gradient descent

Fig. 3: Predictions of the 2-class nuScenes baseline on images attacked with uni-
versal noise, $\epsilon = 5/255$

(a) Untargeted attack with gradient as-
cent on the 2-class model

(b) Untargeted attack with gradient as-
cent on the 1-class model

Fig. 4: Predictions of the nuScenes baselines on images attacked with universal
patch

k. We have therefore used a small $k = 5$ per adversarial attack during training
and drastically increased the learning rate of the Adam optimizer to make the
attacks possible. Training 5-PGD AT both on nuScenes and KITTI thus took
five times longer than the corresponding baseline, both for the adversarial noise
and the adversarial patch.

In the case of the pre-generated patches for adversarial training, we first
generated a pool of patches against the baseline as described above. We then
trained a model, while adding a randomly chosen patch at each training step
with a 50% probability. The AT with reused patches involved no generation of
new patches, therefore its duration is comparable to regular model training.

Finally, R-FGSM AT was trained with adversarial noise with $\epsilon = 5/255$.

5   Evaluation

5.1   Attacks on 1-class vs. on 2-class Baselines

Visual assessment of the adversarial noise patterns helps to understand, how the attack functions. We observed different behavior of attacks on 1-class and 2-class models. In particular, universal noise, generated to attack 1-class baselines evidently contains structures resembling cars, whereas noise attacking 2-class models exhibits no such patterns (see Figure 5). Apparently, the attack aims at mimicking existing objects, if they all belong to one class.

Adversarial patches, generated for both types of models, however, look similar. Patch-based attacks on nuScenes baselines tend to detect non-existing cars in a patch (see Figure 4), whereas attacks against the KITTI 2-class baseline rather find pedestrians in a patch. This might be explained by a different portion of pedestrians in the corresponding datasets. Patches against nuScenes 2-class model never mimicked pedestrians, because they are highly underrepresented in the training data.

5.2   Impact of Temporal Horizon

Temporal fusion models have better performance due to the incorporation of the temporal history. To assess which portion of the history is enough to attack the model, we perform the evaluation exemplary with the $\epsilon = 10/255$ adversarial noise attack on the 1-class KITTI baseline. We have attacked single frames using adversarial noise, which was initially generated for the whole input sequence of length four and with adversarial noise, generated for the corresponding portion of the input (see Table 5). We observed, that perturbations, deliberately generated for specific frames, work better when attacking them, than those generated for the whole input sequence. For both cases, the attack works the best, when frames, immediately preceding the current frame, are attacked. On contrary, attacking only the oldest frame leads to the worst results. Also, perturbing only the frame for which the prediction is done and not attacking the temporal history at all leads to a significantly weaker attack. Finally, the more preceding temporal history frames are attacked, the better the results.

These results confirm, that the later images in the input sequence are more important for the prediction. Furthermore, single attacked images that appear later in the input sequence, cause larger error than those which appear earlier.

Furthermore, we evaluate the impact of the temporal horizon. In addition to the already evaluated models with four input images, we also evaluated models with a smaller sequence length. In particular, for the 1-class nuScenes model, we observe about 10% for each reduction of the number of input images: 73.20% mAP for the temporal history of length four, 63.12% for the length three and 52.18% for the length two. The attack strength also decreases correspondingly. We thus conclude that a larger temporal horizon helps to enhance not only the performance on the clean data, but also the adversarial robustness.

(a) KITTI 1-class baseline   (b) KITTI 2-class baseline

(c) KITTI 1-class robustified   (d) KITTI 2-class robustified

(e) nuScenes 1-class baseline   (f) nuScenes 2-class baseline

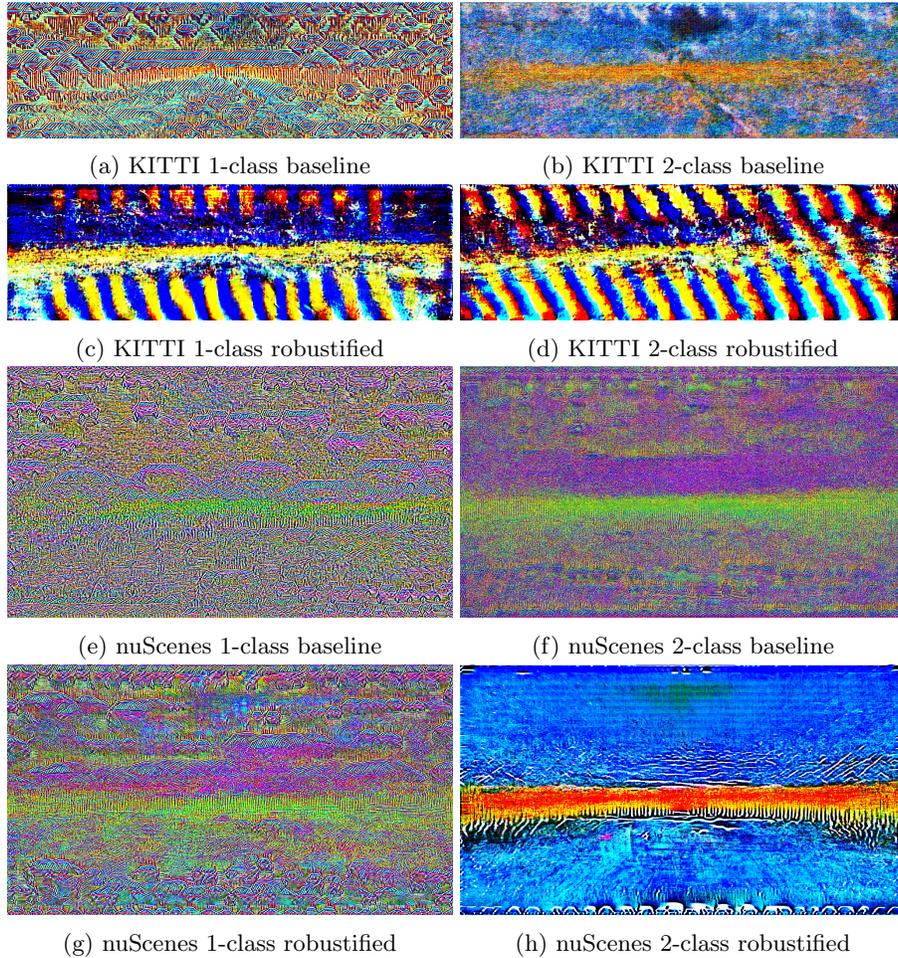(g) nuScenes 1-class robustified   (h) nuScenes 2-class robustified

Fig. 5: Universal noise with $\epsilon = 5/255$, generated to attack baselines and models, robustified via 5-PGD AT with noise, $\epsilon = 5/255$. Original pixel value range [-5,5] mapped to [0,255] for better visibility
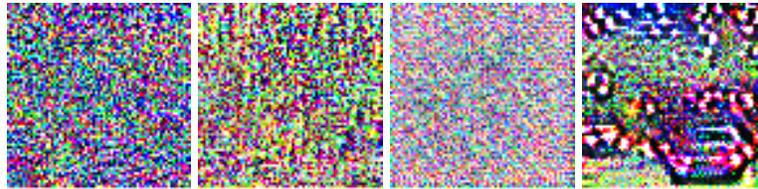
5.3   Robustness of the Adversarially Trained Models

To evaluate the robustness of the robustified models, we attack them again with newly generated universal patches and noises. Tables 1-4 demonstrate the results.

All models robustified via AT demonstrate performance similar to the baseline on non-attacked data and almost no accuracy drop on the malicious data for the nuScenes models and a small drop for the KITTI models. For the KITTI dataset, we have additionally evaluated AT with adversarial noise with $\epsilon = 5/255$ and $\epsilon = 10/255$ (see Tables 1 and 2). The larger epsilon leads to worse performance on clean data due to larger perturbation.

Attacked sequence part				Noise generated for all four inputs	Noise generated for each evaluated case
t_{-3}	t_{-2}	t_{-1}	t		
				74.82	74.82
				70.68	58.46
				67.02	45.43
				53.14	36.21
				23.91	5.77
				50.84	22.14
				32.45	10.13
				6.19	2.96
				3.80	2.74
				2.08	2.08

Table 5: Attacking a part of the sequence. The attacked frames are highlighted red, prediction is performed for the frame t. mAP in % is reported for the 1-class KITTI baseline, attacked with $\epsilon = 10/255$ adversarial noise, generated either for all four inputs or for each evaluated case

(a) Baseline   (b)   AT   with reused patches   (c) 5-PGD   AT with patch   (d) 5-PGD   AT with noise

Fig. 6: Universal patches to attack different nuScenes 2-class models

Moreover, we have evaluated AT with reused patches and R-FGSM AT on the 2-class nuScenes model (see Table 4). As expected, the defended model with reused patches is less robust to attacks than the one which was trained with 5-PGD. Surprisingly, the R-FGSM AT method has completely failed to defend against attacks. We explain this behavior with the *catastrophic overfitting* phenomenon, mentioned in the original work by Wong et al. [31] and in a more recent study by Andriushchenko et al. [1], which challenges the original claim that using randomized initialization prevents this overfitting.

Figure 6 compares patches, generated for the adversarially trained models with the patch generated against the nuScenes 2-class baseline. In the case of patch reuse, the patch contains more green and yellow pixels than the original patch. In the case of K-PGD adversarial training, the patch is brighter and contains more white pixels. Interestingly, the patch generated for a model, which

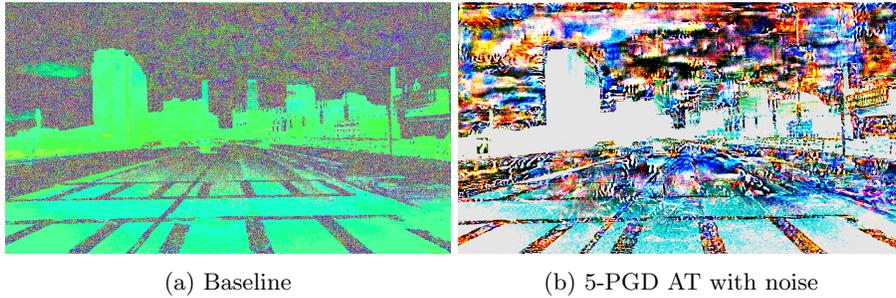(a) Baseline   (b) 5-PGD AT with noise

Fig. 7: Per-instance adversarial noise with $\epsilon = 5/255$ generated to attack nuScenes 2-class model on a single input sequence. Original pixel value range [-5,5] mapped to [0,255] for better visibility
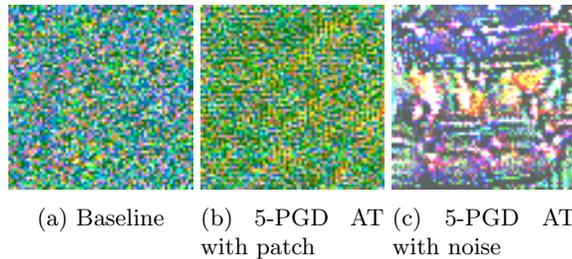

(a) Baseline   (b) 5-PGD AT   (c) 5-PGD AT
                 with patch   with noise

Fig. 8: Per-instance adversarial patch generated to attack nuScenes 2-class model on single input sequence

was adversarially trained with adversarial noise, is the only one that contains structures, resembling a car.

Figure 5 compares universal noises, generated to attack the KITTI and nuScenes baselines and the corresponding model defended via 5-PGD AT with $\epsilon = 5/255$ noise. Both contain a streak of color at the horizon line and wave-like patterns at the bottom. We again conclude, that perturbation attacking robustified models exhibit more complex structure.

5.4   Robustness of the AT-trained Models against Per-instance Attacks

Finally, we examine whether adversarially trained models also become robust against per-instance attacks. For this, we take an exemplary input sequence and generate adversarial perturbations against it. Each attack is trained for 1000 steps.

Per-instance noise attack (see Figure 7) manage to completely suppress all detections of the corresponding model. Analogously to universal attacks, non-universal noise attacks against the hardened model look much more complex. Similarly, per-instance patches (see Figure 8) against the robustified model show

complex structures resembling cars. Interestingly, this patch is also unable to efficiently attack the model, several cars are still correctly detected after applying this patch.

Overall, while models hardened with the evaluated adversarial training strategies are very successful in resisting universal attacks while preserving high accuracy, they are still unprotected against per-instance attacks. Universal attacks are, however, much more feasible with regard to real-life settings.

6   Conclusion

In this work, we have studied the adversarial vulnerability of temporal feature networks for object detection. The architecture proposed by Weber et al. [30] was used as an exemplary model under attack.

Our experiments on KITTI and nuScenes datasets have demonstrated that the studied temporal fusion model is susceptible to both universal patch and noise attacks. Furthermore, we have explored different adversarial training strategies as a defense measure. Out of the three evaluated methods, the 5-PGD approach with a per-instance adversarial noise has proven to be the most powerful. The R-FGSM strategy, however, has failed to defend against the studied attacks. 5-PGD adversarial training was able to withstand newly created universal attacks. The robustified networks have also demonstrated only a slight drop in performance on clean data.

Our experiments with attacking a portion of the temporal history have demonstrated, that the frames, immediately preceding the current frame, have a greater impact on the model decision and thus lead to stronger attacks when manipulated. We have further observed, that reducing the temporal horizon leads to worse performance and adversarial robustness of the model.

We have compared the universal and per-instance perturbations generated to attack the baseline and the robustified models. In all cases, we observed that in order to attack a hardened neural network, the adversarial perturbation has to exhibit a much more complex structure. In particular, a universal patch against the most robust 5-PGD with noise contains a pattern resembling a car.

Our adversarially trained models, however, remain vulnerable to non-universal attacks like per-instance-generated noise or patch. This stresses the need for further research in this area.

Since the computation time for adversarial training is still a bottleneck, adapting gradient re-usage strategies like [21] or [34] for models, which use different loss functions to learn an adversarial perturbation and to update the model weights, might be a promising line of research for future.

References

1. Andriushchenko, M., Flammarion, N.: Understanding and Improving Fast Adversarial Training. Advances in Neural Information Processing Systems (NIPS) (2020)
2. Athalye, A., Carlini, N., Wagner, D.: Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In: International Conference on Machine Learning (ICML). PMLR (2018)
3. Brown, T.B., Mané, D., Roy, A., Abadi, M., Gilmer, J.: Adversarial Patch. In: Advances in Neural Information Processing Systems (NIPS) - Workshops (2017)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., et al.: nuScenes: A multimodal dataset for autonomous driving. Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
5. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: Symposium on Security and Privacy (SP). IEEE (2017)
6. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust Physical-World Attacks on Deep Learning Visual Classification. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2018)
7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012), https://doi.org/10.1109/CVPR.2012.6248074
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2010)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. International Conference on Learning Representations (ICLR) (2015)
10. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. International Conference on Learning Representations (ICLR) (2015)
11. Lee, M., Kolter, Z.: On Physical Adversarial Patches for Object Detection. CoRR **abs/1906.11897** (2019)
12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. International Conference on Learning Representations (ICLR) (2018)
13. Mao, C., Gupta, A., Nitin, V., Ray, B., Song, S., Yang, J., Vondrick, C.: Multitask learning strengthens adversarial robustness. In: European Conference on Computer Vision (ECCV). Springer (2020)
14. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: International Conference on Computer Vision (ICCV). IEEE Computer Society (2017)
15. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal Adversarial Perturbations. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
16. Mummadi, C.K., Brox, T., Metzen, J.H.: Defending Against Universal Perturbations with Shared Adversarial Training. In: International Conference on Computer Vision (ICCV) (2019)
17. Nesti, F., Rossolini, G., Nair, S., Biondi, A., Buttazzo, G.C.: Evaluating the robustness of semantic segmentation for autonomous driving against real-world adversarial patch attacks. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV). IEEE (2022)

18. Pavlitskaya, S., Codau, B., Zöllner, J.M.: Feasibility of inconspicuous gan-generated adversarial patches against object detection. In: International Joint Conference on Artificial Intelligence (IJCAI) - Workshops (2022)
19. Pavlitskaya, S., Ünver, S., Zöllner, J.M.: Feasibility and Suppression of Adversarial Patch Attacks on End-to-End Vehicle Control. In: International Conference on Intelligent Transportation Systems (ITSC). IEEE (2020)
20. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
21. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! In: Advances in Neural Information Processing Systems (NIPS) (2019)
22. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J., Davis, L.S., Goldstein, T.: Universal Adversarial Training. In: AAAI Conference on Artificial Intelligence (2020)
23. Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: DSOD: Learning Deeply Supervised Object Detectors From Scratch. In: International Conference on Computer Vision (ICCV). IEEE (2017)
24. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramèr, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: USENIX Workshop on Offensive Technologies, WOOT 2. USENIX Association (2018)
25. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing Properties of Neural Networks. International Conference on Learning Representations (ICLR) (2014)
26. Thys, S., Ranst, W.V., Goedemé, T.: Fooling automated surveillance cameras: Adversarial patches to attack person detection. In: Conference on Computer Vision and Pattern Recognition (CVPR) - Workshops. Computer Vision Foundation / IEEE (2019)
27. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble Adversarial Training: Attacks and Defenses. International Conference on Learning Representations (ICLR) (2018)
28. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness May Be at Odds with Accuracy. In: International Conference on Learning Representations (ICLR) (2019)
29. Tu, J., Li, H., Yan, X., Ren, M., Chen, Y., Liang, M., Bitar, E., Yumer, E., Urtasun, R.: Exploring Adversarial Robustness of Multi-Sensor Perception Systems in Self Driving. CoRR **abs/2101.06784** (2021)
30. Weber, M., Wald, T., Zöllner, J.M.: Temporal Feature Networks for CNN based Object Detection. In: Intelligent Vehicles Symposium (IV). IEEE (2021)
31. Wong, E., Rice, L., Kolter, J.Z.: Fast is better than free: Revisiting adversarial training. International Conference on Learning Representations (ICLR) (2020)
32. Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P., Wang, Y., Lin, X.: Adversarial t-shirt! evading person detectors in a physical world. In: European Conference on Computer Vision (ECCV). Springer (2020)
33. Yu, Y., Lee, H.J., Kim, B.C., Kim, J.U., Ro, Y.M.: Investigating Vulnerability to Adversarial Examples on Multimodal Data Fusion in Deep Learning. CoRR **abs/2005.10987** (2020)
34. Zhang, D., Zhang, T., Lu, Y., Zhu, Z., Dong, B.: You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle. In: Advances in Neural Information Processing Systems (NIPS) (2019)