

# Adversarial Attacks on Multi-task Visual Perception for Autonomous Driving

Ibrahim Sobh and Ahmed Hamed

Valeo R&D Egypt

Varun Ravi Kumar

Valeo DAR Germany

Technische Universität Ilmenau, Germany

Senthil Yogamani

Valeo Ireland

E-mail: [senthil.yogamani@valeo.com](mailto:senthil.yogamani@valeo.com).

---

**Abstract.** In recent years, deep neural networks (DNNs) have accomplished impressive success in various applications, including autonomous driving perception tasks. However, current deep neural networks are easily deceived by adversarial attacks. This vulnerability raises significant concerns, particularly in safety-critical applications. As a result, research into attacking and defending DNNs has gained much coverage. In this work, detailed adversarial attacks are applied on a diverse multi-task visual perception deep network across distance estimation, semantic segmentation, motion detection, and object detection. The experiments consider both white and black box attacks for targeted and un-targeted cases, while attacking a task and inspecting the effect on all others, in addition to inspecting the effect of applying a simple defense method. We conclude this paper by comparing and discussing the experimental results, proposing insights and future work. The visualizations of the attacks are available at <https://youtu.be/6AixN90budY>. © 2021 Society for Imaging Science and Technology. [DOI: 10.2352/J.ImagingSci.Technol.2021.65.6.060408]

---

## 1. INTRODUCTION

Autonomous Vehicles are expected to significantly reduce accidents [1], where visual perception systems [2–8] are at the core of these vehicles. Despite the notable achievements of DNNs in visual perception, we can easily deceive the networks by adversarial examples that are imperceptible to the human eye but cause the network to fail. Adversarial examples are usually created by deliberately employing imperceptibly small perturbations to the benign inputs resulting in incorrect model outputs. This small perturbation is progressively amplified by a deep network and usually yields inaccurate predictions. Generally speaking, attacks can be a white box or black box depending on the adversary's knowledge (the agent who creates an adversarial example). White box attacks presume full knowledge of the targeted model's design, parameters, and, in some cases, training data. Gradients can thus be calculated efficiently in white box

attacks using the back-propagation algorithm. In contrast, in Black box attacks, the adversary is unaware of the model parameters and has no access to the gradients. Furthermore, attacks can be targeted or un-targeted based on the intention of the adversary. Targeted attacks try to deceive the model into a specific predicted output. In contrast, in the un-targeted attacks, the predicted output itself is irrelevant, and the main goal is to deceive the model into any incorrect output. Fast gradient sign method (FGSM) [9] is an example of a simple yet effective attack for generating adversarial instances. FGSM aims to deceive the classification of the image by adding a small vector obtained by taking the sign of the gradient of the loss function. Moreover, it was shown that robust 3D adversarial objects could deceive deep network classifiers in the physical world [10], despite the combination of viewpoint shifts, camera noise, and other natural transformations. Deceiving surveillance cameras was introduced in Ref. [11] where adversarial patches are designed to attack person detection. Dense Adversary Generation (DAG) algorithm [12] is an example of generating adversarial attacks for semantic segmentation and object detection tasks. It was discovered that the perturbations are exchangeable across different networks, even though they have different training since they share some intrinsic structure that makes them susceptible to a common source of perturbations. In addition to camera sensors, potential vulnerabilities of LiDAR-based autonomous driving detection systems are explored in Ref. [13]. Moreover, the 3D-printed adversarial objects showed effective physical attacks on LiDAR equipped vehicles, raising concerns about autonomous vehicles' safety. Robust Physical Perturbations (RP<sub>2</sub>) [14] is another example that generates robust visual adversarial perturbations under different physical conditions on road sign classifications.

Adversarial robustness [15] and defense methods of neural networks have been studied to improve these networks' resistance to different adversarial attacks. One method for defense is Adversarial training, where adversarial examples besides the clean examples are both used to train the model. Adversarial training can be seen as a sort of simple

---

Received Sept. 8, 2021; accepted for publication Nov. 9, 2021; published online Dec. 3, 2021. Associate Editor: Francisco Imai.

1062-3701/2021/65(6)/060408/9/\$25.00

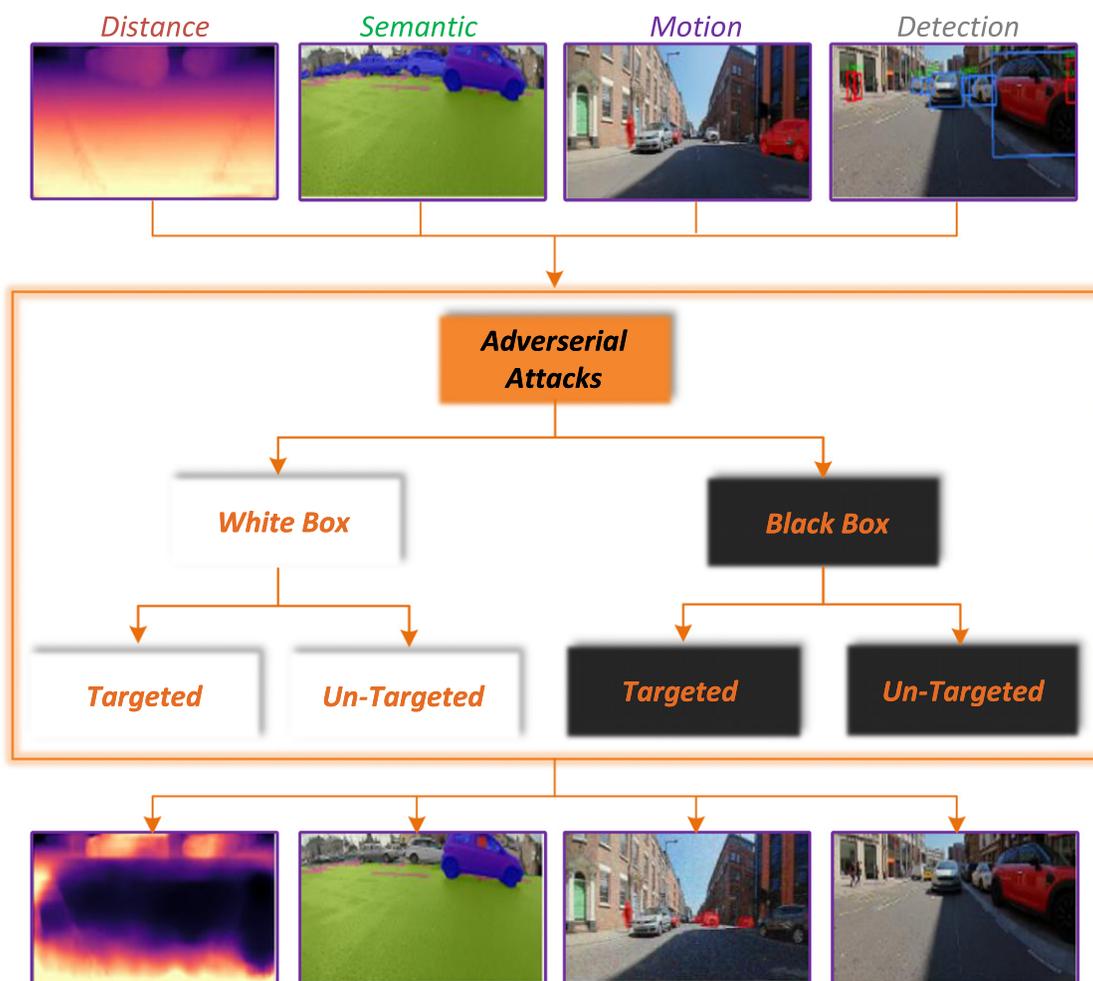


Figure 1. Adversarial attacks on OmniDet [26] MTL model. Distance, segmentation, motion and detection perception tasks are attacked by white and black box methods with targeted and un-targeted objectives, resulting in incorrect model predictions.

data augmentation. Despite being simple, but it cannot cover all the attack cases. In Ref. [16], it is demonstrated that JPEG compression can undo the small adversarial perturbations created by the FGSM. However, this method is not effective for large perturbations. Xu et al. [17] proposed Feature-squeezing for detecting adversarial examples, in which the model is tested on both original input and the input after being pre-processed by feature squeezers such as spatial smoothing. If the difference between the outputs exceeds a certain threshold, we identify the input as an adversarial example. Defense-GAN [18] is another defence technique that employs generative adversarial networks (GAN)s [19], in which it seeks a similar output to a given picture while ignoring adversarial perturbations. It is shown to be a feasible defense that relies on the GAN's expressiveness and generative power. However, training GANs is still a challenging task.

Robust attacks and defenses are still challenging tasks and an active area of research. Most previous works on adversarial attacks focused on single task scenarios. However, in real-life situations, multi-task learning is adopted to solve several tasks at once [20–22]. Accordingly, multi-task

networks are used to leverage the shared knowledge among tasks, leading to better performance, reduced storage, and faster inference [23, 24]. Moreover, it is shown that when models are trained on multiple tasks at once, they become more robust to adversarial attacks on individual tasks [25]. However, defense remains an open challenge. In this work, as shown in Figure 1, white and black box attacks are applied on a multi-task visual perception deep network across distance estimation, semantic segmentation, motion detection, and object detection, taking into consideration both targeted and un-targeted scenarios. For the experiment, while attacking one of the tasks, the attacking curve is plotted to inspect the performance across all tasks over the attacking steps. Additionally, a simple defense method is used across all experiments. Finally, in addition to visual samples of perturbations and performance before and after the attacks, detailed results and comparisons are presented and discussed.

## 2. MULTI-TASK ADVERSARIAL ATTACKS

In this section, the target multi-task network is presented in terms of architecture, data, tasks, and training. The attacks

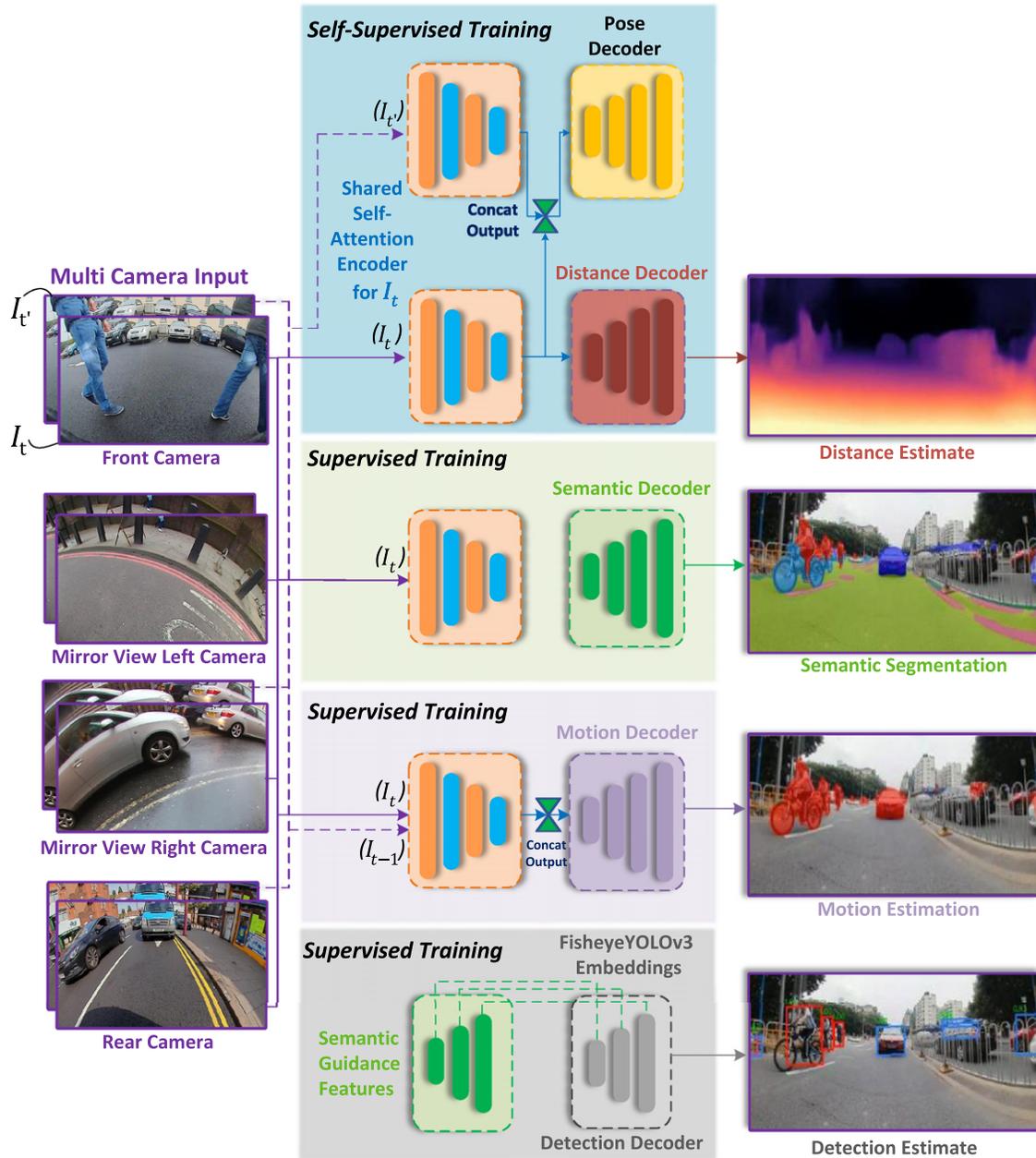


Figure 2. Illustration of baseline multi-task architecture comprising of four tasks [26].

are subsequently detailed for each task, including a white and black box for both targeted and un-targeted cases.

### 2.1 Baseline Multitask Model

We derive the baseline model from the recent work OmniDet [26–29], a six-task complete perception model for surround view fisheye cameras. We focus on the four main perception tasks and skip visual odometry and the soiling detection task. We provide a short overview of the baseline model used and refer to [26] for more details. A high-level architecture of the model is shown in Figure 2. It comprises a shared ResNet18 encoder and four decoders for each task. Motion decoder uses additionally previous encoder feature

in Siamese encoder style. 2D box detection task has the five important objects, namely *pedestrians*, *vehicles*, *riders*, *traffic sign*, and *traffic lights*. Segmentation task has *vehicles*, *pedestrians*, *cyclists*, *road*, *lanes*, and *curbs* categories. Motion task has binary segmentation corresponding to static and moving i.e. dynamic pixels. Depth task provides scale-aware distance in 3D space validated by occlusion corrected LiDAR depth [30, 31]. The model is trained jointly using the public WoodScape [32] dataset comprising 8K samples and evaluated on 2K samples.

In this paragraph, we briefly summarize the loss functions used for training. We construct a self-supervised monocular structure-from-motion (SfM) system for distance

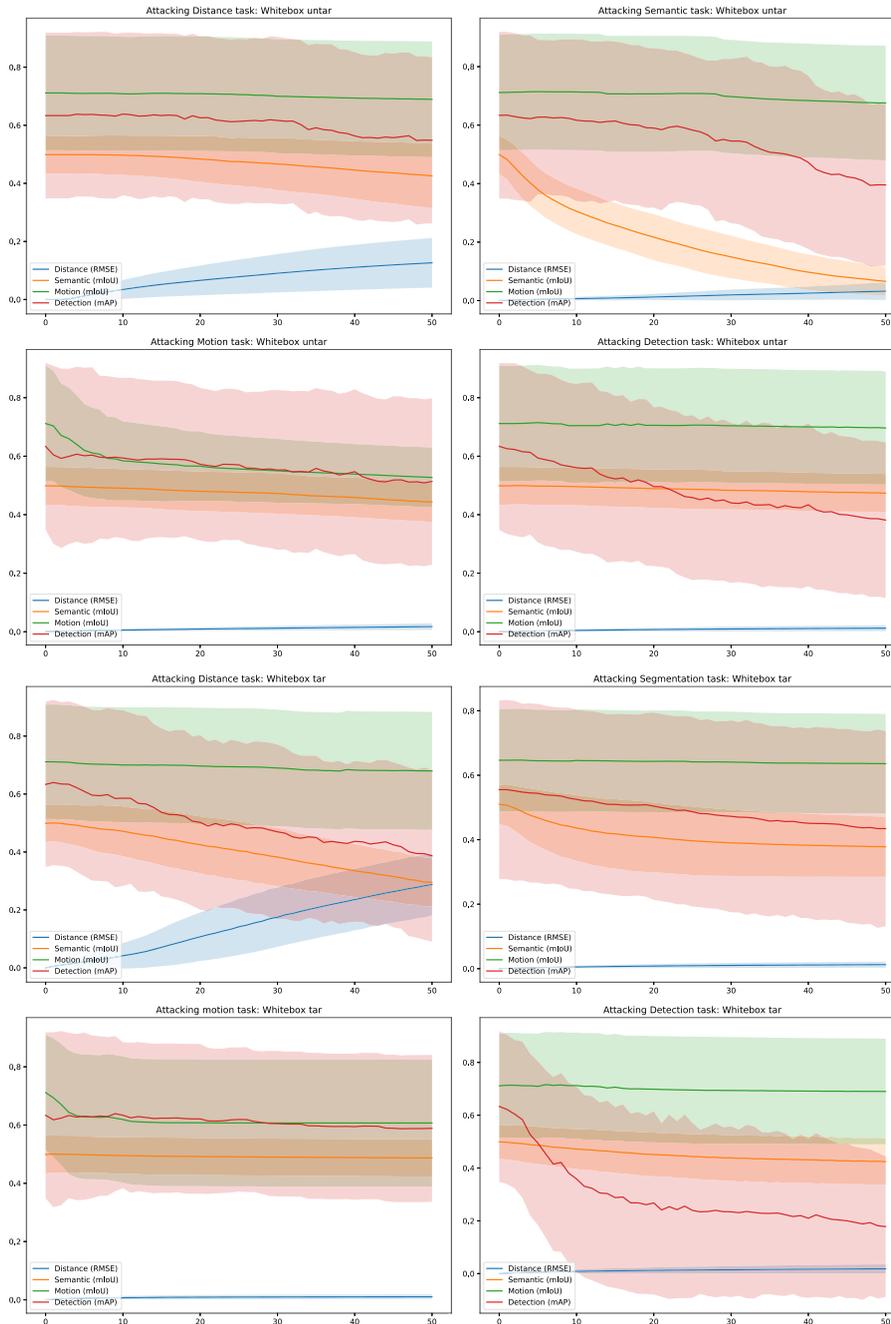


Figure 3. Performance comparison of the White box attacks across different tasks. The first row shows the un-targeted attacks, the second row shows the targeted attacks, and columns represent the tasks.

and pose estimation. The total loss consists of a photometric term  $\mathcal{L}_r$ , a smoothness term  $\mathcal{L}_s$ , that enforces edge-aware smoothness within the distance map  $\hat{D}_t$ , a cross-sequence distance consistency loss  $\mathcal{L}_{dc}$ , and feature-metric losses from [33] where  $\mathcal{L}_{dis}$  and  $\mathcal{L}_{cvt}$  are computed on  $I_t$ . Final loss function for distance estimation is weighted average of all these losses. The segmentation task contains seven classes on the WoodScape and employs *Lovasz-Softmax* [34] loss. Motion segmentation employs two frames and predicts either a binary moving or static mask and employs

*Lovasz-Softmax* [34], and *Focal* [35] loss for managing class imbalance instead of the cross-entropy loss. For object detection, we make use of YOLOv3 loss and add IoU loss using segmentation mask [36, 37].

## 2.2 Experimental Setup for Attacks

In this section, the details of the experiments are described. We conduct the experiments across the four visual perception tasks, on a test set of 100 images i.e. randomly sampled from the original test set of the target network. We generate the

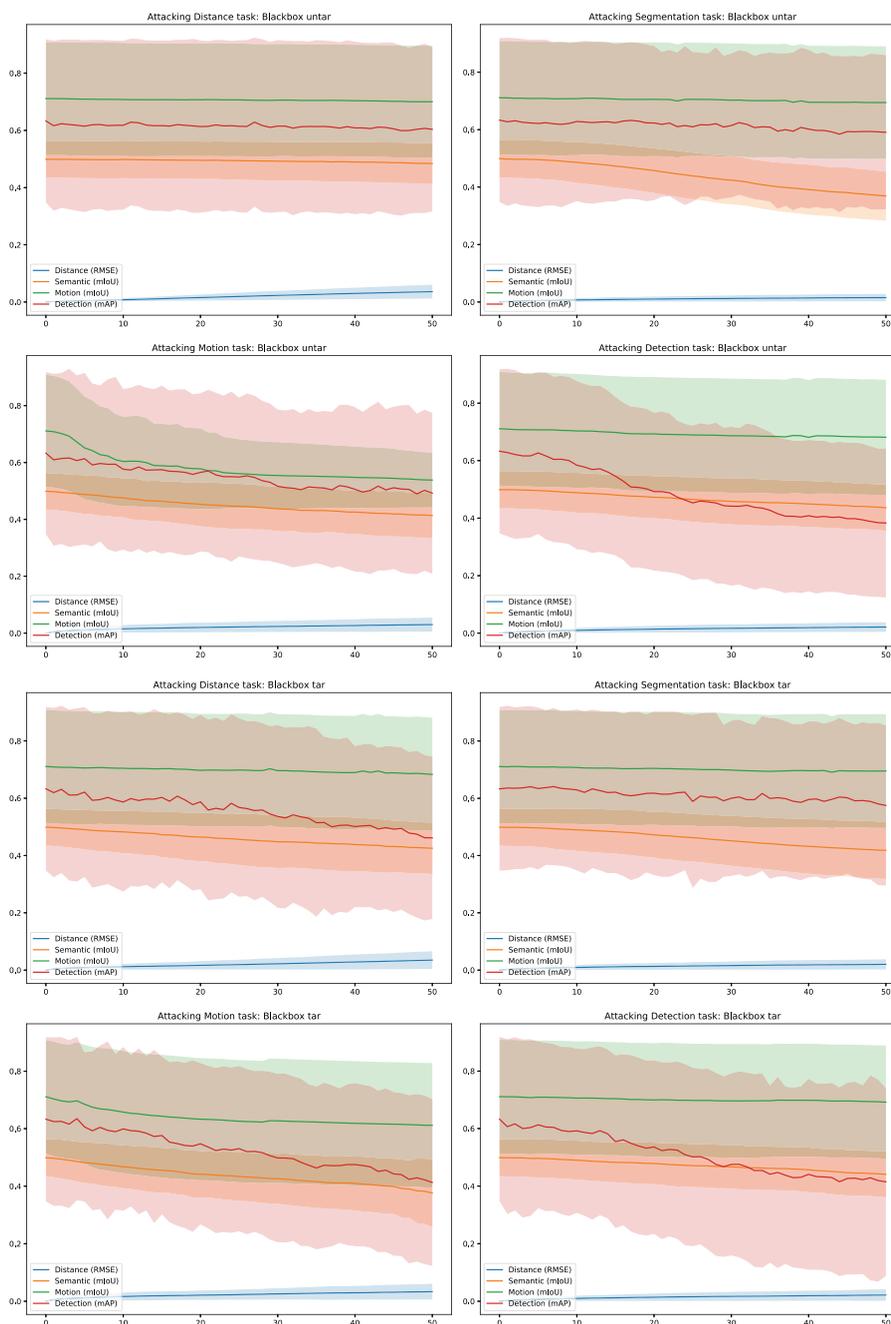


Figure 4. Performance comparison of the Black box attacks across different tasks. The first row shows the un-targeted attacks, the second row shows the targeted attacks, and columns represent the tasks.

Adversarial examples for each image in the test set while attacking one task at a time.

We make use of the standard FGSM method [9] for the white box attack. Using the gradients of the network, we perform an *iterative optimization process* to add perturbation in the input image in a direction to harm the original predictions. For Black box attacks we set up similar protocols established in the white-box; however, the gradients are not given but estimated. As a generic black-box optimization algorithm, we show that Evolution Strategies (ES) can be adopted as a black-box optimization method for generating

adversarial examples. Precisely, ES algorithm is used to update the adversarial example over the attacking steps. At each step, we take the adversarial example vector, i.e. the image, and generate a population of 25 slightly different vectors by adding noise sampled from a normal distribution. Then, we evaluate each of the individuals by feeding it through the network. Finally, the newly updated vector is the weighted sum of the population vectors. Each weight is proportional to the task's desired performance, and the process continues till convergence or stops criteria.

**Table I.** Summary of attacking and defending results across the test data where A and D columns are for attack and defense respectively.

Task		Distance		Segmentation		Motion		Detection	
		RMSE		mIoU		mIoU		mAP	
		A	D	A(%)	D(%)	A(%)	D(%)	A(%)	D(%)
Distance	wb_untarget	0.126	0.047	-14	-7.3	-3.1	-3.0	-13.4	-25.5
	wb_target	0.288	0.031	-40	-7.5	-4.4	-2.7	-38.9	-33.2
	bb_untarget	0.036	0.033	-3.0	-6.5	-1.5	-3.7	-4.6	-25.8
	bb_target	0.035	0.036	-14.8	-13.4	-3.9	-2.9	-27.1	-37.6
Segmentation	wb_untarget	0.032	0.028	-86.8	-14.2	-5.0	-3.5	-37.6	-30.1
	wb_target	0.017	0.027	-32.0	-5.5	-4.0	-2.6	-21.3	-27.5
	bb_untarget	0.015	0.031	-26.1	-11.4	-2.3	-4.9	-6.7	-27.3
	bb_target	0.020	0.034	-16.1	-9.2	-2.2	-2.5	9.2	-28.8
Motion	wb_untarget	0.018	0.027	-11.1	-7.3	-25.9	-9.2	-18.9	-23.1
	wb_target	0.010	0.027	-2.4	-6.0	-14.7	-9.2	-7.1	-30.0
	bb_untarget	0.030	0.039	-17.1	-15.8	-24.3	-17.6	-22.2	-38.6
	bb_target	0.033	0.040	-24.6	-22.5	-13.9	-11.7	-34.7	-47.9
Detection	wb_untarget	0.012	0.027	-5.1	-5.9	-2.1	-2.5	-39.8	-31.4
	wb_target	0.018	0.027	-15.0	-6.2	-3.0	-4.0	-71.9	-30.6
	bb_untarget	0.021	0.033	-12.5	-10.4	-4.2	-5.8	-39.4	-35.3
	bb_target	0.022	0.034	-11.6	-10.6	-2.7	-3.7	-34.4	-37.9

In the un-targeted case, the aim is to harm the predictions the most without considering a certain target prediction:  $f(x_{adv}) \neq y_{true}$ , however in the targeted case, the aim is to harm the predictions in a desired specific way towards a certain target:  $f(x_{adv}) = y_{target}$ . The attack loss is based on the task. Mean square error (MSE) is used for the distance task, while cross-entropy loss is used for motion and semantic segmentation tasks. For the object detection task, only object confidence is attacked, and hence cross-entropy loss is adopted. Regarding un-targeted attacks across all the tasks, the goal is to maximize the distance between the original output of the network and the adversarial example's output. Accordingly, we add the perturbations to achieve this simple goal where the output can be anything but the correct one. This can be formulated as  $\theta = \theta + \alpha dJ/d\theta$  where  $\theta$  is the image parameters i.e. pixels,  $J$  the loss functions and  $\alpha$  is the learning rate. However, for the targeted attacks, the target output is defined. The aim is to minimize the distance between the original output and the target output according to  $\theta = \theta - \alpha dJ/d\theta$ .

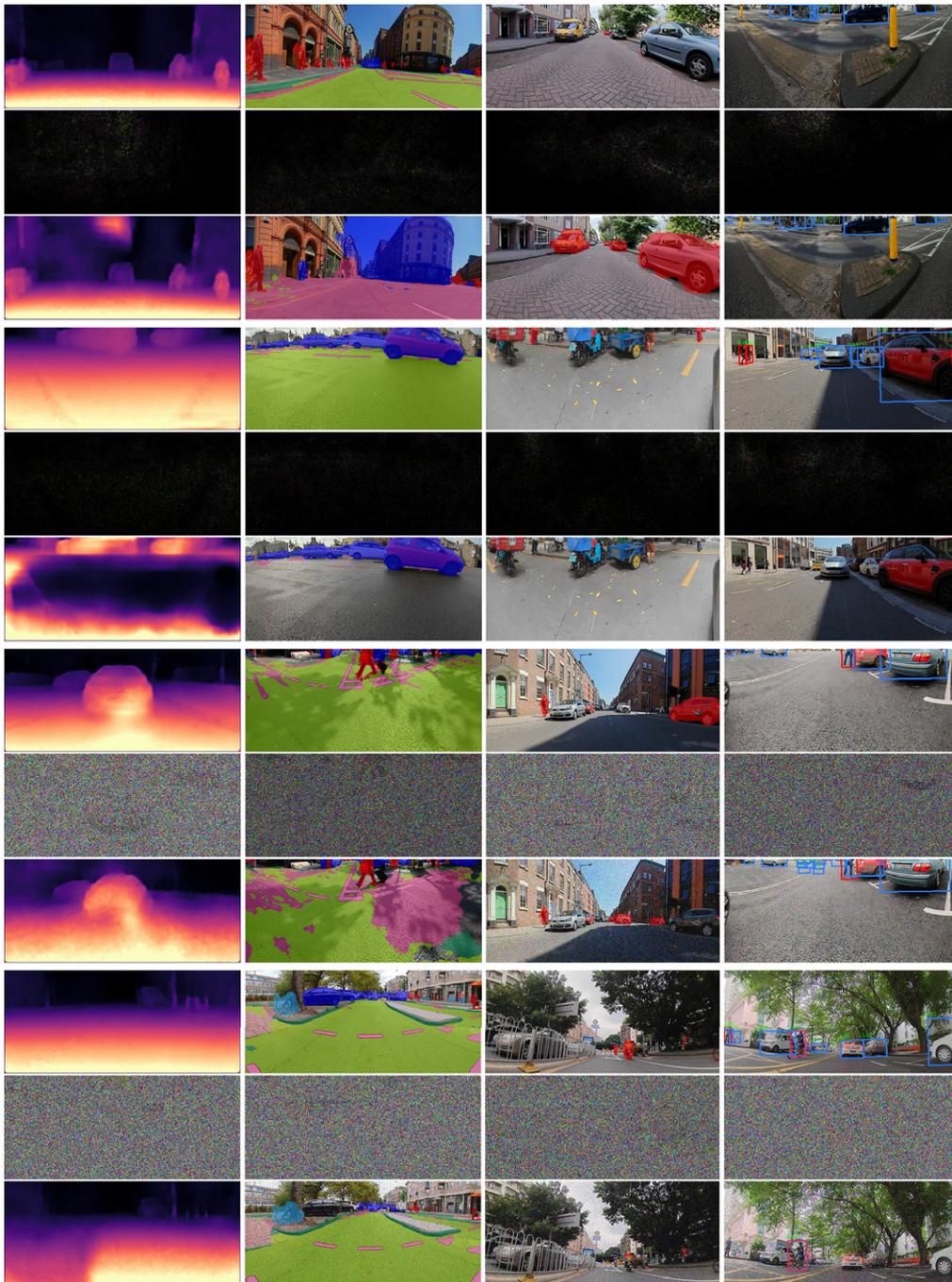
For each perception task, the targets are as follows: *Targeted Depth* attack tries to convert the predicted near pixels to be predicted as far. The *Targeted Segmentation* attack tries to convert the predicted vehicle pixels as void for randomly 50% of the test set. For the other 50%, tries to convert the predicted road pixels as void. The *Targeted Motion* attack tries to convert the predicted dynamic object pixels to be predicted as static. Finally, similar to semantic segmentation, the *Targeted Object Detection* attack tries to increase or decrease the predicted confidence

randomly. In addition to attacks, we apply a simple blurring defense approach across all the attacks. Similar to [16], the intuition is trying to remove the adversarial perturbations and restore the original output as much as possible. The hyperparameters of the attacks are empirically defined based on a very small validation set of three samples. All white-box attacks are conducted with learning rate  $\alpha = 0.00015$ . In black-box attacks, the hyperparameters are chosen to balance the attack effect and the severity of the perturbations, where the learning rates range from 0.0001 to 0.001, and  $\mu = 0$ ,  $\sigma = 0.05$  for ES population generation.

### 3. RESULTS

In this section, we present and discuss the details of the results. As expected, white-box attacks, with the gradients are accessible, were easier to find than the black box case. White box attacks can generate adversarial examples with minimal and localized perturbations across all the tasks. On the other hand, ES black-box attacks have more significant perturbations and require more hyperparameters to optimize.

The attacking curves for white and black box attacks are shown in Figures 3 and 4 respectively. Each plot shows each perception task's performance over the 50 attacking steps where the first step at index 0 represents the actual performance of the target network without applying any attack. Each curve shows the mean performance of a task over the test set, where the shaded area is the mean  $\pm$  standard deviation. Generally, motion and detection tasks have a performance with a large standard deviation



**Figure 5.** From top to bottom: White box Un-targeted, White box Targeted, Black box Un-targeted, & Black box Targeted Attacks. Within each group from top to bottom: Original results, adversarial perturbations, & the impacted results.

indicating the test set's diversity containing easy and hard examples. Across all curves, it is clear that the performance is decreasing along with the attacking steps.

Moreover, attacking one task by generating an adversarial example affects the other tasks' performance in different ways along the attacking curve. These curves enable the adversary to decide at which step the adversarial example is generated according to the required effect on the target task and the other tasks. As shown in Figs. 3 and 4, in most cases, attacking other tasks has a marginal

negative effect on motion task. The main reason is that the motion task takes two frames as input, and only one of them is attacked. Moreover, it is shown that the attacking distance task affects both segmentation and detection tasks. Attacking segmentation or detection showed to affect other tasks. As mentioned, the attack effect depends on the parameters selected for the attack. Moreover, targeted attacks try to optimize the adversarial example to produce the required target prediction. In contrast, the un-targeted attack

**Table II.** Input blurring effect on the tasks.

Task	Metric	Original	Blurred	Effect (%)
Distance	RMSE	0.0	0.026	NA
Segmentation	mIoU	0.499	0.477	-4.4
Motion	mIoU	0.711	0.693	-2.6
Detection	mAP	0.633	0.416	-34.3

continues to apply perturbation to produce as different as possible predictions.

To understand the effect of applying a defense method on the attacks, Gaussian blurring with radius = 1 is applied to the final adversarial examples and then fed into the target network, and performance is reported. As shown in Table I, this simple defense method has a positive effect for both segmentation and motion tasks in most cases compared to depth and detection tasks. Furthermore, the effect of blurring on the network's performance is inspected without applying any attacks, as shown in Table II. Both detection and distance tasks are affected the most. This explains why this defense method is more effective for segmentation and motion tasks. Figure 5 shows different visual samples of the attacks organized into four groups. Each group has three images: the original output, the adversarial perturbations magnified to 10X, and the impacted results are overlaid on the adversarial examples. As expected, perturbations for the white box attacks are much more minor and more localized than the black box case. Moreover, for the un-targeted attacks, the performance is harmed without having a specific goal leading to arbitrary predictions. On the other hand, for targeted attacks, vehicles or roads are removed for the semantic segmentation task. We add false objects or remove true objects for the detection task. Near pixels are converted as far for the distance task. Finally, we convert dynamic objects to static for the motion task.

#### 4. CONCLUSION

In this work, various adversarial attacks are applied to a multi-task target network with shared encoder and different decoders for autonomous driving visual perception. For each perception task, white and black box attacks are conducted for targeted and un-targeted scenarios. Moreover, attacking curves show the interactions between the attacks on different tasks. It is shown how attacking a task has an effect not only on that task but also on the rest. Moreover, by applying blurring on the adversarial examples as a defense method, it is found to have a positive effect on segmentation and motion tasks in contrast to object detection and distance tasks for the considered target network. In the future, we plan to conduct physical attacks, test other sensors such as LiDAR, and attacking multiple tasks jointly. It is obvious that, attacks and defenses are still challenging tasks and an active area of research, especially for autonomous driving applications with multi-task deep networks.

#### REFERENCES

- WHO, "Global status report on road safety 2018", *World Health Organization* (Accessed October-2020).
- A. Dahal, E. Golab, R. Garlapati, V. R. Kumar, and S. Yogamani, "Road-EdgeNet: Road edge detection system using surround view camera images," *Proc. IS&T Electronic Imaging: Autonomous Vehicles and Machines 2021* (IS&T, Springfield, VA, 2021), pp. 210-1-210-8.
- A. Dahal, V. R. Kumar, S. Yogamani, and C. Eising, "An online learning system for wireless charging alignment using surround-view fisheye cameras," Preprint arXiv:2105.12763 (2021).
- A. Das, P. Křížek, G. Sistu, F. Bürger, S. Madasamy, M. Uricar, V. R. Kumar, and S. Yogamani, "TiledSoilingNet: tile-level soiling detection on automotive surround-view cameras using coverage metric," *Proc. ITSC* (IEEE, Piscataway, NJ, 2020), pp. 1-6.
- M. M. Dhananjaya, V. R. Kumar, and S. Yogamani, "Weather and light level classification for autonomous driving: dataset, baseline and active learning," *Proc. ITSC* (IEEE, Piscataway, NJ, 2021).
- L. Gallagher, V. R. Kumar, S. Yogamani, and J. B. McDonald, "A hybrid sparse-dense monocular slam system for autonomous driving," *Proc. ECVPR* (IEEE, Piscataway, NJ, 2021), pp. 1-8.
- M. Uricar, G. Sistu, H. Rashed, A. Vobecky, V. R. Kumar, P. Krizek, F. Burger, and S. Yogamani, "Let's get dirty: GAN based data augmentation for camera lens soiling detection in autonomous driving," *Proc. WACV* (IEEE, Piscataway, NJ, 2021), pp. 766-775.
- M. Yahiaoui, H. Rashed, L. Mariotti, G. Sistu, I. Clancy, L. Yahiaoui, V. R. Kumar, and S. K. Yogamani, "FisheyeMODNet: Moving object detection on surround-view cameras for autonomous driving," *ICCV 2019 Workshop on 360° Perception and Interaction* (2019).
- I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *3rd Int'l. Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA* (2015).
- A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," *Int'l. Conf. on Machine Learning* (2018), pp. 284-293, PMLR.
- S. Thys, W. Van Ranst, and T. Goedemé, "Fooling automated surveillance cameras: adversarial patches to attack person detection," *Proc. CVPR Workshops* (IEEE, Piscataway, NJ, 2019), pp. 49-55.
- C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," *Proc. ICCV* (IEEE, Piscataway, NJ, 2017), pp. 1369-1378.
- Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li, "Adversarial objects against LiDAR-based autonomous driving systems," *CoRR* abs/1907.05418, Preprint arXiv:1907.05418 (2019).
- K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," *Proc. CVPR* (IEEE, Piscataway, NJ, 2018), pp. 1625-1634.
- S. Houben, S. Abrecht, M. Akila, A. Bär, F. Brockherde, P. Feifel, T. Fingscheidt, S. S. Gannamaneni, S. E. Ghobadi, A. Hammam, A. Haselhoff, F. Hauser, C. Heinemann, M. Hoffmann, N. Kapoor, F. Kappel, M. Klingner, J. Kronenberger, F. Küppers, J. Löhdefink, M. Mlynarski, M. Mock, F. Mualla, S. Pavlitskaya, M. Poretschkin, A. Pohl, V. Ravi-Kumar, J. Rosenzweig, M. Rottmann, S. Rüping, T. Sämann, J. D. Schneider, E. Schulz, G. Schwalbe, J. Sicking, T. Srivastava, S. Varghese, M. Weber, S. Wirtz, T. Wirtz, and M. Woehle, "Inspect, understand, overcome: A survey of practical methods for ai safety," Preprint arXiv:2104.14235 (2021).
- G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A study of the effect of jpg compression on adversarial images," Preprint arXiv:1608.00853 (2016).
- W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA* (The Internet Society, 2018).
- P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: protecting classifiers against adversarial attacks using generative models," *6th Int'l. Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada* (OpenReview.net, 2018).

- <sup>19</sup> I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM* (ACM, New York, NY, USA, 2020), Vol. 63, pp. 139–144.
- <sup>20</sup> G. Sistu, I. Leang, and S. Yogamani, "Real-time joint object detection and semantic segmentation network for automated driving," *Proc. NeurIPS Workshop on Machine Learning for Autonomous Driving* (2019).
- <sup>21</sup> S. Chennupati, G. Sistu, S. Yogamani, and S. Rawashdeh, "Auxnet: auxiliary tasks enhanced semantic segmentation for automated driving," *Proc. 14th Int'l. Joint Conf. on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP: VISAPP, Prague, Czech Republic* (SciTePress, 2019), Vol. 5, pp. 645–652.
- <sup>22</sup> V. R. Kumar, M. Klingner, S. Yogamani, S. Milz, T. Fingscheidt, and P. Mader, "SynDistNet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving," *Proc. WACV* (IEEE, Piscataway, NJ, 2021), pp. 61–71.
- <sup>23</sup> I. Leang, G. Sistu, F. Bürger, A. Bursuc, and S. Yogamani, "Dynamic task weighting methods for multi-task networks in autonomous driving systems," *Proc. ITSC* (IEEE, Piscataway, NJ, 2020), pp. 1–8.
- <sup>24</sup> V. R. Kumar, M. Klingner, S. Yogamani, M. Bach, S. Milz, T. Fingscheidt, and P. Mäder, "SVDistNet: Self-supervised near-field distance estimation on surround view fisheye cameras," *Proc. T-ITS* (IEEE, Piscataway, NJ, 2021), Vol. abs/2104.04420.
- <sup>25</sup> C. Mao, A. Gupta, V. Nitin, B. Ray, S. Song, J. Yang, and C. Vondrick, "Multitask learning strengthens adversarial robustness," *ECCV 2020* (Springer International Publishing, Berlin, New York, 2020), pp. 158–174.
- <sup>26</sup> V. R. Kumar, S. Yogamani, H. Rashed, G. Sitsu, C. Witt, I. Leang, S. Milz, and P. Mäder, "Omnidet: Surround view cameras based multi-task visual perception network for autonomous driving," *Proc. ICRA + RA-L* (IEEE, Piscataway, NJ, 2021), Vol. 6, pp. 2830–2837.
- <sup>27</sup> V. R. Kumar, S. A. Hiremath, M. Bach, S. Milz, C. Witt, C. Pinard, S. Yogamani, and P. Mder, "Fisheyedistancenet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving," *Proc. ICRA* (IEEE, Piscataway, NJ, 2020).
- <sup>28</sup> V. R. Kumar, S. Yogamani, M. Bach, C. Witt, S. Milz, and P. Mäder, "UnRectDepthNet: Self-supervised monocular depth estimation using a generic framework for handling common camera distortion models," *Proc. IROS* (IEEE, Piscataway, NJ, 2020), pp. 8177–8183.
- <sup>29</sup> V. R. Kumar, S. Yogamani, S. Milz, and P. Mäder, "FisheyeDistance-Net++: Self-supervised fisheye distance estimation with self-attention, robust loss function and camera view generalization," *Proc. IS&T Electronic Imaging: Autonomous Vehicles and Machine 2021* (IS&T, Springfield, VA, 2021), pp. 181-1–181-11.
- <sup>30</sup> V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse LiDAR data," *CVPR Workshop* (IEEE, Piscataway, NJ, 2018), Vol. 7.
- <sup>31</sup> V. R. Kumar, S. Milz, C. Witt, M. Simon, K. Amende, J. Petzold, S. Yogamani, and T. Pech, "Monocular fisheye camera depth estimation using sparse lidar supervision," *Proc. ITSC* (IEEE, Piscataway, NJ, 2018).
- <sup>32</sup> S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricr, S. Milz, M. Simon, M. Simon, K. Amende, and C. Witt, "Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving," *Proc. ICCV* (IEEE, Piscataway, NJ, 2019), pp. 9308–9318.
- <sup>33</sup> C. Shu, K. Yu, Z. Duan, and K. Yang, "Feature-metric loss for self-supervised learning of depth and egomotion," *Proc. ECCV* (Springer, Berlin, New York, 2020), pp. 572–588.
- <sup>34</sup> M. Berman, A. R. Triki, and M. B. Blaschko, "The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," *Proc. CVPR* (IEEE, Piscataway, NJ, 2018).
- <sup>35</sup> T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *Proc. CVPR* (IEEE, Piscataway, NJ, 2017), pp. 2980–2988.
- <sup>36</sup> H. Rashed, E. Mohamed, V. R. K. Sistu, and Ganesh, C. Eising, A. El-Sallab, and S. Yogamani, "FisheyeYOLO: object detection on fisheye cameras for autonomous driving," *Machine Learning for Autonomous Driving NeurIPS 2020 Virtual Workshop* (2020).
- <sup>37</sup> H. Rashed, E. Mohamed, G. Sistu, V. R. Kumar, C. Eising, A. El-Sallab, and S. Yogamani, "Generalized object detection on fisheye cameras for autonomous driving: Dataset, representations and baseline," *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision* (IEEE, Piscataway, NJ, 2021), pp. 2272–2280.