Counteracting Adversarial Attacks in Autonomous Driving

(Invited Talk)

Qi Sun
Chinese University of Hong Kong
qsun@cse.cuhk.edu.hk

Arjun Ashok Rao
Chinese University of Hong Kong
arjunrao@cse.cuhk.edu.hk

Xufeng Yao
Chinese University of Hong Kong
xfyao@cse.cuhk.edu.hk

Bei Yu
Chinese University of Hong Kong
byu@cse.cuhk.edu.hk

Shiyan Hu
University of Southampton
s.hu@soton.ac.uk

Abstract

In this paper, we focus on studying robust deep stereo vision of autonomous driving systems and counteracting adversarial attacks against it. Autonomous system operation requires real-time processing of measurement data which often contain significant uncertainties and noise. Adversarial attacks have been widely studied to simulate these perturbations in recent years. To counteract these attacks in autonomous systems, a novel defense method is proposed in this paper. A stereo-regularizer is proposed to guide the model to learn the implicit relationship between the left and right images of the stereo-vision system. Univariate and multivariate functions are adopted to characterize the relationships between the two input images and the object detection model. The regularizer is then relaxed to its upper bound to improve adversarial robustness. Furthermore, the upper bound is approximated by the remainder of its Taylor expansion to improve the local smoothness of the loss surface. The model parameters are trained via adversarial training with the novel regularization term. Our method exploits basic knowledge from the physical world, *i.e.*, the mutual constraints of the two images in the stereo-based system. As such, outliers can be detected and defended with high accuracy and efficiency. Numerical experiments demonstrate that the proposed method offers superior performance when compared with traditional adversarial training methods in state-of-the-art stereo-based 3D object detection models for autonomous vehicles.

Keywords

Robust Stereo Vision, Autonomous System, Adversarial Defense, Local Smoothness

1 Introduction

With the arrival of the artificial intelligence era, autonomous driving systems based on deep neural networks (DNN) have triggered a new revolution in traveling, and have a high potential to change the development of cities. An autonomous driving system needs to complete the following tasks: sensing, decision-making, planning, and control. Among these, sensing is considered as the most fundamental task and of vital importance. In recent years, vision and LiDAR-based 3D object detection systems which utilize deep neural networks have been widely used as the sensing systems [1].

Stereo-based 3D object detection is a vision-based system which fully exploits sparse, dense, semantic, and geometrical information in stereo imagery. Most of these models, *e.g.*, Faster R-CNN [2], utilize large feature networks as their backbone to extract features and use region proposal networks (RPNs) to generate object proposals which are then refined in subsequent modules to get the exact bounding boxes and class labels. With this rich information, we can get more accurate keypoints, viewpoints, object dimensions, and bounding boxes [3–6]. Usually, the left and the right images cooperate with each other in the stereo-vision system, as shown in Figure 1. 3D spatial knowledge is highly dependent on the left and right stereo-pair images. Contrary to stereo systems, monocular 3D object detection approaches suffer from the lack of accurate depth information, and as a result, cannot provide comparable performance [5]. In addition to vision-based systems, complex real environments make manufacturers adopt LiDAR-based systems

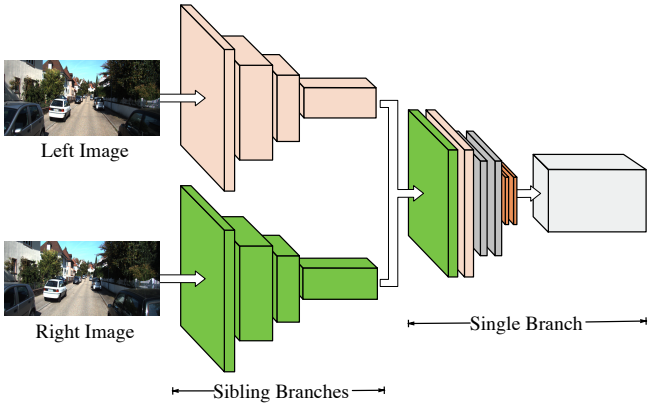

Figure 1: The structure of a typical stereo-based multi-task object detection model. There are two sibling branches, *e.g.*, RPN modules. Each branch takes left and right images as input respectively. The extracted object proposals are concatenated or reshaped into a single feature map for further processing, *e.g.*, regressing 3D boundary cube, and predicting viewpoints.

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11…$15.00
https://doi.org/10.1145/3400302.3415758

at the same time. LiDAR systems generate 3D point cloud data to model the 3D structures of scenes, either by projecting them into a bird's view or directly learning the 3D representations for classification and regression [7–10].

Although deep learning algorithms have demonstrated superior performance in many circumstances, it has been recently shown that these algorithms are vulnerable to perturbations. This security risk is especially dangerous for 3D object detection in autonomous driving. Consequently, the concept of adversarial attacks [11] came into being to measure these perturbations. Typically, adversarial perturbations are crafted to be imperceptible to human observers and indistinguishable from the original image. This is achieved by constraining the ℓ_p norm of the adversarial image to a pre-defined value that ensures human imperceptibility from a pixel-difference perspective. However, adversarial examples can cause large errors in the detection model when added to images. To date, several adversarial attack algorithms have been designed to attack DNN models [11–19]. [11] first demonstrated the existence of perturbations to natural images which can fool DNN models into misclassification. To generate adversarial images more efficiently, [12] proposed a novel method termed 'fast gradient sign method' (FGSM) to generate the perturbations by computing the gradient of the loss function. Intuitively, this means optimizing each input image pixel through its gradient to maximize the loss while model parameters are kept unchanged. FGSM utilizes the linearity hypothesis of DNN models, *i.e.*, designs of deep learning models encourage linear behavior for computational gains. The basic iterative method (BIM) [16] extended FGSM by iteratively take multiple small steps to adjust the perturbation direction. Projected gradient descent (PGD) [17] further studied the adversarial perturbations from the perspective of optimization. PGD initializes the search for an adversarial image at a random point within the perturbation range. The noisy initialization creates a stronger attack than previous methods. Attacking the object detection model is more challenging compared to attacking the classification model as it needs to mislead the multiple region proposals. [20] attacks detectors via expectation over transformation (EOT) technique – a method that computes the perturbation by adding random distortions (*e.g.*, resizing, rotation, *etc.*) to natural images. [18] attacks the shapes of bounding boxes and classification labels simultaneously. [19] and [21] focus on attacking more relevant objects by splitting the whole image into subregions, *e.g.*, foreground and background, or several superpixels. Adversarial examples also exist in the physical world. Some adversarial images and road signs are printed to fool deep vision models [16, 22]. Adversarial T-shirts can evade person detection systems, even with only a few adversarial patches on the clothing [23, 24]. [25] generates adversarial 3D objects via transformation-based methods.

Correspondingly, to improve robustness against attacks, certain adversarial defense algorithms have been proposed. Currently, defense methods develop along three directions: using modified training or modified inputs, modifying networks, and using external add-on networks [26]. A majority of the literature that introduced new adversarial attack methods simultaneously train the models with their attacked inputs [12, 16, 17] - a practice termed as adversarial training. Some modified inputs by conducting preprocessing operations, *e.g.*, random resizing [27] and data compression [28]. SafetyNet [29] proposed to append an SVM classifier to the models such that SVM can use the discrete codes computed by ReLUs. For an input image, its discrete codes are compared against the codes of training data to determine whether it is an adversarial image. Generative adversarial networks (GANs) [30, 31], composed of a generator and a discriminator, add two novel modules to help generate perturbations and discriminate adversarial inputs. Outside of these outstanding works, to the best of our knowledge, there has been no work done on defending against attacks on stereo-based 3D object detection models. Although we can directly impart reasonable robustness via brute-force adversarial training with adversarial images as inputs, this strategy ignores the physical characteristics of stereo vision.

In this paper, we propose a defense method based on adversarial training with a novel and physically meaningful regularization term. Stereo-based detection models normally utilize the implicit spatial information from the left and right images to regress proposals independently, *i.e.*, the sibling branches in Figure 1. Meanwhile, the concatenated features from these two images are further fused to learn model information, *i.e.*, the single branch in Figure 1. Considering these two types of mechanisms that can be modeled as univariate and multivariate functions, a novel stereo-based regularizer is proposed. The regularizer is further relaxed to its upper bounds which ease the optimization process. To maximize the local smoothness of the loss surface, the upper bound is further approximated by the remainders of Taylor expansions. With these features, our novel defense method can counteract adversarial attacks efficiently.

The rest of our paper is organized as follows. Section 2 introduces the problem to be addressed and preliminaries. Section 3 explains our proposed defense techniques in detail. Section 4 summarizes the overflow defense flow. Section 5 demonstrates the experiments and results, followed by conclusion in Section 6.

2 Preliminaries

2.1 Adversarial Training

Adversarial training can be traced back to the rise of adversarial attack algorithms. The typical form of most adversarial training algorithms involve training the target model on adversarial images generated via the attack method. Notably, most adversarial training methods perform the following min-max training strategy shown in Equation (1).

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\delta}} L(\boldsymbol{x}+\boldsymbol{\delta}, \boldsymbol{\theta}; \boldsymbol{y}), \\ &\text{s.t. } \|\boldsymbol{\delta}\|_p \leq \epsilon, \end{aligned} \tag{1}$$

where $\boldsymbol{\theta}$ represents the model parameters, $\boldsymbol{\delta}$ is the perturbation, $\boldsymbol{y}$ is the ground truth and $L(\boldsymbol{x}+\boldsymbol{\delta}, \boldsymbol{\theta}; \boldsymbol{y})$ is the loss function. $\|\cdot\|_p$ is the ℓ_p-norm, which constrains the perturbation within ϵ such that the perturbation is imperceptible to cameras and human eyes.

2.2 Stereo-based 3D Object Detection

Stereo-based 3D object detection [5, 6] has proved a success in object detection in autonomous driving systems. Stereo-based systems can detect and associate objects simultaneously using the left and right images through exploiting semantic and geometric information in stereo imagery. The network architecture can be briefly divided into two parts [5], as shown in Figure 1. The first module contains two sibling Stereo RPNs which extract features and generate object bounding proposals for the left and right images

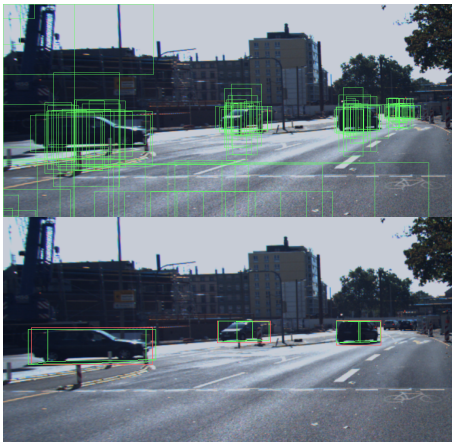

Figure 2: The generated object proposals and the final detected objects.

independently. Relying on the sibling features extracted in the first module, the subsequent module fuses the features and predicts the boundary cube, keypoint, and other related spatial information. The final detection results are jointly determined by the region proposals. An example is shown in Figure 2.

2.3 Problem Formulation

Denote $\boldsymbol{x}_l$ and $\boldsymbol{x}_r$ as the input left and right images respectively. The object bounding boxes in the left and right images are $\boldsymbol{b}_l$ and $\boldsymbol{b}_r$ respectively and the object class label is y. Given a stereo 3D object detection model with parameters $\boldsymbol{\theta}$ and loss function L, our task is to solve the following min-max problem:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\delta}_l, \boldsymbol{\delta}_r} & L(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r, \boldsymbol{\theta}; \boldsymbol{b}_l, \boldsymbol{b}_r, y), \\ \text{s.t. } & \|\boldsymbol{\delta}_l\|_p \leq \epsilon, \quad \|\boldsymbol{\delta}_r\|_p \leq \epsilon, \end{aligned} \tag{2}$$

where $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$ represent the perturbations on the left and right images. $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$ are both constrained within the manipulation budget ϵ. In the following sections, we use L_o to denote the above original $L(\cdot)$ loss function for brevity.

3 Defense Algorithm

As previously mentioned, the stereo-based 3D object detection model can handle various tasks. Different tasks can be modeled as different forms of functions. For example, the sibling RPN modules generate bounding boxes for the left and right images respectively (as shown in Figure 3). Therefore we can model this part as two independent univariate functions. The regularization term should constrain both of these two functions. Regressing the 3D bounding box or predicting the viewpoint can be represented as a multivariate function. The embedded features which are learned from the left-right stereo pair are jointly used as inputs to the multivariate function. Consequently, the regularization term should be able to handle multivariate functions. Both of the two regularization terms are optimized by relaxation and approximation, to improve local smoothness of the loss surface.

3.1 Stereo-based Regularizer

The two regressed bounding boxes from the left and right images share a high intersection over union (IoU). This phenomenon is

Figure 3: Bounding boxes regressed from the left and right images.

consistent with the pre-existing understanding that stereo cameras capture the same field of view from a rectified stereo-pair with a small level of disparity. However, the two resulting bounding boxes contain differences that are influenced by physical factors such as the distance between the car and the object, the object orientation with respect to the stereo camera, *etc.* These physical factors vary with environments, which make them expensive to be measured accurately. For simplicity, we compute the distance between the two bounding boxes to characterize the effects of the practical physical factors.

Let $f_l(\boldsymbol{x}_l)$ and $f_r(\boldsymbol{x}_r)$ represent two univariate functions, to represent the features extracted from the left image $\boldsymbol{x}_l$ and right image $\boldsymbol{x}_r$ respectively. Therefore, the distance between the bounding boxes predicted from the two images is defined as:

$$d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right) = \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r)\|_n. \tag{3}$$

As mentioned before, the physical characteristics are measured with $d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right)$. After attacking the images, the corresponding distance is computed as:

$$d\left(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r\right) = \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r)\|_n. \tag{4}$$

To improve the robustness of the detection system, we hope that these physical characteristics are well reserved. Therefore, the regularization term for the sibling branches is defined as:

$$L_b = \mid d\left(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r\right) - d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right) \mid, \tag{5}$$

where $|\cdot|$ computes the absolute value. With L_b and the original loss function L_o, the updated optimization objective function is $L = L_b + L_o$. Note that as shown in Equation (2), the regularization term is minimized with respect to $\boldsymbol{\theta}$. Minimizing Equation (5) would possibly result in inflexible optimization and ambiguous convergence status [32]. The straightforward hazard is that pushing $d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right)$ close to zero makes the model confuse the left and right images. So is for $d\left(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r\right)$. For example, $d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right) = 0$ would result in $f_l(\boldsymbol{x}_l) = f_r(\boldsymbol{x}_r)$. Although the original loss term L_o would alleviate this hazard as it computes the errors between the predicted bounding boxes and ground truths, L_b would no longer be a helper and would become a burden. This contradicts our initial intuition.

We add a margin $\boldsymbol{m}$ to reinforce the optimization of the distance functions [32, 33]. Take $d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right)$ as an example. $f_l(\boldsymbol{x}_l)$ and $f_r(\boldsymbol{x}_r)$ are in symmetric positions in $d\left(\boldsymbol{x}_l, \boldsymbol{x}_r\right)$. This means that adding a positive margin to $f_l(\boldsymbol{x}_l)$ is equivalent to adding a negative margin

to $f_r(\boldsymbol{x}_r)$. The margin-based distance function is shown in Equation (6).

$$\begin{aligned} d\,(\boldsymbol{x}_l, \boldsymbol{x}_r) &= \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}\|_n, \\ d\,(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r) &= \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m}\|_n. \end{aligned} \tag{6}$$

The same margin $\boldsymbol{m}$ is shared in the two distance metrics because we believe that the model should be able to recover the same results after been attacked.

The tasks which use the fused features learned from the early module can be modeled as multivariate functions. For example, the viewpoint prediction function can be represented as $f_m(\boldsymbol{x}_l, \boldsymbol{x}_r)$, and the resultant vector with perturbation becomes $f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r)$. We hope the model can get the same result after the images are attacked, therefore the regularization term L_m to be minimized is defined as:

$$L_m = \|f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_m(\boldsymbol{x}_l, \boldsymbol{x}_r)\|_n. \tag{7}$$

Different from Equation (6), we do not add a margin here because the features learned from the perturbated images should be equal to the original features. With L_m, the update optimization objective function is $L = L_o + L_b + L_m$.

3.2 Local Smoothness Optimization

Recent work has demonstrated that the robustness of models usually suffers from the non-linearity of loss surface and gradient obfuscation. Many methods have been proposed to improve the local smoothness [34–36]. Equation (6) and Equation (5) is transformed to a nested $\|\cdot\|$ formulation shown in Equation (8).

$$L_b = \| \, \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m}\|_n - \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}\|_n \, \|_1. \tag{8}$$

Equation (8) with nested norm parameters is challenging to be solved. Moreover, $\boldsymbol{m}$ is a hyper parameter that needs to be determined through adversarial training. Besides, the difference between two terms in $\|\cdot\|$ is at a high magnitude, while the loss surface usually has a low magnitude. Inspired by recent work which approximates the regularization term by the remainder of its Taylor expansion [34, 35], we propose to relax Equation (8) as Equation (9). The detailed relaxation process is attached in Appendix A.

$$\begin{aligned} L_b &= \| \, \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m}\|_n - \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}\|_n \, \|_1 \\ &\le \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_l) \, \|_n + \| \, f_l(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_r(\boldsymbol{x}_r) \, \|_n \\ &\le \| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n + \gamma_l(\epsilon, \boldsymbol{x}_l) + \| \, \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r) \, \|_n + \gamma_r(\epsilon, \boldsymbol{x}_r), \end{aligned} \tag{9}$$

where $\boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l)$ is the first-order term in the Taylor expansion of $f_l(\boldsymbol{x}_l)$, and $\boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r)$ is the first-order term in the Taylor expansion of $f_r(\boldsymbol{x}_r)$. $\gamma_l(\epsilon, \boldsymbol{x}_l)$ and $\gamma_r(\epsilon, \boldsymbol{x}_r)$ are the maximums of the high-order remainders of the Taylor expansions. According to the inner maximization operation in Equation (2), they are defined as:

$$\begin{aligned} h_l(\epsilon, \boldsymbol{x}_l) &= \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) - \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n, \\ h_r(\epsilon, \boldsymbol{x}_r) &= \| \, f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_r(\boldsymbol{x}_r) - \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r) \, \|_n, \\ \gamma_l(\epsilon, \boldsymbol{x}_l) &= \max_{\|\boldsymbol{\delta}_l\|_p \le \epsilon} h_l(\epsilon, \boldsymbol{x}_l), \quad \gamma_r(\epsilon, \boldsymbol{x}_r) = \max_{\|\boldsymbol{\delta}_r\|_p \le \epsilon} h_r(\epsilon, \boldsymbol{x}_r), \end{aligned} \tag{10}$$

where h_l and h_r represent the high-order remainders for the left and right images respectively.

With Equation (9), we can not only erase $\boldsymbol{m}$, but also relax Equation (8) to its upper bound. Considering the trade-off between computational workload and model accuracy, the higher order remainders, *e.g.*, the 2-nd gradient is not computed. The insights behind Equation (9) is straightforward: the difference between $f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l)$ and $f_l(\boldsymbol{x}_l)$ is constrained by the first-order gradient term and the high-order remainder of the Taylor expansion of $f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l)$. γ_l and γ_r are good measures of how linear the surfaces are within the perturbation range ϵ. This kind of quality is called *local smoothness measure*. By minimizing the smoothness term, we will maximize the smoothness of the loss surface and therefore improve the model robustness.

As to the classification regularizer L_m, it follows a similar relaxation strategy. $f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r)$ is approximated by:

$$f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r) \approx f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) + \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) + \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r). \tag{11}$$

Thus we can form the following bound:

$$\begin{aligned} L_m &= \| \, f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) \, \|_n \\ &\le \| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) + \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) \, \|_n + \gamma_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r), \end{aligned} \tag{12}$$

where $\gamma_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r)$ is the maximum of the high-order remainder $h_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r)$. They are defined as follows:

$$\begin{aligned} h_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r) &= \| \, f_m(\boldsymbol{x}_l + \boldsymbol{\delta}_l, \boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) \\ &\qquad - \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) - \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) \, \|_n, \\ \gamma_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r) &= \max_{\|\boldsymbol{\delta}_l\|_p \le \epsilon, \|\boldsymbol{\delta}_r\|_p \le \epsilon.} h_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r). \end{aligned} \tag{13}$$

Combining Equation (10) and Equation (13) together, we define the regularization term for high-order remainder as L_h, as shown in Equation (14).

$$L_h = h_l(\epsilon, \boldsymbol{x}_l) + h_r(\epsilon, \boldsymbol{x}_r) + h_m(\epsilon, \boldsymbol{x}_l, \boldsymbol{x}_r). \tag{14}$$

Similarly, we combine all of the first-order gradient term together, and then we have the regularization term L_∇ defined as follows:

$$\begin{aligned} L_\nabla &= \| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n + \| \, \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r) \, \|_n \\ &\quad + \| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) + \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_m(\boldsymbol{x}_l, \boldsymbol{x}_r) \, \|_n \end{aligned} \tag{15}$$

The overall stereo-based regularizer is $L_h + L_\nabla$. Together with the original loss function L_o in Equation (2), we can derive the following min-max problem formulation:

$$\begin{aligned} &\min_{\boldsymbol{\theta}} L_a = L_o + L_\nabla + [\max_{\boldsymbol{\delta}_l, \, \boldsymbol{\delta}_r} L_h] \\ &\text{s.t. } \|\boldsymbol{\delta}_l\|_p \le \epsilon, \quad \|\boldsymbol{\delta}_r\|_p \le \epsilon, \end{aligned} \tag{16}$$

where L_a is defined as the summation of the training error together with the regularization terms.

4 Overall Flow

In the previous section, we discuss the stereo-based regularizer in detail. Afterward, local smoothness is considered and the originally proposed regularizer is relaxed to obtain the local smoothness. An adversarial training strategy is adopted in this paper.

We iteratively optimize perturbations $\boldsymbol{\delta}_l$, $\boldsymbol{\delta}_r$, and model parameters $\boldsymbol{\theta}$. The pseudo-code of the overall optimization training flow is shown in Algorithm 1. The advantages of using ℓ_1-norm over ℓ_2-norm in terms of robustness analysis procedures are largely recognized across the scientific literature [37]. To improve model

Table 1: Statistical Results of Adversarial Attacks

Model	AP_{2d} (%)			AOS (%)			AP_{3d} (%)			AP_{bv} (%)		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
No Attack [5]	99.28	91.09	78.62	98.42	89.43	76.94	54.10	34.44	28.15	68.24	46.84	39.34
FGSM, $\epsilon = 0.7$	88.29	76.45	62.39	87.54	74.11	60.36	40.52	32.94	27.56	15.52	12.19	10.05
FGSM, $\epsilon = 2$	76.82	60.49	49.67	74.73	57.84	47.35	26.21	21.35	16.81	13.64	7.7	6.14
PGD, $\epsilon = 0.7$	69.55	58.94	48.04	66.72	56.04	45.59	22.52	18.88	15.32	7.02	5.53	4.29
PGD, $\epsilon = 2$	53.01	43.11	34.21	51.48	40.23	31.80	9.60	7.61	6.23	3.82	2.22	1.95

Table 2: Statistical Results of Adversarial Defenses

Testing Images	Defense Method	AP_{2d} (%)			AOS (%)			AP_{3d} (%)			AP_{bv} (%)		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
FGSM, $\epsilon = 0.7$	Direct + FGSM	87.58	81.54	71.53	87.25	80.11	62.42	41.95	30.62	**28.89**	21.57	19.62	16.56
	SmoothStereo	**88.38**	**82.74**	**73.94**	**88.89**	**81.87**	**63.63**	**45.51**	**31.01**	26.61	**24.50**	**20.88**	**18.26**
FGSM, $\epsilon = 2$	Direct + FGSM	84.73	70.82	57.90	**84.13**	69.19	55.61	40.15	30.57	**24.42**	16.21	13.03	10.54
	SmoothStereo	**85.95**	**72.64**	**61.22**	81.65	**74.83**	**60.00**	**41.43**	**31.63**	23.79	**18.25**	**14.76**	**12.53**
PGD, $\epsilon = 0.7$	Direct + PGD	73.37	**61.82**	56.66	73.04	60.46	50.04	**27.47**	20.08	**18.74**	**13.77**	7.10	9.30
	SmoothStereo	**75.67**	61.58	**59.73**	**73.43**	**62.27**	**52.82**	24.88	**20.90**	16.99	12.44	**11.73**	**9.46**
PGD, $\epsilon = 2$	Direct + PGD	54.46	49.11	40.44	53.37	46.23	38.07	14.39	10.38	9.32	5.84	**4.65**	3.29
	SmoothStereo	**55.29**	**49.38**	**41.92**	**53.47**	**47.27**	**40.60**	**18.11**	**12.42**	**9.43**	**6.82**	4.52	**3.94**

Algorithm 1 Adversarial Training of Stereo-based Object Detection Model

Input: Training set $\{(\boldsymbol{x}_l^1, \boldsymbol{x}_r^1, \boldsymbol{b}_l^1, \boldsymbol{b}_r^2, y^1), \ldots, (\boldsymbol{x}_l^N, \boldsymbol{x}_r^N, \boldsymbol{b}_l^N, \boldsymbol{b}_r^N, y^N)\}$, batch size B, # of iterations for outer optimization I_o, # of iterations for inner optimization I_i, model parameters $\boldsymbol{\theta}$, learning rate η, perturbation range ϵ.

for $i = 1 \rightarrow I_o$ **do**
  Sample a batch B from the training set;
  Generate Initial $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$ for B, in perturbation range ϵ;
  for $j = 1 \rightarrow I_i$ **do**
    Calculate L_h in Equation (14) for the batch B;
    Update $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$ via back-propagation with L_h as the loss function;
  end for
  Compute L_∇ with $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$, according to Equation (15);
  Compute $L_a = L_o + L_\nabla + L_h$ with $\boldsymbol{\delta}_l$ and $\boldsymbol{\delta}_r$;
  Update $\boldsymbol{\theta}$ via back-propagation with L_a as the loss function;
end for

robustness, ℓ_1-norm is used as the norm in Equation (6) and Equation (7).

5 Experimental Results

In this section, we evaluate our defense method on the challenging KITTI object detection benchmark [38]. KITTI set is divided into three categories: Easy, Moderate, and Hard, which reflect the difficulties of the object detection tasks. The state-of-the-art Stereo-based 3D object detection model from [5] is used as the target detection model. Two popular and powerful attack methods are implemented to attack the detection model, *i.e.*, FGSM [12] and PGD [17]. Direct adversarial training, proposed in [17] is used to defend against the adversarial attacks, and results are compared with our novel defense method. For brevity, our method is shorted as SmoothStereo.

The result statistics are listed in Table 1. AP_{2d} represents the average detection precision of the 2D bounding box. AOS represents the average orientation similarity of the joint 3D detection [38]. AP_{3d} represents the average detection precision of the 3D bounding box. AP_{bv} represents the average localization precision of bird's eye view. The error statistics are computed according to boxes with IoU ≥ 0.7. Note that in real environments, the perturbations are usually not overly abnormal. Greater perturbation ranges lead to stronger attacks. For balance, in our experiments, we take two perturbation values as examples, *i.e.*, $\epsilon = 0.7$ and $\epsilon = 2$. In each experiment, the left and right images share the same perturbation range ϵ. The optimization iteration of PGD images is 2. It is evident from Table 1 that PGD produces much lower accuracy rates, and is hence a much stronger adversary compared to FGSM. This phenomenon is consistent with people's experience. Even with a moderate $\epsilon = 2$, PGD can degrade the model performance by nearly half.

5.1 Defense against FGSM Attacks

Figure 4 shows an example of the FGSM attack and the results of different defense methods. The adversarial image misleads the model to misclassify one car and incorrectly predict the object orientation. Direct adversarial training still loses that car, while misclassifying the granite steps as a car. In comparison, our method can correctly predict the locations and directions of the cars. Moreover, our regularization and smoothness terms are also able to outperform natural detections in some cases. This is shown in Figure 4 where our robust model correctly detects a car which was previously misclassified on the unperturbed model. This proves the

Figure 4: Examples of results on FGSM attacks. The images from left to right are: original detection results (ground-truth), adversarial images generated via FGSM with $\epsilon = 2$, defense results via direct adversarial training, and defense results via our `SmoothStereo`.

Figure 5: Example of results on PGD attacks. The images from left to right are: original detection results (ground-truth), adversarial images generated via PGD with $\epsilon = 2$, defense results via direct adversarial training, and defense results via our `SmoothStereo`.

local smoothness of our method. The statistical results are listed in Table 2.

5.2 Defense against PGD Attacks

Figure 4 shows an example of the PGD attack and the results of defense methods. The original model loses the car in adversarial images. Intuitively, this can be considered a misclassification and the model incorrectly perceives class 'Car' as class 'Background'. Direct adversarial training [17] can correct the model and predict the car successfully. In comparison, our `SmoothStereo` method not only predicts the car, but also finds the nearest object which hinders the car. The results prove that our method can also improve robustness of the model while improving the local smoothness. The statistical results are listed in Table 2.

In summary, the results show that our method can efficiently improve local smoothness of the detection model and improve prediction results. It is also shown that out novel regularization, which considers local smoothness and stereo information, can significantly boost detection performance of the original model as well.

6 Conclusion

To counteract adversarial attacks and improve the robustness of object detection models for autonomous driving systems, a novel defense method which specifically considers the physical meaning of the Stereo-based 3D object detection model is proposed in this paper. Our regularizer can help the model learn the relative relationship of the bounding boxes between the left and right images, which can be modeled as two univariate functions. The regularizer is also capable of handling the branch which is modeled as a multivariate function. These regularizers are further relaxed to their upper bounds and approximated by first-order remainders of Taylor expansions. With this relaxation and approximation, we can maximize the local smoothness of the loss surface to improve the robustness. It is also shown in the results that our novel regularization considering local smoothness and stereo information can boost the detection performance of the original model as well.

7 Acknowledgment

This work is partially supported by Tencent Technology, SmartMore, and The Research Grants Council of Hong Kong SAR (Project No. CUHK14209420),

References

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[3] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals using stereo imagery for accurate object class detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1259–1272, 2017.
[4] P. Li, T. Qin *et al.*, "Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 646–661.
[5] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7644–7652.
[6] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 536–12 545.
[7] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099–5108.
[8] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1907–1915.
[9] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 3569–3577.
[10] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *International Conference on Learning Representations (ICLR)*, 2014.
[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Conference on Learning Representations (ICLR)*, 2015.
[13] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.
[14] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.
[15] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1369–1378.
[16] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.
[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learning Representations (ICLR)*, 2018.
[18] Y. Li, D. Tian, X. Bian, S. Lyu *et al.*, "Robust adversarial perturbation on deep proposal-based models," *British Machine Vision Conference (BMVC)*, 2018.
[19] Y. Li, X. Bian, M. Chang, and S. Lyu, "Exploring the vulnerability of single shot module in object detectors via imperceptible background patches," in *British Machine Vision Conference (BMVC)*, 2019.
[20] S.-T. Chen, C. Cornelius, J. Martin, and D. H. P. Chau, "Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 52–68.
[21] X. Dong, J. Han, D. Chen, J. Liu, H. Bian, Z. Ma, H. Li, X. Wang, W. Zhang, and N. Yu, "Robust superpixel-guided attentional adversarial attack," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 895–12 904.
[22] 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," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1625–1634.
[23] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," *arXiv*, pp. arXiv–1910, 2019.
[24] Z. Wu, S.-N. Lim, L. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," *European Conference on Computer Vision (ECCV)*, 2020.
[25] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *International conference on machine learning*, 2018, pp. 284–293.
[26] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.
[27] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *International Conference on Learning Representations (ICLR)*, 2018.
[28] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *arXiv preprint arXiv:1705.02900*, 2017.
[29] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," in *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.
[31] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1222–1230.
[32] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.
[33] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks." in *ICML*, vol. 2, no. 3, 2016, p. 7.
[34] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, "Adversarial robustness through local linearization," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 847–13 856.
[35] J. Xu, Y. Li, Y. Bai, Y. Jiang, and S.-T. Xia, "Adversarial defense via local flatness regularization," *arXiv preprint arXiv:1910.12165*, 2019.
[36] B. Yu, J. Wu, J. Ma, and Z. Zhu, "Tangent-normal adversarial regularization for semi-supervised learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 676–10 684.
[37] S. A. Flores, "Robustness of ℓ_1-norm estimation: From folklore to fact," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1640–1644, 2018.
[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3354–3361.
[39] R. Kress, *Numerical Analysis*, ser. Graduate Texts in Mathematics. Springer New York, 1998. [Online]. Available: https://books.google.com.hk/books?id=e7ZmHRIxum0C

A Relaxation of Equation (8)

According to the triangle inequality:

$$||a| + |b|| \leq |a \pm b| \leq |a| + |b|, \tag{17}$$

which is one of the defining property of the normed vector space [39], Equation (8) can be relaxed to a upper bound:

$$\begin{aligned} L_b &= \| \, \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m}\|_n - \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}\|_n \, \|_1 \\ &\leq \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m} - (f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}) \, \|_n \\ &= \| \, (f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l)) - (f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_r(\boldsymbol{x}_r)) \, \|_n \\ &\leq \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) \, \|_n + \| \, f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_r(\boldsymbol{x}_r) \, \|_n . \end{aligned} \tag{18}$$

The left and right images are in the symmetric positions in Equation (18), *i.e.*, $f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r)$ leads to the same deduced results with $f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l)$. Further, $f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l)$ can be approximated by its first-order Taylor expansion $f_l(\boldsymbol{x}_l) + \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l)$. Thus we can have the following bound:

$$\begin{aligned} &\| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) \, \|_n \\ \approx &\| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) + f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) - \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n \\ \leq &\| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n + \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) - \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n \\ \leq &\| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n + \gamma_l(\boldsymbol{x}_l, \epsilon), \end{aligned} \tag{19}$$

where $\gamma_l(\boldsymbol{x}_l, \epsilon)$ is defined as the maximum of the remainder of the first-order Taylor expansion of $f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l)$, *i.e.*:

$$\gamma_l(\boldsymbol{x}_l, \epsilon) = \max_{\|\boldsymbol{\delta}_l\|_p \leq \epsilon} \| \, f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_l(\boldsymbol{x}_l) - \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n . \tag{20}$$

Similarly, the term for the right image is relaxed as follow:

$$\| \, f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) - f_r(\boldsymbol{x}_r) \, \|_n \leq \| \, \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r) \, \|_n + \gamma_r(\boldsymbol{x}_r, \epsilon). \tag{21}$$

Given Equation (19) and Equation (21), L_b is further relaxed to it upper bound, as shown in Equation (22).

$$\begin{aligned} L_b &= \| \, \|f_l(\boldsymbol{x}_l + \boldsymbol{\delta}_l) - f_r(\boldsymbol{x}_r + \boldsymbol{\delta}_r) + \boldsymbol{m}\|_n - \|f_l(\boldsymbol{x}_l) - f_r(\boldsymbol{x}_r) + \boldsymbol{m}\|_n \, \|_1 \\ &\leq \| \, \boldsymbol{\delta}_l \nabla_{\boldsymbol{x}_l} f_l(\boldsymbol{x}_l) \, \|_n + \gamma_l(\boldsymbol{x}_l, \epsilon) + \| \, \boldsymbol{\delta}_r \nabla_{\boldsymbol{x}_r} f_r(\boldsymbol{x}_r) \, \|_n + \gamma_r(\boldsymbol{x}_r, \epsilon). \end{aligned} \tag{22}$$