drones

MDPI

Article

Categorical-Parallel Adversarial Defense for Perception Models on Single-Board Embedded Unmanned Vehicles

Yilan Li [1,*], Xing Fan [1], Shiqi Sun [2], Yantao Lu [2] and Ning Liu [3]

[1] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China; xing.fan@xaut.edu.cn
[2] School of Computer Science, Northwestern Polytechnical University, Xi'an 710060, China; shiqisun@nwpu.edu.cn (S.S.); yantaolu@nwpu.edu.cn (Y.L.)
[3] Midea Group, Beijing 100070, China; ningliu1220@gmail.com
* Correspondence: liyilan@xaut.edu.cn

Abstract: Significant advancements in robustness against input perturbations have been realized for deep neural networks (DNNs) through the application of adversarial training techniques. However, implementing these methods for perception tasks in unmanned vehicles, such as object detection and semantic segmentation, particularly on real-time single-board computing devices, encounters two primary challenges: the time-intensive nature of training large-scale models and performance degradation due to weight quantization in real-time deployments. To address these challenges, we propose Ca-PAT, an efficient and effective adversarial training framework designed to mitigate perturbations. Ca-PAT represents a novel approach by integrating quantization effects into adversarial defense strategies specifically for unmanned vehicle perception models on single-board computing platforms. Notably, Ca-PAT introduces an innovative categorical-parallel adversarial training mechanism for efficient defense in large-scale models, coupled with an alternate-direction optimization framework to minimize the adverse impacts of weight quantization. We conducted extensive experiments on various perception tasks using the Imagenet-te dataset and data collected from physical unmanned vehicle platforms. The results demonstrate that the Ca-PAT defense framework significantly outperforms state-of-the-art baselines, achieving substantial improvements in robustness across a range of perturbation scenarios.

Keywords: adversarial defense; robustness improvement; single-board computing; unmanned vehicles perception models

Citation: Li, Y.; Fan, X.; Sun, S.; Lu, Y.; Liu, N. Categorical-Parallel Adversarial Defense for Perception Models on Single-Board Embedded Unmanned Vehicles. *Drones* **2024**, *8*, 438. https://doi.org/10.3390/drones8090438

Academic Editor: Yushu Yu

Received: 23 July 2024
Revised: 23 August 2024
Accepted: 27 August 2024
Published: 28 August 2024

1. Introduction

In the context of simultaneous localization and mapping (SLAM) of unmanned vehicles tasks, deep neural networks (DNNs) have achieved remarkable performance in providing proficient and adaptable solutions for perception modules. These modules encompass multiple computer vision sub-tasks, including object detection, semantic segmentation, and agent localization. However, recent studies have highlighted that DNNs can be vulnerable to specific perturbation patterns during inference. For instance, traffic signs or QR codes can be incorrectly identified due to imperceptible graffiti or obstructions [1–4]. This vulnerability poses a significant threat to the robustness of perception models utilizing DNNs. When unmanned vehicles build a map of an unknown environment while keeping track of their location within it, perception models that incorporate DNNs may make incorrect predictions when their input sensing data are subjected to disturbances such as common corruption, noise, or adversarial perturbations.

To address the robustness issues of deep neural networks (DNNs), several methodologies have been proposed, including data augmentation, random smoothing, and adversarial training. Among these approaches, we emphasize the application of adversarial training due to its promising performance and implementation efficiency. Adversarial training

equips DNNs with the ability to defend against perturbations by integrating adversarial examples (AEs) into the training process [5]. Initially, AEs are created by applying carefully crafted perturbations to benign inputs. The DNN is subsequently trained on data augmented with these AEs. Extensive research has been devoted to improving defense mechanisms against adversarial attacks [5–8]. As noted in [9–11], the process of generating adversarial examples (AEs), which requires numerous gradient ascent iterations, can significantly increase computational overhead. Furthermore, the difficulty of managing ambiguous gradients can lead to a false sense of security. To mitigate these challenges, several regularizers, such as TRADES [12] and LLR [13,14], have been utilized to strike a better balance between the classification accuracy of adversarially trained models and their robustness against attacks.

While adversarial training shows promise in enhancing the robustness of DNNs, significant challenges arise when applying existing methods to unmanned vehicles perception models within the context of SLAM tasks. Specifically, two critical difficulties impede the application of conventional adversarial training to unmanned vehicles perception models deployed on real-time single-board computing devices: (i) Onboard device limitations—to meet real-time demands while deploying unmanned vehicles models on real-time single-board computing devices, it is necessary to quantize model weights into half-precision floats or integers. However, this weight quantization process can potentially compromise the effectiveness of adversarial training. (ii) Constraints on model size—adversarial training can be computationally intensive, especially when implemented on practical DNN models. This leads to prohibitive computational costs when using existing methods for real-world unmanned vehicles models, which continually grow in weight complexity.

To address these challenges, we introduce Ca-PAT, a comprehensive adversarial training framework designed to enhance the resilience of unmanned vehicles perception models implemented on real-time single-board computing devices. As shown in Figure 1, when compared to traditional training schemes, our approach features a novel categorical gradient-based adversarial training method, termed the Parallel Adversarial Training (PAT) pipeline. This pipeline adeptly manages feature variations while simultaneously reducing computational demands. Additionally, to alleviate the robustness loss associated with weight quantization in single-board computing deployment, we present Ca-PAT as an alternative-direction optimization framework. This framework coordinates an iterative and cooperative interaction between adversarial training and weight quantization processes.

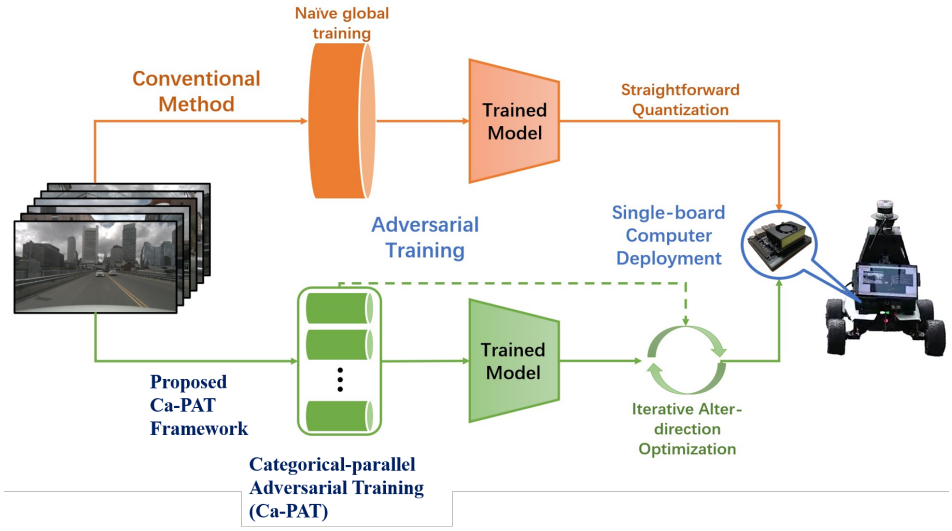


Figure 1. Comparison between our proposed framework and conventional adversarial training framework.

The main contributions of our work are summarized as follows:

- *Identification of significant limitations:* We identified substantial limitations in the adversarial training of unmanned vehicles perception models, including challenges associated with quantization for single-board computers and computational complexity arising from the large number of parameters in these models.
- *Proposal of a comprehensive framework:* To address these limitations, we introduce an adversarial training framework specifically tailored for perception models, termed Ca-PAT, designed for deployment on real-time single-board computing devices. This framework utilizes the alter-direction optimizer in combination with a novel categorical-parallel adversarial training method to simultaneously train and quantize the models.
- *Rationale of the proposed Ca-PAT:* We provide a rational proof of the effectiveness of Ca-PAT by deriving its theoretical upper bound during backward propagation.

The rest of the paper is organized as follows: Section 2 offers an overview of the background and definitions pertinent to this work. Section 3 describes the proposed framework in detail. The results of our experiments are presented in Section 4. Finally, Section 6 provides the conclusions of the work.

2. Background and Definition

2.1. Overview of Unmanned Vehicles Perception Models

Unmanned vehicles have been extensively deployed across various domains, leveraging advanced perception models to autonomously perform complex tasks. These models enable unmanned vehicles to accurately interpret their surroundings, facilitating critical functions, such as navigation and object recognition. By employing sophisticated algorithms that process sensor data from cameras, LiDAR, and other sources, unmanned vehicles perception models construct detailed environmental understandings. The applications of these models are diverse, ranging from agriculture, where they monitor crop health and manage livestock, to search and rescue operations, where they locate survivors and assess disaster-stricken areas. Additionally, unmanned vehicles with advanced perception capabilities are instrumental in infrastructure inspection, environmental monitoring, and military reconnaissance, highlighting the versatility and growing importance of robust and reliable perception systems for ensuring the efficacy and safety of unmanned vehicles operations.

Unmanned vehicles perception models play a pivotal role in executing essential tasks, such as simultaneous localization and mapping (SLAM) [15], and object detection. These models enable unmanned vehicles to create and update maps of unknown environments while simultaneously tracking their location within those maps, which is crucial for autonomous navigation in dynamic or uncharted territories. Object detection models allow unmanned vehicles to identify and classify various objects in their surroundings, facilitating tasks like surveillance, environmental monitoring, and target recognition. Motion planning models compute optimal flight paths to navigate through complex environments, avoiding obstacles and ensuring efficient route planning. The integration of these models enhances the operational capabilities of unmanned vehicles, making them indispensable tools in diverse applications, such as autonomous delivery, precision agriculture, disaster response, and urban planning. Continuous improvement of perception models is, therefore, critical to advancing the autonomy and reliability of unmanned vehicles systems.

However, unmanned vehicles perception models face several significant challenges, primarily due to computational limitations and the need for real-time processing. One of the foremost issues is the constrained computational capacity of onboard hardware, often limited to single-board computing devices with restricted processing power and memory. These limitations make it challenging to implement complex and large-scale deep learning models without sacrificing performance or accuracy. Additionally, unmanned vehicles must process sensory data in real time to navigate dynamically changing environments and make instantaneous decisions. This real-time requirement imposes stringent demands on computational efficiency and latency, as delays can lead to critical failures in navigation and obstacle avoidance. Moreover, the power consumption associated with high-performance

computations is another constraint, as unmanned vehicles typically operate on limited battery life, necessitating energy-efficient algorithms. Balancing the trade-offs between computational load, processing speed, and power efficiency poses a significant challenge in developing and deploying effective unmanned vehicles perception models. Addressing these challenges is essential for enhancing the autonomy, reliability, and operational lifespan of unmanned vehicles in various applications.

2.2. Adversarial Training to Enhance Network Robustness

Adversarial attacks involve intentionally manipulating input data to mislead deep learning models, resulting in incorrect outputs. These attacks exploit weaknesses in the model's decision-making process by introducing subtle perturbations to the input data, which are often imperceptible to humans but can significantly affect the model's predictions. The impact of adversarial attacks can be profound, especially in high-stakes applications, such as autonomous driving, medical diagnosis, and financial systems, where inaccurate outputs can lead to safety risks, misdiagnoses, and financial damage. To counteract these threats, adversarial training has recently gained prominence as an effective defense strategy. This approach involves enhancing the training dataset with adversarial examples, inputs specifically designed to trick the model, allowing it to learn to correctly classify these perturbed inputs. Adversarial training aims to improve the model's robustness and resilience by exposing it to adversarial examples during the training process, typically through iterative optimization to reduce loss on both normal and adversarial examples.

Here, we use the target detection classification problem as an illustrative example. Let $\{\mathcal{X}, \mathcal{Y}\}$ be the training dataset with C classes. $\mathcal{X} \subseteq \mathbb{R}^N$ represents the set of input images, where N denotes the number of pixels in each image. $\mathcal{Y} \subseteq \mathbb{R}^C$ denotes the label set corresponding to $\mathcal{X}$. $\mathcal{X}_c$ denotes the images labeled with class c, and $\boldsymbol{x} \in \mathcal{X}$ is used to represent a single image sampled from $\mathcal{X}$. Each single label $\boldsymbol{y} \in \mathcal{Y}$ is represented by a one-hot vector.

To solve the image classification problem on dataset $\{\mathcal{X}, \mathcal{Y}\}$, a neural network classifier $f(\boldsymbol{x}; \boldsymbol{\theta}) : \boldsymbol{x} \mapsto \mathbb{R}^C$ is trained to map an input sample $\boldsymbol{x} \in \mathbb{R}^N$ to the distribution of inference probability $p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta})$. $f(\boldsymbol{x}; \boldsymbol{\theta})$ denotes the approximation function of the neural network with weights $\boldsymbol{\theta}$ for predicting the class label given an input image $\boldsymbol{x}$. Following the stochastic gradient descent (SGD) process [16], we denote the mini batch as $\mathcal{D}_{\mathcal{B}}$ with a batch size of s, and use $\mathcal{L}$ to denote the objective function of the problem. A standard and general objective function is the cross-entropy loss function defined by:

$$\mathcal{L}_{obj}(\boldsymbol{x}; \boldsymbol{y}, \boldsymbol{\theta}) = -(\boldsymbol{y}^T \log_e(p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta})). \tag{1}$$

Since the classifier $f(\boldsymbol{x}; \boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$, to render a model with high accuracy and enhanced adversarial robustness with respect to L_p norm restriction, the objective function with perturbations (i.e., the robustness of a network) can be formulated as follows:

$$\begin{aligned} &\arg\max_{i \in \mathcal{C}} f_i(\boldsymbol{x}; \boldsymbol{\theta}) = \arg\max_{i \in \mathcal{C}} f_i(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{\theta}) \\ &\text{s.t.} \quad \forall \boldsymbol{\delta} \in B_p(\epsilon) = \{\boldsymbol{\delta} : \|\boldsymbol{\delta}\|_p \leq \epsilon\} \end{aligned} \tag{2}$$

where ϵ is the maximum magnitude for adversarial perturbation $\boldsymbol{\delta}$, p is the paradigm, and $B(\epsilon)$ is short for $B_\infty(\epsilon)$ when p is ∞, which is a key factor affecting the adversarial robustness of a network.

However, neural networks are notoriously vulnerable [1], even under simple gradient-based attacks, such as projected gradient descent (PGD) [8]. PGD and its gradient-based series aim to generate perturbations, which could maximize the objective function (1) to attack inference results. The training steps that PGD employs to generate effective perturbations and degrade the performance of networks is defined as follows:

$$\begin{aligned} \boldsymbol{\delta} \leftarrow &\text{Proj}(\boldsymbol{\delta} + \eta \bigtriangledown_{\boldsymbol{\delta}} \mathcal{L}_{obj}(\boldsymbol{x} + \boldsymbol{\delta}; \boldsymbol{y}, \boldsymbol{\theta})) \\ &\text{s.t. } \forall \boldsymbol{\delta} \in B_p(\epsilon) = \{\boldsymbol{\delta} : |\boldsymbol{\delta}|_p \leq \epsilon\} \end{aligned} \tag{3}$$

where $\bigtriangledown$ denotes a gradient derivation operation.

To defend against attacks and protect the neural networks, adversarial training [8] has been used, which randomly combines original examples and AEs generated from (3) as the training data. The cost of adversarial training is mainly focused on the optimization step iteration when maximizing the objective function, but when the optimization step is reduced, the defense capability is weakened. Thus, the trade-off between defense capability and computation cost becomes an inevitable problem. Another inconvenient fact is that the inference accuracy and adversarial robustness of networks may be at odds [14]. In other words, it is imperative to find an approach that maximizes the defense performance without reducing the inference accuracy. Therefore, several works, such as TRADES and local linear regularization (LLR) [12,13], have explored alternative ways to enhance adversarial training. For instance, LLR tries to smooth the surface of networks, i.e., minimizing the following quantity γ,

$$\gamma = \max_{\delta \in B_p(\epsilon)} \| \mathcal{L}_{obj}(\boldsymbol{x} + \boldsymbol{\delta}) - \mathcal{L}_{obj}(\boldsymbol{x}) - \boldsymbol{\delta}^T \bigtriangledown_{\boldsymbol{x}} \mathcal{L}_{obj}(\boldsymbol{x}) \| \tag{4}$$

which represents the norm distance between the adversarial loss and its first-order Taylor expansion. Methods such as LLR focus on the inherent properties of networks and effectively address the limitations of conventional adversarial training. By enhancing network characteristics, such as achieving smoother decision boundaries, these methods can help mitigate the trade-offs between accuracy and adversarial robustness. Consequently, in this work, we propose analyzing and improving defensive performance through the lens of Lipschitz continuity.

2.3. Motivation of Ca-PAT

Adversarial training significantly enhances the robustness of machine learning models but introduces challenges, such as increased computational demands and potential trade-offs between accuracy and robustness. Despite these challenges, it remains a vital strategy in the ongoing effort to develop secure and reliable machine learning systems. The primary motivation for adversarial training has been to focus on optimizing model parameters by comparing adversarial outputs with corresponding ground truth labels. This approach, however, evaluates the learning process solely on the basis of prediction accuracy, without accounting for robustness or imposing constraints on the model's sensitivity to perturbations during training. Consequently, achieving comprehensive defense against adversarial attacks remains difficult. Selvaraju et al. [17] explored the influence of adversarial attacks on models by analyzing attention maps. Building on this foundation, subsequent research [18–21] proposed analyzing the output range of neural networks using formal methods and linear programming. However, these studies primarily address simpler networks, such as multi-layer perceptrons, and do not fully demonstrate the effectiveness of high-level features in complex classification and detection tasks. By globally constraining output and attention distortions during training, the impact of adversarial examples can be significantly reduced. Our approach quantifies logits distortion estimation, making it tractable during adversarial training iterations, particularly for large-scale deep neural network (DNN)-based defenses. Additionally, we aim to constrain variations in gradient-based attention maps [17], especially in the presence of subtle perturbations and distortions.

To summarize existing adversarial training approaches, solving (3) is computationally intractable, particularly when the parameter set $\boldsymbol{\theta}$ is large, as is often the case with perception models in autonomous vehicles. These models feature extensive and intricate parameter spaces, rendering direct optimization both challenging and resource-intensive. To overcome this issue, we have developed Ca-PET, a novel methodology that systematically decomposes the perception model f into several smaller, more manageable subcomponents. This decomposition is achieved by partitioning the overall parameter set $\boldsymbol{\theta}$

into a series of subsets $\boldsymbol{\theta_i}$, each corresponding to specific, decoupled components of the model. By identifying and utilizing decoupling conditions, Ca-PET significantly mitigates the computational demands associated with adversarial training, thus facilitating more efficient and scalable model optimization. The detailed methodology and implementation of this framework are described comprehensively in Section 3.

Furthermore, to adapt general adversarial training for single-board computing devices, we approach the problem as a constrained optimization challenge. Our objective is to enhance the robustness of the target model while maintaining its performance post-parameter quantization. This complex task involves multiple constraints and is not easily resolved by conventional stochastic gradient descent methods. Consequently, Ca-PAT employs the Alternating Direction Method of Multipliers (ADMM) [22] to create a robustness optimization framework. This approach transforms the constrained optimization problem into an unconstrained one using augmented Lagrangian factors, and subsequently decomposes the problem into manageable sub-problems: parameter quantization and adversarial training. These sub-problems are well-suited for efficient stochastic gradient descent methods, such as ADAM.

3. Categorical-Parallel Adversarial Training for Perception

In this section, we firstly propose the efficient categorical-parallel (CaP) mechanism for large-scale models. Furthermore, we propose Ca-PAT, the quantization-friendly adversarial training framework for perception module on single-board embedded unmanned vehicles. To facilitate understanding, we summarize the parameter notations in this work as shown in Table 1.

Table 1. Parameters notations.

Parameter	Notation
$\boldsymbol{x}$	input
$\boldsymbol{y}$	ground truth with respect to $\mathcal{X}$
$\mathcal{D}$	Dataset
$\mathcal{L}_{obj}$	the objective function of the network
$f(\boldsymbol{x}, \boldsymbol{\theta})$	DNN-based classifier
$\boldsymbol{\delta}$	adversarial perturbation
ϵ	maximum magnitude of $\boldsymbol{\delta}$
$\mathfrak{L}$	Lipschitz Constant
$\mathfrak{L}_i$	Lipschitz Constant over category i
$\mathbb{R}^N$	N dimensional space
sup	supreme
$\|\|\|_p$	p norm distance, $\|\boldsymbol{x}\|_p = [\sum_{k=1}^{N} \|x^k\|^p]^{\frac{1}{p}}$
α,β	the order of norm
$\mathbb{F}$	the feature map from any conv layer

3.1. Objective

The primary idea of this work is to limit feature deviations with Lipschitz continuity-based strategies [23,24] from a quantitative perspective, and thus enhance the robustness of large-scale autonomous driving perception (ADP) models. In other words, the objective is to avoid subtle perturbations in the input space causing deviations in the output, especially for large-scale networks. Notably, adversarial examples can be defended against by smoothing output biases for subtle input changes. However, due to the non-linearity and massive number of parameters of neural networks, it is not straightforward to perform reachability

analysis of neural networks, especially during training. Inspired by the concept of Lipschitz continuity, we exploit Lipschitz-based optimization strategies to quantize and restrict the perturbation sensitivity of neural networks. This viewpoint provides a mathematical interpretation for the abstract correlation between the robustness and the variations of corresponding logits and attention map values of the input. Furthermore, we leverage a systematic end-to-end training framework, to satisfy computational feasibility requirements in addition to providing satisfactory model accuracy and adversarial robustness.

3.2. Categorical-Parallel Mechanism

Without loss of generality, given a neural network $\mathcal{N}$, we formalize its mapping function $f(\boldsymbol{x};\boldsymbol{\theta})$ as a feature extraction problem with C feature channels. $\boldsymbol{x}$ is an arbitrary input from the input set $\mathcal{X}$. The global Lipschitz constant $\mathfrak{L}$ of the network represents the rate of gradient changes of $\mathcal{N}$ after training all images. $\mathfrak{L}$ of a well-designed adversarially trained network can be obtained by solving the problem of the following form:

$$\mathfrak{L}^{\alpha,\beta}(f,\mathcal{X}) = \sup_{\boldsymbol{x_p}\neq\boldsymbol{x_q}} \frac{\left\|f(\boldsymbol{x_p};\boldsymbol{\theta}) - f(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_\alpha} \tag{5}$$

where $\boldsymbol{x_p}$ and $\boldsymbol{x_q}$ denote two different input images sampled from the input set $\mathcal{X}$, and $\|\|_\beta$ and $\|\|_\alpha$ denote the β-norm and α-norm, respectively. "sup" indicates the supremum and it ensures the minimal rate of function change and uniform continuity. $\mathfrak{L}(f,\mathcal{X})$ is a constant that indicates the adversarial perception sensitivity of the network.

In order to facilitate its design, we divide the overall adversarial training into sub-phases. *Phase I* employs $\mathcal{L}_{obj}$, which denotes the task-wise objectives of the network. *Phase II* is represented by $\mathfrak{L}_{lip}$, focusing on the optimization of the logits variances under the continuity property of the model and improves the learning performance through parallel training. The objective is to eliminate the negative influence of perturbations on inputs, by strengthening the attention information and limiting feature distortion of the discriminative image regions. Figure 2 provides an overview of the proposed Ca-PAT defense method. $\mathfrak{L}_{lip}$ enforces the prediction stability of each category with respect to their output features from the macro-level. Given $\mathcal{L}_{obj}$ and $\mathfrak{L}_{lip}$, the optimization goal is directly correlated with the adversarial robustness of practical large-scale DNNs, and the overall network continuity constraint needs to be satisfied.

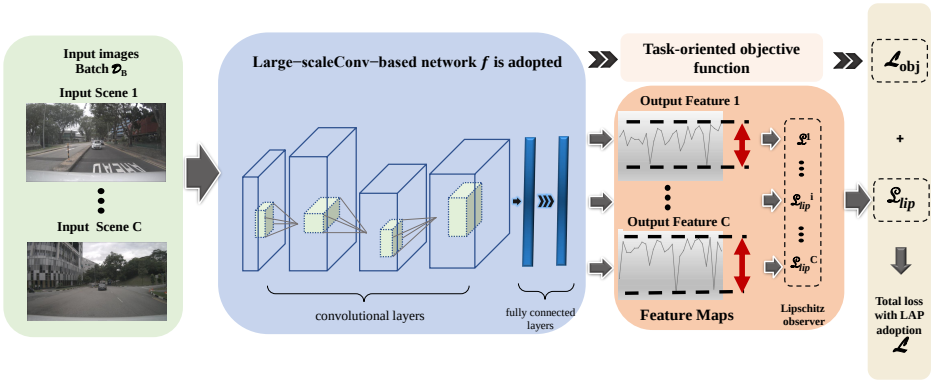


Figure 2. Overview of the proposed CaP mechanism. $\mathcal{L}_{obj}$ is the original objective function of the task. $\mathfrak{L}_{\mathrm{lip}}$ categorically measures the difference between the expected output and adversarial output predicted by the network f. The primary objective is to minimize the total loss $\mathcal{L}$, which is a linear combination of $\mathcal{L}_{obj}$ and $\mathfrak{L}_{\mathrm{lip}}$, and is back-propagated through SGD to update the weights of the network. Here, we use a convolutional DNN to illustrate the idea, but f can be any practical DNN of any scale.

Due to the continuous, non-linear and non-convex properties of the model $f(\boldsymbol{x},\boldsymbol{\theta})$, the global Lipschitz constant $\mathfrak{L}(f,\mathcal{X})$ in (5) cannot be calculated practically by derivation

or other mathematical methods. Hence, the key idea of the categorical-parallel Lipschitz optimization is to approximate the adversarial property of the global network with a group of scalars, where each scalar represents the Lipschitz constant of one category. From the perspective of representing adversarial property, we demonstrate that this involves the same or even stricter regularization requirement compared to the global Lipschitz constant of the network.

Assume there are C feature channels in the input set $\mathcal{X}$ and, correspondingly, the calculation of the global Lipschitz constant can be partitioned in C groups. Each categorical-parallel Lipschitz constant is denoted by a scalar $\mathfrak{L}_{C_i}$. The calculation of $\mathfrak{L}_{C_i}$ only considers the difference between the expected output and the current output within the same category C_i. As a result, the categorical-parallel Lipschitz constant search space is significantly reduced, thereby significantly decreasing the computational complexity. The categorical-parallel Lipschitz constant over category i is calculated as follows:

$$\mathfrak{L}_{C_i}^{\alpha,\beta}(f,\mathcal{X}_i) = \sup_{\substack{x_p \neq x_q \\ x_p, x_q \in \mathcal{X}_i}} \frac{\left\| f_i(x_p;\theta) - f_i(x_q;\theta) \right\|_\beta}{\left\| x_p - x_q \right\|_\alpha} \tag{6}$$

where $f_i(x;\theta)$ is the output value for the i^{th} channel, and $\mathcal{X}_i \subseteq \mathbb{R}^N$ is the input space of the i^{th} channel. Finally, the approximation of the global Lipschitz constant can be performed by combining the results of each feature channel. It is important to note that, during training, the categorical-parallel Lipschitz optimization estimates the global Lipschitz constant by restricting the distortion of features for each channel individually.

3.3. Constraint on Categorical-Parallel Training

Given a classifier $f(x;\theta) : \mathbb{R}^N \to \mathbb{R}^C$ over an open set $\mathcal{X} \subseteq \mathbb{R}^N$, the constraint between the parallel $\mathfrak{L}_i$ and the global Lipschitz observer $\mathfrak{L}$ is as follows:

$$\mathfrak{L}^{\alpha,\beta}(f,\mathcal{X}) \leq \sum_{i=1}^{C} \mathfrak{L}_i^{\alpha,\beta}(f,\mathcal{X}_i) \tag{7}$$

Assume the global Lipschitz constant is obtained at two specific points x_p and x_q, where $x_p \neq x_q$. Then,

$$\begin{aligned} \mathfrak{L}^{\alpha,\beta}(f,\mathcal{X}) &= \sup_{x_p \neq x_q} \frac{\left\| f(x_p;\theta) - f(x_q;\theta) \right\|_\beta}{\left\| x_p - x_q \right\|_\alpha} \\ &= \frac{\left\| f(x_p;\theta) - f(x_q;\theta) \right\|_\beta}{\left\| x_p - x_q \right\|_\alpha} \end{aligned} \tag{8}$$

Suppose that there is a set $\mathcal{U} = \{({x_p}^1, {x_q}^1), \ldots, ({x_p}^C, {x_q}^C)\}$, such that the pair $({x_p}^i, {x_q}^i)$, ${x_p}^i \neq {x_q}^i$ satisfies the requirement for the parallel Lipschitz constant $\mathfrak{L}_i^{\alpha,\beta}(f,\mathcal{X})$. Therefore,

$$\begin{aligned} \mathfrak{L}_{C_i}^{\alpha,\beta}(f,\mathcal{X}) &= \sup_{x_p \neq x_q} \frac{\left\| f_i(x_p;\theta) - f_i(x_q;\theta) \right\|_\beta}{\left\| x_p - x_q \right\|_\alpha} \\ &= \frac{\left\| f_i({x_p}^i;\theta) - f_i({x_q}^i;\theta) \right\|_\beta}{\left\| {x_p}^i - {x_q}^i \right\|_\alpha} \end{aligned} \tag{9}$$

Since Lipschitz is a computation of the supremum, we can rewrite the equation as follows:

$$\begin{aligned}\mathfrak{L}_{C_i}^{\alpha,\beta}(f,\mathcal{X}) &= \frac{\left\|f_i(\boldsymbol{x_p}^i;\boldsymbol{\theta}) - f_i(\boldsymbol{x_q}^i;\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p}^i - \boldsymbol{x_q}^i\right\|_\alpha} \\ &\geq \frac{\left\|f_i(\boldsymbol{x_p};\boldsymbol{\theta}) - f_i(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_\alpha}\end{aligned} \tag{10}$$

Taking the summation over the set $\mathcal{U}$:

$$\begin{aligned}\sum_{i=1}^{C}\mathfrak{L}_{C_i}^{\alpha,\beta}(f,\mathcal{X}) &= \sum_{i=1}^{C}\frac{\left\|f_i(\boldsymbol{x_p}^i;\boldsymbol{\theta}) - f_i(\boldsymbol{x_q}^i;\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p}^i - \boldsymbol{x_q}^i\right\|_\alpha} \\ &\geq \sum_{i=1}^{C}\frac{\left\|f_i(\boldsymbol{x_p};\boldsymbol{\theta}) - f_i(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_\alpha} \\ &= \frac{\sum_{i=1}^{C}\left\|f_i(\boldsymbol{x_p};\boldsymbol{\theta}) - f_i(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_\alpha} \\ &\geq \frac{\left\|f(\boldsymbol{x_p};\boldsymbol{\theta}) - f(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_\beta}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_\alpha} \\ &= \mathfrak{L}^{\alpha,\beta}(f,\mathcal{X})\end{aligned} \tag{11}$$

As can be seen, the summation of categorical-parallel Lipschitz constants represents an upper bound on the sensitivity of large-scale deep neural networks to perturbations when represented by the global Lipschitz constant. Although we place the emphasis on the difference between the features and the input in each channel, we show that the categorical-parallel Lipschitz observer helps to represent the adversarial robustness of the global network and it can be computed categorical practically and more efficiently. Therefore, the categorical-parallel mechanism can be effectively utilized in the training process of large-scale DNN models.

3.4. Categorical-Parallel Adversarial Training for Perception Module

It is demonstrated in Section 3.2 that the computation complexity increases to $O(C^2)$ after calculating $\mathfrak{L}_{C_i}^{\alpha,\beta}(f,\mathcal{X}_i)$ for all channels. The computation complexity further increases with increasing number of input channels. Thus, we do not quantify the robustness in (6) using computationally inefficient theoretical methods. Different from conventional work, we leverage stochastic sampling in every small batch to estimate the robustness through parallel-Lipschitz optimization. Given a mini-batch $\mathcal{D}_\mathcal{B}$ with C channels, under the adversarial perturbation $\boldsymbol{\delta}$, the corresponding adversarial batch is specified as follows:

$$\mathcal{D}_\mathcal{B}{}^{adv} = \mathcal{D}_{\mathcal{B}1}^{adv} \cup \mathcal{D}_{\mathcal{B}2}^{adv} \cup \ldots \cup \mathcal{D}_{\mathcal{B}C}^{adv} \tag{12}$$

where

$$\mathcal{D}_{\mathcal{B}i}^{adv} = \{x + \boldsymbol{\delta} | \boldsymbol{x} \in \mathcal{X}_i \cap \mathcal{D}_\mathcal{B}, \boldsymbol{\delta} \in B(\epsilon)\}, i = 1, \ldots, C. \tag{13}$$

$\boldsymbol{\delta}$ is optimized by

$$\begin{aligned}&\boldsymbol{\delta} = \arg\max_{\boldsymbol{\delta}} \mathcal{L}_{SSIM}(f(\boldsymbol{x})||f(\boldsymbol{x}+\boldsymbol{\delta})) \\ &\text{s.t.} \quad \boldsymbol{\delta} \in B(\epsilon), \boldsymbol{x} \in \mathcal{D}_{\mathcal{B}i}\end{aligned} \tag{14}$$

where $\mathcal{L}_{SSIM}$ indicates the structural similarity loss [25], which is devised to evaluate the similarity between the expected output feature maps and the adversarial output feature maps. Given two images, i.e., $f(\boldsymbol{x})$ and $f(\boldsymbol{x}+\boldsymbol{\delta})$, we separate them and obtain their K corresponding patches. With the corresponding patch pair, for instance, $\boldsymbol{p}$ from $f(\boldsymbol{x})$ and

$\boldsymbol{q}$ from $f(\boldsymbol{x}+\boldsymbol{\delta})$, the SSIM index combines three loss modules, respectively, a luminance term, a contrast term and a structure term, which are defined as follows:

$$l(p,q) = \frac{2\mu_p\mu_q + C_1}{\mu_p^2 + \mu_q^2 + C_1}, \tag{15}$$

$$c(p,q) = \frac{2\sigma_p\sigma_q + C_2}{\sigma_p^2 + \sigma_q^2 + C_2} \tag{16}$$

$$s(p,q) = \frac{\sigma_{pq} + C_3}{\sigma_p\sigma_q + C_3} \tag{17}$$

where μ_p and μ_q represent the mean of $\boldsymbol{p}$ and $\boldsymbol{q}$ while σ represents the variance. σ_{pq} is the covariance of $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively. The positive constants C_1, C_2, and C_3 are referred to stabilize each module. Therefore, the SSIM index loss could be defined as follows:

$$\begin{aligned}&\mathcal{L}_{SSIM}(f(\boldsymbol{x})||f(\boldsymbol{x}+\boldsymbol{\delta})) = \\ &\sum_{i=1}^{C} \sum_{\forall p\in f(\boldsymbol{x}), \forall q\in f(\boldsymbol{x}+\boldsymbol{\delta})} [l(p,q)]^{\alpha}[c(p,q)]^{\gamma}[s(p,q)]^{\beta}\end{aligned} \tag{18}$$

where parameters α, γ, and β are positive and are used to adjust the importance of three modules.

We introduce auxiliary batch $\hat{\mathcal{D}_{\mathcal{B}}}$, which contains twice as many images compared to $\mathcal{D}_{\mathcal{B}}$, including each input image and the corresponding adversarial image. $\hat{\mathcal{D}_{\mathcal{B}}}$ is also grouped into C channels. Thus, the execution of the parallel-Lipschitz optimization in the discrete domain reduces the computational complexity from $O(C^2)$ to $O(C)$ and it is determined by the sum of the parallel Lipschitz constants. Hence, we have the following:

$$\mathfrak{L}_{lip}(\hat{\mathcal{D}_{\mathcal{B}}};\boldsymbol{\theta}) = \sum_{i=1}^{C} \mathfrak{L}_{C_i}^{\alpha,\beta}(f, \hat{\mathcal{D}_{\mathcal{B}i}}) \tag{19}$$

where

$$\begin{aligned}&\mathfrak{L}_{\mathfrak{C}_i}{}^{\alpha,\beta}(f, \hat{\mathcal{D}_{\mathcal{B}i}}) = \\ &\qquad \max_{\substack{\boldsymbol{x_p},\boldsymbol{x_q}\in\hat{\mathcal{D}_{\mathcal{B}i}} \\ \boldsymbol{x_p}\neq\boldsymbol{x_q}}} \frac{\left\|f_i(\boldsymbol{x_p};\boldsymbol{\theta}) - f_i(\boldsymbol{x_q};\boldsymbol{\theta})\right\|_{\beta}}{\left\|\boldsymbol{x_p} - \boldsymbol{x_q}\right\|_{\alpha}}\end{aligned} \tag{20}$$

$$\begin{aligned}\hat{\mathcal{D}_{\mathcal{B}}} &= \cup_{i=1}^{C}\{\hat{\mathcal{D}_{\mathcal{B}i}}\} \\ &= \cup_{i=1}^{C}\{\boldsymbol{x_i}, \boldsymbol{x_i} + \boldsymbol{\delta}_i\}\end{aligned} \tag{21}$$

We have proven that this is the optimal analytical solution of the Lipschitz-based attribute quantization. Essentially, the optimization effectively overcomes the limitation of infeasible global quantization for adversarial perception and achieves higher robustness compared to baselines.

The overall loss for adversarial training is as follows:

$$\mathcal{L} = \lambda\mathcal{L}_{obj} + \gamma\mathfrak{L}_{lip} \tag{22}$$

where λ and γ are the coefficients of each term in the objective function. $\mathcal{L}_{obj}$ is the original objective function, and $\mathfrak{L}_{lip}$ can be calculated from (19).

3.5. Overall Optimization Process

To adapt adversarially trained models for quantization deployment on unmanned vehicles embedded with a single-board computer, we propose a unified optimization frame-

work that concurrently performs categorical-parallel adversarial training and quantization adaptation. In this approach, adversarial training is formulated as an alternating direction method of multipliers (ADMM) optimization problem. This framework treats the adversarial training of perception models as an optimization problem with quantization constraints. The optimization process then iteratively updates the parameters to converge towards the optimal solution.

To simplify the optimization process and make it suitable for efficient algorithmic solutions, following the primary principles of the ADMM algorithm, we decompose the initial optimization problem into two smaller sub-problems. These sub-problems can then be systematically solved through iterative procedures. To be specific, the problem can be formulated as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} [\mathcal{L}(\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\phi})], \\ s.t.\ \ A\boldsymbol{\theta} + B\boldsymbol{\phi} = \boldsymbol{c}. \end{aligned} \tag{23}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ represent the network parameters requiring optimization, and both $\mathcal{L}(\cdot)$ and $\mathcal{R}(\cdot)$ are convex functions. The matrices A and B, as well as the vector c, are predefined. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are identity weights at different training stages. Therefore, in this work, A, B, c are set to 1, -1, 0, respectively. $\mathcal{L}(\boldsymbol{\theta})$ is defined as the adversarial training loss function, while $\mathcal{R}(\boldsymbol{\phi})$ represents an auxiliary constraint function as follows:

$$\mathcal{R}(\boldsymbol{\phi}) = \begin{cases} 0, & if\ \boldsymbol{\theta} \in \mathbf{S}, \\ +\infty, & otherwise \end{cases} \tag{24}$$

where $\mathbf{S}$ denotes the quantization operation as defined by the deployment procedure for single-board computers on unmanned vehicles. In this context, $\mathcal{R}(\boldsymbol{\phi})$ is defined as a mandatory constraint, and all parameters must adhere to this quantization projection to avoid additional loss values. Failure to comply will result in an infinite loss value.

By introducing the Lagrange multiplier $\mathbf{U}$ associated with the constraint $\boldsymbol{\theta} - \boldsymbol{\phi} = 0$ in (23), the objective function can be formulated as follows:

$$\hat{\boldsymbol{\theta}} = \mathcal{L}(\boldsymbol{\theta}) + \mathcal{R}(\boldsymbol{\phi}) + \mathbf{U}^T(\boldsymbol{\theta} - \boldsymbol{\phi}) \tag{25}$$

where T denotes the transpose of the Lagrange multiplier $\mathbf{U}$. Consequently, the optimization process involves iteratively updating the variables $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, and $\mathbf{U}$ until convergence is achieved. This iterative procedure at the k^{th} iteration can be carried out as follows:

$$\min_{\boldsymbol{\theta}} \ \mathcal{L}(\boldsymbol{\theta}) + \frac{\rho}{2}||\boldsymbol{\theta} - \boldsymbol{\phi}^k + \mathbf{U}^k||_F^2, \tag{26}$$

$$\min_{\boldsymbol{\phi}} \ \mathcal{R}(\boldsymbol{\phi}) + \frac{\rho}{2}||\boldsymbol{\theta}^{k+1} - \boldsymbol{\phi} + \mathbf{U}^k||_F^2, \tag{27}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \rho(\boldsymbol{\theta}^{k+1} - \boldsymbol{\phi}^{k+1}). \tag{28}$$

where ρ stands for a positive parameter referred to as the penalty parameter.

In (26), $\mathcal{L}(\cdot)$ is a differentiable loss function of the DNN, and the second term $||\boldsymbol{\theta} - \boldsymbol{\phi}^k + \mathbf{U}^k||_F^2$ is both differentiable and convex. Therefore, during the training process, (26) can be solved by stochastic gradient descent, with the complexity remaining equivalent to that of training the original DNN. Next, the quantization operation $\mathcal{R}(\cdot)$, as represented in (27), is viewed as a vector space projection and is also differentiable. The update process for $||\boldsymbol{\theta}^{k+1} - \boldsymbol{\phi} + \mathbf{U}^k||_F^2$ remains consistent with that in (26). The derived $\boldsymbol{\theta}^{k+1}$ and $\boldsymbol{\phi}^{k+1}$ will be fed into (28); all three updated values are used in the next iteration until convergence is achieved. This iterative process ultimately yields an optimal solution to the original optimization problem.

Solution to the joint problem of adversarial training and quantization: Adversarial training and quantization can be cooperatively solved using the above-mentioned ADMM

framework. In Equation (25), $\mathcal{L}(\boldsymbol{\theta})$ represents the adversarial training task and $\mathcal{R}(\boldsymbol{\phi})$ represents the quantization task. At the beginning, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are identity weights. We first perform adversarial training by Equation (26). The training weights are marked as $\boldsymbol{\theta}^{k+1}$. Then, we perform quantization on $\boldsymbol{\phi}$ given $\boldsymbol{\theta}^{k+1}$. This step is illustrated in (27) and (24). The updated weights are marked as $\boldsymbol{\phi}^{k+1}$. Finally, $\mathbf{U}^{k+1}$ is able to be updated using (28).

4. Experimental Results

In this section, we conduct a thorough evaluation of our proposed framework. We begin with an analysis of learning performance, including learning accuracy and deployment effectiveness. Following this, we conduct the time complexity to examine its speed performance. Next, we evaluate the framework's effectiveness against various DNN sizes and different attack strengths. Finally, we carry out an end-to-end evaluation to compare the practical performance of our framework with state-of-the-art methods.

4.1. Datasets

In the experiments, we evaluate the robustness and generalizability of the proposed Ca-PAT framework for the object detection task, using three datasets as follows: ImageNet-te [26], NuImages [27], and a real-world captured dataset. ImageNet-te is a subset of ImageNet consisting of 10 classes, with 9000 training images and 3000 test images, all of size $320 \times 320 \times 3$. This dataset provides a controlled environment to test basic object detection capabilities. NuImages, on the other hand, offers a more challenging and realistic dataset derived from real-world driving scenarios. It comprises 93,000 images meticulously selected to represent a diverse range of driving situations, addressing the limitations of redundant annotations found in other datasets. NuImages includes challenging scenarios, diverse weather conditions, and temporal dynamics, making it a valuable resource for advancing object detection in autonomous driving applications. Additionally, we use a real-world captured dataset to further assess the framework's performance in practical settings. This dataset enhances our evaluation by providing real-world data, ensuring that our results are not only theoretical but also applicable in practical, real-world situations. This comprehensive evaluation demonstrates the strong adversarial awareness and generalizability of our Ca-PAT framework across different datasets and environments.

4.2. Experimental Settings

- **Baselines**. For attack methods, we select five state-of-the-art methods, including Common Corruptions (CC) [28], Projected Gradient Descent (PGD) [29], and Ensemble Attack (Ens) that includes APGD, FAB, and Square Attack [8]. All attacks are conducted with a perturbation ϵ set to $8/255$ and a stride length α_1 of $\epsilon/10$. Our adversarial images follow the above settings and are generated by iterative gradient ascent of the structural similarity loss, which is shown in (14). In this experiment, we compare the adversarial robustness of our strategy with five different baseline adversarial defense methods, configured in various ways, resulting in eight distinct adversarial defense approaches. The selected baseline defense methods include Robust Library [30], Salman [31], LLR [13], and TRADES [12]. Additionally, we include a baseline referred to as "Clean", which represents the training without any adversarial examples (AEs). This provides a comparison point for the performance of models trained without adversarial perturbations.
- **Implementation Details**. All input images are cropped to a size of $224 \times 224 \times 3$ and processed using a WideResNet [32] for the task. The network architecture includes an initial convolution layer with a stride of 2, followed by three residual blocks with strides $[2, 2, 2]$. This configuration is designed to demonstrate the effectiveness of our proposed Ca-PAT framework in enabling large-scale DNNs to achieve strong adversarial awareness. Following the established settings for LLR on ImageNet, we set the initial learning rate at 0.1, with a decay factor of 0.1 applied at epochs 35, 70, and 95. Our training regimen spans 100 epochs. For the proposed Ca-PET method, we

specified λ as 1.0 and γ as 0.5 as the hyper-parameters. Additionally, we employed L2-regularization with a strength of 1e-4 and a batch size of 64 for the training process. The experiments are performed on an RTX 3090 GPU paired with an Intel(R) Xeon(R) Gold 5218R CPU, utilizing CUDA version 11.0. This setup provides the necessary computational power and efficiency to validate our approach.

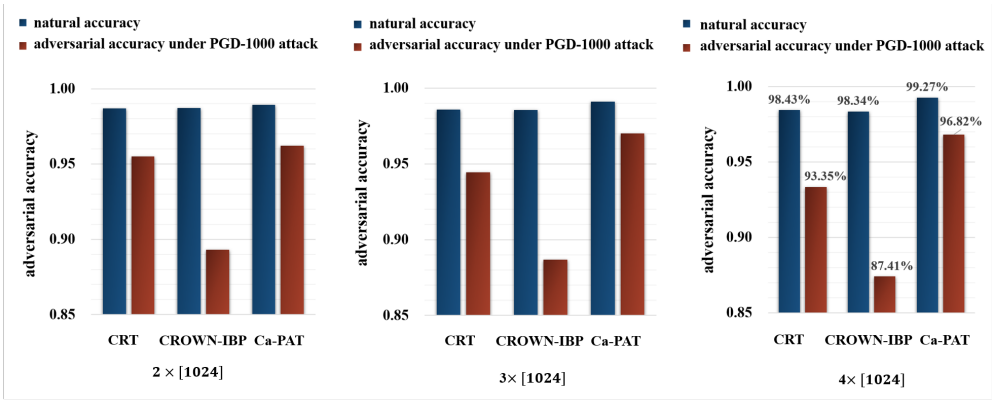
- **Evaluation Metrics**. To ensure fair comparisons, we applied consistent hyperparameter configurations across all experimental groups. The methodologies evaluated include Robust Library [30], Salman [31], LLR [13], and TRADES [12]. These established approaches were rigorously assessed within the context of the proposed framework to confirm their compatibility and performance. This systematic evaluation allows for a thorough comparison and the identification of potential performance enhancements.
  - Robust Library. The Robust Library is a comprehensive toolkit designed for evaluating and improving the robustness of deep learning models against adversarial attacks. It provides a collection of pre-implemented adversarial attack methods and defense strategies, allowing researchers to systematically assess the resilience of their models. This library facilitates reproducible and consistent evaluations, enabling a standardized comparison of different approaches to adversarial robustness.
  - Salman. The Salman approach, named after the researchers Salman et al., focuses on adversarial training techniques to enhance model robustness. This methodology incorporates adversarial examples during the training process to improve the model's ability to resist adversarial perturbations. By systematically exposing the model to challenging adversarial scenarios, the Salman method aims to strengthen the model's defenses and improve its overall robustness against attacks.
  - Lipschitz Linear Regularization. LLR is a defense mechanism that aims to improve the robustness of neural networks by enforcing Lipschitz continuity. This approach introduces a regularization term that controls the Lipschitz constant of the network, ensuring that small changes in the input lead to proportionally small changes in the output. By doing so, LLR helps in mitigating the model's sensitivity to adversarial perturbations, leading to enhanced stability and robustness against adversarial attacks.
  - TRADES. TRADES is an adversarial training framework designed to balance the trade-off between model accuracy and robustness. It introduces a surrogate loss function that explicitly quantifies and manages this trade-off. By optimizing this surrogate loss, TRADES aims to achieve a desirable balance where the model maintains high accuracy on clean data while also being robust against adversarial examples. This approach is particularly effective in scenarios where both accuracy and robustness are critical.

4.3. Learning Performance of Ca-PAT Framework

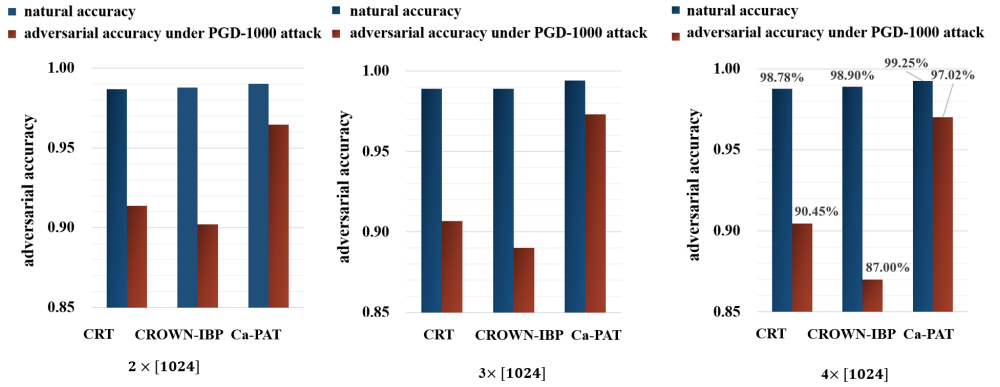
We commence with a practical evaluation of the proposed LAP-DNN framework. The objective is to verify the effectiveness of deploying the Ca-PAT defense framework in comparison to other methods, such as CRT [33], CROWN-IBP [34], and COAP [35], within an end-to-end framework that includes non-differentiable activation functions. A comprehensive comparison of the adversarial perception performance across different model scales is illustrated in Figure 3.

Figure 3a illustrates the change in accuracy under adversarial attack as the neural network size increases. Initially, a differentiable activation function is utilized. Our proposed Ca-PAT framework achieves an accuracy of 96.82% under the PGD-1000 attack with a $4 \times [1024]$ DNN structure. The accuracy degradation for our framework is negligible as the model size increases, especially when compared to baseline methods. Figure 3b presents the performance of the Ca-PAT framework with a non-differentiable activation function on the ImageNet-te dataset. Neural networks with increasing scales are adopted, and

the Softmax layer is replaced by ReLU. The Ca-PAT framework achieves higher accuracy under adversarial attacks, even with larger-scale DNNs. Specifically, under the PGD-1000 attack with a $4 \times [1024]$ DNN structure, the accuracy is 10.02% higher than COAP [35] and 6.57% higher than CROWN-IBP [34]. Furthermore, Ca-PAT exhibits much more moderate accuracy degradation as the model complexity increases. These results clearly demonstrate the significant advantages and potential of Ca-PAT, making it suitable for any end-to-end deep learning application. Compared to state-of-the-art Lipschitz-based methods, Ca-PAT achieves higher accuracy under attack with complex networks, irrespective of whether the mapping function is differentiable or non-differentiable.



(a) Adversarial defense performance comparison of multi-layer DNNs with softplus activation function as the scale of DNN increases.



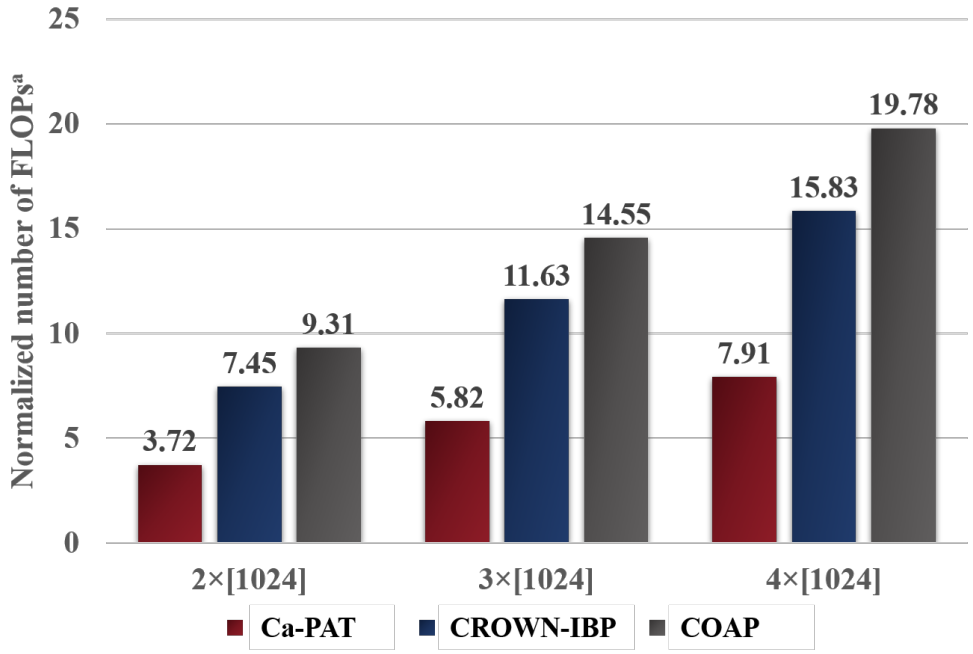
(b) Adversarial defense performance comparison of multi-layer DNNs with ReLU activation function as the scale of DNN increases.

Figure 3. Illustration of adversarial defense effectiveness on practical-scale DNNs on the ImageNette dataset for CRT [33], CROWN-IBP [34], COAP [35] and our Ca-PAT framework; (**a**) Comparison of accuracy under adv. attack for different scale DNNs with a differentiable activation function (i.e., softplus); (**b**) Comparison of accuracy under attack for different scale DNNs with non-differentiable activation function (i.e., ReLU). Here, $i \times [1024]$ refers to having i layers with 1024 neurons per layer. "Natural accuracy" indicates test results without adversarial attacks.

4.4. Time Complexity Analysis

Next, we compare the number of floating point operations (FLOPs) across different approaches. In this comparison, we exclude approaches that cannot be deployed with non-differentiable activation functions. Figure 4 illustrates the normalized number of FLOPs for CROWN-IBP [34], COAP [35], and our Ca-PAT framework. For instance, Ca-PAT requires 3.72 million FLOPs when deployed in a two-layer fully connected network, representing a 50.07% and 60.04% reduction compared to CROWN-IBP and COAP, respectively. As the model size increases, the computational complexity gap between the proposed Ca-PAT

framework and the benchmarks widens to as much as 2.5 times. It is noteworthy that the computational increase for the Ca-PAT framework is minimal as the model size grows. This efficiency is attributed to our approach of categorically separating the overall optimization into sub-problems, as demonstrated in Section 3.3, without compromising the convergence and correctness of the final objectives.



[a]: This comparison does not include approaches that can only compute the curvature bounds for DNN with differentiable activation functions.

Figure 4. Comparison of normalized number of FLOPs among CROWN-IBP, COAP, and our Ca-PAT with different network scales.

4.5. Effectiveness against Various DNN Sizes

We conclude our evaluation by assessing the performance of our Ca-PAT defense framework on a larger-scale network using the Imagenet-te dataset, which is a subset of the ImageNet dataset encompassing 10 categories and containing 9000 training images and 3000 test images, each sized $320 \times 320 \times 3$. For this experiment, we adjusted the stride of the convolutional layer to 2, while maintaining the strides of the three residual blocks at $[2, 2, 2]$. All training images were resized to $224 \times 224 \times 3$ before being input into the model. The total number of parameters reached 36.5 million. As demonstrated in Table 2, our proposed defense approach consistently outperforms the benchmark methods across all attack settings. The performance improvement becomes increasingly significant as the attack strength intensifies. Notably, our defense framework achieves 60.73% accuracy under the PGD-20 attack with a 0.21% error margin. When the attack strength is increased to PGD-40, our approach shows a 16.4% improvement in accuracy compared to TRADES, reducing the error margin by 86.2%. Under the PGD-100 attack, our framework achieves at least 47.01% accuracy, which is 17.35% higher than the best adversarial test accuracy achieved by TRADES [12]. It is important to note that this test result is conservative, considering the worst-case scenario for applying our Ca-PAT defense framework.

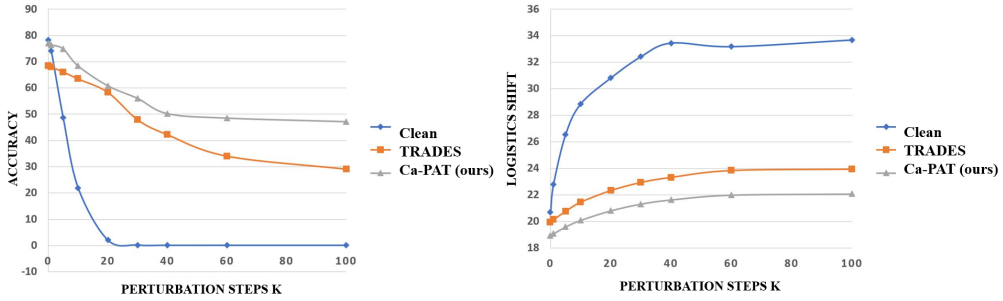
Table 2. Comparison of defense accuracy among different defense approaches, (including clean, TRADES, and Ca-PAT), under different attack configurations on the ImageNette dataset.

	Nature [b]	CC-20	CC-40	CC-100
Clean [a]	78.29 ± 0.53%	16.20 ± 0.28%	9.20 ± 0.25%	5.01 ± 0.14%
TRADES [12]	68.46 ± 0.83%	57.43 ± 0.13%	44.04 ± 1.23%	29.15 ± 0.51%
LLR [13]	71.19 ± 0.36%	60.71 ± 0.45%	48.58 ± 0.36%	37.01 ± 0.29%
RobustLib [30]	68.46 ± 0.83%	64.18 ± 0.49%	49.56 ± 0.31%	39.69 ± 0.18%
Salman [31]	70.74 ± 0.40%	64.58 ± 0.35%	52.37 ± 0.31%	45.18 ± 0.17%
Ca-PAT (ours)	**76.88 ± 0.41**%	**70.73 ± 0.21**%	**61.27 ± 0.17**%	**50.25 ± 0.14**%
	Nature [b]	PGD-20	PGD-40	PGD-100
Clean [a]	78.29 ± 0.53%	0.20 ± 0.08%	0.01 ± 0.05%	0.01 ± 0.01%
TRADES [12]	68.46 ± 0.83%	58.33 ± 0.28%	42.23 ± 0.29%	29.15 ± 0.21%
LLR [13]	71.19 ± 0.36%	57.71 ± 0.30%	45.58 ± 0.39%	33.99 ± 0.36%
RobustLib [30]	68.46 ± 0.83%	58.18 ± 0.41%	45.89 ± 0.26%	36.10 ± 0.19%
Salman [31]	70.74 ± 0.40%	59.73 ± 0.35%	46.37 ± 0.22%	39.18 ± 0.29%
Ca-PAT (ours)	**76.88 ± 0.41**%	**60.81 ± 0.39**%	**50.21 ± 0.21**%	**47.15 ± 0.17**%
	Nature [b]	Ens-20	Ens-40	Ens-100
Clean [a]	78.29 ± 0.53%	0.09 ± 0.05%	0.01 ± 0.04%	0.01 ± 0.01%
TRADES [12]	68.46 ± 0.83%	49.43 ± 0.21%	31.04 ± 0.12%	25.32 ± 0.19%
LLR [13]	71.19 ± 0.36%	49.71 ± 0.30%	35.72 ± 0.29%	27.81 ± 0.26%
RobustLib [30]	68.46 ± 0.83%	53.18 ± 0.36%	35.89 ± 0.33%	29.10 ± 0.11%
Salman [31]	70.74 ± 0.40%	51.73 ± 0.31%	41.75 ± 0.28%	34.01 ± 0.13%
Ca-PAT (ours)	**76.88 ± 0.41**%	**59.73 ± 0.31**%	**49.27 ± 0.19**%	**41.07 ± 0.09**%

[a]: Natural training without AEs or additional regularization. [b]: Test results without any attack.

4.6. Effectiveness against Attack Strengths

In what follows, we examine the effectiveness of the Ca-PAT defense framework in mitigating feature distortion under various attack strengths for the same DNN model. To ensure a fair comparison, we utilize the same network structure previously employed on the Imagenet dataset. We apply the PGD-k attack with a stride length of $\alpha_1 = \epsilon/10$. Figure 5 illustrates the relationship between the learning performance of different models under varying attack strengths. Specifically, Figure 5a displays the correlation between adversarial test accuracy and attack strength, while Figure 5b shows the correlation between logits distortion and different attack strengths for various defense approaches. The blue curves indicate the defense performance changes of the model trained without adversarial examples (AEs). As depicted in Figure 5b, the Ca-PAT defense framework maintains a 47.15% accuracy under the strong PGD-100 attack, whereas the performance of TRADES drops sharply to 29.15% under the same attack strength. By analyzing these curves, we observe that the performance of models trained with adversarial examples exhibits slight variations and faster convergence. Additionally, as shown in Figure 5b, the Ca-PAT defense framework effectively restricts logits distortion, significantly reducing the model's sensitivity to perturbations. The moderate trend in logit variations supports the conclusion that our approach is effective even for large-scale neural networks against adversarial attacks.

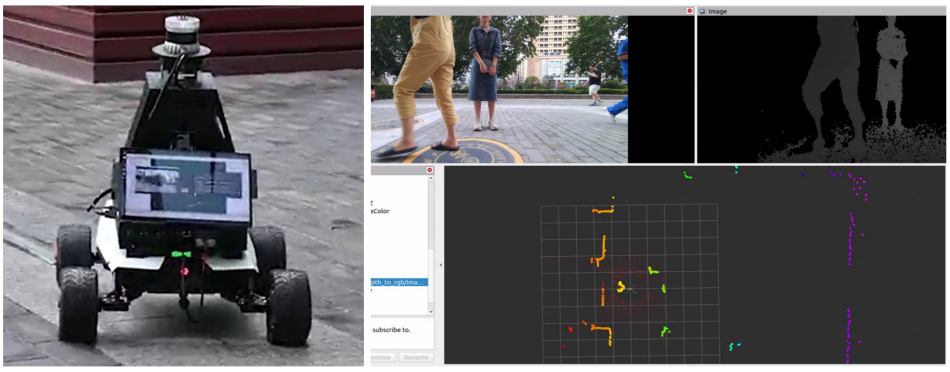


(a) Accuracy - Attack strength **(b) Logistics Shift - Attack strength**

Figure 5. Relationship between defense performance under different attack strengths. *k* indicates the attack strength. "Clean" refers to training without AEs or additional regularization.

4.7. End-to-End Evaluation

To evaluate the practical performance of our proposed Ca-PAT defense framework, we deployed it on a real-world autonomous car, specifically, the Scout-2.0 model [36], which measures 930 mm × 699 mm × 751 mm and is equipped with a 4k RGB camera and 64-ray LiDAR sensors. We used Scout-2.0 to gather 40 h of perception data, including LiDAR point clouds and RGB images, at a perception rate of 1 Hz, with vehicle coordinates following the NuScenes setting [27]. The hardware platform and sensing visualization are shown in Figure 6. During the training process, we first pretrain the models on the NuImages dataset. We then fine-tune the models using the real-world collected data, selecting FasterRCNN-Resnet50 as the detection model. Vehicles, pedestrians, and cyclists were labeled in the selected data. For inference, we captured and labeled 1000 samples to evaluate the mean average precision (mAP) performance of the trained detection model. Common Corruptions, PGD, and Ensemble attacks, as previously described, were used as the attack methods. TRADES, LLR, RobustLib, and Salman were selected as baseline defense methods. The quantitative comparison results are shown in Table 3. It can be observed that the proposed Ca-PAT defense framework achieves the best performance across all experiments. Furthermore, our framework demonstrates a significant performance advantage, particularly under 100-iteration attack scenarios. Based on various experimental settings, the results clearly indicate that the Ca-PAT defense framework can substantially enhance real-world detection robustness.



(a) Hardware platform **(b) Sensing Visualization**

Figure 6. Hardware platform and sensing visualization for real-world evaluation. (**a**) shows the hardware platform for gathering camera-LiDAR aligned data and onboard evaluating the fine-tuned 3D object detectors. (**b**) shows the GUI while gathering perception data.

Table 3. Comparison of object detection performance among Ca-PAT defense and baselines under different attacks on the practical real-world data.

Attacks	Def. Mtd.	Clean	Itr-20	Itr-40	Itr-100
CC	Clean	32.49	9.24	6.54	3.78
	TRADES [12]	23.05	14.71	11.62	9.97
	LLR [13]	22.95	15.73	11.73	10.07
	RobustLib [30]	22.38	13.83	10.81	8.63
	Salman [31]	21.71	21.64	16.19	13.06
	Ca-PAT (ours)	22.33	20.62	19.71	18.13
PGD	Clean	-	2.61	1.16	0.97
	TRADES [12]	-	13.69	10.86	9.36
	LLR [13]	-	14.63	10.71	9.44
	RobustLib [30]	-	12.85	10.14	7.92
	Salman [31]	-	20.05	15.08	12.17
	Ca-PAT (ours)	-	19.07	17.97	17.12
Ens	Clean	-	1.96	0.98	0.58
	TRADES [12]	-	19.61	17.90	17.31
	LLR [13]	-	19.54	18.00	17.18
	RobustLib [30]	-	19.18	17.59	16.94
	Salman [31]	-	18.57	17.11	16.43
	Ca-PAT (ours)	-	19.02	17.44	16.83

5. Discussion and Future Work

Currently, we focus on perception models in autonomous vehicles. We are planning to conduct further research in two directions as follows: (i) extend the proposed training scheme to a multi-modality perception model, rather than relying solely on cameras; (ii) go beyond perception models and perform adversarial training on downstream tasks, such as multi-object tracking, trajectory prediction, and motion planning.

6. Conclusions

In this paper, we introduce a novel optimization mechanism called categorical-parallel optimization, designed to defend against perturbations by estimating and measuring both logits distortion and attention distortion of input images. We also present a comprehensive defense framework, termed Ca-PAT, which systematically optimizes parameters to enhance the robustness of large-scale DNNs deployed on single-board unmanned vehicles. Adversarial perception in large-scale DNNs has been relatively underexplored; thus, our work represents a significant advancement in developing a robust pipeline that offers both sophisticated adversarial metrics and a complete end-to-end training framework. Extensive experiments were conducted on various public datasets. Empirical results demonstrate that Ca-PAT outperforms state-of-the-art baselines in classification and object detection tasks. Notably, Ca-PAT shows a significant performance advantage, particularly in scenarios involving aggressive quantization strategies. Additionally, by effectively controlling logit deviations and attention distortion, our approach reduces the sensitivity of large-scale DNNs to perturbations and ensures stable prediction accuracy. The Ca-PAT defense framework is versatile and can be adapted for a wide range of applications involving single-board unmanned vehicles.

Author Contributions: Conceptualization, Y.L. (Yilan Li) and X.F.; software, Y.L. (Yilan Li); validation, S.S.; formal analysis, Y.L. (Yilan Li) and S.S.; investigation, X.F.; resources, Y.L. (Yantao Lu); original draft preparation, Y.L. (Yilan Li); supervision, N.L.; funding acquisition, Y.L. (Yilan Li). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 62106202, in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2022JQ-579, and in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2023-JC-QN-0741.

Data Availability Statement: The original data presented in the study are openly available in Imagenet-te at https://github.com/fastai/imagenette (accessed on 21 July 2024). No new data were created or analyzed in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Ca-PAT	categorical-parallel adversarial training
DNN	deep neural networks
SLAM	simultaneous localization and mapping
FLOPs	floating point operations

References

1. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
2. Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; Swami, A. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi, United Arab Emirates, 2–6 April 2017; pp. 506–519.
3. 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 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1625–1634.
4. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 99–112.
5. Buckman, J.; Roy, A.; Raffel, C.; Goodfellow, I. Thermometer encoding: One hot way to resist adversarial examples. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April 30–3 May 2018.
6. Huang, L.; Joseph, A.D.; Nelson, B.; Rubinstein, B.I.; Tygar, J.D. Adversarial machine learning. In Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence, Chicago, IL, USA, 21 October 2011; pp. 43–58.
7. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
8. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning, Vienna, Austria, 13–18 July 2020.
9. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the International Conference on Machine Learning, Stockholm Sweden, 10–15 July 2018; PMLR: Birmingham, UK, 2018; pp. 274–283.
10. Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J.; Studer, C.; Davis, L.S.; Taylor, G.; Goldstein, T. Adversarial training for free! *arXiv* **2019**, arXiv:1904.12843.
11. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
12. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: Birmingham, UK, 2019; pp. 7472–7482.
13. Qin, C.; Martens, J.; Gowal, S.; Krishnan, D.; Dvijotham, K.; Fawzi, A.; De, S.; Stanforth, R.; Kohli, P. Adversarial robustness through local linearization. *arXiv* **2019**, arXiv:1907.02610.
14. Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; Madry, A. Robustness may be at odds with accuracy. *arXiv* **2018**, arXiv:1805.12152.
15. Thrun, S. Simultaneous Localization and Mapping. In *Robotics and Cognitive Approaches to Spatial Mapping*; Jefferies, M.E., Yeap, W.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 13–41. [CrossRef]
16. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27 August 2010; pp. 177–186.
17. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

18. Dutta, S.; Jha, S.; Sankaranarayanan, S.; Tiwari, A. Output range analysis for deep feedforward neural networks. In Proceedings of the NASA Formal Methods Symposium, Newport News, VA, USA, 17–19 April 2018; Springer: Cham, Switzerland, 2018; pp. 121–138.
19. Ehlers, R. Formal verification of piece-wise linear feed-forward neural networks. In Proceedings of the International Symposium on Automated Technology for Verification and Analysis, Pune, India, 3–6 October 2017; Springer: Cham, Switzerland, 2017; pp. 269–286.
20. Ruan, W.; Huang, X.; Kwiatkowska, M. Reachability analysis of deep neural networks with provable guarantees. *arXiv* **2018**, arXiv:1805.02242.
21. Sun, S.; Zhang, Y.; Luo, X.; Vlantis, P.; Pajic, M.; Zavlanos, M.M. Formal Verification of Stochastic Systems with ReLU Neural Network Controllers. *arXiv* **2021**, arXiv:2103.05142.
22. Ghadimi, E.; Teixeira, A.; Shames, I.; Johansson, M. Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems. *IEEE Trans. Autom. Control* **2014**, *60*, 644–658. [CrossRef]
23. Fazlyab, M.; Robey, A.; Hassani, H.; Morari, M.; Pappas, G.J. Efficient and accurate estimation of lipschitz constants for deep neural networks. *arXiv* **2019**, arXiv:1906.04893.
24. Latorre, F.; Rolland, P.; Cevher, V. Lipschitz constant estimation of neural networks via sparse polynomial optimization. *arXiv* **2020**, arXiv:2004.08688.
25. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
26. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
27. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. *arXiv* **2019**, arXiv:1903.11027.
28. Hendrycks, D.; Dietterich, T. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
29. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
30. Engstrom, L.; Ilyas, A.; Salman, H.; Santurkar, S.; Tsipras, D. Robustness (Python Library), 2019. Volume 4, pp. 3–4. Available online: https://github.com/MadryLab/robustness (accessed on 21 July 2024).
31. Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; Madry, A. Do Adversarially Robust ImageNet Models Transfer Better? *arXiv* **2020**, arXiv:2007.08489.
32. Zagoruyko, S.; Komodakis, N. Wide residual networks. *arXiv* **2016**, arXiv:1605.07146.
33. Singla, S.; Feizi, S. Second-order provable defenses against adversarial attacks. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; PMLR: Birmingham, UK, 2020; pp. 8981–8991.
34. Zhang, H.; Chen, H.; Xiao, C.; Gowal, S.; Stanforth, R.; Li, B.; Boning, D.; Hsieh, C.J. Towards stable and efficient training of verifiably robust neural networks. *arXiv* **2019**, arXiv:1906.06316.
35. Wong, E.; Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 5286–5295.
36. Scout-2.0. Available online: https://www.iquotient-robotics.com/fuzhishouye.html (accessed on 22 May 2024).