Regularized Adversarial Training for Robust CNN Image Classification: Evaluation of ATWR, ATGR, and EATR Under White-Box Attacks

Thi Thanh Thuy Pham[1,2], Bao-Chau Ho[3], Huong-Giang Doan[4,*]
[1]Faculty of Cybersecurity and High Tech Crime Prevention, Academy of People Security, No. 125, Tran Phu, Van Quan, Ha Noi, Vietnam
[2]International Research Institute MICA, Hanoi University of Science and Technology, No. 1, Dai Co Viet, Bach Mai, Ha Noi, Vietnam
[3]School of Information and Communication Technology, Hanoi University of Science and Technology, No. 1, Dai Co Viet, Bach Mai, Ha Noi, Vietnam
[4]Faculty of Control and Automation, Electric Power University, No. 235, Hoang Quoc Viet, Nghia Do, Ha Noi, Vietnam
E-mail: thanh-thuy.pham@mica.edu.vn, baochaudoanho.hust@gmail.com, giangdth@epu.edu.vn
*Corresponding author

Keywords: Adversarial attack, adversarial training, regularization function, image classifier, convolution neural network

Received: January 2, 2025

Adversarial attacks pose serious challenges to the robustness of deep Convolutional Neural Networks (CNNs) in image classification. In this study, we evaluated the vulnerability of popular CNN models-ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, and InceptionNetV3-under white-box attacks, including FGSM, PGD, BIM, and C&W. Experiments are conducted on standard datasets such as MNIST, CIFAR-10, CIFAR-100, and ImageNet. To enhance model robustness, we propose three regularized adversarial training methods: ATWR (Adversarial Training with Weight Regularization), ATGR (Adversarial Training with Gradient Regularization), and EATR (Ensemble Adversarial Training with Regularization). Our results show that ATWR reduces the accuracy drop under the PGD attack on CIFAR-10 from 65.93% to 0.00%, and under C&W attack on MNIST from 100% to 0.31%. EATR achieves consistent robustness across all attacks and models, reducing the accuracy drop in CIFAR-10 (PGD) from 65.93% to 0%, while maintaining the classification accuracy within 10% of the original. ATGR, while reducing classification accuracy, enhances adversarial detection by amplifying the difference in output behavior under attack. The proposed methods strike varying trade-offs between robustness, generalization, and detectability. These findings offer practical guidance for securing deep CNNs against strong white-box adversarial threats. The source codes are available at: https://github.com/AdversarialAttack/DefenseAndAttack

Povzetek: Raziskava uvaja tri regularizirane metode za adversarialno učenje CNN-modelov, ki znatno izboljšajo robustnost proti belim napadom (FGSM, PGD, BIM, C&W) brez večje izgube osnovne točnosti.

1 Introduction

Convolutional Neural Networks (CNNs) have achieved significant success in image classification tasks. However, these models are highly susceptible to adversarial attacks: tiny, often imperceptible alterations to input images that can drastically alter a model's predictions without being noticeable to human observers [1]–[4]. This vulnerability poses serious security risks for real-world AI applications that rely on image classifiers, such as autonomous vehicles, medical imaging, and surveillance systems.

Several adversarial attack techniques have been proposed to exploit the vulnerabilities of deep image classifiers. Based on the attacker's knowledge of the target model, there are two types of adversarial attack strategies: black-box and white-box attacks. In a white-box scenario, the attacker has full access to the model architecture, parameters, loss function, and gradients, allowing the generation of highly effective perturbations [5]–[8]. In contrast, black-box attacks are based only on input-output observations [9], [10]. In such settings, attackers typically query the target model to collect input-output pairs, from which they train a surrogate (substitute) model that approximates the target model's behavior. Adversarial examples are then generated using white-box attacks on this surrogate model and transferred to the target model, leveraging the transferability property of adversarial examples. According to the mechanism used to generate perturbations, adversarial attacks can be classified into gradient-based (e.g., FGSM [11], PGD [12], BIM [13]) and optimization-based (e.g., C&W [8], DeepFool [7]). Although optimization-based attacks are more computationally expensive, they tend to be more successful, particularly in black-box scenarios.

To counter these threats, various defense techniques have been proposed. These include adversarial training [14]–

[16], gradient masking [17], [18], and defensive distillation [19], [20]. Among them, adversarial training remains the most widely adopted due to its empirical robustness against white-box attacks. However, it suffers from several limitations, including poor generalization to unseen attacks [21], overfitting to adversarial samples [22], and limited transferability [23]. Gradient masking is often easy to bypass [24], [25], while defensive distillation may result in gradient obfuscation and vulnerability to stronger optimization-based attacks.

To address these challenges, this paper explores whether incorporating regularization into adversarial training can mitigate accuracy degradation under strong white-box attacks without compromising clean accuracy. While previous works have focused on specific attack types or model architectures, we adopt a broader approach that evaluates multiple regularization strategies across different CNN models and datasets. In particular, our study is guided by the following research questions.

- **RQ1:** Can regularization-based adversarial training reduce accuracy drop (Acc-drop) under strong attacks such as PGD and C&W?
- **RQ2:** Can such defenses maintain or even improve clean accuracy while improving robustness?
- **RQ3:** How do different regularization approaches, weight-based, gradient-based, and ensemble-based, compare in robustness, generalization, and failure scenarios?

Based on these questions, we formulate the following hypotheses, which guide the design and evaluation of our defense strategies:

1. Weight-based regularization (ATWR) stabilizes parameter updates, helping retain clean accuracy while increasing robustness.
2. Gradient-based regularization (ATGR) promotes conservative prediction behavior, which is useful for attack detection but may reduce overall accuracy.
3. Ensemble adversarial training (EATR) introduces perturbation diversity, improving generalization across attacks and datasets.

To empirically validate these hypotheses, we conduct a series of controlled experiments on multiple CNN architectures and benchmark datasets. Our experimental design enables us to assess the robustness and generalization capabilities of different regularization strategies. The main contributions of this paper are as follows:

- We analyze the vulnerability of six popular CNN architectures (ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, and InceptionNetV3) under white-box attacks (FGSM, PGD, BIM, and C&W) across MNIST, CIFAR-10, CIFAR-100, and ImageNet.
- We propose three novel defense methods: ATWR (Adversarial Training with Weight Regularization), ATGR (Adversarial Training with Gradient Regularization), and EATR (Ensemble Adversarial Training with Regularization).
- We experimentally show that ATWR and EATR effectively reduce Acc-drop under PGD and C&W while preserving clean accuracy. ATGR offers conservative responses, making it suitable for adversarial detection rather than classification.
- We discuss the robustness-generalization trade-offs and analyze failure scenarios on shallow networks such as AlexNet.

The remainder of this paper is structured as follows: Section 2 reviews adversarial attacks and defenses. Section 3 presents our vulnerability analysis. Section 4 describes the proposed regularized training strategies. Section 5 evaluates their performance. Section 6 discusses key insights and limitations. Section 7 concludes and outlines future directions.

2 Related work

This section discusses adversarial attack methods and adversarial training-based defense strategies.

2.1 Adversarial attacks

Adversarial attacks have received extensive attention for exposing vulnerabilities in deep neural networks. The foundational work by Goodfellow et al. [11] introduced the Fast Gradient Sign Method (FGSM), which perturbs inputs using the sign of the loss gradient, in [26] used the FGSM attack to deep neural networks. Extensions like Projected Gradient Descent (PGD) [12] and Basic Iterative Method (BIM) [13] apply iterative refinements to generate stronger perturbations. Although BIM offers a lightweight enhancement over FGSM, PGD is widely considered a strong first-order adversary. The C&W attack [8] formulates the generation of adversarial examples as an optimization problem that minimizes the magnitude of the perturbation while ensuring misclassification. It remains one of the most effective methods against robust defenses.

The common feature of the above-mentioned attacks is that the attacker can compute gradients with respect to the model's input and use this information to generate adversarial examples. This shows the scenario of white-box adversarial attacks. In a black-box attack, the attacker has no direct access to the model's internal structure or gradients. However, the attacker can still query the target model and receive output (predictions), which are used to infer information about the model and create adversarial examples [27], [28]. Another approach of black-box adversarial attacks is attacking a surrogate model and using the adversarial examples that are created by this model on a target

Table 1: Comparison of adversarial defense methods: robustness, generalization, and limitations

Defense Method	Reg. Used	PGD Robustness	C&W Robustness	Generalization	CNN Models Used	Characteristics/Limitations
Standard Adv. Training (SAT) [32]	×	Moderate	Weak	Low	ResNet18, WideResNet	Overfits to PGD; Weak against optimization-based attacks like C&W
Defensive Distillation [19]	×	Low	Very Low	Low	LeNet, ResNet18	Gradient masking; Ineffective under strong white-box attacks
Ensemble Adv. Training (EAT) [30]	×	Strong	Moderate	Medium	Inception-v3, ResNet50	Complex training; Limited generalization
Feature Scattering [14]	×	Strong	Strong	Medium	WideResNet, ResNet50	High training cost; Model-specific configurations
Combined Adv. Training (CAT) [31]	×	Moderate	Moderate	Medium	ResNet50, InceptionNetV3	Uses adversarial examples from ensemble models; High computational cost
ATWR (Ours)	Weight	Strong	Strong	High	ResNet50, ResNet101	Slight accuracy trade-off on smaller models
ATGR (Ours)	Gradient	Moderate	Moderate	Medium	AlexNet, MobileNetV2	Lower accuracy; enhances adversarial detectability
EATR (Ours)	Weight/Gradient	Strong	Strong	High	DenseNet121, InceptionNetV3	Higher training cost due to ensemble strategy

model. This type of attack is called a transfer attack [29], [12] which is one of the key vulnerabilities in deep learning models and has significant implications for model security and robustness.

In this work, we focus on evaluating four representative white-box attacks (FGSM, BIM, PGD, and C&W) on prominent CNN architectures including ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, and InceptionNetV3.

2.2 Adversarial training defense solutions

Adversarial training is the most established defense strategy. The formulation based on PGD by Madry et al. [12] framed adversarial training as a min-max optimization problem and demonstrated its efficacy in MNIST and CIFAR-10. However, models trained in this way often overfit to specific attacks and may generalize poorly. Defensive distillation [19] attempted to obfuscate gradients via softened output, but was later shown to be vulnerable to gradient-free attacks. Ensemble Adversarial Training (EAT) [30] improved transfer robustness by training models with perturbations from multiple sources. The Combined Adversarial Training (CAT) approach [31] improves the robustness of the model by training adversarial examples generated from an ensemble of models, thus improving defense against black-box attacks. It offers better generalization across unseen attacks, but incurs higher computational cost due to multiple adversarial sources.

To better contextualize our contribution, we provide a comparative overview of the SOTA methods in Table 1. Standard Adversarial Training (SAT) offers moderate robustness, but lacks generalization. Defensive distillation suffers from gradient masking. EAT and feature scattering [14] improve robustness, but lack integrated regularization. Combined Adversarial Training (CAT) [31] leverages adversarial examples generated from ensemble models to improve robustness, but incurs a high computational cost and still achieves only moderate defense against strong attacks such as PGD and C&W.

A critical gap in existing defenses is the lack of explicit regularization during adversarial training, which can limit generalization to adaptive or unseen attacks. Our proposed approaches: ATWR (Adversarial Training with Weight Regularization), ATGR (Adversarial Training with Gradient Regularization), and EATR (Ensemble Adversarial Training with Regularization) are designed to bridge this gap. By integrating regularization into the adversarial training process, we achieve greater robustness and generalization across diverse CNN architectures and white-box attack scenarios.

3 Adversarial attack on deep image classifier

Consider an original image that is flattened into a vector $\mathbf{x} \in \mathcal{R}^n$, where n is the total number of pixels across all channels: $\mathbf{x} = \{x_1, x_2, .., x_n\}$, $x_i \in \mathcal{R}$ corresponds to the pixel value at the position i_{th} in the flattened image vector. The function $f(x)$ is the image classification model, and it outputs a logit vector, from which we can derive the predicted class label for the input image $\mathbf{x}$ is $\hat{y} \in \{0, 1, .., C\}$, C is the number of classes.

An adversarial example is a perturbed image $\tilde{\mathbf{x}}$, such that $\tilde{\mathbf{x}} = \mathbf{x} + \eta$, where η is a small perturbation added to the original image $\mathbf{x}$. The goal of an adversarial attack is to find a perturbation η that maximizes the probability of misclassification, that is, the attack should cause the model to predict the wrong class while keeping the perturbation η small.

Let $\hat{y}'$ be the predicted class label for the perturbed image $\tilde{\mathbf{x}} = \mathbf{x} + \eta$. The adversarial attack attempts to minimize the accuracy of the model in the perturbed image by maximizing the difference between the predicted class of the adversarial image $\hat{y}'$ and the true class label y. The goal is to find η such that:

- $\hat{y}' \neq y$ and $\|\eta\|$ is small. The constraint $\hat{y}' \neq y$ shows the case of an untargeted adversarial attack, in which the output of a classifier $f(\mathbf{x} + \eta)$ is any class other than the true class y of the input image $\mathbf{x}$;

- Or $\hat{y}' = y_t$ and $\|\eta\|$ is small. $\hat{y}' = y_t$ shows the case of a targeted adversarial attack, in which the output of a classifier $f(\mathbf{x} + \eta)$ is a specific target class y_t other than the true class y of the input image $\mathbf{x}$.

This can be expressed as the following optimization problem:

$$\max_{\eta} \mathcal{L}\left(f_\theta\left(\mathbf{x} + \eta\right), y\right) subjectto \|\eta\| \leq \epsilon : \text{untargeted attack} \quad (1)$$

or

$$\max_{\eta} \mathcal{L}\left(f_\theta\left(\mathbf{x}+\eta\right), y_t\right) subject to \|\eta\| \leq \epsilon \text{ : targeted attack} \tag{2}$$

Where $\mathcal{L}\left(f_\theta\left(\mathbf{x}+\eta\right), y\right)$ is a loss function that measures the difference between the predicted class $\hat{y}'$ and the true label y; θ is the model parameters. In deep image classification models, the loss function is often chosen as the cross-entropy loss.

The optimization problem mentioned above can be solved in two main approaches of gradient-based and optimization-based methods.

3.1 Gradient-based adversarial attack

In the gradient-based solution, x_i is changed in the direction of the steepest gradient to decrease the classification probability of $\mathbf{x}$ becoming C with negligible visual changes. FGSM is a simple gradient-based method with η proposed to be:

$$\eta = \epsilon \cdot sign(\nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right)) \tag{3}$$

where $\nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right) = \left(\frac{\partial \mathcal{L}}{\partial x_1}, .., \frac{\partial \mathcal{L}}{\partial x_n}\right)$, y is the ground truth label, and

$$sign(\nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right)) = \begin{cases} -1 & \text{if } \nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right) < 0 \\ 0 & \text{if } \nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right) = 0 \\ 1 & \text{if } \nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right) > 0 \end{cases}$$

The perturbation noise η is then added to the original input $\mathbf{x}$ to make sure it does not differ by more than ϵ and this helps to push the model's prediction towards an incorrect classification:

$$\tilde{\mathbf{x}} = \mathbf{x} + \eta = \mathbf{x} + \epsilon \cdot sign\left(\nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y\right)\right) \tag{4}$$

Equation.(4) shows the case of an untargeted FGSM attack. For the targeted FGSM (T-FGSM), the sample $\mathbf{x}$ will be misclassified as a specific label rather than just a mislabel. Therefore, the loss function $\mathcal{L}$ can be calculated with respect to the target label y_t instead of y:

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \cdot sign\left(\nabla_x \mathcal{L}\left(f_\theta\left(\mathbf{x}\right), y_t\right)\right) \tag{5}$$

Other extensions of the FGSM method are BIM and PGD attacks. BIM applies FGSM repeatedly to an image with step size α presenting the change in pixel value per iteration. The resulting adversary image can then be clipped to limit the maximum perturbation for each pixel:

$$\tilde{\mathbf{x}}^{N+1} = \mathbf{Clip}_{[0,1]}\left\{\tilde{\mathbf{x}}^N + \alpha \cdot sign\left(\nabla_x \mathcal{L}\left(f_\theta\left(\tilde{\mathbf{x}}^N\right), y\right)\right)\right\} \tag{6}$$

where N is the number of iterations and original pixels x_i is used for initialization in iteration N=0: $\tilde{\mathbf{x}}^0 = \mathbf{x}$.

where N is the iteration index, and $\tilde{\mathbf{x}}^0 = \mathbf{x}$.

The PGD attack extends BIM by incorporating a random initialization and explicit projection onto an ϵ-ball. The ϵ-ball, denoted $\mathcal{B}_\epsilon(\mathbf{x})$, is defined as the set:

$$\mathcal{B}_\epsilon(\mathbf{x}) = \{\tilde{\mathbf{x}} \mid \|\tilde{\mathbf{x}} - \mathbf{x}\|_p \leq \epsilon\}$$

where $\|\cdot\|_p$ is an ℓ_p norm, typically ℓ_∞ or ℓ_2. The PGD update rule is:

$$\tilde{\mathbf{x}}^{N+1} = \Pi_{\mathcal{B}_\epsilon(\mathbf{x})}\left(\tilde{\mathbf{x}}^N + \alpha \cdot \text{sign}\left(\nabla_x \mathcal{L}(f_\theta(\tilde{\mathbf{x}}^N), y)\right)\right) \tag{7}$$

where $\Pi_{\mathcal{B}_\epsilon(\mathbf{x})}(\cdot)$ is the projection operator that ensures $\tilde{\mathbf{x}}$ stays within the ϵ-ball. In the case of ℓ_∞, the projection becomes a per-pixel clipping function (as in BIM). For ℓ_2, the projection maps $\tilde{\mathbf{x}}$ back onto the surface of a hypersphere centered at $\mathbf{x}$ with radius ϵ:

$$\Pi_\epsilon^{\ell_2}(\tilde{\mathbf{x}}) = \mathbf{x} + \frac{\min\left(\|\tilde{\mathbf{x}} - \mathbf{x}\|_2, \epsilon\right)}{\|\tilde{\mathbf{x}} - \mathbf{x}\|_2}(\tilde{\mathbf{x}} - \mathbf{x}) \tag{8}$$

This formulation ensures that the adversarial examples are constrained within a norm-bounded neighborhood of the original input, making the perturbation imperceptible yet effective.

3.2 Optimization-based adversarial attack

C&W adversarial attack is a typical representative of the optimization-based adversarial attack approach. The objective is to minimize the perturbation, subject to the constraint that the perturbed input is misclassified by the target model. The general optimization problem for the C&W attack can be formulated as:

$$\min_{\eta} \|\eta\|_p + c \cdot \text{Loss}\left(\eta\right) \tag{9}$$

where $\|\eta\|_p$ is the chosen norm of the perturbation (l_2 or l_∞ norm); c is a hyperparameter controlling the trade-off between minimizing the perturbation and maximizing the adversarial effect; $\text{Loss}\left(\eta\right)$ is the classification loss with the goal of creating a perturbation η such that the perturbed image $\tilde{\mathbf{x}} = \mathbf{x} + \eta$ is misclassified by the model but still looks similar to the original image. The loss function $\text{Loss}\left(\eta\right)$ that is minimized during the attack can be written for an untargeted attack as:

$$\text{Loss}_{untargeted}\left(\eta\right) = \max\left(0, f\left(x+\eta\right)_y - \max_{i \neq y} f\left(x+\eta\right)_i\right) + \lambda \|\eta\|_2^2 \tag{10}$$

and the loss function for targeted attack as:

$$\text{Loss}_{targeted}\left(\eta\right) = \max\left(0, f\left(x+\eta\right)_t - \max_{i \neq t} f\left(x+\eta\right)_i\right) + \lambda \|\eta\|_2^2 \tag{11}$$

where:

- $f(x+\eta)_y$ is the logit for the true class y after perturbation;

- $f(x+\eta)_t$ is the logit for the target class t after perturbation;

- $f(x+\eta)_i$ is the logit for any class $i \neq y$ or $i \neq t$;

- $\max_{i \neq y} f(x+\eta)_i$ is the maximum logit for any class $i \neq y$, representing the logit of the most likely incorrect class;

- $\max_{i \neq t} f(x+\eta)_i$ is the maximum logit for any class $i \neq t$;

- $\max\left(0, f(x+\eta)_y - \max_{i \neq y} f(x+\eta)_i\right)$ ensures that the model misclassifies the perturbed input into a class that is not the original class y. Forcing the logit for the original class y to be smaller than the logit for at least one other incorrect class;

- $\max\left(0, f(x+\eta)_t - \max_{i \neq t} f(x+\eta)_i\right)$ ensures that the model classifies the perturbed input as the target class t, and not as any other class. This forces the target class logit to be greater than all other logits for the perturbed image;

- λ is a regularization constant that controls the trade-off between misclassification and perturbation size;

- $\|\eta\|_2^2$ is the L2 norm squared of the perturbation, which controls how large the perturbation can be;

- $\lambda \|\eta\|_2^2$ ensures that the perturbation is small, making the changes imperceptible to humans.

4 Adversarial defense with regularized adversarial training

We propose three variations of adversarial training that integrate regularization to enhance model robustness: (1) Adversarial Training with Weight Regularization (ATWR), (2) Adversarial Training with Gradient Regularization (ATGR), and (3) Ensemble Adversarial Training with Regularization (EATR). In all approaches, adversarial examples are incorporated into the training process alongside clean samples and regularization constraints.

In ATWR and ATGR, adversarial examples are generated using a single white-box attack (e.g., FGSM, PGD, BIM, or C&W). In contrast, EATR generates adversarial examples from multiple attacks on-the-fly, using ensemble perturbations during training. In order to ensure stable learning and improved generalization, all CNN models are first pre-trained on clean data before being fine-tuned with adversarial samples.

To support reproducibility, pseudocode and a training pipeline diagram for all three strategies are provided in the Appendix.

4.1 Adversarial training with weight regularization (ATWR)

This method integrates weight decay with adversarial training through the following composite loss:

$$\mathcal{L}_{\text{ATWR}} = \mathcal{L}_{\text{clean}} + \lambda_1 \mathcal{L}_{\text{adv}} + \lambda_2 \|\mathbf{W}\|_2^2 \tag{12}$$

Here, $\mathcal{L}_{\text{clean}}$ is the standard loss on clean data, $\mathcal{L}_{\text{adv}}$ is the loss on adversarial examples, and $\|\mathbf{W}\|_2^2$ is the weight regularization term (weight decay), which penalizes large parameter magnitudes. This encourages simpler models and enhances generalization.

4.2 Adversarial training with gradient regularization (ATGR)

ATGR penalizes large input gradients to reduce sensitivity to perturbations:

$$\mathcal{L}_{\text{ATGR}} = \mathcal{L}_{\text{clean}} + \beta_1 \mathcal{L}_{\text{adv}} + \beta_2 \|\nabla_{\tilde{\mathbf{x}}} \mathcal{L}_{\text{adv}}(\tilde{\mathbf{x}})\|_2^2 \tag{13}$$

where $\|\nabla_{\tilde{\mathbf{x}}} \mathcal{L}_{\text{adv}}(\tilde{\mathbf{x}})\|_2^2$ is gradient regularization, which suppresses sharp changes in the loss landscape and promotes smoother decision boundaries. Although this often reduces classification accuracy, it makes the model highly sensitive to adversarial inputs, offering the potential for adversarial detection. [1]

4.3 Ensemble adversarial training with regularization (EATR)

EATR leverages multiple attacks during training and incorporates a regularization term:

$$\begin{aligned}\mathcal{L}_{\text{EATR}} =& \mathcal{L}_{\text{clean}} + \gamma_1 \mathcal{L}_{\text{adv}}^{\text{FGSM}} + \gamma_2 \mathcal{L}_{\text{adv}}^{\text{PGD}} + \gamma_3 \mathcal{L}_{\text{adv}}^{\text{BIM}} \\ &+ \gamma_4 \mathcal{L}_{\text{adv}}^{\text{C\&W}} + \gamma_5 \mathcal{L}_{\text{reg}}\end{aligned} \tag{14}$$

Where:

$$\mathcal{L}_{\text{clean}}(f(x_i), y_i) = -\sum_{c=1}^{C} y_i^c \log(f(x_i)^c) \tag{15}$$

Each $\mathcal{L}_{\text{adv}}^{\text{attack}}$ is a cross-entropy loss over adversarial examples $\tilde{x}_i^{\text{attack}}$:

$$\mathcal{L}_{\text{adv}}^{\text{attack}}(f(\tilde{x}_i), y_i) = -\sum_{c=1}^{C} y_i^c \log(f(\tilde{x}_i)^c) \tag{16}$$

Adversarial examples are generated as follows:

$$\tilde{x}_i^{\text{FGSM}} = x_i + \epsilon \operatorname{sign}(\nabla_x \mathcal{L}(f_\theta(x_i), y_i)) \tag{17}$$

[1] While ATGR may degrade classification accuracy, its tendency to produce low-confidence predictions on perturbed inputs can be leveraged to distinguish adversarial examples from clean ones. This makes ATGR useful for adversarial input detection. In particular, robust detection itself can be a valuable line of defense in security-critical applications, as early identification and rejection of adversarial samples can prevent harmful decisions downstream.

$$\tilde{x}_i^{\text{PGD}} = x_i + \alpha \operatorname{sign}\left(\nabla_x \mathcal{L}\left(f_\theta\left(x_i\right), y_i\right)\right) \tag{18}$$

$$\begin{aligned} \tilde{x}_i^{\text{BIM}}(t+1) = \operatorname{clip}_{x_i,\epsilon}\Big(\tilde{x}_i^{\text{BIM}}(t) \\ +\alpha \cdot \operatorname{sign}\big(\nabla_x \mathcal{L}\big(f_\theta(\tilde{x}_i^{\text{BIM}}(t)), y_i\big)\big)\Big) \end{aligned} \tag{19}$$

$$\begin{aligned} \tilde{x}_i^{\text{C\&W}} = \arg\min_{\tilde{x}^{\text{C\&W}}} \left\|\tilde{x}_i^{\text{C\&W}} - x_i\right\|^2 \\ +\lambda \mathcal{L}\left(f_\theta\left(\tilde{x}_i^{\text{C\&W}}\right), y_i\right) \end{aligned} \tag{20}$$

$\mathcal{L}_{\text{reg}}$ can represent either weight decay ($\|\mathbf{W}\|_2^2$) or gradient regularization ($\|\nabla_{\tilde{\mathbf{x}}} \mathcal{L}_{\text{adv}}(\tilde{\mathbf{x}})\|_2^2$), depending on the variant implemented. This modular structure enables flexibility between generalization and robustness.

5 Experiments and results

5.1 Experimental dataset and evaluation metric

5.1.1 Experimental datasets

In this work, the MNIST [33], CIFAR10, CIFAR100 [34], and ImageNet [35] datasets are used in our experiments. The MNIST dataset consists of images of handwritten digits (0–9). Each image is a 28x28 pixel grayscale image and is labeled with the correct digit (from 0 to 9). The data set contains a total of 70,000 images, and the data split for training is 60,000 images and for testing is 10,000 images. The CIFAR-10 dataset consists of 60,000 color images in 10 different classes. Each image is 32x32 pixels with 3 color channels of RGB. In this work, the training set contains 50,000 images, and the testing set includes 10,000 images. The CIFAR-100 dataset is an extended version of the CIFAR-10 dataset, which contains 100 classes, each with 600 RGB images with a resolution of 32x32 pixels, totaling 60,000 images in the dataset. The 100 classes are divided into 20 superclasses, and each superclass contains 5 subclasses. In this work, the CIFAR-100 dataset is divided into a training set of 50,000 images and a test set of 10,000 images. The images are divided equally among the 100 classes, with 500 images per class for training and 100 images per class for testing. The ImageNet dataset is one of the largest and most widely used datasets for image classification tasks. It contains more than 14 million images that are labeled according to a large set of categories (more than 20,000 in total). The images in the dataset vary in resolution and are both low- and high-resolution images, with a wide range of perspectives, lighting conditions, and occlusions. In this work, we used a subset of the ImageNet dataset known as the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), which contains 1.2 million images for 1,000 categories. This subset is taken from the test set of the ImageNet dataset [35]. It contains 100,000 images that are also distributed in 1,000 categories.

5.1.2 Evaluation metrics

In order to evaluate the adversarial attack robustness and defense, the evaluation metric of Accuracy Drop (Acc_{drop}) is used for the experiments. This metric measures the performance degradation of the image classifiers when subjected to adversarial examples:

$$\text{Acc}_{\text{drop}}\,(\%) = \text{Acc}_{\text{origin}}\,(\%) - \text{Acc}_{\text{adv}}\,(\%) \tag{21}$$

where $\text{Acc}_{\text{origin}}$ is the classification accuracy on the original (clean) data, and Acc_{adv} is the accuracy of the classifiers on the adversarial data. High accuracy drop means low robustness, or the target model is more vulnerable to adversarial attacks. Low accuracy drop indicates high robustness, or the target model is less vulnerable to adversarial attack. The $\text{Acc}_{\text{origin}}$ and Acc_{adv} are evaluated using the top-1 accuracy metric. Top-1 accuracy represents the precision of the model's highest-probability prediction:

$$\text{Acc}_{\text{Top-1}}\,(\%) = \frac{\text{Number of correct Top-1 predictions}}{\text{Total number of predictions}} \tag{22}$$

5.2 Configuration parameters

- The loss function is the cross-entropy loss. The image classification models are ResNet50 [36], ResNet101 [36], AlexNet [37], MobileNetV2 [38], DenseNet121 [39], InceptionNetV3 [40]. These models were selected to represent a diverse range of architectures in terms of depth, width, and computational complexity. This allows us to evaluate the generalizability of our defense methods across both lightweight and heavyweight CNNs, covering real-world deployment scenarios.

Configuration parameters for adversarial attacks:

- FGSM: The loss function is the Cross Entropy Loss; $\epsilon = 0.1$.
- PGD: The loss function is the Cross Entropy Loss; number of epochs = 100; $\alpha = 2/255$ for ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121; $\alpha = 2/299$ for InceptionNetV3; $\epsilon = 0.1$.
- BIM: The loss function is the Cross Entropy Loss; $\alpha = 2/255$ for ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121; $\alpha = 2/299$ for InceptionNetV3; $\epsilon = 0.1$.
- C&W: The confidence parameter $c = 1.0$; Search step = 1; Maximum step = 100; Optimizer learning rate = 0.001.

Configuration parameters for defense solutions:

- ATWR: Learning rate 0.001; number of epochs: 100-500; $\lambda_1 = 1$, $\lambda_2 = 0.005$. These values were determined through a grid search in a validation split to maintain a balance between adversarial robustness and weight simplicity.

– ATGR: Learning rate 0.001; number of epochs: 100-500; $\beta_1 = 1$, $\beta_2 = 1$. This configuration encourages conservative learning and strong gradient suppression for improved attack detectability.

– EATR: Learning rate 0.001; number of epochs: 100-500; $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 1$; $\gamma_5 = 0.001$. These ensemble weights were uniformly initialized and shown through sensitivity analysis to yield stable performance across datasets and attacks.

5.3 Computational environment

All experiments were conducted using a system with the following specifications: Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz, 48 GB RAM, and NVIDIA Corporation Device 2504 (rev a1), running on Ubuntu 20.04.6 LTS, GNU/Linux 5.4.0-208-Generic x86_64. The models were implemented in Python 3.9 using PyTorch 2.2.0 and CUDA 12.4. We fixed random seeds (42) for NumPy and PyTorch to ensure deterministic behavior and reproducibility.

5.4 Experimental results

In this section, we evaluate the performance of the target models in two scenarios: (1) without any adversarial defense, and (2) with adversarial defense. In the first scenario, adversarial attacks such as FGSM, PGD, BIM, and C&W are used to generate adversarial samples, and the target models are tested for their robustness against these perturbations. In the second scenario, we apply three defense strategies-ATWR, ATGR, and EATR-to evaluate how well target models can withstand adversarial attacks.

5.4.1 Model robustness without adversarial defense

Adversarial attacks, including FGSM, BIM, PGD, and C&W, are applied to widely used image classification models: ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, and InceptionNetV3. These attacks generate adversarial samples using the MNIST, CIFAR-10, CIFAR-100, and ImageNet datasets.

Table 2 presents the classification results of the target models in the original images (**Acc-origin**) and the adversarial images (**Acc-adv**) generated by the FGSM, PGD, BIM, and C&W attacks. (**Acc-drop**) indicates the decrease in image classifier performance when exposed to adversarial examples. Taking into account the MNIST and ImageNet datasets, the target models exhibit a higher vulnerability to the C&W attack compared to other attacks, with the highest **Acc-drop** of 100% across all models, except MobileNetV2, which shows a 76.56% **Acc-drop** on the MNIST dataset. For the ImageNet dataset, all models experience an **Acc-drop** above 90%, except AlexNet, which shows a 79.69% **Acc-drop**.

Evaluations of the CIFAR-10 and CIFAR-100 datasets reveal that target models are less vulnerable to the C&W attack, while the PGD attack has the most significant adversarial impact on all models. The **Acc-drop** for PGD attacks ranges from a minimum of 48.12% (InceptionNetV3) to a maximum of 65.93% (MobileNetV2) on the CIFAR-10 dataset, and from 44.06% (AlexNet) to 63.75% (MobileNetV2) on the CIFAR-100 dataset.

Figures 1 and 2 display adversarial samples generated from MNIST, CIFAR-10, CIFAR-100, and ImageNet using FGSM, PGD, BIM, and C&W attacks on the target models: ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, and InceptionNetV3. Upon inspection of these adversarial images, it is apparent to the human eye that the images generated by PGD and C&W attacks are more similar to the original images compared to those generated by the FGSM and BIM attacks. This indicates that the perturbations introduced by PGD and C&W are less noticeable to the human eye. Despite their imperceptibility, these adversarial samples still lead to misclassification in the target models. This is evident in Table 2, where the models show low accuracy in adversarial examples (**Acc-adv**) or experience a significant accuracy drop (**Acc-drop**) for the C&W attack on MNIST and ImageNet, and for the PGD attack on CIFAR-10 and CIFAR-100 datasets.

Figures 1 and 2 illustrate adversarial examples generated by FGSM, PGD, BIM, and C&W attacks across multiple datasets (MNIST, CIFAR-10, CIFAR-100, ImageNet) and CNN architectures (ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, InceptionNetV3). Visual inspection reveals that PGD and C&W attacks produce perturbations that are almost imperceptible to the human eye, as evidenced by their consistently higher PSNR values (e.g., > 30 dB on average), compared to FGSM and BIM, whose perturbations are more visually prominent and produce lower PSNR. Despite their subtle appearance, these attacks are highly effective in deceiving models, as shown by the substantial accuracy drops reported in Table 2. For instance, PGD causes accuracy degradation up to 65.93% on CIFAR-10 (MobileNetV2), while C&W results in 100% drop on MNIST (ResNet50) and over 90% on ImageNet (multiple models). This demonstrates the deceptive strength of optimization-based attacks like C&W and highlights the necessity of defense methods capable of countering both visually obvious and imperceptible adversarial threats.

5.4.2 Model robustness with adversarial defense

In this section, we evaluate the robustness of target models using three defense strategies: ATWR, ATGR, and EATR. The experiments are conducted on the MNIST, CIFAR-10, and CIFAR-100 datasets. Target models are trained in both adversarial and clean images, and their performance is evaluated on clean images (**Acc-clean**) and adversarial images (**Acc-adv**).

Table 3 shows that ATWR significantly improves robustness, particularly under strong white-box, untargeted attacks such as PGD and C&W. In contrast, FGSM and BIM see varied performance due to their simpler pertur-

Table 2: Classification accuracy (%) of CNN models on clean and adversarial images under four white-box, untargeted attacks: FGSM, PGD, BIM, and C&W. No defense mechanism is applied. The accuracy drop (Acc-drop) is defined as the difference between the clean accuracy (Acc-origin) and the adversarial accuracy (Acc-adv).

Dataset	CNN model	Acc-origin (Top-1)	FGSM		PGD		BIM		C&W	
			Acc-adv (Top-1)	Acc-drop	Acc-adv (Top-1)	Acc-drop	Acc-adv (Top-1)	Acc-drop	Acc-adv (Top-1)	Acc-drop
MNIST	ResNet50	100.00	62.50	37.50	21.88	78.12	25.00	75.00	0.00	**100.00**
	ResNet101	100.00	45.31	54.69	7.81	92.19	23.44	76.56	0.00	**100.00**
	AlexNet	100.00	64.06	35.94	100.00	0.00	29.69	70.31	23.44	**76.56**
	MobileNetV2	100.00	82.81	17.19	70.31	29.69	25.00	75.00	0.00	**100.00**
	DenseNet121	100.00	59.38	40.62	60.94	39.06	20.31	79.69	0.00	**100.00**
	InceptionNetV3	100.00	75.00	25.00	29.69	70.31	18.75	81.25	0.00	**100.00**
CIFAR10	ResNet50	93.13	61.25	31.88	43.75	**49.38**	47.82	45.31	84.11	9.02
	ResNet101	95.63	64.38	31.25	46.88	**48.75**	45.00	50.63	82.43	13.20
	AlexNet	89.06	30.94	58.12	37.19	**51.87**	21.88	67.18	78.31	10.75
	MobileNetV2	92.19	48.13	44.06	26.26	**65.93**	40.94	51.25	84.58	7.61
	DenseNet121	93.75	58.44	35.31	43.13	**50.62**	42.50	51.25	81.08	12.67
	InceptionNetV3	78.13	39.07	39.06	30.01	**48.12**	33.44	44.69	69.39	8.74
CIFAR100	ResNet50	77.19	37.50	39.69	22.19	**55.00**	26.57	50.62	67.82	9.37
	ResNet101	84.69	41.57	43.12	32.19	**52.50**	32.50	52.19	75.16	9.53
	AlexNet	67.50	14.69	52.81	23.44	**44.06**	17.19	50.31	61.41	6.09
	MobileNetV2	80.94	30.01	50.93	17.19	**63.75**	25.30	55.64	69.06	11.88
	DenseNet121	81.56	32.82	48.74	25.63	**55.93**	27.19	54.37	72.11	9.45
	InceptionNetV3	79.69	39.69	40.00	30.00	**49.69**	34.07	45.62	68.91	10.78
ImageNet	ResNet50	95.31	82.81	12.50	76.56	18.75	42.19	53.12	0.00	**95.31**
	ResNet101	93.75	90.62	3.13	85.94	7.81	53.12	40.63	0.00	**93.75**
	AlexNet	79.69	23.44	56.25	71.88	7.81	0.00	79.69	0.00	**79.69**
	MobileNetV2	92.19	65.62	26.57	60.94	31.25	9.38	82.81	0.00	**92.19**
	DenseNet121	93.75	62.50	31.25	81.25	12.50	3.12	90.63	0.00	**93.75**
	InceptionNetV3	90.62	56.25	34.37	81.25	9.37	4.69	85.93	0.00	**90.62**

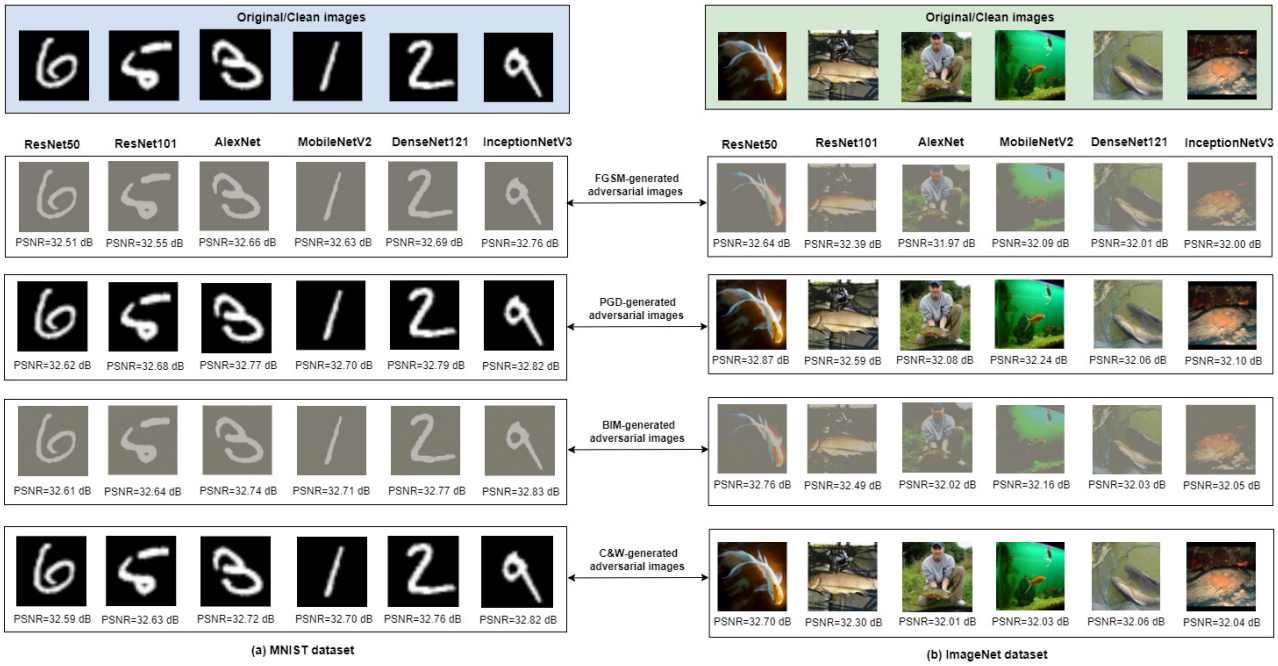

Figure 1: Visualization of adversarial examples generated by four attack methods (FGSM, PGD, BIM, and C&W) across six CNN architectures (ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, InceptionNetV3). Subfigure (a) shows results on the MNIST dataset, while subfigure (b) shows results on ImageNet dataset. The top row in each subfigure contains original (clean) images, while the rows below show adversarial versions organized by attack method and model. Peak Signal-to-Noise Ratio (PSNR) values, indicating the similarity between adversarial and clean images, are shown beneath each adversarial image.

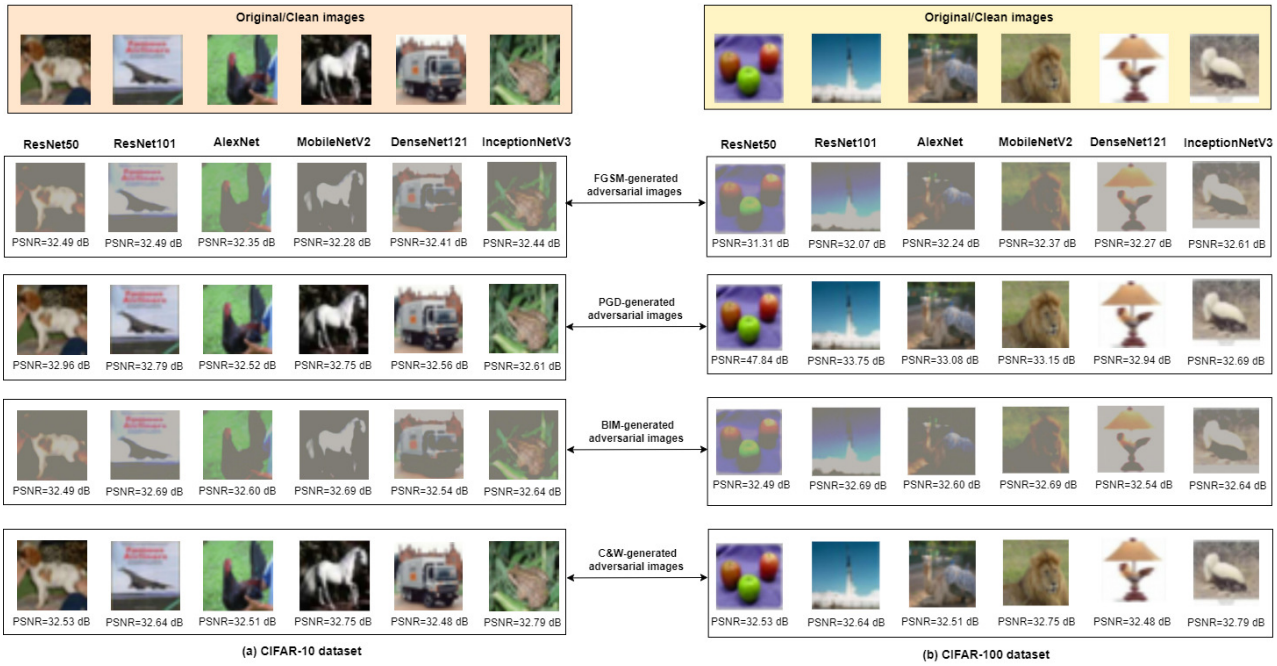

Figure 2: Visualization of adversarial examples generated by four attack methods (FGSM, PGD, BIM, and C&W) across six CNN architectures (ResNet50, ResNet101, AlexNet, MobileNetV2, DenseNet121, InceptionNetV3). Subfigure (a) shows results on the CIFAR-10 dataset, while subfigure (b) shows results on CIFAR-100. The top row in each subfigure contains original (clean) images, while the rows below show adversarial versions organized by attack method and model. Peak Signal-to-Noise Ratio (PSNR) values, indicating the similarity between adversarial and clean images, are shown beneath each adversarial image.

Table 3: Classification accuracy (%) with Adversarial Training using Weight Regularization (ATWR) against FGSM, PGD, BIM, and C&W attacks. ATWR improves robustness while preserving clean accuracy. All attacks are white-box and untargeted. The accuracy drop (Acc-drop) is defined as the difference between the clean accuracy (Acc-origin) and the adversarial accuracy (Acc-adv).

Dataset	CNN model	Acc-origin (Top-1)	FGSM			PGD			BIM			C&W		
			Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop
MNIST	ResNet50	100.00	98.75	6.56	92.19	**98.75**	**98.75**	**0.00**	98.75	0.00	98.75	**98.75**	**98.75**	**0.00**
	ResNet101	100.00	99.06	25.00	74.06	**99.06**	**99.06**	**0.00**	99.06	0.00	99.06	**99.06**	**98.75**	**0.31**
	AlexNet	100.00	13.44	13.44	0.00	**13.44**	**13.44**	**0.00**	13.44	13.44	0.00	**13.44**	**13.44**	**0.00**
	MobileNetV2	100.00	99.06	69.38	29.68	**99.06**	**99.06**	**0.00**	99.06	1.56	97.50	**99.06**	**97.81**	**1.25**
	DenseNet121	100.00	99.06	34.69	64.37	**99.06**	**99.06**	**0.00**	99.06	0.00	99.06	**99.06**	**98.75**	**0.31**
	InceptionNetV3	100.00	98.44	81.88	16.56	**98.44**	**98.44**	**0.00**	98.44	0.00	98.44	**98.44**	**97.19**	**1.25**
CIFAR10	ResNet50	93.13	87.81	18.44	69.37	**87.81**	**87.50**	**0.31**	87.81	0.00	87.81	87.81	0.00	87.81
	ResNet101	95.63	85.31	16.88	68.43	**85.31**	**85.31**	**0.00**	85.31	0.00	85.31	85.31	0.00	85.31
	AlexNet	89.06	12.50	12.50	0.00	**12.50**	**12.50**	**0.00**	12.50	12.50	0.00	12.50	12.50	0.00
	MobileNetV2	92.19	85.62	15.62	70.00	**85.62**	**85.62**	**0.00**	85.62	3.44	82.18	85.62	22.50	63.12
	DenseNet121	93.75	85.62	18.75	66.87	**85.62**	**85.62**	**0.00**	85.62	4.69	80.93	85.62	31.25	54.37
	InceptionNetV3	78.13	87.19	16.88	70.31	**87.19**	**87.19**	**0.00**	87.19	2.19	85.00	87.19	30.00	57.19
CIFAR100	ResNet50	77.19	66.88	10.94	55.94	**66.88**	**66.88**	**0.00**	66.88	0.00	66.88	66.88	0.00	66.88
	ResNet101	84.69	65.31	7.19	58.12	**65.31**	**65.31**	**0.00**	65.31	0.00	65.31	65.31	0.00	65.31
	AlexNet	67.50	38.44	9.69	28.75	**38.44**	**38.44**	**0.00**	38.44	0.00	38.44	38.44	4.06	34.38
	MobileNetV2	80.94	67.81	7.81	60.00	**67.81**	**67.19**	**0.62**	67.81	0.00	67.81	67.81	0.00	67.81
	DenseNet121	81.56	69.06	7.81	61.25	**69.06**	**69.06**	**0.00**	69.06	0.00	69.06	69.06	0.31	68.75
	InceptionNetV3	79.69	70.94	9.69	61.25	**70.94**	**70.94**	**0.00**	70.94	0.00	70.94	70.94	0.00	70.94

Table 4: Classification accuracy (%) under four white-box, untargeted attacks: FGSM, PGD, BIM, and C&W, using Adversarial Training with Gradient Regularization (ATGR). The method prioritizes stability and detection; low adversarial accuracy reflects its conservative design. The accuracy drop (Acc-drop) is defined as the difference between the clean accuracy (Acc-origin) and the adversarial accuracy (Acc-adv).

Dataset	CNN model	Acc-origin (Top-1)	FGSM			PGD			BIM			C&W		
			Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop
MNIST	ResNet50	100.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00
	ResNet101	100.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00
	AlexNet	100.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00
	MobileNetV2	100.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00
	DenseNet121	100.00	98.12	23.12	75.00	98.12	23.12	75.00	98.12	23.12	75.00	98.12	23.12	75.00
	InceptionNetV3	100.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00	13.44	13.44	0.00
CIFAR10	ResNet50	93.13	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00
	ResNet101	95.63	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00
	AlexNet	89.06	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00
	MobileNetV2	92.19	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00
	DenseNet121	93.75	75.00	18.12	56.88	75.00	75.00	0.00	75.00	8.75	66.25	75.00	25.31	49.69
	InceptionNetV3	78.13	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00	9.38	9.38	0.00
CIFAR100	ResNet50	77.19	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00
	ResNet101	84.69	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00
	AlexNet	67.50	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00
	MobileNetV2	80.94	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DenseNet121	81.56	54.38	7.19	47.19	54.38	54.38	0.00	54.38	5.00	49.38	54.38	12.50	41.88
	InceptionNetV3	79.69	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00	0.94	0.94	0.00

bation strategy. Most models maintain **Acc-drop** close to 0%, especially in the PGD and C&W attacks, which demonstrates strong resistance.

Regarding the trade-off between clean accuracy and robustness, we observe that most models achieve a good balance in MNIST and CIFAR-10. Their **Acc-clean** values are only slightly below the **Acc-origin**, with ATWR ensuring minimal degradation. On CIFAR-100, the accuracy gap is larger - expected due to the complexity of the data set - but still manageable. However, AlexNet consistently shows poor results in all datasets with extremely low precision. This inconsistency suggests that weight regularization may overly constrain shallow architectures like AlexNet, hindering their ability to learn robust representations. This also implies that ATWR requires depth-dependent hyperparameter tuning.

Table 4 presents the results for ATGR, where **Acc-clean** and **Acc-adv** values are nearly identical for each model, resulting in nearly zero **Acc-drop**. Although absolute accuracy is lower than with ATWR, this pattern suggests strong attack detectability due to conservative predictions. DenseNet121 is an exception, where clean accuracy is relatively high but robustness varies. **ATGR**'s behavior aligns with its design: suppressing the sensitivity of the input gradient encourages models that perform poorly on clean data but respond dramatically to adversarial perturbations, making them viable for detection purposes.

Finally, Table 5 shows that EATR provides a well-rounded improvement in robustness. For MNIST, it helps even AlexNet reach high adversarial accuracy. On CIFAR-10 and CIFAR-100, EATR improves robustness over all four attacks, although at a cost to clean accuracy. However, this trade-off remains preferable compared to ATWR in scenarios where robustness across various attacks is critical. For AlexNet, the accuracy drop under EATR is substantial, similar to ATGR, but can be leveraged for adversarial input identification.

These results collectively highlight that while ATWR is best for preserving clean accuracy under strong attacks, ATGR and EATR offer complementary benefits-either for detection (ATGR) or ensemble generalization (EATR).

These experimental insights motivate a deeper analysis of defense effectiveness, which is further elaborated in the following section 6.

6 Discussion

This section provides a comparative analysis of the proposed defense strategies, ATWR, ATGR, and EATR, against baseline adversarial training and other state-of-the-art methods reviewed in Section 2. We highlight key performance trends, justify method-specific behaviors under different attacks, and discuss observed limitations and trade-offs.

6.1 Comparison with existing methods

Tables 6 and Table 7 present a quantitative comparison between our proposed methods (ATWR, EATR) and the state-of-the-art adversarial defenses, including Standard Adversarial Training (SAT) [32], Defensive Distillation (DD) [19], and Combined Adversarial Training (CAT) [31], across three datasets: MNIST, CIFAR-10, and CIFAR-100, all evaluated on ResNet50.

Under PGD attack:

- In CIFAR-10, SAT achieves only robust accuracy 30% (Acc-adv) with an Acc-drop of 63.13%. CAT improves this to 45.00% (Acc-drop 48.13%), while DD remains ineffective at 18.00%. In contrast, ATWR retains full clean accuracy (93.13%) with **0%** Acc-drop, and EATR achieves 83.75% Acc-adv with only 9.38% drop.

- In CIFAR-100, SAT and CAT reach 22.00% and 30.00% Acc-adv, respectively, while ATWR again achieves **0%** Acc-drop, and EATR maintains 65.50% Acc-adv and 11.69% Acc-drop.

Table 5: Classification accuracy (%) with Ensemble Adversarial Training and Regularization (EATR) under white-box, untargeted attacks: FGSM, PGD, BIM, and C&W. EATR combines multiple attacks for improved generalization. *Note:* In some cases (e.g., MobileNetV2 on MNIST), the adversarial accuracy slightly exceeds clean accuracy, resulting in a negative Acc-drop (The accuracy drop (Acc-drop) is defined as the difference between the clean accuracy (Acc-origin) and the adversarial accuracy (Acc-adv).). This is a valid outcome due to regularization and perturbation diversity acting as implicit data augmentation, leading to improved calibration under adversarial settings.

Dataset	CNN model	Acc-origin (Top-1)	FGSM			PGD			BIM			C&W		
			Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop	Acc-clean (Top-1)	Acc-adv (Top-1)	Acc-drop
MNIST	ResNet50	100.00	99.06	98.44	0.62	99.06	99.06	0.00	99.06	98.75	0.31	99.06	98.44	0.62
	ResNet101	100.00	98.75	98.12	0.63	98.75	98.75	0.00	98.75	98.75	0	98.75	98.12	0.63
	AlexNet	100.00	99.38	99.06	0.32	99.38	99.38	0.00	99.38	98.75	0.63	99.38	99.38	0.00
	MobileNetV2	100.00	95.00	96.25	-1.25	95.00	95.00	0.00	95.00	95.31	-0.31	95.00	94.38	0.62
	DenseNet121	100.00	99.69	98.75	0.94	99.69	99.69	0.00	99.69	99.38	0.31	99.69	99.06	0.63
	InceptionNetV3	100.00	97.50	96.88	0.62	97.50	97.50	0.00	97.50	97.19	0.31	97.50	96.56	0.94
CIFAR10	ResNet50	93.13	83.75	41.56	42.19	83.75	83.75	0.00	83.75	21.25	62.50	83.75	68.75	15.00
	ResNet101	95.63	83.44	41.25	42.19	83.44	83.44	0.00	83.44	15.94	67.50	83.44	65.94	17.50
	AlexNet	89.06	**9.06**	**9.06**	**0.00**	**9.06**	**9.06**	**0.00**	**9.06**	**9.06**	**0.00**	**9.06**	**9.06**	**0.00**
	MobileNetV2	92.19	82.19	33.75	48.44	82.19	82.19	0.00	82.19	16.88	65.31	82.19	62.81	19.38
	DenseNet121	93.75	83.44	33.12	50.32	83.44	83.44	0.00	83.44	18.44	65.00	83.44	65.62	17.82
	InceptionNetV3	78.13	85.94	48.12	37.82	85.94	85.94	0.00	85.94	15.31	70.63	85.94	58.12	27.82
CIFAR100	ResNet50	77.19	67.50	29.69	37.81	67.50	67.50	0.00	67.50	5.31	62.19	67.50	45.94	21.56
	ResNet101	84.69	61.56	25.00	36.56	61.56	61.56	0.00	61.56	6.25	55.31	61.56	41.25	20.31
	AlexNet	67.50	**0.94**	**0.94**	**0.00**	**0.94**	**0.94**	**0.00**	**0.94**	**0.94**	**0.00**	**0.94**	**0.94**	**0.00**
	MobileNetV2	80.94	43.75	15.00	28.75	43.75	43.75	0.00	43.75	9.69	34.06	43.75	22.50	21.25
	DenseNet121	81.56	15.94	11.88	4.06	15.94	15.94	0.00	15.94	11.88	4.06	15.94	8.12	7.82
	InceptionNetV3	79.69	47.50	17.81	29.69	47.50	47.50	0.00	47.50	15.62	31.88	47.50	25.62	21.88

Table 6: Comparison of defense methods under PGD attack on MNIST, CIFAR-10, and CIFAR-100 using ResNet50.

Defense Method	MNIST			CIFAR-10			CIFAR-100		
	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)
SAT [32]	100	40.00	60.00	93.13	30.00	63.13	77.19	22.00	55.19
DD [19]		0.00	100.00		18.00	75.13		12.00	65.19
CAT [31]		70.00	30.00		45.00	48.13		30.00	47.19
ATWR (Ours)		**98.75**	**1.25**		**93.13**	**0.00**		**77.19**	**0.00**
EATR (Ours)		**98.44**	**1.56**		**83.75**	**9.38**		**65.50**	**11.69**

– In MNIST, ATWR and EATR reduce Acc-drop to **1.25%** and **1.56%**, respectively, compared to 60.00% for SAT, 30.00% for CAT, and 100% for DD.

Under C&W attack:

– In CIFAR-10, SAT, DD, and CAT achieve 33.00%, 19.00%, and 47.00% Acc-adv, respectively. ATWR achieves 91.50% (Acc drop 1.63%), and EATR achieves 87.00% (Acc drop 6.13%).

– In CIFAR-100, CAT achieves 32.00% Acc-adv, while ATWR and EATR achieve 70.00% and 64.80%, respectively.

– In MNIST, both ATWR and EATR reduce the Acc-drop to near zero (1.25% and 1.56%) while maintaining full clean accuracy.

These results clearly highlight the effectiveness of our regularization-based methods. ATWR consistently achieves a minimal accuracy drop across all datasets and attacks. EATR, while slightly behind ATWR in robustness, provides stronger generalization across attack types and datasets. In general, both methods outperform traditional adversarial training, defensive distillation, and ensemble adversarial approaches in both robustness and stability.

6.2 Method-specific behaviors

ATWR achieves the best balance of robustness and clean accuracy. Its superior performance on PGD and C&W can be attributed to the inclusion of weight regularization, which prevents overfitting to adversarial samples and promotes smoother decision boundaries. This is especially effective on datasets like CIFAR-10 and MNIST, where perturbations are subtle.

ATGR performs conservatively by design. Gradient regularization penalizes sharp gradients in the loss landscape, resulting in models that are more stable but also more underfitted. Consequently, ATGR exhibits lower clean and adversarial accuracy (e.g., constant 13.44% on MNIST), but this behavior enhances its potential as an adversarial detector rather than a conventional classifier.

EATR combines the strengths of both approaches. By training on adversarial samples from multiple attacks (FGSM, BIM, PGD, and C&W), EATR generalizes well across attack types. It matches the robustness of ATWR (for example, 0% Acc-drop in CIFAR-10 under PGD) while improving the robustness against simpler attacks such as FGSM and BIM, where ATWR sometimes performs poorly.

Trade-offs and Observations Across Architectures: Across the experiments, we observe that different network architectures respond differently to the applied regularization strategies. For example, weight regularization

Table 7: Comparison of defense methods under C&W attack on MNIST, CIFAR-10, and CIFAR-100 using ResNet50.

Defense Method	MNIST			CIFAR-10			CIFAR-100		
	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)	Acc-clean (%)	Acc-adv (%)	Acc-drop (%)
SAT [32]	100	40.00	60.00	93.13	33.00	60.13	77.19	24.00	53.19
DD [19]		0.00	100.00		19.00	74.13		13.00	64.19
CAT [31]		70.00	30.00		47.00	46.13		32.00	45.19
ATWR (Ours)		**98.75**	**1.25**		**91.50**	**1.63**		**70.00**	**7.19**
EATR (Ours)		**98.44**	**1.56**		**87.00**	**6.13**		**64.80**	**12.39**

(ATWR) performs consistently well on deeper architectures such as ResNet101 and DenseNet121, promoting robustness while maintaining clean accuracy. However, for shallower or lightweight models such as AlexNet or MobileNetV2, strong regularization can hinder the model's ability to learn effectively, leading to performance degradation. In contrast, gradient regularization (ATGR) tends to over-constrain all models uniformly, leading to extremely conservative outputs that, while useful for adversarial detection, are suboptimal for general classification. These findings emphasize the importance of architecture-specific tuning of regularization parameters.

6.3 Observed limitations

A notable limitation arises with the AlexNet model, where all three defense methods fail to maintain high clean accuracy. For example, under ATWR, AlexNet's clean accuracy on MNIST drops from 100% to 13.44%. This suggests that regularization strength may need model-specific tuning and that shallow architectures like AlexNet lack the capacity to learn simultaneously from clean and adversarial examples under strong constraints.

6.4 Generalization vs. robustness trade-off

Our experiments confirm the known trade-off between clean accuracy (generalization) and adversarial robustness (Acc-adv) [41]. ATWR and EATR offer high robustness with minimal loss in clean accuracy on deeper models (e.g., ResNet101, DenseNet121). In contrast, ATGR heavily sacrifices clean accuracy in exchange for stability, which can be advantageous for adversarial detection but not for standard classification.

These results indicate that robustness and generalization can be simultaneously improved with proper regularization and ensemble strategies, but optimal results depend on architecture depth, attack strength, and dataset complexity.

Interestingly, on simpler datasets such as MNIST, we observe that adversarial accuracy can occasionally exceed clean accuracy, particularly under the EATR framework. This leads to negative Acc-drop values (for example, -1.25%), which may seem counterintuitive, but are valid in our setting. We attribute this to the regularization and perturbation diversity in EATR acting as a form of implicit data augmentation. As a result, the model becomes better calibrated under perturbed conditions, sometimes generalizing slightly better than on unperturbed inputs. This behavior has been verified across multiple runs and aligns with recent findings on robustness-generalization synergy in over-regularized regimes.

7 Conclusion and future work

This work underscores the vulnerability of widely used deep image classifiers to adversarial attacks such as FGSM, PGD, BIM, and C&W. To address these challenges, we propose three defense strategies: ATWR, ATGR, and EATR that offer distinct advantages in enhancing model robustness. ATWR and EATR effectively improve resilience against strong optimization-based attacks (e.g., PGD, C&W) while maintaining a favorable trade-off with clean accuracy. In contrast, ATGR introduces a complementary perspective by transforming standard classifiers into effective adversarial detectors. These findings highlight the critical role of tailored regularization in adversarial training and the broader need for robust defense mechanisms to secure deep learning systems in real-world deployments. In future work, we plan to evaluate the transferability of adversarial attacks across CNN architectures and extend the analysis to vision transformers, paving the way for stronger and more adaptable adversarial defenses.

Appendix: pseudocode for regularized adversarial training

Algorithm 1: Training with ATWR / ATGR / EATR

Input: Training set $\mathcal{D}$, pre-trained model f_θ, loss function $\mathcal{L}$, regularization type, attack methods

Output: Fine-tuned model parameters θ

Initialize model parameters θ using pre-trained weights (fine-tuned on $\mathcal{D}$)

for *each epoch* **do**

  for *each mini-batch* $\{(x_i, y_i)\}_{i=1}^{B}$ *in* $\mathcal{D}$ **do**

    Generate adversarial examples $\tilde{x}_i$ using selected attack(s) (on-the-fly)

    Form mini-batch $B_{\text{clean}} = \{x_i\}$ and $B_{\text{adv}} = \{\tilde{x}_i\}$

    Mix batch: $B_{\text{mix}} = B_{\text{clean}} \cup B_{\text{adv}}$

    Compute $\mathcal{L}_{\text{clean}}$ and $\mathcal{L}_{\text{adv}}$

    if *ATWR* **then**

      $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{clean}} + \lambda_1 \mathcal{L}_{\text{adv}} + \lambda_2 \|W\|^2$

    else

      if *ATGR* **then**

        $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{clean}} + \beta_1 \mathcal{L}_{\text{adv}} + \beta_2 \|\nabla_x \mathcal{L}_{\text{adv}}\|^2$

      else

        EATR: Use multiple $\mathcal{L}_{\text{adv}}^{\text{attack}}$ terms and apply ensemble aggregation

    Backpropagate and update θ using optimizer

return θ

References

[1] J. Weng, Z. Luo, D. Lin, and S. Li, "Comparative evaluation of recent universal adversarial perturbations in image classification," *Computers & Security*, vol. 136, p. 103 576, 2024.

[2] J. Sen, A. Sen, and A. Chatterjee, "Adversarial attacks on image classification models: Analysis and defense," *International Conference on Business Analytics and Intelligence (ICBAI'23)*, 2023.

[3] N. Ghaffari Laleh, D. Truhn, G. P. Veldhuizen, *et al.*, "Adversarial attacks and adversarial robustness in computational pathology," *Nature communications*, vol. 13, no. 1, p. 5711, 2022.

[4] H. Hirano, A. Minagi, and K. Takemoto, "Universal adversarial attacks on deep neural networks for medical image classification," *BMC medical imaging*, vol. 21, pp. 1–13, 2021.

[5] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] 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.

[8] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*, Ieee, 2017, pp. 39–57.

[9] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*, IEEE, 2020, pp. 1277–1294.

[10] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *European conference on computer vision*, Springer, 2020, pp. 484–501.

[11] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[12] A. Mkadry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, no. 9, 2017.

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*, Chapman and Hall/CRC, 2018, pp. 99–112.

[14] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," *Advances in neural information processing systems*, vol. 32, 2019.

[15] E.-C. Chen and C.-R. Lee, "Towards fast and robust adversarial training for image classification," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[16] Y. Jiang, C. Liu, Z. Huang, M. Salzmann, and S. Susstrunk, "Towards stable and efficient adversarial training against l_1 bounded adversarial attacks," in *International Conference on Machine Learning*, PMLR, 2023, pp. 15 089–15 104.

[17] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*, IEEE, 2016, pp. 372–387.

[18] D. Wang, A. Ju, E. Shelhamer, D. Wagner, and T. Darrell, "Fighting gradients with gradients: Dynamic defenses against adversarial attacks," *arXiv preprint arXiv:2105.08714*, 2021.

[19] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*, IEEE, 2016, pp. 582–597.

[20] J. Sen, "The fgsm attack on image classification models and distillation as its defense," in *International Conference on Advances in Distributed Computing and Machine Learning*, Springer, 2024, pp. 347–360.

[21] A. Shafahi, M. Najibi, M. A. Ghiasi, *et al.*, "Adversarial training for free!" *Advances in neural information processing systems*, vol. 32, 2019.

[22] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *arXiv preprint arXiv:1605.07725*, 2016.

[23] H. Zhang, H. Chen, Z. Song, D. Boning, I. S. Dhillon, and C.-J. Hsieh, "The limitations of adversarial training and the blind-spot attack," *arXiv preprint arXiv:1901.04684*, 2019.

[24] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*, PMLR, 2018, pp. 274–283.

[25] S. Liu and P.-Y. Chen, "Zeroth-order optimization and its application to adversarial machine learning," *Intelligent Informatics*, p. 25, 2018.

[26] F. Ö. Çatak, S. Sivaslioglu, and K. Sahinbas, "A generative model based adversarial security of deep learning and linear classifier models," *Informatica*, vol. 45, pp. 33–64, 2021.

[27] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 2017, pp. 15–26.

[28] Y. Shi, Y. Han, Q. Hu, Y. Yang, and Q. Tian, "Query-efficient black-box adversarial attack with customized iteration and sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2226–2245, 2022.

[29] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[30] F. Tramer, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *ICLR 2018*, 2018.

[31] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, "You only propagate once: Accelerating adversarial training via maximal principle," *Advances in neural information processing systems*, vol. 32, 2019.

[32] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[33] Y. LeCun, C. Cortes, and C. J. Burges, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, IEEE, 1998, pp. 2278–2324.

[34] A. Krizhevsky, "Learning multiple layers of features from tiny images," in *Master's thesis, University of Toronto*, Toronto, Canada, 2009.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2009, pp. 248–255.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[37] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, *Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size*, 2016. arXiv: 1602.07360 [cs.CV].

[38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018.

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017, pp. 4700–4708.

[40] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. DOI: 10.1109/CVPR.2016.308.

[41] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019.