



ARTICLE

Adversarial Defense Method Based on Dual Mode Pixel Transformation and Multi-Objective Spatial Optimization

Jiaying Li¹, Xiujuan Wang^{1,*}, Shuhan Han², Liya Xu¹ and Changxing Wang¹

¹College of Computer Science, Beijing University of Technology, Beijing, China

²School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding Author: Xiujuan Wang. Email: xjwang@bjut.edu.cn

Received: 19 March 2026; Accepted: 18 June 2026; Published: 13 August 2026

ABSTRACT: Deep neural networks are widely applied in computer vision tasks but remain highly vulnerable to adversarial attacks. Tiny and imperceptible perturbations can cause severe model misclassification. Most existing defense methods improve robustness but significantly reduce model accuracy on clean examples. To address this issue, we propose a defense framework combining pixel value transformation and spatial transformation. The proposed method divides the input image into two complementary regions. Feature compression is applied to one region to reduce model sensitivity to subtle perturbations. Intense reversible pixel transformation is applied to the other region to disrupt the spatial distribution of the perturbations. Furthermore, a spatial transformation module is introduced to enhance defense capabilities against strong attacks. This module adjusts the spatial structure of the image, increases nonlinear relationships among pixels, and suppresses perturbation transfer effects. A Multiple Objective Particle Swarm Optimization (MOPSO) is employed to adaptively search for optimal transformation parameters, achieving a balance between robustness and accuracy. An inverse transformation mechanism is proposed to restore attacked examples back to a form recognizable to human eyes, which effectively destroys the imperceptible nature of adversarial examples. Experimental results demonstrate that the proposed method successfully resists various adversarial attacks while maintaining high recognition accuracy on clean images. The proposed reversible defense framework demonstrates theoretical universality and scalability, providing an effective new approach for adversarial defense.

KEYWORDS: Adversarial examples; adversarial defense; pixel-wise transformation; model security

1 Introduction

In recent years, deep neural networks have made great progress in many computer vision tasks, such as image classification, object detection, and semantic segmentation. But many studies show that these models have clear weaknesses when confronted with adversarial examples [1,2]. Adversarial examples are created by adding small and carefully perturbations changes to the original images, and these changes are hard for the human eye to detect, they can cause the model to give incorrect results. These small changes are almost invisible to humans, but they can greatly reduce the model's ability to make correct predictions. This problem shows that deep models are not strong in security and generalization, and it also makes their use risky in safety related tasks such as autonomous driving, medical image diagnosis, and biometric recognition [3].

Adversarial attacks are often divided into white-box and black-box attacks based on how much the attacker knows about the model. In white-box attacks, the attacker has full access to the model, including its structure, parameters, and gradients. Common white-box methods include the Fast Gradient Sign Method (FGSM) [1], the Basic Iterative Method (BIM) [2], and the Projected Gradient Descent (PGD) [4]. In

black-box attacks, the attacker cannot access the model's gradients or parameters, so the attacker uses model transfer or repeated queries to attack the model [5]. Gray-box attacks are between white-box and black-box attacks [6]. In this case, the attacker only knows part of the model information, such as the structure or some parameters, but does not know the full gradients and weights [7].

To deal with adversarial perturbations, researchers have proposed several types of defense methods such as adversarial training [8], input transformation [9], gradient masking [10], and model regularization [11]. Input transformation methods process the input before it is fed into the model so that the effect of adversarial perturbations can be reduced. Common methods include JPEG compression [9], High level Representation Guided Denoiser (HGD) [12], and random resizing and padding [13]. Input transformation methods often cannot fully remove adversarial perturbations, and they may also damage useful features in clean examples. For example, JPEG compression can reduce small perturbations, but it can also lower the classification accuracy of clean images. This problem is one of the main challenges in adversarial machine learning.

We propose an adversarial defense framework that combines pixel-level transformations and spatial transformations to improve robustness while maintaining high accuracy on clean examples. The method divides the input image into two complementary regions in a checkerboard pattern. In one region, the method applies feature compression to reduce the model's sensitivity to small perturbations while preserving important semantic information. In the other region, the method uses an invertible high intensity pixel transformation to break gradient paths and reduce the spread of adversarial perturbations. This dual pixel transformation enables the model to maintain strong classification ability on clean examples and resist different types of adversarial perturbations. The method also adds a spatial transformation module to improve defense performance under strong attacks. This module changes the spatial order of pixels and increases nonlinear dependencies between neighboring regions, thereby reducing the transferability and local consistency of adversarial perturbations. We use the MOPSO to search for appropriate spatial transformation parameters; this strategy helps the model reach a good balance between defense ability and classification performance. The method also includes a reversible transformation mechanism. If a white-box attack is carried out in the transformed domain, the inverse transformation can change the adversarial perturbations into visible changes in the image, rendering them perceptible to the human eye. This process removes the hidden nature of adversarial examples and reduces their attack effect.

In summary, the main contributions of this paper are as follows:

- (1) We propose a checkerboard style dual mode pixel transformation method that integrates feature compression with an invertible high intensity transformation, achieving a balanced trade-off between clean example accuracy and adversarial robustness.
- (2) We introduce a spatial transformation module and employ MOPSO to automatically search for the optimal transformation parameter configurations, thereby enabling adaptive adjustment between robustness and accuracy.
- (3) We design an inverse transformation mechanism that breaks the hidden nature of adversarial examples and enlarges the perturbations produced by white-box attacks. The proposed defense framework is general in theory and can adapt to different types and strengths of adversarial attacks.

2 Related Work

2.1 Adversarial Attacks

Let the original input be denoted as x , with its true label y . Consider a classifier $f: \mathbb{R}^d \rightarrow \mathbb{R}^K$, where $f_j(x)$ represents the logit corresponding to class j , d is the input dimensionality, and K is the number of classes. The decision function is defined as $C(x) = \operatorname{argmax}_j f_j(x) = y$. The generation of an adversarial

example can be formulated as finding a perturbation $\delta \in \mathbb{R}^d$ satisfying $\|\delta\| \leq \varepsilon$, such that $C(x + \delta) \neq y$ (where $\varepsilon > 0$ is a small constant bounding the perturbation magnitude), thereby misleading the model while ensuring that the perturbation remains sufficiently small to be imperceptible to human observers. Depending on the attacker's level of access to the target model's internal information, adversarial attacks are typically categorized into white-box and black-box attacks.

In white-box attack settings, the attacker knows the full structure, parameters, and gradients of the target model, so the attacker can use gradient information to create adversarial examples directly. Goodfellow et al. proposed FGSM [1], which adds perturbations by using a step gradient update based on a linear assumption. Kurakin et al. proposed BIM [2], which uses small iterative gradient steps with clipping and makes the attack stronger. Madry et al. then proposed PGD [4], which adds random initialization to iterative gradient updates and is often seen as the strongest first order attack baseline. Croce and Hein extended PGD by introducing two adaptive step size methods and employed an optimization process to reduce the risk of getting stuck in poor local solutions [14]. They also combined Fast Adaptive Boundary (FAB) and Square attacks to build the AutoAttack (AA) framework [15], which is now a standard tool for testing model robustness. For pixel-level prediction tasks, researchers also proposed an efficient white-box attack [16]. Carlini and Wagner proposed another attack method that changes the original constrained optimization problem into an unconstrained form that can be solved by common optimizers [17]. The Carlini & Wagner (C&W) attack works well under different distance measures such as L_2 , L_0 , and L_∞ , and it has broken many early gradient masking defense methods, which led researchers to design stronger and more reliable defense methods.

In black-box attack settings, the attacker can only obtain input-output pairs by querying the target model. Early studies mainly used the idea of transferability. Papernot et al. built the basic framework for black-box attacks by training a substitute model to imitate the target model and then using white-box attack methods on the substitute model to create adversarial examples that can transfer to the target model [18]. Later studies focused on query-based attacks. Wang and He proposed the Variance-tuning Momentum Iterative Fast Gradient Sign Method (VMI-FGSM) and the Variance-tuning Nesterov Iterative Fast Gradient Sign Method (VNI-FGSM), which use momentum correction to improve gradient estimation and make black-box attacks stronger [19]. In recent years, researchers have also proposed many black-box attack methods based on differential evolution [20,21]. The proliferation of such attack methods has created serious security risks for deep learning systems [22].

2.2 Adversarial Defense

Researchers have proposed many defense methods to address with adversarial examples; these methods can be divided into two main types: adversarial training and input feature purification methods.

Adversarial training is one of the most common and well developed defense methods. Kurakin et al. proposed a foundational adversarial training method [8], which incorporates adversarial examples into the training data to improve robustness. Since adversarial training can reduce accuracy on clean examples, Helper-based Adversarial Training (HAT) adds extra mislabeled examples during training to alleviate this problem and improve the balance between accuracy and robustness [23]. Liao et al. proposed HGD [12]; while this method shows good defense performance, it is computationally expensive. Laidlaw et al. leveraged perceptual distance measures to create perturbations so that the trained model can better handle different types of adversarial perturbations [24]. Adversarial Training with Generated Data (AT-GD) uses additional generated images as extra training data and achieves improved robustness [25]. Fast Adversarial Training (FAT) uses randomized smoothing to solve the inner optimization problem efficiently and introduces a new initialization method called backward smoothing to make single step robust training more stable and improve robustness [26]. Adversarial training can greatly improve robustness against perturbations, but it requires high computational cost and often reduces accuracy on clean examples.

Other studies have tried to remove adversarial perturbations by using preprocessing methods and model optimization instead of adversarial training. Guo et al. used simple image transformations, but the defense effect was limited [27]. Feature squeezing methods reduce color bit depth and smooth the image to limit the attack space [28]. Pixel deflection defends against perturbations by forcing the image to follow natural image statistics [29]. Shi et al. added a self-supervised task during training and adjusted the input during testing to better fit this task and reduce perturbations [30]. Li et al. proposed a boundary aware transformation method to improve purification accuracy [31]. Hwang et al. proposed Adversarial Input Defense (AID) as an extra purification module that can be added to common classifiers for adversarial defense [32]. Wang et al. used color space transformations to enlarge adversarial perturbations, but this method greatly reduced accuracy on clean examples [33]. Zhao et al. improved the robustness of defense models against adversarial perturbations by exploiting richer multi-scale features across different image scales [34].

Adversarial training achieves strong defense performance, but it has poor generalization ability and needs very high computation cost. Preprocessing purification methods are easier to deploy, but they often remove important information during transformation and lower accuracy on clean examples.

3 Method

Many existing adversarial defense methods based on image preprocessing improve model robustness, but they often cause a large drop in classification accuracy on clean examples. When the attack strength is high, the classification performance of these methods decreases quickly, and this result shows that preprocessing defenses are still weak under strong attacks. We propose a method that combines pixel-level transformations and spatial feature transformations. The method keeps high recognition accuracy on clean examples and greatly improves the model's robustness against many types of adversarial attacks.

3.1 Pixel-Level Transformations

3.1.1 Feature Squeezing

Feature Squeezing (FS) is a classic adversarial defense approach based on input transformation. Its core objective is to suppress adversarial perturbations by reducing the degrees of freedom in the input features. Specifically, FS compresses input features to decrease data dimensionality or precision, thereby restricting the perturbation space available to an adversary. In practical applications, FS typically employs operations such as color bit depth reduction or spatial smoothing to eliminate subtle pixel level distortions that are imperceptible to human vision but may mislead deep neural networks. By mapping similar input values to identical or proximal representations, feature squeezing effectively weakens adversarial noise while preserving the original semantic information of the image to the greatest extent possible. Fig. 1 shows the effects of different pixel bit-depth quantization levels on images.

Let the original input be denoted as $x \in \mathbb{R}^{H \times W \times C}$ with ground label y . The classifier output for class j is represented by $f_j(x)$, and the decision function is defined as $C(x) = \arg\max_j f_j(x) = y$. Let the adversarial perturbation be $\delta \in \mathbb{R}^d$, constrained by $\|\delta\| \leq \epsilon$. The feature squeezing error is defined as in Eq. (1), where $S(\cdot)$ denotes the feature squeezing operator. Assume that f is $L > 0$ -Lipschitz continuous within the relevant neighborhood, $\forall u, v$ in the neighborhood, Eq. (2) holds. Define the classification margin γ as in Eq. (3). Based on the Lipschitz continuity, we obtain Eq. (4). If the condition in Eq. (5) is satisfied, then for any $j \neq y$, Eq. (6) holds, which ensures $C(S(x + \delta)) = y$.

$$\Delta_S(x, \delta) := \|S(x + \delta) - S(x)\| \quad (1)$$

$$\|f(u) - f(v)\| \leq L \|u - v\| \quad (2)$$

$$\gamma(x) := f_y(x) - \max_{j \neq y} f_j(x) \quad (3)$$

$$\|f(S(x + \delta)) - f(S(x))\| \leq L \|S(x + \delta) - S(x)\| = L\Delta_S(x, \delta) \quad (4)$$

$$L\Delta_S(x, \delta) < \frac{\gamma(x)}{2} \quad (5)$$

$$f_y(S(x + \delta)) - f_j(S(x + \delta)) \geq f_y(S(x)) - f_j(S(x)) - 2L\Delta_S(x, \delta) \geq \gamma(x) - 2L\Delta_S(x, \delta) > \gamma(x) - \gamma(x) > 0 \quad (6)$$

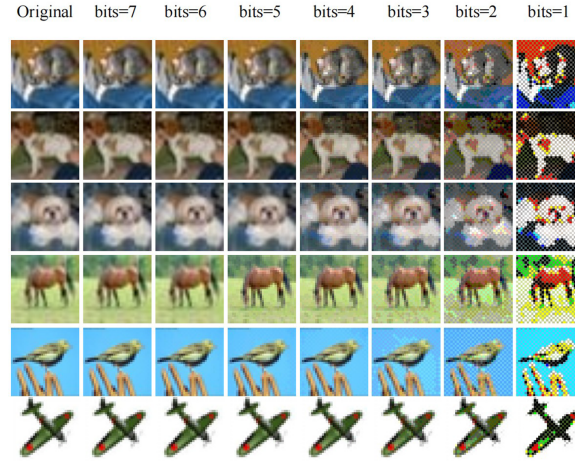


Figure 1: Effects of different compression bit depths on images.

For a classifier f that is locally Lipschitz continuous, if the approximation error introduced by the feature compression operator S a clean example x satisfies $\|S(x) - x\| < \gamma(x)/2L$, then the compression does not alter the classification decision. Given the complexity of dataset distributions and model architectures, it is currently infeasible to derive a unified theoretical upper bound for b -bit quantization. Nevertheless, our experimental results demonstrate that, under moderate reductions in pixel bit depth, the defensive model incurs only negligible degradation in classification accuracy on clean inputs. This observation suggests that, with appropriately tuned compression strength, it is possible to achieve an optimal trade-off in practice between enhanced robustness and preservation of the model's original recognition performance.

3.1.2 Checkerboard Transformation

A major limitation of employing FS as a standalone defense mechanism is its inability to withstand white-box attacks. Under the white-box threat model, the adversary has full access to the model architecture, parameters, and gradient information, enabling adaptive optimization that explicitly targets the defense. By injecting adversarial perturbations that anticipate the bit depth compression stage, a white-box attacker can effectively circumvent FS defenses, particularly those relying on fixed parameters or a single transformation scheme. To address this vulnerability, we introduce a checkerboard pixel-value transformation that injects structured perturbations into the spatial domain. When combined with FS, this spatial transformation increases the model's robustness against adaptive white-box adversaries by disrupting the gradient alignment leveraged during attack generation.

The checkerboard transformation assigns different pixel-value operations to image coordinates satisfying $(i + j) \bmod 2 = 0$ and $(i + j) \bmod 2 \neq 0$, where i and j denote the row and column indices of each pixel,

respectively. Specifically, for pixels in the R channel whose spatial locations satisfy $(i + j) \bmod 2 = 0$, a large magnitude pixel-value transformation is applied, as expressed in Eq. (7). Conversely, the G and B channels undergo an opposite transformation pattern, as shown in Eq. (8), where $pix \in \{0, 1, \dots, 255\}$ denotes the original pixel intensity value, and pix' denotes the transformed pixel value. This operation induces mirrored reflections and periodic folding within the low and high ranges of pixel intensity, thereby constructing a discontinuous and non-differentiable input mapping.

$$pix' = (2 \cdot pix) \bmod 256 - 255 \cdot \left\lfloor \frac{pix}{128} \right\rfloor \quad (7)$$

$$pix' = \left\lfloor \frac{pix}{2} \right\rfloor + 128 \cdot (pix \bmod 2) \quad (8)$$

When employed in isolation, Feature Squeezing fails to provide sufficient protection against specialized white-box attacks. This vulnerability arises because an adversary with full knowledge of the defense mechanisms and parameters can generate deceptive adversarial perturbations using gradient approximation or backward pass adjustment strategies. In contrast, the checkerboard transformation proposed in this work, which integrates nonlinear pixel transformation with feature squeezing, effectively defends against adversarial attacks in white-box scenarios. As illustrated in Fig. 2, the proposed method applies a high folding pixel mapping transformation where pixel coordinates satisfy $i + j \equiv 0 \pmod{2}$, while performing 4 bits depth compression where $i + j \equiv 1 \pmod{2}$. Following the inverse transformation, a significant discrepancy becomes observable between the attacked pixels of the adversarial example (d) and the original image (a). As shown in Fig. 3, the adversarial perturbations no longer maintain their imperceptible quality. Consequently, by disrupting the smoothness and stealth of adversarial noise, the checkerboard pixel transformation significantly enhances the robustness of the defense model under white-box attack conditions.

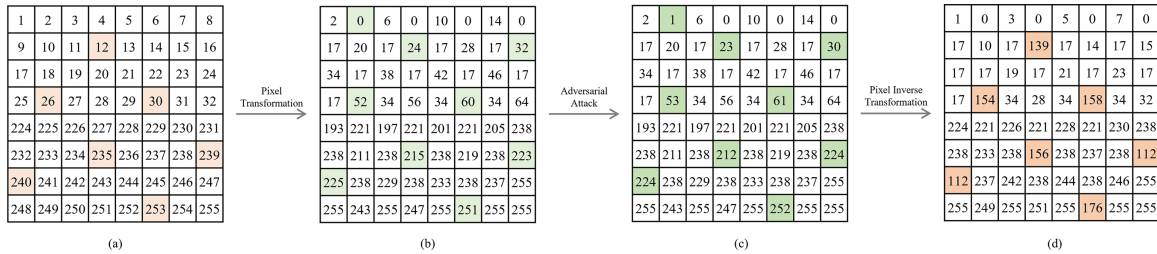


Figure 2: Significant pixel value differences (quantitative analysis) between clean examples and inversely transformed white-box attacks after pixel transformation. (a) Original image; (b) pixel-transformed image; (c) adversarially attacked image after pixel transformation; (d) inverse pixel-transformed image after adversarial attack.

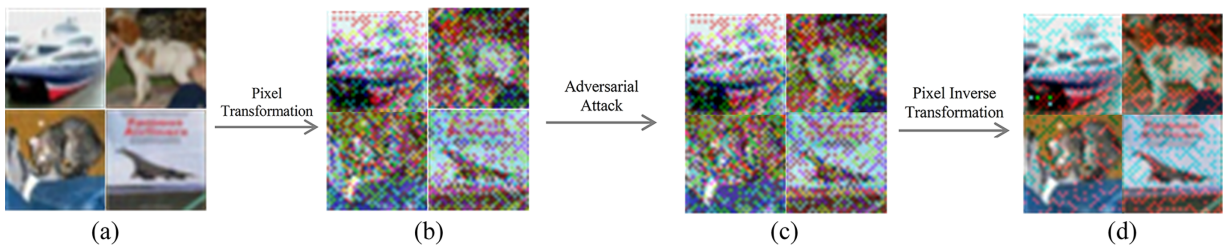


Figure 3: Significant pixel value differences (visual comparison) between clean examples and inversely transformed white-box attacks after pixel transformation. (a) Original image; (b) pixel-transformed image; (c) adversarially attacked image after pixel transformation; (d) inverse pixel-transformed image after adversarial attack.

3.2 Spatial Transformation

3.2.1 Row-Column Dual Transformation

When the perturbation magnitude increases (e.g., $\varepsilon \geq 8/255$), pixel-level transformation defense strategies still exhibit notable vulnerabilities. To address this limitation, we propose a row-column transformation strategy. For an input image $x \in \mathbb{R}^{H \times W \times C}$, the row-column transformed representation is defined as Eq. (9), where $P_{row} \in \mathbb{R}^{H \times H}$ and $P_{col} \in \mathbb{R}^{W \times W}$ denote the linear transformation matrices along the row and column dimensions, respectively. Adversarial attacks fundamentally rely on the input gradient $\nabla_x \mathcal{L}(C(x), y)$. After the proposed transformation, the gradient becomes Eq. (10), where $\mathcal{L}$ denotes the classification loss, its direction is jointly perturbed by the permutation matrices P_{row}^T and P_{col} , leading to spatial misalignment of the gradient flow. This phenomenon introduces gradient distortion, preventing the attacker from optimizing along the correct gradient direction and thereby significantly weakening the attack effectiveness. Meanwhile, the adversarial perturbation after applying the spatial transformation becomes Eq. (11). Fig. 4 illustrates the complete architecture of the proposed defense model for both training and testing.

$$x' = P_{row} x P_{col}^T \quad (9)$$

$$\nabla_{x'} \mathcal{L}(C(P_{row} x P_{col}^T), y) = P_{row}^T \nabla_x \mathcal{L}(C(x), y) P_{col} \quad (10)$$

$$\delta' = P_{row} \delta P_{col}^T \quad (11)$$

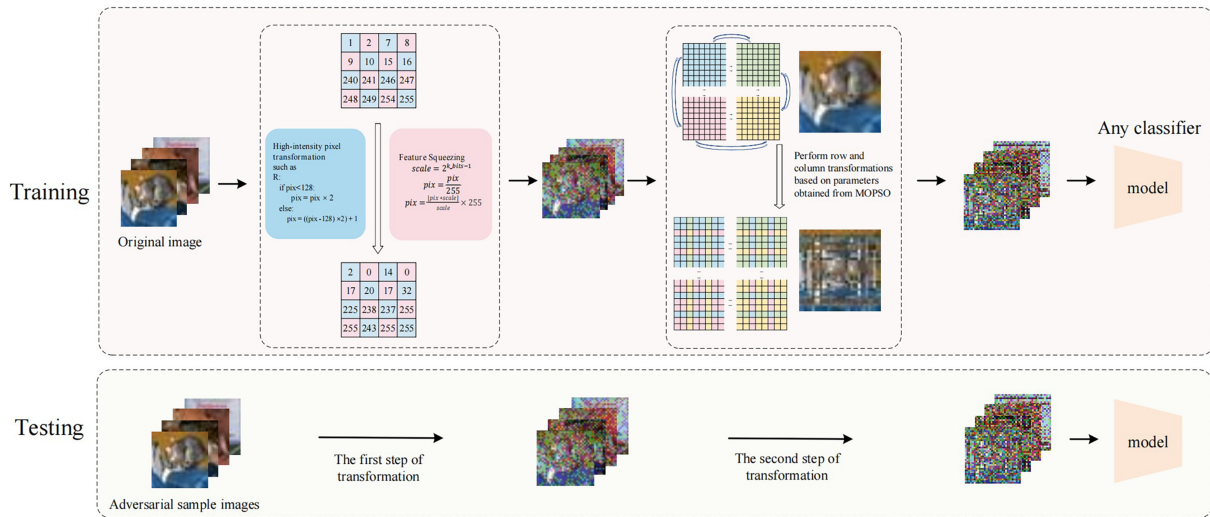


Figure 4: Schematic illustration of the training and testing pipeline for the proposed defense method.

3.2.2 Multiple Objective Particle Swarm Optimization

Different configurations of spatial transformation parameters can exert a substantial influence on both the model's defensive capability and its classification accuracy on clean examples. To achieve an optimal balance between these two competing objectives, this study employs a Multiple Objective Particle Swarm Optimization (MOPSO) to jointly optimize the spatial transformation parameters [35]. MOPSO treats the classification accuracy on clean inputs and the robustness against adversarial examples as simultaneous optimization objectives, thereby preserving both model fidelity and security during the parameter search process.

In multiple objective optimization, each particle must simultaneously account for multiple objective functions. The optimization problem can be formulated as Eq. (12), where m denotes the number of

objectives, $\mathbf{z}_i$ denotes the position vector of the i -th particle, $\mathbf{Q}(\mathbf{z}_i)$ denotes the objective function vector. For the defense task considered in this work, we define four primary optimization objectives: $1 - Acc_{clean}$, $1 - Acc_{FGSM}$, $1 - Acc_{PGD}$, $1 - Acc_{AA}$, where Acc_{clean} is the classification accuracy of the model on clean examples after applying the spatial transformation, and Acc_{FGSM} , Acc_{PGD} , Acc_{AA} denote the accuracies under adversarial examples generated using FGSM, PGD, and AA attacks, respectively, with perturbation magnitude $\epsilon = 8/255$. To balance multiple objectives, the algorithm incorporates a non-dominated sorting mechanism to maintain the Pareto Front P . For two candidate solutions z_a and z_b , if Eq. (13) holds, then z_a is said to dominate z_b . All individuals not dominated by any other solutions constitute the current Pareto set P , where $q_k(\cdot)$ denotes the k -th objective function. During particle updates, the global best solution g is selected from the Pareto Front based on the crowding distance criterion to preserve solution diversity.

$$\min_{\mathbf{z}_i} \mathbf{Q}(\mathbf{z}_i) = [q_1(\mathbf{z}_i), q_2(\mathbf{z}_i), \dots, q_m(\mathbf{z}_i)] \quad (12)$$

$$\forall q, q_k(\mathbf{z}_a) \leq q_k(\mathbf{z}_b); \exists v, q_v(\mathbf{z}_a) < q_v(\mathbf{z}_b) \quad (13)$$

In the proposed method, each particle $\mathbf{z}_i$ encodes a set of spatial transformation parameters. Specifically, $\mathbf{z}_i$ consists of 16 binary elements, where a value of “1” indicates that the i -th row of the image is swapped with the $(32 - i)$ -th row, while a value of “0” denotes that the corresponding rows remain unchanged. The spatial transformation, together with the aforementioned pixel-level transformations, is applied to the training set for model training. After 35 epochs of training, the resulting model weights are evaluated on clean examples and three sets of adversarial examples to obtain the objective values, which subsequently guide the iterative optimization. Finally, the optimal parameter configuration is selected as the particle achieving the highest value of $0.5 \times Acc_{clean} + 0.5 \times average(Acc_{FGSM} + Acc_{PGD} + Acc_{AA})$.

4 Experiments

To comprehensively evaluate the effectiveness of the proposed defense method, a series of experiments were conducted. The original CIFAR-10 dataset, without any preprocessing or data augmentation, was adopted as the evaluation benchmark. This dataset consists of 10 classes, with 50,000 images in the training set and 10,000 images in the test set. Six widely used classification models: ResNet18, SENet18, VGG19, GoogLeNet, RegNetX-200MF, and WideResNet-28-10. All models were retrained incorporating the proposed adversarial defense method, and no additional fine-tuning or auxiliary training strategies were applied beyond this defense mechanism.

4.1 Classification Capability

We first evaluate the classification performance of the proposed method on clean examples, which serves as the primary criterion for assessing the practical applicability of any defense mechanism. Both the original models and the defended models based on the six architectures mentioned above were trained and evaluated, and their performance was further compared against five baseline defense methods. The quantitative results are reported in Table 1. The missing values are attributed to the core mechanism of SOAP, which relies on $\nabla_x \mathcal{L}_{aux}$ to guide pixel updates—a purification strategy that is heavily gradient dependent. For deeper and more complex architectures, back propagating self-supervised loss to the input pixel space frequently encounters the vanishing gradient problem, resulting in weak or noisy ∇_x signals during the purification phase. As noted in the original SOAP study, certain self-supervised tasks are incompatible with specific architectures, leading to the observed omissions. Furthermore, Var-nmODE is a specialized variant of ResNet that reformulates discrete hierarchical layers as continuous-time ordinary differential equations (ODEs); consequently, it cannot be extended to other backbone models.

Table 1: Classification accuracy (%) on clean examples for the original model (shown in the first row) and the proposed defense model.

	ResNet	SeNet	RegNet	GoogLeNet	VGG	WideResNet
Original	93.96	93.85	94.05	94.38	92.59	95.11
PAT	80.20	80.17	80.26	80.29	72.44	–
AT-GD	87.35	87.3	87.17	87.24	81.74	85.97
SOAP	84.07	–	–	–	–	91.89
DiffPure	85.2	85.08	85.2	85.29	80.2	89.02
Wang	86.91	86.12	85.77	79.10	80.96	90.38
var-nmODE	50.21	–	–	–	–	–
Ours	91.56	90.54	92.02	92.61	86.66	93.31

As evidenced by the results reported in Table 1, the proposed pixel-transformation defense consistently achieves higher classification accuracy on clean examples across all six evaluated architectures compared with the other defense methods. Except for VGG-19, the defended models obtained after applying the proposed transformations maintain clean example accuracies above 90%. This indicates that the proposed defense introduces negligible degradation in classification performance on clean data and yields a substantial improvement over existing defense strategies. The proposed adversarial defense method disrupts the input space of the original image. To maintain classification performance under such conditions, the model must possess the capability to learn feature transformations. VGG-19 is a representative parameter intensive model that relies heavily on fully connected layers. This architecture struggles to adjust feature dependencies dynamically, which leads to a diminished ability to discriminate images after preprocessing operations.

The essence of adversarial defense is typically to filter or smooth non robust features contained in images. The trade off between robustness and accuracy is a well recognized phenomenon in the adversarial machine learning community, as highlighted by Tsipras et al. [36]. Nevertheless, the proposed method achieves higher clean sample accuracy than other adversarial defense methods across different models, with an average improvement ranging from 3.69% to 12.22%. In practical applications such as autonomous driving and face recognition based payment systems, the consequences of adversarial attacks can be catastrophic. Therefore, sacrificing 2% to 4% of clean accuracy to obtain stronger resistance against malicious attacks can be regarded as a necessary and acceptable security premium.

4.2 Gray-Box Testing

To evaluate the effectiveness of the proposed adversarial defense, we designed and conducted a series of comparative experiments under gray-box attack settings. This setting effectively simulates realistic threat models encountered in practical applications and avoids overly strong attack assumptions. In the experiments, we selected five widely adopted adversarial attack methods, including FGSM, PGD, BIM, AutoAttack (AA), and the C&W attack. Adversarial examples were generated on the 10,000 test images of the CIFAR-10 dataset for defense evaluation, resulting in a total of 130,000 adversarial examples (i.e., 13 attack configurations $\times$ 10,000 images). Specifically, FGSM, PGD, BIM, and AA were conducted under an L_∞ -norm constraint with perturbation budgets set to $\epsilon = 2/255, 8/255$, and $16/255$, respectively, while the C&W attack was performed under an L_2 -norm constraint with the perturbation parameter set to $c = 0.3/255$.

All experiments employed ResNet-18 as the standard classifier, and the proposed method was compared against five defense approaches: PAT [24], AT-GD [25], SOAP [30], color defense [33], DiffPure [37], and var-nmODE [38]. The experimental results are summarized in Table 2. The proposed defense not only

achieves higher classification accuracy on clean examples than the competing methods, but also consistently outperforms the baseline defenses across multiple attack types and perturbation strengths. Notably, under high intensity attacks ($\epsilon \geq 8/255$), the proposed method demonstrates a more pronounced advantage. For instance, against the BIM attack with $\epsilon = 8/255$, our method attains an accuracy of 81.89%, representing an average improvement of 25.7% over the other four defenses. Similarly, for BIM with $\epsilon = 16/255$, the proposed method yields an average accuracy gain of 38.23% compared to the competing approaches. Although DiffPure achieves slightly better robustness than our method under several strong attack settings, it relies on an iterative diffusion denoising process, which introduces substantial computational overhead during inference. In contrast, our approach only applies lightweight spatial transformations and therefore incurs almost no additional inference latency. When evaluating the same set of 10,000 images, DiffPure requires approximately 55 min, whereas our method completes the process in only 2.27 min. This demonstrates that our approach is approximately 24 times more efficient than DiffPure, highlighting its significant practical advantage in terms of computational efficiency and real world deployment.

Table 2: The original and defense models achieve accuracy (%) when evaluated against gray-box attacks. Bold values indicate the best performing model under each column's condition.

Model	Clean	FGSM				PGD		BIM			AA		CW	
Original	93.96	47.83	16.93	10.81	24.85	0.38	0.00	24.66	0.28	0.00	14.89	2.96	1.84	7.31
PAT	80.20	73.91	54.02	36.13	73.34	40.91	10.67	73.31	41.05	11.25	73.42	39.51	9.29	72.42
AT-GD	87.35	82.67	66.59	47.65	82.64	67.19	40.75	82.65	63.48	34.15	82.75	66.04	33.77	25.56
SOAP	84.07	–	51.02	–	–	51.42	–	–	51.42	–	–	–	–	73.95
DiffPure	85.2	78.41	77.82	76.1	79.85	79.74	77.22	77.06	76.72	75.81	80.04	79.15	75.98	81.19
Color	86.91	82.28	57.71	34.70	82.44	64.59	46.09	82.75	68.81	57.21	81.11	68.85	48.64	85.96
var-nmODE	50.21	–	58.52	–	–	57.02	–	–	45.30	–	–	–	–	–
Ours	91.56	87.88	74.03	49.90	88.22	77.58	58.65	88.16	81.89	72.43	87.95	83.47	70.12	90.36

4.3 Black-Box Testing

For black-box attack testing, we employ ResNet-18 as the substitute model. Adversarial examples were generated using the same five attack methods as in the gray-box experiments: FGSM, PGD, BIM, AA, and C&W. In Table 3, the upper row of each entry reports the classification accuracy of the adversarial examples on the original undefended model, while the lower row shows the accuracy of the defended model on the same adversarial inputs. As indicated by the data in the table, the decision boundaries of the original model are severely compromised when subjected to adversarial attacks. This degradation causes the classification performance to drop to the level of random guessing, with the average accuracy falling to approximately 10%. Notably, once the perturbation magnitude reaches the critical threshold at which the model fails (e.g., $\epsilon = 2/255$ for FGSM), further increasing the perturbation budget ($\epsilon = 8/255$ or $16/255$) does not lead to a substantial additional drop in accuracy. After incorporating the proposed adversarial defense, the robustness of the target models is significantly improved. For example, under a PGD attack with $\epsilon = 2/255$, the original SENet model achieves an accuracy of only 10.55%, whereas the defended model attains an accuracy of 87.39%. When the perturbation strength increases (e.g., $\epsilon = 16/255$), the adversarial noise may exceed the robustness threshold of the pixel transformation, leading to a degradation in defense performance. In such cases, the structure of some adversarial examples is not completely disrupted and can still deceive the mode. Compared with the original model, the defended model continues to provide a non-negligible level of protection.

Table 3: Original and defense models achieve accuracy (%) under black-box attack conditions. For each model, the first row is the original model, and the second is the defense model (Bold values).

Model		FGSM				PGD			BIM		AA		CW	
SeNet	Original	10.92	10.76	10.32	10.55	10.71	10.09	11.13	10.58	10.63	11.02	11.04	10.64	10.57
	Defense	86.47	73.61	51.25	87.39	76.12	57.57	87.16	81.08	71.71	86.83	82.50	69.87	89.33
RegNet	Original	10.68	10.62	10.81	10.48	10.50	10.46	10.59	10.50	10.71	10.57	10.57	10.66	10.64
	Defense	88.57	74.73	49.39	89.16	78.45	59.17	88.99	82.75	73.42	88.71	84.39	70.41	91.21
Goog LeNet	Original	10.17	10.07	10.17	10.14	10.23	9.85	10.22	9.79	9.88	9.98	10.01	9.98	9.92
	Defense	88.48	73.65	48.40	88.98	77.07	56.18	88.87	81.67	71.07	88.76	83.25	69.50	91.47
VGG	Original	10.65	10.58	10.19	10.28	10.17	9.99	10.25	10.18	9.96	10.22	10.27	9.85	10.30
	Defense	83.64	75.58	60.16	84.09	78.01	66.74	84.25	80.18	74.67	83.87	81.10	73.56	85.23
Wide-ResNet	Original	10.40	10.33	10.48	10.20	10.09	10.03	10.31	10.21	10.10	10.35	10.22	10.08	10.39
	Defense	89.68	74.13	48.97	90.46	77.88	55.30	90.45	83.39	72.36	90.15	85.29	70.35	92.27

4.4 White-Box Testing

In adversarial defense research, white-box testing refers to scenarios in which the attacker has full knowledge of the defense model's architecture, parameters, and defense mechanisms, and generates optimal adversarial examples based on this information. In this study, white-box attacks are conducted under two settings: attacks applied before the pixel transformation and attacks applied after the pixel transformation.

4.4.1 White-Box Attacks before Pixel Transformation

We first conducted experiments on white-box attacks applied prior to pixel transformation, using the five attack methods described above under an L_2 -norm constraint with a perturbation budget of $\epsilon = 0.3$. The C&W attack is based on the L_2 norm constraint with a perturbation parameter of $c = 0.3/255$. Adversarial examples were generated for all six classification, resulting in a total of $5 \times 6 = 30$ adversarial sets.

As shown in Table 4, across different attack algorithms, the majority of adversarial perturbations were nearly neutralized after undergoing pixel and spatial feature reconstruction, allowing the models to maintain high classification accuracy. Compared with the performance on clean examples, the adversarial examples generated under white-box settings caused only minor decreases in accuracy, with overall reductions controlled within 3%. This indicates that the defense method can ensure adversarial robustness while still keeping near normal recognition performance.

Table 4: Accuracy (%) of defense models (rows) under white-box attacks (columns) on adversarial examples before pixel transformation.

	BIM	FGSM	PGD	AA	CW	Clean
ResNet	88.08	89.12	87.39	88.99	90.19	91.56
SeNet	87.17	88.18	87.78	87.18	89.01	90.54
RegNet	90.17	89.27	90.2	89.78	90.9	92.06
VGG	84.35	84.69	84.52	85.14	85.33	86.66
GoogLeNet	90.8	90.78	90.73	90.9	91.38	92.61
Wide-ResNet	91.23	91.27	90.51	91.55	92.07	93.31

4.4.2 White-Box Attacks after Pixel Transformation

Perturbation visibility and classification prevention are two distinct concepts. The proposed defense model fails to correctly classify white-box adversarial examples generated after pixel transformation. However, the inverse transformation disrupts the imperceptibility of the adversarial perturbations, rendering them visible to the human eye. This destroys the characteristic of adversarial examples being imperceptible to humans. As shown in Fig. 5, adding adversarial noise after pixel transformation leads to color changes across the entire image once the inverse transformation is applied.

					LPIPS
Clean examples and adversarial examples					0.0061
Clean examples and adversarial examples after pixel transformation					0.0008
Clean examples and examples after pixel transformation and inverse transformation					0.000419
Clean examples and examples subjected to white-box attacks after pixel transformation and then inverse transformation					0.495923

Figure 5: LPIPS values before and after adversarial attacks under different pixel transformation settings.

To quantitatively assess the perceptibility of adversarial examples generated after pixel transformation and subsequently reversed, we employ the Learned Perceptual Image Patch Similarity (LPIPS) metric. LPIPS [39] measures the perceptual similarity between two images and has become a key evaluation tool in the image processing domain. Its main advantage lies in leveraging high level features extracted from pretrained networks, making it more aligned with human visual perception compared to traditional metrics such as PSNR or SSIM. Eq. (14) formalizes the computation of the perceptual distance between the original image x and its transformed counterpart x_0 using a network F . Features are extracted from layer l and normalized along the channel dimension to produce representations $\hat{y}^l, \hat{y}_0^l \in \mathbb{R}^{H_l \times W_l \times C_l}$. These features are then scaled channel wise by a vector $w_l \in \mathbb{R}^{C_l}$, followed by computation of the L_2 distance. The resulting distances are spatially averaged and summed across channels.

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \odot (\hat{y}_{hw}^l - \hat{y}_{0hw}^l)\|_2^2 \quad (14)$$

Leveraging this perceptual metric, Karras et al. demonstrated that the improved StyleGAN2 model yields LPIPS values below 0.2 in most cases when comparing generated images to their projected counterparts [40], validating its effectiveness in preserving source image attributes. This finding establishes 0.2 as a natural perceptual boundary in the static image domain: LPIPS values exceeding this threshold correspond to visually perceptible differences detectable by human observers. Hou et al. further corroborated the validity of this perceptual range from a complementary perspective, reporting that when the LPIPS difference between two images exceeds 0.15, consistency with human subjective perception reaches 97.6% [41]. Although the two thresholds were derived in distinct research contexts, image generation and video frame interpolation. Respectively, they are consistent in magnitude and mutually reinforce the conclusion that LPIPS values in this range reliably indicate humanly perceptible distortion. We therefore adopt 0.2 as our reference criterion,

as it was established under experimental conditions most closely aligned with our static image defense setting, and because it represents a more conservative bound whose exceedance provides stronger evidence of perceptual distinguishability. In our experiments, the LPIPS distance between the original clean image and the inversely-transformed white-box adversarial example reaches $0.4959\text{--}2.48\times$ above this threshold, confirming that the pixel-level distortions introduced by our checkerboard transformation under white-box attack are unambiguously perceptible to human observers, irrespective of which threshold serves as the reference.

We computed the LPIPS distances between the inversely transformed adversarial images and the original clean images. Across 10,000 image pairs, the mean perceptual distance was 0.4959, as shown in the fourth row of Fig. 5, significantly exceeding the human perceptibility threshold. In the figure, the first row represents the LPIPS distance between clean examples and standard adversarial examples; the second row corresponds to clean and adversarial examples after pixel transformation; the third row shows the distance between original clean examples and those subjected to pixel transformation followed by inverse transformation, where the nonzero distance is due to the irreversible nature of bit depth feature compression. Fig. 6 provides representative visual examples.



Figure 6: Comparison of some CIFAR-10 sample images. (a) Original Images, (b) images after white-box attacks on pixel transformed inputs followed by inverse transformation.

4.5 Adaptive Attack Testing

To rigorously evaluate the robustness of the proposed defense method, we adopt adaptive attacks based on Backward Pass Differentiable Approximation (BPDA) [42] combined with Expectation Over Transformation (EOT) [43]. In the experimental setting, BPDA replaces the non differentiable transformations with the identity function during backpropagation. Since the transformations in the proposed method are deterministic operations, EOT under this setting is equivalent to a standard gradient based attack on a fixed transformation pipeline. We nevertheless retain the EOT framework for consistency with the baseline methods, and the number of EOT samples is set to 20. Under five evaluation settings, we compare the proposed method with two baseline defenses, DeR [44] and Color [33]. All attacks are conducted under L_∞ , $\epsilon = 8/255$ with 30 PGD iterations and 20 EOT samples. The results in Table 5 demonstrate that the proposed defense achieves substantially stronger robustness against adaptive attacks than the baseline methods. As the number of attack iterations increases, the attack success rate rises monotonically and the model accuracy correspondingly decreases. Furthermore, we observe that the accuracy under black box transfer attacks (Table 3) is higher than that under white box BPDA attacks (54.69%), which is consistent with the expectation

that white box attacks should be stronger than black box attacks. The pixel level remapping changes the distribution of perturbations across channels, leading perturbations crafted in the original space to lose alignment in the transformed space. Therefore, the robustness of the proposed method originates from the genuine disruption of the input space rather than merely obscuring gradients.

Table 5: Defense accuracy (%) of different methods under various attacks. Bold values indicate the best performing model under each column's condition.

	None	FGSM + BPDA	PGD + BPDA	FGSM + BPDA + EOT	PGD + BPDA + EOT
DeR	86.10	–	–	38.30	19.50
Color	86.91	50.00	39.84	44.69	25.66
Ours	91.56	72.03	54.69	67.27	35.35

4.6 Ablation study

4.6.1 Module Ablation

To further validate the effectiveness of the proposed dual-mode pixel transformation strategy, ablation studies were conducted on the two constituent submodules: chessboard style feature squeezing ($bits = 4$) and high intensity pixel transformation. Fig. 7 presents qualitative visualizations and quantitative performance comparisons under three different pixel processing schemes. It can be observed that when feature squeezing is applied alone, the model achieves slightly higher classification accuracy on clean examples than the dual-mode strategy; however, it exhibits pronounced vulnerability to adversarial examples, particularly under strong perturbations with $\epsilon \geq 8/255$, where the defense performance deteriorates rapidly. In contrast, applying high intensity pixel transformation alone improves robustness to a certain extent, yet its overall performance remains inferior to that of the combined dual-mode approach.

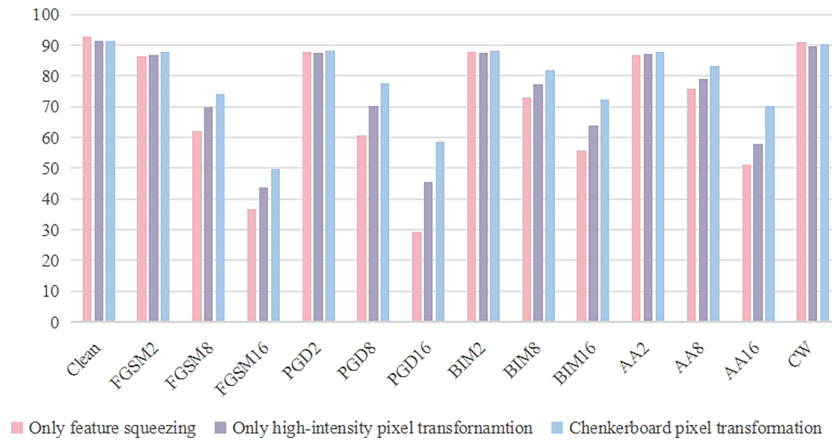


Figure 7: Experimental results using each pixel transformation method individually.

To assess the contribution of the spatial transformation module within the overall defense framework, we conduct comparative experiments on three spatial transformation strategies: (i) no spatial transformation, (ii) row wise transformation only, and (iii) joint row and column transformation. Fig. 8 reports the classification performance on clean examples and the robustness under different attack strengths for the three schemes. The results show that the model without spatial transformation attains slightly higher accuracy on clean examples; however, its defensive capability is markedly insufficient, particularly under

high intensity attacks. For instance, under a BIM attack with a perturbation magnitude of 16/255, the defense performance of this scheme drops by 27.25% compared with the complete model. In contrast, applying row wise transformation alone improves robustness to some extent across varying attack strengths, yet its overall performance remains inferior to that of the full spatial transformation module that jointly applies row and column transformations.

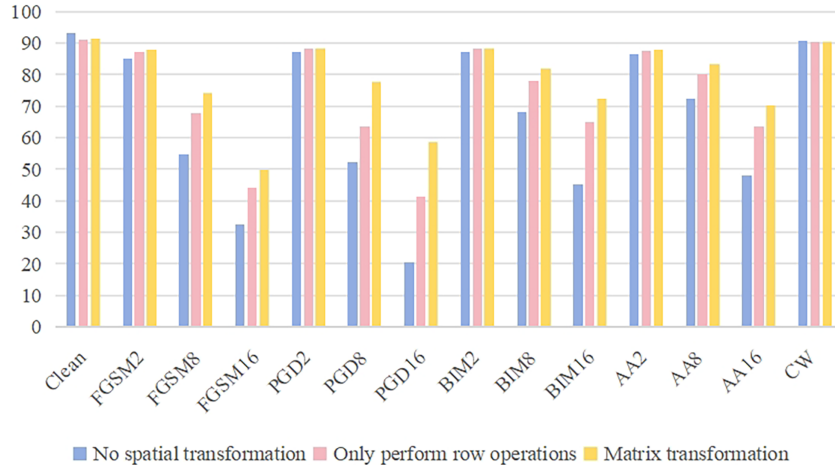


Figure 8: Experimental results under different spatial transformation modes.

4.6.2 Feature Squeezing Strategy

To further analyze the impact of feature squeezing bit depth on defense performance, we conduct a systematic evaluation of feature compression models with different bits ($bits = 1 \sim 8$) under a unified experimental setting. We evaluate the defense using the five attack methods mentioned above. FGSM, PGD, BIM, and AA are based on the L_∞ norm with $\epsilon = 8/255$. The C&W attack is based on the L_2 norm constraint with a perturbation parameter of $c = 0.3/255$. As illustrated in Fig. 9, when $bits = 4$, the model achieves the highest classification accuracy on clean examples and exhibits the best overall defensive performance against BIM, AA, and C&W attacks. In contrast, when $bits = 3$, although a slight robustness improvement is observed under FGSM and PGD attacks, this gain comes at the expense of clean example accuracy. With $bits = 3$, the model attains only 89.36% accuracy on clean examples, representing a 4.6% drop compared to the original model.

Fig. 1 provides an intuitive visualization of the discrepancy in pixel information preservation between $bits = 3$ and $bits = 4$. The $bits = 3$ setting discards substantially more pixel-level details, leading to more severe visual degradation. When $bits = 8$, without feature compression, the model attains slightly higher accuracy on clean examples than with $bits = 4$; however, its defense performance is consistently inferior to that achieved with $bits = 4$. To validate the selection of the 4-bit quantization depth, we conducted paired t -tests across all evaluation metrics. The results indicate that $bits = 4$ achieves significantly superior overall performance compared to $bits = 5$ ($p = 0.00024 < 0.001$), while the difference between $bits = 4$ and $bits = 3$ does not reach statistical significance ($p = 0.064$). Nevertheless, the $bits = 3$ configuration incurs an approximate 2.2% degradation in clean accuracy relative to $bits = 4$. Given that lower bit depths introduce greater risk of information loss despite higher compression ratios, $bits = 4$ represents a more principled engineering trade-off between classification fidelity and feature compression strength, and is therefore adopted as the default quantization depth in our method.

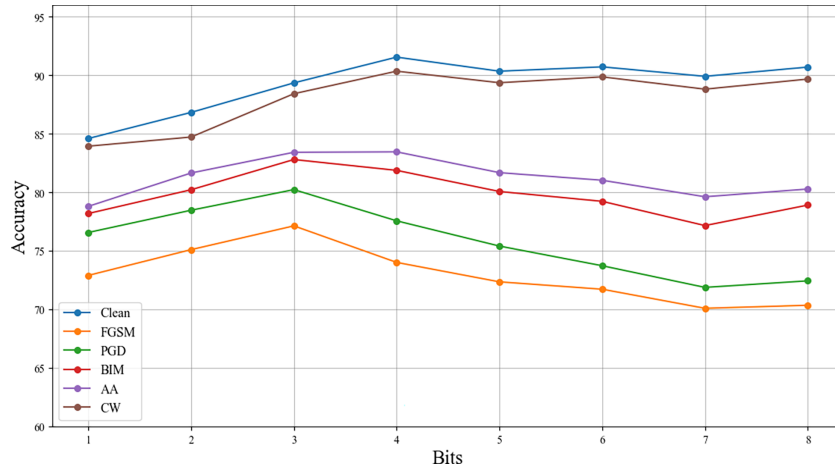


Figure 9: Defense performance across different pixel compression bit depths.

4.6.3 Spatial Transformation Scheme

Each particle's position is initialized by uniform random sampling over the continuous search space, i.e., $\mathbf{z}_i \sim \mathcal{U}(0,1)^{16}$, and the initial velocity is sampled from $\mathbf{v}_i \sim \mathcal{U}(-0.1, 0.1)^{16}$. Each evaluation involves training a ResNet-18 model for 35 epochs on the transformed training set, followed by inference on three test sets. All experiments were conducted on a single NVIDIA RTX 3060 GPU. Each fitness evaluation per particle requires approximately 0.96 h. Also, the inertia weight decreases linearly from $w_{max} = 0.9$ to $w_{min} = 0.4$ across iterations to balance global exploration and local exploitation. The velocity of each particle is clamped to $[-v_{max}, v_{max}]$ with $v_{max} = 0.2$. The archive capacity is set to 50 with crowding distance truncation to preserve diversity. Optimization terminates early if the archive size varies by no more than 2 individuals over 4 consecutive iterations. The particle swarm size is set to 30, and the number of iterations is set to 20, resulting in a total of 600 fitness evaluations. The selection of these parameters is primarily governed by the substantial computational cost, given that every fitness evaluation requires a full model retraining. Wang et al. noted that, following the common practice in the evolutionary computation community for computationally expensive optimization tasks [45], the swarm size is typically selected within the range of 20 to 50 particles. Due to time constraints, they adopted a configuration of 20 particles over 20 generations. Therefore, we selected an intermediate setting with 30 particles over 20 generations.

We further visualize the optimization process of the MOPSO algorithm. As shown in Fig. 10, the reported results correspond to the defense model associated with the best transformation scheme at the current iteration, evaluated in terms of Acc_{clean} , Acc_{FGSM} , Acc_{PGD} and Acc_{AA} . The optimal scheme is selected according to the accuracy trade-off criterion $0.5 \times Acc_{clean} + 0.5 \times \text{average}(Acc_{FGSM}, Acc_{PGD}, Acc_{AA})$, with statistics recorded every 30 iterations. As can be observed from Fig. 10, the optimization process gradually converges and becomes stable in the later stages. Compared with the suboptimal solution, the final transformation scheme adopted in this work results in only a 0.65% decrease in clean examples classification accuracy, while achieving an average improvement of 2.17% in defensive performance.

Fig. 11 illustrates the evolution of the hypervolume (HV) indicator throughout the MOPSO optimization process. The HV value increases rapidly during the first 240 iterations and stabilizes after iteration 480, with a relative change rate below 0.1% over iterations 480–600, indicating that the Pareto front has reached sufficient convergence.

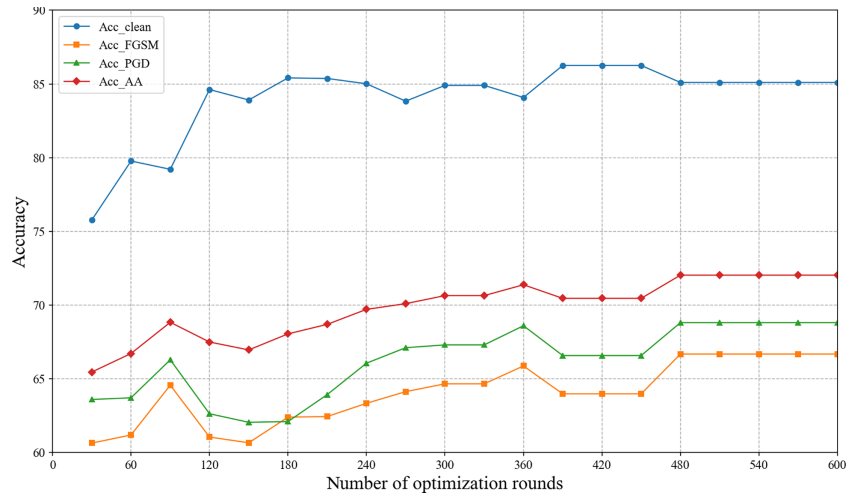


Figure 10: Visualization of the MOPSO optimal solution selection process.

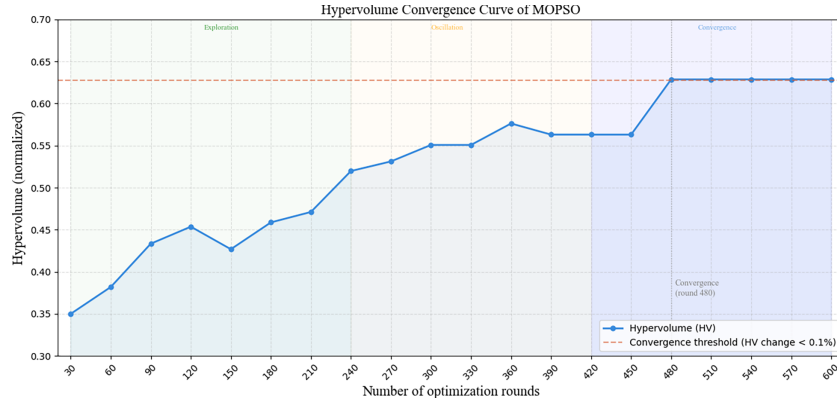


Figure 11: MOPSO process hypervolume iteration curve.

4.6.4 Additional Dataset

To evaluate the generalizability of the proposed method on higher resolution datasets, we conduct additional experiments on the mini-ImageNet dataset using ResNet18. mini-ImageNet is a subset of ImageNet comprising 60,000 images across 100 categories, with 600 images per category. The experimental setup follows that of CIFAR-10, with attack configurations fixed at L_∞ , $\epsilon = 8/255$, and the number of iterations for both PGD and BIM set to 30. As reported in Table 6, the undefended model is almost completely compromised under white-box attacks, whereas the proposed defense method restores classification accuracy by approximately 50%. Under black-box attack settings, the defense gain exceeds 35% across all evaluated attack methods. These results demonstrate that the proposed spatial transformation and pixel transformation strategies remain effective in disrupting adversarial perturbations in higher resolution images.

4.7 Computational Cost Analysis

All three preprocessing modules exhibit a computational complexity of $O(HWC)$. The feature compression module performs pixel level bit depth quantization, the checkerboard transformation relies on integer operations, and the spatial transformation is implemented through two tensor indexing rearrangement

operations. For a standard $32 \times 32 \times 3$ input, the total preprocessing overhead is less than 0.25 ms on a modern GPU.

The MOPSO based parameter search is conducted offline before deployment. Once the optimal permutation is determined, it remains fixed during inference and introduces no additional computational cost. Compared with diffusion based defense methods such as DiffPure, which require hundreds of neural function evaluations per image and incur inference latency on the order of several seconds, the proposed method achieves substantially higher efficiency. Furthermore, since the defense is implemented as a pre-processing wrapper without modifying the model architecture, it can be directly integrated with existing classifiers, making it suitable for practical deployment scenarios.

Table 6: Classification accuracy (%) on mini-ImageNet under white-box and black-box adversarial attacks

	Clean	White-Box Attack			Black-Box Attack		
		FGSM	PGD	BIM	FGSM	PGD	BIM
Original	78.81	3.18	0	0	19.16	14.71	16.96
Defense	71.00	52.45	50.71	49.01	60.08	57.22	55.41

5 Conclusion

we propose a defense framework that integrates a checkerboard pixel-level transformation with a spatial transformation to effectively counter diverse types of adversarial perturbations. While preserving the classification accuracy of the model on clean examples, the proposed method substantially enhances robustness against high intensity adversarial attacks. Extensive experimental results demonstrate that our approach consistently defends against gray-box, black-box, and white-box adversarial attacks across a wide range of attack methods.

For white-box attacks conducted after pixel transformation, adversarial perturbations can be amplified through an inverse transformation, rendering them perceptible to the human eye. With respect to the spatial transformation component, we employ MOPSO to identify transformation parameters that achieve an optimal trade-off between clean examples accuracy and adversarial robustness. The defense models trained using our approach maintain clean examples classification accuracy above 90%. It is worth noting that the result under the BPDA+EOT attack (35.35%) represents a realistic upper bound for a fully informed adversary; the primary strength of the proposed method lies in its effectiveness under practical black-box and gray-box threat models. In future work, we aim to establish a more unified theoretical framework for the proposed transformations and derive more principled transformation strategies through theoretical analysis, with the goal of further improving model robustness while preserving a higher degree of the original image information. We also plan to analyze the influence of checkerboard transformations on the adversarial optimization space and derive theoretical bounds for the gradient misalignment introduced by spatial permutations.

Acknowledgement: Not applicable

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Xiujuan Wang: Writing—review & editing, Supervision, Resources, Conceptualization. Jiaying Li: Writing—original draft, Validation, Software, Methodology, Visualization, Conceptualization, Formal analysis. Shuhan Han: Data curation, Visualization. Liya Xu: Supervision, Validation. Changxing Wang: Methodology, Validation. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. The data that support the findings of this study are openly available in [cifar-10] at <https://tensorflow.google.cn/datasets/catalog/cifar10>.

Ethics Approval: Not applicable.

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

References

1. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations; 2015 May 7–9; San Diego, CA, USA. p. 1–11.
2. Kurakin A, Goodfellow IJ, Bengio S. Adversarial examples in the physical world. In: Artificial intelligence safety and security. Boca Raton, FL, USA: Chapman and Hall/CRC; 2018. p. 99–112.
3. Shaye GG, Zabil MHM, Habeeb MA, Khaleel YL, Albahri AS. Strategies for protection against adversarial attacks in AI models: an in-depth review. *J Intell Syst.* 2025;34(1):20240277. doi:10.1515/jisys-2024-0277.
4. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6th International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada. p. 1–23.
5. Bai Y, Wang Y, Zeng Y, Jiang Y, Xia ST. Query efficient black-box adversarial attack on deep neural networks. *Pattern Recognit.* 2023;133(1):109037. doi:10.1016/j.patcog.2022.109037.
6. Vivek BS, Mopuri KR, Babu RV. Gray-box adversarial training. In: Proceedings of the 15th European Conference on Computer Vision; 2018 Sep 8–14; Munich, Germany. p. 213–28.
7. Han X, Yang S, Wang W, Li Y, Dong J. Probing unlearned diffusion models: a transferable adversarial attack perspective. *Pattern Recognit.* 2026;174(3):112916. doi:10.1016/j.patcog.2025.112916.
8. Kurakin A, Goodfellow IJ, Bengio S. Adversarial machine learning at scale. In: Proceedings of the 5th International Conference on Learning Representations; 2017 Apr 24–26; Toulon, France. p. 1–17.
9. Dziugaite GK, Ghahramani Z, Roy DM. A study of the effect of JPG compression on adversarial images. *arXiv:1608.00853.* 2016.
10. Mosbach M, Andriushchenko M, Trost T, Hein M, Klakow D. Logit pairing methods can fool gradient-based attacks. *arXiv:1810.12042;* 2018.
11. Talvitie E. Model regularization for stable sample rollouts. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence; 2014 Jul 23–27; Quebec City, QC, Canada. p. 780–9.
12. Liao F, Liang M, Dong Y, Pang T, Hu X, Zhu J. Defense against adversarial attacks using high-level representation guided denoiser. In: Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–22; Salt Lake City, UT, USA. p. 1778–87.
13. Xie C, Wang J, Zhang Z, Ren Z, Yuille A. Mitigating adversarial effects through randomization. In: Proceedings of the 6th International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada. p. 1–16.
14. Croce F, Hein M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of the 37th International Conference on Machine Learning; 2020 Jul 13–8; Virtual. p. 2206–16.
15. Croce F, Hein M. Minimally distorted adversarial examples with a fast adaptive boundary attack. In: Proceedings of the 37th International Conference on Machine Learning; 2020 Jul 13–8; Virtual. p. 2196–205.
16. Agnihotri S, Jung S, Keuper M. CosPGD: an efficient white-box adversarial attack for pixel-wise prediction tasks. In: Proceedings of the 41st International Conference on Machine Learning; 2024 Jul 21–27; Vienna, Austria. p. 416–51.
17. Carlini N, Wagner D. Towards evaluating the robustness of neural networks. In: Proceedings of the 2017 IEEE Symposium on Security and Privacy; 2017 May 22–26; San Jose, CA, USA. p. 39–57.
18. Papernot N, McDaniel P, Goodfellow IJ, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security; 2017 Apr 2–6; Abu Dhabi, United Arab Emirates. p. 506–19.

19. Wang X, He K. Enhancing the transferability of adversarial attacks through variance tuning. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2021 Jun 19–25; Virtual Event. p. 1924–33. doi:10.1109/cvpr46437.2021.00196.
20. Qiu H, Custode L, Iacca G. Black-box adversarial attacks using evolution strategies. In: Proceedings of the 2021 Genetic and Evolutionary Computation Conference Companion; 2021 Jul 10–14; Lille, France. p. 1827–33.
21. Li C, Yao W, Wang H, Jiang T, Zhang X. Bayesian evolutionary optimization for crafting high-quality adversarial examples with limited query budget. *Appl Soft Comput.* 2023;142(5):110370. doi:10.1016/j.asoc.2023.110370.
22. Li C, Jiang T, Wang H, Yao W, Wang D. Optimizing latent variables in integrating transfer and query based attack framework. *IEEE Trans Pattern Anal Mach Intell.* 2024;47(1):161–71. doi:10.1109/tpami.2024.3461686.
23. Rade R, Moosavi-Dezfooli SM. Helper-based adversarial training: reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In: Proceedings of the Tenth International Conference on Learning Representations; 2021 Apr 25–29; Virtual.
24. Laidlaw C, Singla S, Feizi S. Perceptual adversarial robustness: defense against unseen threat models. arXiv:2006.12655. 2020.
25. Goyal S, Rebuffi SA, Wiles O, Stumberg F, Calian DA, Mann TA. Improving robustness using generated data. *Adv Neural Inf Process Syst.* 2021;34:4218–33.
26. Chen J, Cheng Y, Gan Z, Gu Q, Liu J. Efficient robust training via backward smoothing. *Proc AAAI Conf Artif Intell.* 2022;36(6):6222–30. doi:10.1609/aaai.v36i6.20571.
27. Guo C, Rana M, Cisse M, van der Maaten L. Countering adversarial images using input transformations. In: Proceedings of the 6th International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada. p. 1–12.
28. Xu W, Evans D, Qi Y. Feature squeezing: detecting adversarial examples in deep neural networks. In: Proceedings of the 25th Annual Network and Distributed System Security Symposium; 2018 Feb 18–21; San Diego, CA, USA. p. 1–15.
29. Prakash A, Moran N, Garber S, DiLillo A, Storer J. Deflecting adversarial attacks with pixel deflection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–22; Salt Lake City, UT, USA. p. 8571–80.
30. Shi C, Holtz C, Mishne G. Online adversarial purification based on self-supervised learning. In: Proceedings of the 9th International Conference on Learning Representations; 2021 May 3–7; Virtual Event. p. 1–5.
31. Li J, Zhang S, Cao J, Tan M. Learning defense transformations for counterattacking adversarial examples. *Neural Netw.* 2023;164:177–85. doi:10.1016/j.neunet.2023.03.008.
32. Hwang D, Lee E, Rhee W. Aid-purifier: a light auxiliary network for boosting adversarial defense. *Neurocomputing.* 2023;541:126251.
33. Wang H, Wu C, Zheng K. Defense against adversarial attacks based on color space transformation. *Neural Netw.* 2024;173(8):106176. doi:10.1016/j.neunet.2024.106176.
34. Zhao Y, Zheng J, Gao X, Liu L, Zhang Y, Zhang Q. Enhancing the transferability of adversarial attacks via scale enriching. *Neural Netw.* 2025;189(11):107549. doi:10.1016/j.neunet.2025.107549.
35. Coello CC, Lechuga MS. MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600); 2002 May 12–17; Honolulu, HI, USA. p. 1051–6.
36. Tsipras D, Santurkar S, Engstrom L, Turner A, Madry A. Robustness may be at odds with accuracy. In: Proceedings of the 7th International Conference on Learning Representations; 2019 May 6–9; New Orleans, LA, USA. p. 1–23.
37. Nie W, Guo B, Huang Y, Xiao C, Vahdat A, Anandkumar A. Diffusion models for adversarial purification. In: Proceedings of the 39th International Conference on Machine Learning; 2022 Jul 17–23; Baltimore, MD, USA. p. 16805–27.
38. Yang Q, Hu X, Yu J, Sun Q, Shu L, Yi Z, et al. Model with L2-stability based on nmODE for defending against adversarial attacks. *Neurocomputing.* 2025;648:130605. doi:10.2139/ssrn.5165687.

39. Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 586–95.
40. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T. Analyzing and improving the image quality of styleGAN. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 8110–9.
41. Hou Q, Ghildyal A, Liu F. A perceptual quality metric for video frame interpolation. In: European Conference on Computer Vision. Cham, Switzerland: Springer Nature; 2022. p. 234–53.
42. Athalye A, Carlini N, Wagner D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning; 2019 Jul 10–15; Stockholm, Sweden. p. 274–83.
43. Athalye A, Engstrom L, Ilyas A, Kwok K. Synthesizing robust adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning; 2019 Jul 10–15; Stockholm, Sweden. p. 284–93.
44. Pang Z, Yan X, Guo S, Lu Y. Diversity-enhanced reconstruction as plug-in defenders against adversarial perturbations. *Front Artif Intell.* 2025;8:1665106. doi:10.3389/frai.2025.1665106.
45. Wang B, Sun Y, Xue B, Zhang M. Evolving deep neural networks by multi-objective particle swarm optimization for image classification. In: Proceedings of the Genetic and Evolutionary Computation Conference; 2019 Jul 13–17; Prague, Czech Republic. p. 490–8.