

# OrthoCNN: Mitigating Adversarial Noise in Convolutional Neural Networks via Orthogonal Projections

Aristeidis Bifis<sup>a</sup> and Emmanouil Psarakis<sup>b</sup>

*Computer Engineering & Informatics Department, University of Patras, Patras, Greece*  
fi

**Keywords:** Adversarial Defense, Adversarial Training, Neural Network Robustness, Adversarial Robustness, Deep Learning, Convolutional Layers, Null-Space Projection, Range-Space Projection, Orthogonal Projection, PGD, White Box, Feature Manipulation.

**Abstract:** Adversarial training is the standard method for improving the robustness of neural networks against adversarial attacks. However, a well-known trade-off exists: while adversarial training increases resilience to perturbations, it often results in a significant reduction in accuracy on clean (unperturbed) data. This compromise leads to models that are more resistant to adversarial attacks but less effective on natural inputs. In this paper, we introduce an extension to adversarial training by applying novel constraints on convolutional layers, that address this trade-off. Specifically, we use orthogonal projections to decompose the learned features into clean signal and adversarial noise, projecting them onto the range and null spaces of the network's weight matrices. These constraints improve the separation of adversarial noise from useful signals during training, enhancing robustness while preserving the same performance on clean data as adversarial training. Our approach achieves significant improvements in robust accuracy while maintaining comparable clean accuracy, providing a balanced and effective adversarial defense strategy.


## 1 INTRODUCTION


Adversarial attacks pose a significant threat to the reliability and security of neural networks, particularly in critical real-world applications such as autonomous driving, healthcare, and finance (Wu et al., 2023; Selvakumar et al., 2022; Chen et al., 2021). These attacks, which introduce small but deliberate perturbations to input data, can lead to incorrect predictions or system failures, undermining the trustworthiness of AI systems. Although adversarial training has become the standard defense mechanism, it often results in a trade-off: models become more robust to adversarial perturbations but suffer decreased performance on clean (unperturbed) data (Tsipras et al., 2018; Zhang et al., 2019). This trade-off limits the practicality of adversarially robust models, highlighting the need for methods that enhance robustness without sacrificing accuracy on natural inputs. Such methods are crucial for improving the overall reliability and usability of AI systems in real-world scenarios.

The standard adversarial training framework (Goodfellow et al., 2014) seeks to improve the robust-

ness of machine learning models by explicitly training them on adversarial examples—inputs that have been intentionally perturbed to mislead the model. The core idea is to augment the training data with these adversarial examples, forcing the model to learn from both the original clean data and the perturbations that challenge its decision boundaries. This process typically involves generating adversarial examples using methods like the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) or Projected Gradient Descent (PGD) (Mađry et al., 2017) and incorporating them into the training procedure. The goal is to enhance the model's ability to classify both clean and adversarial inputs correctly, thereby increasing its resilience to attacks. By exposing the model to a diverse set of adversarial perturbations, adversarial training helps it develop more stable decision boundaries, ultimately improving its generalization and robustness against malicious attacks.

A significant limitation of existing adversarial training methods is the trade-off between improving robustness to adversarial attacks and maintaining clean accuracy on unperturbed data. While adversarial training helps models become more resilient to adversarial examples by exposing them to perturbations during training, it often leads to a degradation in clean

<sup>a</sup>  <https://orcid.org/0000-0003-0246-1209>

<sup>b</sup>  <https://orcid.org/0000-0002-9627-0640>

accuracy—the model’s performance on natural, unmodified inputs. This occurs because the model’s decision boundaries are adjusted to become more robust to adversarial perturbations, sometimes at the cost of over fitting to the adversarial examples or losing its ability to generalize well on normal data. This trade-off represents a key challenge in adversarial training, as improving robustness to attacks can compromise the model’s overall performance, making it less effective in real-world, unperturbed scenarios. Researchers are actively exploring ways to mitigate this issue, such as through more sophisticated loss functions, regularization techniques, or hybrid training strategies that balance both clean and adversarial accuracy.

This paper introduces a novel extension to the adversarial training (AT) framework by applying orthogonal projection constraints to the clean signal and adversarial noise representations, mapping them onto the range and null spaces of the network’s weight matrices. This approach separates the adversarial perturbations from the clean data signal during training, effectively enhancing the model’s robustness to adversarial attacks while preserving its performance on unperturbed data. By doing so, it addresses the common trade-off between adversarial defense and clean accuracy, offering a more balanced and practical solution.

## 2 RELATED WORK

Adversarial training using Projected Gradient Descent (PGD) has become one of the most widely adopted techniques for improving model robustness against adversarial attacks. In (Mađry et al., 2017), authors demonstrated the effectiveness of PGD-based adversarial training, showing that iteratively applying adversarial perturbations during training could significantly improve model performance on adversarial examples. PGD attacks involve multiple steps of gradient updates, which are projected onto a specified norm ball to ensure the adversarial perturbations remain within a predefined limit. The authors showed that this method could help train models that are robust to a variety of adversarial attacks, particularly white-box attacks like PGD itself, providing a foundation for many subsequent works in adversarial defense (Rade and Moosavi-Dezfooli, 2022; Kumari et al., 2019; Sitawarin et al., 2021). The approach has been widely used and adapted to various architectures and datasets, becoming a standard benchmark for evaluating adversarial robustness. However, while PGD-based adversarial training is effective, it often involves a trade-off between robustness and clean data accuracy, as the process can lead to overfitting on the

adversarial perturbations.

TRADES (TRadeoff-inspired Adversarial Defense via Surrogate-loss minimization) (Zhang et al., 2019), is a prominent adversarial defense technique designed to improve the robustness of neural networks against adversarial attacks while preserving their generalization to clean data. TRADES introduces a novel approach to adversarial training by minimizing a surrogate loss that balances between adversarial robustness and clean data accuracy. Specifically, it incorporates a trade-off term that penalizes the difference in output distributions between clean and adversarial examples, using the Kullback-Leibler (KL) divergence to measure this discrepancy. The method encourages the model to behave similarly on both clean and adversarially perturbed data, which leads to improved performance against adversarial attacks, especially in terms of transferability and robustness in black-box settings. One of the key strengths of TRADES is its ability to improve the trade-off between adversarial robustness and clean accuracy, addressing a common challenge in adversarial training methods, where boosting one often results in a decline in the other. TRADES has shown superior performance over standard adversarial training methods, such as PGD-based training, by achieving better generalization and robustness. However, despite its effectiveness, TRADES introduces additional computational overhead and requires careful tuning of the trade-off parameter to maintain a balance between adversarial robustness and clean data accuracy. Subsequent research has extended TRADES by exploring alternative regularization techniques and improving its efficiency in large-scale models (Pang et al., 2022; Levi and Kontorovich, 2024).

In (Bifis et al., 2023) a novel adversarial defense strategy that leverages orthogonal constraints applied to denoising autoencoders (DAEs) was introduced. The proposed approach demonstrated that tied-weight DAEs, which have half the complexity of full-weight models, offer substantial improvements in adversarial robustness without compromising on computational efficiency. By enforcing orthogonality during training, the model becomes more resilient to adversarial perturbations while maintaining low inference overhead. Building upon this foundation, we extend that approach to more complex architectures, specifically exploring the application of that theory to convolutional layers. Furthermore, we are investigating the potential of applying the orthogonal constraints outside the denoising framework, broadening their applicability to other areas of adversarial defense.

In this paper limitations of the approach presented in (Bifis et al., 2023) are addressed; namely:

- its focus on applying constraints exclusively to

fully connected layers in smaller neural networks. This approach aimed to reduce the number of parameters while maintaining a comparable level of robustness to larger, more complex robust networks, but it restricted the scope of the technique in parameter-aware contexts, in addition

- its reliance on noise from known distributions to minimize the impact on training time, earning it the label of attack-agnostic. However, while the method performed as intended in black- and gray-box setups, it failed to achieve the same level of robustness as adversarial training in white-box scenarios.

### 3 PROBLEM FORMULATION

The objective of our technique is to train a neural network capable of substantially mitigating adversarial noise embedded in input signals. This approach improves classification accuracy on tampered data while preserving performance on clean data, achieving accuracy comparable to models trained exclusively on pristine inputs. The first step in extending the above-mentioned technique (Bifis et al., 2023) is to find out a way for applying constraints to layer types beyond the fully connected ones. This shift is necessary because, in large networks, fully connected layers are less effective at capturing localized features and are computationally inefficient. A more promising approach is to adapt the orthogonality constraints for convolutional layers, which are widely used in state of the art networks, computationally efficient, and well-suited for producing localized features. This process differs fundamentally from the matrix-vector multiplication performed in a network that is constructed by fully connected layers. Therefore, how can we establish and extend the theoretical framework presented in (Bifis et al., 2023) to convolutional layers?

In convolutional layers, their inputs and building blocks are represented by tensors. A convolutional layer, depending on the kind of its input can be seen as:

- **a feature map generator**, if the input is an image or
- **a feature map transform**, if the input is a feature map itself.

An input, independently of its kind, typically can be considered as a  $C_{in}$  channels **image** of size  $H_{in} \times W_{in}$  each. For example, a typical RGB image has  $C_{in} = 3$  and a gray-scale  $C_{in} = 1$  and they can be stored in an input tensor of appropriate size. The convolutional layer **kernels** (filters) can also be stored in tensors. The number of filters defines the number

of output channels  $C_{out}$  and each one can be denoted by an  $C_{in} \times H_{l_k} \times W_{l_k}$  tensor  $K_{c_{out}}$  with  $H_{l_k} \leq H_{in}$  and  $W_{l_k} \leq W_{in}$ . Thus, for a given input image or feature map  $\mathcal{X}$ ,  $\in \mathbb{R}^{C_{in} \times H_{in} \times W_{in}}$  from its convolution with the kernel  $K_{c_{out}}$  results the  $c_{out}$ -th output image or feature map  $\mathcal{Y}_{c_{out}}$  of size  $H_{out} \times W_{out}$  whose each element is a RV given by the following relation:

$$\mathcal{Y}_{c_{out}}(\mathbf{n}) = \sum_{c_{in}=1}^{C_{in}} \sum_{\mathbf{m} \in S_{kern}} \mathcal{X}_{c_{in}}(\mathbf{n} - \mathbf{m}) K_{c_{out}}(c_{in}, \mathbf{m}) \quad \mathbf{n} \in S_{out}, c_{out} = 1, \dots, C_{out} \quad (1)$$

with:

$$S_{kern} = \{[0, H_{l_k} - 1] \times [0, W_{l_k} - 1]\}$$

$$S_{out} = \{[0, H_{out} - 1] \times [0, W_{out} - 1]\}$$

the supports of the  $c_{out}$ -th kernel, and the output respectively, that can be equivalently written as follow:

$$\mathbf{y}_{c_{out}}^t = \mathbf{k}_{c_{out}}^t \mathcal{X}_r, c_{out} = 1, \dots, C_{out} \quad (2)$$

where  $\mathbf{y}_{c_{out}}^t$ ,  $\mathbf{k}_{c_{out}}^t$  the flatten versions of  $\mathcal{Y}_{c_{out}}$  and  $K_{c_{out}}$  of length  $H_{out}W_{out}$  and  $C_{in}H_{l_k}W_{l_k}$  respectively and  $\mathcal{X}_r$  an appropriate rearrangement of the input whose the size depends on the setting of the convolutional parameters, i.e., stride, dilation etc.. Using Eq. (2) the linear convolution can be expressed as the product of a deterministic matrix  $K$  of size  $C_{out} \times C_{in}H_{l_k}W_{l_k}$  with a random matrix  $\mathcal{X}_r$  of size  $C_{in}H_{l_k}W_{l_k} \times H_{out}W_{out}$  as follow:

$$\mathcal{Y} = K \mathcal{X}_r \quad (3)$$

with the random matrix  $\mathcal{Y}$  of size  $C_{out} \times H_{out}W_{out}$  and matrix  $K$  defined as follow:

$$\mathcal{Y} = \begin{pmatrix} \mathbf{y}_1^t \\ \mathbf{y}_2^t \\ \vdots \\ \mathbf{y}_{C_{out}}^t \end{pmatrix} \text{ and } K = \begin{pmatrix} \mathbf{k}_1^t \\ \mathbf{k}_2^t \\ \vdots \\ \mathbf{k}_{C_{out}}^t \end{pmatrix}. \quad (4)$$

Having defined the linear convolution as a multiplication of matrices, let us make some comments about the specific form of the random matrix  $\mathcal{Y}$  defined in Eq. (4), how it depends on the form of matrix  $K$  and how we can apply the desired constraints on the range and the null space of the filters coefficient matrix  $K$ .

#### 3.1 The Proposed Solution

For the purposes of this paper, we model adversarial attacks as the addition of a correlated perturbation (Goodfellow et al., 2014) to the input  $\mathcal{X}$ , that is:

$$\mathcal{X}_A = \mathcal{X} + \mathcal{W}_A \quad (5)$$

In our pipeline, as we can see from Fig. 1, we first apply a non-linear transformation to the inputs RVs.

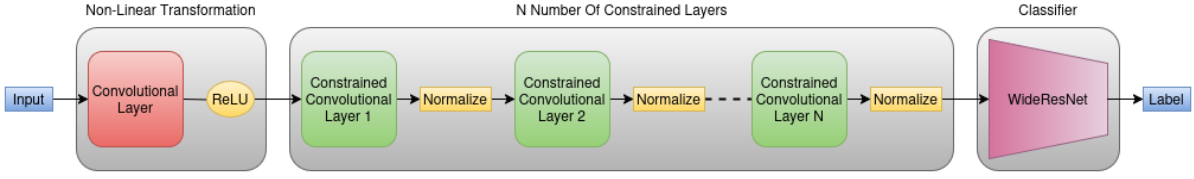


Figure 1: Proposed pipeline, consisting of three key components: (1) the non-linear transformation, (2) the  $N$  number of constrained layers, each followed by the output normalization step, where novel defense constraints are applied, and (3) the WideResNet classifier, which performs the final classification task.

The motivation behind this initial non-linear transformation is to enhance the data representation, improve the feature set, and address the complexity of adversarial noise, which follows intricate patterns.

To perform this nonlinear transformation, we use a simple convolutional layer followed by a non-linear activation function  $f(\cdot)$ . Consequently, using Eq. (3) the pristine data input  $\mathcal{X}$  and the adversarial input  $\mathcal{X}_A$  are non-linearly transformed to representations  $\mathcal{Z}$  and  $\mathcal{Z}_A$  respectively, as follow:

$$\mathcal{Z} = f_0(K\mathcal{X}_r) \quad (6)$$

$$\mathcal{Z}_A = f_0(K\mathcal{X}_{A,r}) \quad (7)$$

We can still claim that the non-linear representation relationship for the above mentioned transformed RVs is:

$$\mathcal{Z}_A = \mathcal{Z} + \mathcal{R} \quad (8)$$

where the term  $\mathcal{R}$  can be viewed as a residual noise perturbation that affects the attacked classifier, shifting the representation of the pristine data towards regions that lead to misclassification and incorrect results. We must stress at this point that the noise perturbation  $\mathcal{R}$  is correlated with the representation of the pristine data  $\mathcal{Z}$ . In order to quantify this dependency, let us rewrite Eq. (8) in a column-wise manner, i.e.:

$$\mathbf{z}_{A_l} = \mathbf{z}_l + \mathbf{r}_l, \quad l = 1, \dots, H_{out}W_{out} \quad (9)$$

Then, we have the following proposition.

**Proposition 1:** Let  $\mathbf{z}_l, \mathbf{z}_{A_l}$  be the non-linear representations of the pristine and adversarial attacked RVs respectively. Then, the following relation holds:

$$\mathbf{z}_{A_l} = (1 + \mu_l)\mathbf{z}_l + \mathbf{v}_l, \quad l = 1, \dots, H_{out}W_{out} \quad (10)$$

with constant  $\mu_l$  bounded by unity and defined by:

$$\mu_l = \frac{\langle \mathbf{z}_{A_l} - \mathbf{z}_l, \mathbf{z}_l \rangle}{\|\mathbf{z}_l\|_2 \|\mathbf{z}_{A_l} - \mathbf{z}_l\|_2}$$

with  $\langle \cdot, \cdot \rangle$  denoting the inner product operator, and the vectorized RV  $\mathbf{v}_l$  being orthogonal to  $\mathbf{z}_l$ , that is,  $\langle \mathbf{z}_l, \mathbf{v}_l \rangle = 0$ .

**Proof:** The proof is easy and thus omitted.  $\square$

We are going to exploit Proposition 1 in order to properly define the activation functions of the next

convolutional layers shown in Fig. 1. The output of these layers can be defined as follow:

$$Q_k = f_k(K_k Q_{k-1}), \quad k = 1, 2, \dots, N \quad (11)$$

with RV  $Q_k$  being either  $\mathcal{Z}_k$  and  $Q_0 = \mathcal{Z}$ , if the pristine data feed the input of the DNN, or  $\mathcal{Z}_{A_k}$  with  $Q_0 = \mathcal{Z}_A$  if the adversarial one, and  $f_k(\cdot)$ ,  $K_k$  the activation function, that acts in a column-wise manner, and kernel's matrix of the  $k$ -th convolutional layer respectively.

In order to achieve our goal, let us adopt the framework proposed in (Bifis et al., 2023); namely the goal is to produce pristine representations by constraining the weights  $K_k$ ,  $k = 1, 2, \dots, N$  of kernels in each convolutional layer of our network to:

- project adversarial residual noise perturbation representations onto the null space of the net's weights
- while preserving all the information from the pristine data representations in the range of the weights.

To this end we focus on finding weights  $K_k$ ,  $k = 1, 2, \dots, N$  for each convolutional layer of the pipeline shown in Fig. 1 such that the corresponding residual noise  $\mathbf{v}_{k,l}$  and  $\mathbf{z}_{k,l}$  be orthogonal to specific parts of the weights matrix. To satisfy these conditions, we can utilize the null space and the range of the matrix  $K_k$ . Next lemma gives us the solution to all the above mentioned requirements.

**Lemma 1:** Let us consider that the following orthogonality constraints:

$$U_{R_k}^T \mathbf{v}_{k,l} = \mathbf{0} \quad (12)$$

$$U_{N_k}^T \mathbf{z}_{k,l} = \mathbf{0} \quad (13)$$

are imposed on the  $k$ -th convolutional layer,  $k = 1, 2, \dots, N$ , of the pipeline shown in Fig. 1 during the training phase of the network, with  $U_{R_k}$ ,  $U_{N_k}$  denoting the range and null space of the  $K_k$  kernel's weights respectively, that can be obtained from the SVD of the corresponding matrix  $K_k = V_k \Sigma_k U_k^T$ . Let us also consider that the following activation function:

$$f_k(K_k \mathbf{q}_{k,l}) = \frac{K_k \mathbf{q}_{k,l}}{\|K_k \mathbf{q}_{k,l}\|_2}, \quad l = 1, \dots, H_{out}W_{out} \quad (14)$$

is acting on the output of the corresponding convolutional layer, with  $\mathbf{q}_{k,l}$  denoting the  $l$ -th column of the random matrix  $Q_k$  defined in Eq. (11). Then, the pristine and adversarial representations match.

**Proof:** The proof of Lemma 1 is easy. Note that if we denote by  $\mathbf{o}_{k,l} = K_k \mathbf{z}_{k-1,l}$  the output of the  $k$ -th convolutional layer to the  $l$ -th column of the pristine matrix  $Z_{k-1}$  then, using Proposition 1 we easily obtain the following relation:

$$K_k \mathbf{z}_{A_{k,l}} = (1 + \mu_{k,l}) \mathbf{o}_{k,l}. \quad (15)$$

By applying the activation function defined in Eq. (14), we can easily prove the lemma.  $\square$

Concluding, by following our pipeline and applying the proposed constraints during training, we ideally would like to acquire a model where, for each pair of pristine and adversarial data their representations to match.

### 3.2 Loss Function

Since we want the whole network be a classifier, we utilize adversarial training and as the loss function we propose the use of a cross entropy based one. Specifically:

$$L(W) = \mathbb{E}_{\mathcal{X}, \mathcal{L}} [CE(g(\mathcal{X}; W), \mathcal{L}) + CE(g(\mathcal{X}_A; W), \mathcal{L})] \quad (16)$$

where  $g(\cdot; W)$  the output of the whole net,  $W = \{K, \{U_{R_k}, U_{N_k}\}_{k=1}^N, W_C\}$  the weights of the non-linear transformation, the  $N$  convolutional layers, the weights of the WideResnet based Classifier and  $\mathcal{L}$  the ground truth labels's set. In addition we would like to impose the following constraints:

$$\mathbb{E}_{\mathcal{V}_k} [\|U_{R_k}^T \mathcal{V}_k\|_F^2] = 0, \quad k = 1, \dots, N \quad (17)$$

$$\mathbb{E}_{Z_k} [\|U_{N_k}^T Z_k\|_F^2] = 0, \quad k = 1, \dots, N. \quad (18)$$

with  $\|X\|_2$  denoting the Euclidean  $l_2$  norm or Frobenious norm of matrix  $X$ . Then, we define the following Lagrangian function:

$$\begin{aligned} J(W_+) &= L(W) + \sum_{k=1}^N \lambda_{R_k} \mathbb{E}_{\mathcal{V}_k} [\|U_{R_k}^T \mathcal{V}_k\|_F^2] \\ &+ \sum_{k=1}^N \lambda_{N_k} \mathbb{E}_{Z_k} [\|U_{N_k}^T Z_k\|_F^2] \end{aligned}$$

or equivalently:

$$\begin{aligned} J(W_+) &= L(W) + \sum_{k=1}^N \lambda_{R_k} \text{tr}\{U_{R_k}^T \mathbb{E}_{\mathcal{V}_k} [\mathcal{V}_k \mathcal{V}_k^T] U_{R_k}\} \\ &+ \sum_{k=1}^N \lambda_{N_k} \text{tr}\{U_{N_k}^T \mathbb{E}_{Z_k} [Z_k Z_k^T] U_{N_k}\} \quad (19) \end{aligned}$$

where  $W_+ = \{W, \{\lambda_{R_k}, \lambda_{N_k}\}_{k=1}^N\}$  and  $\text{tr}\{A\}$  denoting the trace of matrix  $A$ , and minimize it over the weights and the Lagrange multipliers  $\lambda_{R_k}, \lambda_{N_k}, k = 1, \dots, N$  of the network, and this concludes the section.

## 4 EXPERIMENTAL SETUP

All our experiments were conducted, using an NVIDIA A100 with 40GB of vram. As a backbone neural network we utilized the WideResNet architecture from (Zagoruyko, 2016). This architecture has been widely adopted by other researchers for adversarial defense in classification tasks (Bartoldson et al., 2024; Amini et al., 2024; Peng et al., 2023), has been proven effective and is frequently used in the literature, as demonstrated by RobustBench (Croce et al., 2021), a famous benchmark in the context of adversarial robustness for adversarial defenses in the CIFAR-10, CIFAR-100 (Krizhevsky and Hinton, 2009) and ImageNet (Deng et al., 2009) datasets. For our experiments we used a small version of WideResNet with 10 layers and a widen factor of 2 (namely WideResNet-10-2). We ran our tests for two datasets, MNIST (Deng, 2012) & Fashion-MNIST (Xiao et al., 2017). As learning rate, we used  $10^{-5}$  for weights and 1 and 0.01 for lamdas on each dataset respectively. We also tested our theory with different attack hyperparameters. In each network we also added a non-linear transformation layer in the beginning, which consisted of an appropriate size convolutional layer followed by a *ReLU*. We then added two layers ( $N = 2$ ) on which we enforced our constraints during training.

To perform adversarial training, we used PGD (Mađry et al., 2017). For MNIST, we used 40 PGD steps with  $e = 75/255$ , and a step size of  $2/255$ . For Fashion-MNIST, we used 10 PGD steps with  $e = 8/255$ , and a step size of  $2/255$ . We evaluated our trained models under various classical as well as more recent adversarial attacks, namely FGSM (Goodfellow et al., 2014), PGD (Mađry et al., 2017), C&W (Carlini and Wagner, 2017), MIM (Dong et al., 2017), APGD (Croce and Hein, 2020), APGDT (Croce and Hein, 2020), FAB (Croce and Hein, 2019), Square (Andriushchenko et al., 2019), SPSA (Gao et al., 2020), Jitter (Schwinn et al., 2021), VMIFGSM & VNIFGSM (Wang and He, 2021). For the attack implementations, we utilized the widely-used torchattacks library (Kim, 2020), applying the default parameters for each attack, as well as using the same values (where applicable) for  $e$ , step size, and number of steps as in the adversarial training.

## 5 RESULTS

In this section we compare our constraint results with the baseline adversarial training (Mađry et al., 2017). Our approach does not necessitate direct comparison with state-of-the-art defenses, as its primary value

lies in its versatility and lightweight nature. Unlike many specialized techniques, our method is not confined to specific threat models or architectures; instead, it seamlessly integrates with any convolutional layer-based system, enhancing its robustness. This generality distinguishes our approach, as it complements rather than competes with existing defenses. Moreover, adversarial training serves as a universal baseline in this domain due to its ubiquity and established effectiveness across diverse models and attack scenarios. By focusing on comparisons with adversarial training, we highlight the adaptability of our method while avoiding the pitfalls of narrow, scenario-specific evaluations that may not reflect its true potential. This emphasis underscores our contribution as a foundational enhancement to robust learning, capable of synergizing with state-of-the-art techniques to achieve even greater resilience.

## 5.1 MNIST Results

We begin by testing our hypothesis using the MNIST dataset, which consists of 60,000 training images and 10,000 test images of handwritten digits, spanning 10 classes. While defending against adversarial attacks of small magnitude on this dataset is relatively straightforward, particularly through adversarial training, we aim to demonstrate the robustness of our approach under more challenging conditions. To this end, we increase the number of iterations for PGD and use a higher perturbation magnitude  $\epsilon$  (compared to 10 iterations and perturbation magnitude of  $8/255$  typically used in other datasets) to generate adversarial examples that impose a stronger challenge, which is common practice in adversarial defense research. More details can be found in Section 4.

We first compare the performance of our model on clean, unperturbed data. As shown in Table 1, the clean accuracies for both nets, the baseline and the proposed, are comparable. This demonstrates that the constraint we apply to enforce weight orthogonality does not adversely affect the model’s ability to correctly classify clean examples. Our method’s primary contribution lies in improving the robustness of the model against adversarial examples. The enforced orthogonality through our proposed constraint enhances the model’s defense capability against adversarial perturbations, without negatively impacting its performance on clean data.

### 5.1.1 Robustness Against Adversarial Attacks

Next, we evaluate the performance of our trained models against a variety of adversarial attacks in a white-box context. The results, presented in Table

Table 1: Classification accuracies of the two compared classifiers on MNIST.

Classifier	Clean Accuracy
WResNet-10-2	99.30%
Constr. WResNet-10-2 (Ours)	99.33%

Table 2: Robust accuracies under white-box attacks for the two compared classifiers on some typical adversarial attacks on MNIST.

Attack	WResNet	Constr. WResNet (Ours)
FGSM	97.06 %	<b>98.74 %</b>
PGD	95.81 %	<b>96.96 %</b>
C&W	98.17 %	<b>98.33 %</b>
MIM	95.77 %	<b>97.06 %</b>
APGD	92.03 %	<b>98.26 %</b>
APGDT	91.97 %	<b>98.22 %</b>
FAB	94.42 %	<b>98.95 %</b>
Square	97.32 %	<b>98.45 %</b>
SPSA	99.20 %	<b>99.32 %</b>
Jitter	97.25 %	<b>98.02 %</b>
VMIFGSM	95.88 %	<b>96.97 %</b>
VNIFGSM	95.78 %	<b>96.82 %</b>

2, clearly demonstrate that our method outperforms the baseline by nearly 2% average across all attack types, reaching up to 6% at multiple attacks. We must stress at this point that this improvement is achieved solely by imposing our orthogonality constraint during training—without altering the network’s architecture or introducing additional computational overhead during inference.

In other words, our proposed constraint enhances the model’s robustness without causing overfitting to any specific attack, such as PGD, and generalizes well to other adversarial attacks. This highlights the versatility and effectiveness of our approach in strengthening the model’s resistance to adversarial perturbations, marking a significant contribution to the field of adversarial defense.

In summary, our method showcases a simple yet powerful enhancement to adversarial training, improving robustness across a variety of attacks while maintaining similar performance on clean data and avoiding the typical trade-offs associated with more complex defense mechanisms.

## 5.2 Fashion-MNIST Results

We extend our evaluation to the Fashion-MNIST dataset, which consists of 60,000 training images and 10,000 test images representing 10 classes of clothing items. The complexity of Fashion-MNIST lies in the similarity between certain classes (e.g., t-shirts vs. pullovers, trousers vs. dresses), making it a

Table 3: Classification accuracies of the two compared classifiers on Fashion-MNIST.

Classifier	Clean Accuracy
WRResNet-10-2	90.16%
Constr. WRResNet-10-2 (Ours)	90.4%

more challenging benchmark for adversarial defense techniques. Unlike the MNIST experiment, defending against adversarial perturbations through adversarial training in this dataset is relatively more difficult; thus, we aim to demonstrate that our approach is also effective in this scenario.

When testing on clean data, as shown in Table 3, our method exhibits similar performance to the baseline in terms of clean accuracy. This highlights that enforcing orthogonality through our proposed constraints does not degrade the model’s ability to classify clean examples. Despite the increased difficulty of Fashion-MNIST due to more complex class distributions, our method maintains its performance on clean data while significantly enhancing robustness against adversarial attacks. The orthogonality constraint does not overfit the model to adversarial perturbations from the training attacks but rather provides a generalized defense across various attack scenarios, further confirming its effectiveness in improving overall adversarial robustness.

Table 4: Robust accuracies under white-box attacks for the two compared classifiers on some typical adversarial attacks on Fashion-MNIST.

Attack	WRResNet	Constr. WRResNet (Ours)
FGSM	82.51 %	<b>84.25 %</b>
PGD	81.20 %	<b>83.62 %</b>
C&W	85.30 %	<b>86.61 %</b>
MIM	81.36 %	<b>83.68 %</b>
APGD	82.80 %	<b>88.12 %</b>
APGDT	82.50 %	<b>88.10 %</b>
FAB	82.64 %	<b>90.24 %</b>
Square	87.06 %	<b>90.38 %</b>
SPSA	88.97 %	<b>90.23 %</b>
Jitter	82.58 %	<b>86.09 %</b>
VMIFGSM	82.75 %	<b>85.37 %</b>
VNIFGSM	82.88 %	<b>85.25 %</b>

### 5.2.1 Robustness Against Adversarial Attacks

We also evaluate the performance of our method against the same attacks as in section 5.1.1 in a white-box setting. The results, summarized in Table 4, show that our method yields a notable improvement in defense performance, achieving an approximate 3.5% increase in accuracy on average compared to the baseline, across all attack types, reaching up to 7.5%. This improvement is consistent with the results on MNIST,

underscoring that our approach is not tailored to a specific dataset but generalizes well across different data distributions and adversarial settings.

As with the MNIST experiments, the key advantage of our method is that it does not require any architectural modifications or additional computational overhead during inference. The orthogonality constraint, imposed during training, provides robust adversarial defense without introducing significant complexity. Moreover, it helps the model maintain its resistance to various adversarial attacks, demonstrating a consistent performance boost without sacrificing clean data accuracy.

## 6 CONCLUSION

In this paper, we extend our previous work in (Bifis et al., 2023), by introducing a novel defense technique that can be applied to convolutional layers. We demonstrate its effectiveness compared to traditional adversarial training. Our experiments on the MNIST and Fashion-MNIST datasets show consistent improvements of approximately 2% to 7.5% in adversarial robustness across various attacks, compared to classical adversarial training, without sacrificing accuracy on pristine data. These results suggest that incorporating the defense strategy directly into the convolutional layers significantly enhances robustness, providing an efficient and effective improvement over adversarial training in specific settings. It is important to note that, while our technique was tested on an extended network compared to the backbone WideResNet from (Zagoruyko, 2016), the observed improvements are due to the addition of our constraints to the loss function, while maintaining identical network architectures and training conditions. Furthermore, this technique can be seamlessly integrated into existing architectures and combined with state-of-the-art systems to further improve adversarial robustness. In future work, we will explore its application to additional networks and evaluate its performance against a broader range of adversarial attacks. Moreover, optimizing the computational complexity of our method could increase its practicality for deployment in resource-constrained environments.

In conclusion, our proposed defense technique presents a promising avenue for improving the robustness of convolutional neural networks against adversarial attacks. With further refinement, we believe it could become an integral component of future defense strategies in deep learning.

## REFERENCES

- Amini, S., Teymorianfard, M., Ma, S., and Houmansadr, A. (2024). Meansparse: Post-training robustness enhancement through mean-centered feature sparsification. *arXiv preprint arXiv:2406.05927*.
- Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. (2019). Square attack: a query-efficient black-box adversarial attack via random search. *CoRR*, abs/1912.00049.
- Bartoldson, B. R., Diffenderfer, J., Parasyris, K., and Kailkhura, B. (2024). Adversarial robustness limits via scaling-law and human-alignment studies. *arXiv preprint arXiv:2404.09349*.
- Bifis, A., Psarakis, E. Z., and Kosmopoulos, D. (2023). Developing robust and lightweight adversarial defenders by enforcing orthogonality on attack-agnostic denoising autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1272–1281.
- Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57.
- Chen, Y.-Y., Chen, C.-T., Sang, C.-Y., Yang, Y.-C., and Huang, S.-H. (2021). Adversarial attacks against reinforcement learning-based portfolio management strategy. *IEEE Access*, 9:50667–50685.
- Croce, F., Andriushchenko, M., Schwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. (2021). Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Croce, F. and Hein, M. (2019). Minimally distorted adversarial examples with a fast adaptive boundary attack. *CoRR*, abs/1907.02044.
- Croce, F. and Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *CoRR*, abs/2003.01690.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255.
- Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142.
- Dong, Y., Liao, F., Pang, T., Hu, X., and Zhu, J. (2017). Discovering adversarial examples with momentum. *CoRR*, abs/1710.06081.
- Gao, L., Zhang, Q., Song, J., and Shen, H. T. (2020). Patchwise++ perturbation for adversarial targeted attacks. *CoRR*, abs/2012.15503.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- Kim, H. (2020). Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950*.
- Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.
- Kumari, N., Singh, M., Sinha, A., Machiraju, H., Krishnamurthy, B., and Balasubramanian, V. N. (2019). Harnessing the vulnerability of latent layers in adversarially trained models. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 2779–2785.
- Levi, M. and Kontorovich, A. (2024). Splitting the difference on adversarial training. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 3639–3656.
- Mađry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9).
- Pang, T., Lin, M., Yang, X., Zhu, J., and Yan, S. (2022). Robustness and accuracy could be reconcilable by (proper) definition. In *International Conference on Machine Learning*, pages 17258–17277.
- Peng, S., Xu, W., Cornelius, C., Hull, M., Li, K., Duggal, R., Phute, M., Martin, J., and Chau, D. H. (2023). Robust principles: Architectural design principles for adversarially robust cnns. *arXiv preprint arXiv:2308.16258*.
- Rade, R. and Moosavi-Dezfooli, S.-M. (2022). Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*.
- Schwinn, L., Raab, R., Nguyen, A., Zanca, D., and Eskofier, B. M. (2021). Exploring misclassifications of robust neural networks to enhance adversarial attacks. *CoRR*, abs/2105.10304.
- Selvakkumar, A., Pal, S., and Jadidi, Z. (2022). Addressing adversarial machine learning attacks in smart healthcare perspectives. In *Sensing Technology: Proceedings of ICST 2022*, pages 269–282. Springer.
- Sitawarin, C., Chakraborty, S., and Wagner, D. (2021). Sat: Improving adversarial training via curriculum-based loss smoothing. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, pages 25–36.
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2018). Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*.
- Wang, X. and He, K. (2021). Enhancing the transferability of adversarial attacks through variance tuning. *CoRR*, abs/2103.15571.
- Wu, H., Yunas, S., Rowlands, S., Ruan, W., and Wahlström, J. (2023). Adversarial driving: Attacking end-to-end autonomous driving. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7.
- Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747.
- Zagoruyko, S. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.
- Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482.