Multi-way Encoding for Robustness

Donghyun Kim
Boston University
donhk@bu.edu

Sarah Adel Bargal
Boston University
sbargal@bu.edu

Jianming Zhang
Adobe Research
jianmzha@adobe.com

Stan Sclaroff
Boston University
sclaroff@bu.edu

Abstract

Deep models are state-of-the-art for many computer vision tasks including image classification and object detection. However, it has been shown that deep models are vulnerable to adversarial examples. We highlight how one-hot encoding directly contributes to this vulnerability and propose breaking away from this widely-used, but highly-vulnerable mapping. We demonstrate that by leveraging a different output encoding, multi-way encoding, we decorrelate source and target models, making target models more secure. Our approach makes it more difficult for adversaries to find useful gradients for generating adversarial attacks. We present robustness for black-box and white-box attacks on four benchmark datasets: MNIST, CIFAR-10, CIFAR-100, and SVHN. The strength of our approach is also presented in the form of an attack for model watermarking, raising challenges in detecting stolen models.

1. Introduction

Deep learning models are vulnerable to adversarial examples [24]. Evidence shows that adversarial examples are transferable [20, 17]. This weakness can be exploited even if the adversary does not know the target model under attack, posing severe concerns about the security of the models. This is because an adversary can use a substitute model for generating adversarial examples for the target model, also known as *black-box* attacks.

Black-box attacks such as gradient-based attacks [9, 18] rely on perturbing the input by adding an amount dependent upon the gradient of the loss function with respect to the input (input gradient) of a substitute model. An example adversarial attack is $x^{adv} = x + \epsilon sign(\nabla_x Loss(f(x))$, where $f(x)$ is the model used to generate the attack. This added "noise" can fool a model although it may not be visually evident to a human. The assumption of such gradient-based approaches is that the gradients with respect to the input, of the substitute and target models, are correlated.

Our key observation is that the setup of conventional deep classification frameworks aids in the correlation of such gradients, and thereby makes these models more susceptible to black-box-attacks. Typically, a cross-entropy loss, softmax layer, and one-hot vector encoding for the target label are used when training deep models. These conventions constrain the encoding length and number of possible non-zero gradient directions at the encoding layer. This makes it easier for an adversary to pick a harmful gradient direction and perform an attack from a substitute model.

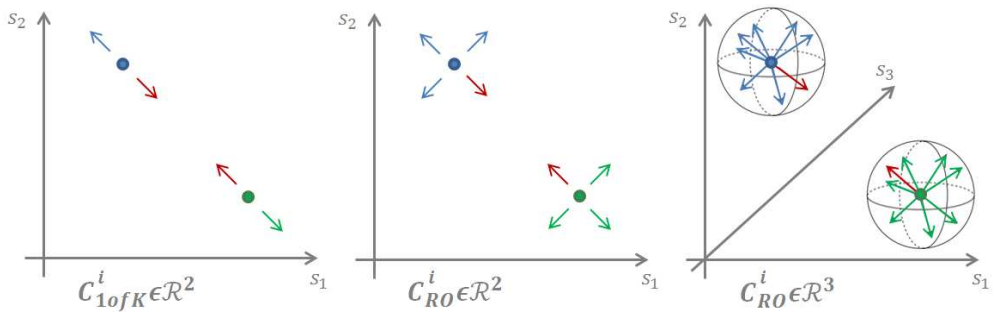
We aim to increase the adversarial robustness of deep models through *model decorrelation*. Our multi-way encoding representation relaxes the one-hot encoding to a real number encoding, and embeds the encoding in a space that has a dimension that is higher than the number of classes. These encoding methods lead to an increased number of possible gradient directions, as illustrated in Fig. 1. This makes it more difficult for an adversary to pick a harmful direction that would cause a misclassification. Multi-way encoding also helps improve a model's robustness in cases where the adversary has full knowledge of the target model under attack: a *white-box* attack. The benefits of multi-way encoding are demonstrated in experiments with four benchmark datasets.

We also demonstrate the strength of *model decorrelation* by introducing an attack for the recent model watermarking algorithm of Zhang *et al*. [29], which deliberately trains a model to misclassify certain watermarked images. We interpret such watermarked images as transferable adversarial examples. We demonstrate that the multi-way encoding reduces the transferability of the watermarked images. Our code is publicly available[1].

We summarize our contributions as follows:

- We propose a novel solution using multi-way encoding to alleviate the vulnerability caused by the $1ofK$ mapping through *model decorrealtion*.
- We empirically show that the proposed approach improves model robustness against both black-box attacks, white-box attacks, and general corruptions.
- We also show the strength of our encoding by attacking a recently proposed model watermarking algorithm.

[1] http://cs-people.bu.edu/donhk/research/Multiway_encoding.html



(a) (b) (c)

Figure 1. Demonstration of the benefit of relaxing and increasing the encoding dimensionality, for a binary classification problem at the final encoding layer. C_i is the codebook encoding for class i, axis s_i represents the output activation of neuron i in the output encoding layer, where $i = 1, \ldots, l$ and l is the encoding dimensionality. The depicted points are correctly classified points of the green and blue classes. The arrows depict the possible non-zero perturbation directions $sign(\frac{\partial Loss}{\partial s_i})$. *(a)* $2D$ $1ofK$ *softmax-crossentropy setup:* Only two non-zero gradient directions exist for a $1ofK$ encoding. Of these two directions, only one is an adversarial direction, depicted in red. *(b)* $2D$ *multi-way encoding:* Four non-zero perturbation directions exist. The fraction of directions that now move a point to the adversarial class (red) drops. *(c)* $3D$ *multi-way encoding:* A higher dimensional encoding results in a significantly lower fraction of gradient perturbations whose direction would move an input from the green ground-truth class to the blue class, or vice versa.

2. Related Work

Attacks. Adversarial examples are crafted images for fooling a classifier with small perturbations. Recently, many different types of attacks have been proposed to craft adversarial examples. We focus on gradient-based attacks which deploy the gradient of the loss with respect to the input [9, 13, 1]. Goodfellow *et al*. [9] propose the Fast Gradient Sign Method (FGSM) which generates adversarial images by adding the sign of the input gradients scaled by ϵ, where the ϵ restricts ℓ_∞ of the perturbation. Kuerakin *et al*. [13] propose the Basic Iterative Method (BIM), which is an iterative version of FGSM and is also called Projected Gradient Descent (PGD). Madry *et al*. [18] show that PGD with randomly chosen starting points within allowed perturbation can make an attack stronger. Gradient-free attacks [2, 15, 26] which do not use gradients from the target model can be used to check whether a defense relies on obfuscated gradients [1].

Defenses. The goal of the defense is to make a correct prediction on adversarial examples. However, adversarial defenses can cause obfuscated gradients (*e.g*. [27, 15]) which are easily broken by Backward Pass Differentiable Approximation attack [1]. Athalye *et al*. [1] recommend performing several sanity tests to check obfuscated gradients for a defense. Madry *et al*. [18] propose a defense based on the minimax formulation of adversarial training which has been extensively evaluated and justified. We also combine our method with the adversarial training and empirically show that our method does not rely on these fragile obfuscated gradients by following evaluations in [1]. However, the previous approach uses the conventional one-hot ($1ofK$) encoding for both source and target models, while we propose a higher dimensional multi-way encoding that obstructs the adversarial gradient search. Our goal is to mitigate the weakness of the transferability of adversarial examples by model decorrelation with our proposed encoding while not relying on obfuscated gradients and compromising white-box robustness at the same time.

Output encoding. There have been attempts to use alternate output encodings for image classification in deep models. Yang *et al*. [28] and Rodriguez *et al*. [21] use an output encoding that is based on Error-Correcting Output Codes (ECOC), for increased performance and faster convergence. In contrast, we use an alternate output encoding scheme, multi-way encoding, to make models more robust to adversarial attacks.

3. Our Approach

In this section we will explain our approach using the following notation: $g(x)$ is the target model to be attacked, and $f(x)$ is the substitute model used to generate a black-box attack for $g(x)$. In the case of a white-box attack, $f(x)$ is $g(x)$. Canonical attacks like FGSM and PGD are gradient-based methods. Such approaches perturb an input x by an amount dependent upon $sign(\nabla_x Loss(f(x)))$. An adversarial example x^{adv} is generated as follows:

$$x^{adv} = x + \epsilon sign(\nabla_x Loss(f(x))), \tag{1}$$

where ϵ is the strength of the attack. Therefore x^{adv} would be a translated version of x, in a vicinity further away from that of the ground-truth class, and thus becomes more likely to be misclassified, resulting in a successful adversarial attack. If the attack is a targeted one, x could be deliberately moved towards some other specific target class.

This is conventionally accomplished by using the adversarial class as the ground truth when back-propagating the loss, and subtracting the perturbation from the original input. The assumption being made in such approaches is that their input gradient direction is similar: $\nabla_x Loss(f(x)) \approx \nabla_x Loss(g(x))$.

We now present the most widely used setup for training state-of-the-art deep classification networks comprising of one-hot encoding and softmax. Let the output activation of neuron i in the final encoding (fully-connected) layer be s_i, where $i = 1, 2, \ldots, k$ and k is the encoding length and the number of classes at the same time. Then, the softmax probability y_i of s_i, and the cross-entropy loss are:

$$y_i = \frac{e^{s_i}}{\sum_{c=1}^{k} e^{s_c}}, \quad \text{and} \quad Loss = -\sum_{i=1}^{k} t_i log(y_i), \quad (2)$$

respectively, where $t_i \in \{0, 1\}$ is the corresponding ground-truth one-hot vector encoding. The partial derivative of the loss with respect to the pre-softmax logit output is:

$$\frac{\partial Loss}{\partial s_i} = y_i - t_i. \quad (3)$$

Combined with the most widely used one-hot ($1ofK$) encoding scheme, the derivative in Eq. 3 makes the gradients of substitute and target models strongly correlated. We demonstrate this as follows: Given a ground-truth example belonging to class [1,0,...,0], non-zero gradients of neuron 1 of the encoding layer will always be negative, while all other neurons will always be positive since $0 < y_i < 1$. So, regardless of the model architecture and the output, the signs of the partial derivatives are determined by the category, and thus the gradients for that category only lie in a limited hyperoctant (see Fig. 1 for the 2D case). This constraint causes strong correlation in gradients in the final layer for different models using the $1ofK$ encoding. Our experiments suggest that this correlation can be carried all the way back to the input perturbations, making these models more vulnerable to attacks.

In this work, we aim to make $\nabla_x Loss(f(x))$ and $\nabla_x Loss(g(x))$ less correlated by encouraging *model decorrelation*. We do this by introducing multi-way encoding instead of the conventional $1ofK$ encoding used by deep models for classification. Multi-way encoding significantly reduces the correlation between the gradients of the substitute and target models, making it more challenging for an adversary to create an attack that is able to fool the classification model.

The multi-way encoding we propose in this work is the Random Orthogonal (RO) output vector encoding generated via Gram-Schmidt orthogonalization. Starting with a random matrix $\mathbf{M} = [a_1|a_2|\ldots|a_n] \in \mathbb{R}^{k \times l}$, the first, second, and k^{th} orthogonal vectors are computed as follows:

$$\begin{aligned} u_1 &= a_1, \quad e_1 = \frac{u_1}{||u_1||}, \\ u_2 &= a_2 - (a_2 \cdot e_1)e_1, \quad e_2 = \frac{u_2}{||u_2||}, \\ u_k &= a_k - \ldots - (a_k \cdot e_{k-1})e_{k-1}, \quad e_k = \frac{u_k}{||u_k||}. \end{aligned} \quad (4)$$

For a classification problem of k classes, we create a codebook $C_{RO} \in \mathbb{R}^{k \times l}$, where $C^i = \beta e_i$ is a length l encoding for class i, and $i \in 1, \ldots, k$, and β is a scaling hyper-parameter dependent upon l. A study on the selection of the length l is presented in the experiments section.

By breaking away from the $1ofK$ encoding, softmax and cross-entropy become ill-suited for the model architecture and training. Instead, we use the loss between the output of the encoding-layer and the RO ground-truth vector, $Loss(f(x), t_{RO})$, where $f(x) \in \mathbb{R}^l$ and $Loss$ measures the distance between $f(x)$ and t_{RO}. In our multi-way encoding setup, the final encoding (s) and $f(x)$ become equivalent. Classification is performed using $\arg\min_i Loss(f(x), t^i_{RO})$. We use Mean Squared Error (MSE) Loss.

Fig. 1 illustrates how using the multi-way and longer encoding results in an increased number of possible gradient directions, reducing the probability of an adversary selecting a harmful direction that would cause misclassification. For simplicity we consider a binary classifier. Axis s_i in each graph represents the output activation of neuron i in the output encoding layer, where $i = 1, \ldots, l$. The depicted points are correctly classified points for the green and blue classes. The arrows depict the sign of non-zero gradients $\frac{\partial Loss}{\partial s_i}$. (a) Using a $1ofK$ encoding and a softmax-cross entropy classifier, there are only two directions for a point to move, a direct consequence of $1ofK$ encoding together with Eq. 3. Of these two directions, only one is an adversarial direction, depicted in red. (b) Using 2-dimensional multi-way encoding, we get four possible non-zero gradient directions. The fraction of directions that now move a correctly classified point to the adversarial class is reduced. (c) Using a higher dimension multi-way encoding results in a less constrained gradient space compared to that of $1ofK$ encoding. In the case of attacks formulated following Eq. 1, this results in 2^l possible gradient directions, rather than l in the case of $1ofK$ encoding. The fraction of gradients whose direction would move the input from the green ground-truth class to the blue class, or vice versa, decreases significantly. In addition, multi-way encoding provides additional robustness by increasing the gradients' dimensionality. The effect of increasing dimensionality is shown in Table 1.

We also combine multi-way encoding with adversarial training for added robustness. We use the following for-

	10	20	40	80	200	500	1000	2000	3000
Black-box	45.4	52.4	62.4	71.3	73.7	78.0	79.6	83.6	82.9
White-box	18.1	23.5	27.4	38.8	40.3	39.5	45.8	54.9	45.3
Clean	96.8	97.0	97.9	98.3	98.5	98.8	98.8	99.1	98.9

Table 1. This table presents the effect of increasing the dimension (10, 20, ..., 3000) of the output encoding layer of the multi-way encoding on the classification accuracy (%) for MNIST on FGSM black-box, white-box attacks ($\epsilon = 0.2$) and clean data. As the dimension increases, accuracy increases up to a certain point; We use 2000 for the length of our multi-way encoding layer.

mulation to solve the canonical min-max problem [18, 11] against adversarial perturbations δ from PGD attacks:

$$\arg\min_{\theta}[\mathbb{E}_{(x,y)\in p_{train}} \max_{\delta}(\mathcal{L}(\theta, x+\delta, y)) + \lambda\mathbb{E}_{(x,y)\in p_{train}}(\mathcal{L}(\theta, x, y))] \quad (5)$$

where p_{train} is the training data distribution, (x, y) are the training points, and λ determines a weight of the loss on clean data together with the adversarial examples at train time. For generating white-box adversarial attacks to our method, we minimize a variant of Carlini-Wagner (CW) loss [6]:

$$\max\left(\min_{i\neq t} Loss\left(x, e_i\right) - Loss\left(x, e_t\right), -\kappa\right) \quad (6)$$

where e_t is the ground-truth vector, κ is a confidence, and $Loss$ is MSE loss.

4. Experiments

We conduct experiments on four commonly-used benchmark datasets: MNIST [14], CIFAR-10 [12], CIFAR-100 [12], and SVHN [19]. **MNIST** is a dataset of handwritten digits. It has a training set of 60K examples and a test set of 10K examples. **CIFAR-10** is a canonical benchmark for image classification and retrieval, with 60K images from 10 classes. The training set consists of 50K images, and the test set consists of 10K images. **CIFAR-100** is similar to CIFAR-10 in format, but has 100 classes containing 600 images each. Each class has 500 training images and 100 testing images. **SVHN** is an image dataset for recognizing street view house numbers obtained from Google Street View images. The training set consists of 73K images, and the test set consists of 26K images.

In this work we define a ***black-box*** attack as one where the adversary knows the architecture and the output encoding used but not learned weights. We use two substitute models using $1ofK$ and RO encodings respectively to evaluate our method. We define a ***white-box*** attack as one where the adversary knows full information about our model, including the learned weights. The threat model is a ℓ_∞ bounded attack within the allowed perturbation ϵ: 0.3 MNIST, 8/255.0 CIFAR-10, 8/255.0 CIFAR-100, 10/255.0 SVHN by following [18, 3].

Layer	Pearson Correlation Coefficient		
	A_{1ofK}, A'_{1ofK}	A_{RO}, A'_{RO}	A_{RO}, A_{1ofK}
Conv1	0.29	0.06	0.0
Conv2	0.24	0.15	0.01
Input	0.35	0.08	0.02

Table 2. Correlation of gradients between models of different encodings. gradients of the loss with respect to the intermediate features of Conv1 and Conv2, and with respect to the input. Then, we compute the correlation coefficient of the sign of the gradients with respect to the intermediate features.

4.1. Multi-way Encoding on MNIST

In this section we provide an in-depth analysis of our multi-way encoding on the MNIST dataset. We conduct experiments to examine how multi-way output encodings can decorrelate gradients (Sec. 4.1.1) and increase adversarial robustness (Sec. 4.1.2). We compare models trained on $1ofK$ output encodings with models having the same architecture but trained on multi-way output encodings. In all experiments we use RO encoding as the multi-way encoding with dimension 2000 determined by Table 1 and $\beta = 1000$. All models achieve $\sim$99% on the clean test set. Models A and C are LeNet-like CNNs and inherit their names from [25]. We use their architecture with dropout before fully-connected layers. We trained models A and C on MNIST with the momentum optimizer and an initial learning rate of 0.01, $momentum = 0.5$ with different weight initializations. It should be noted that, in this section, substitute and target models are trained on clean data and do not undergo any form of adversarial training.

4.1.1 Model Decorrelation

In this section we present how multi-way encoding results in gradient decorrelation. Fig. 2 visualizes the value of the gradients of the loss with respect to input from the models: A_{1ofK}, A'_{1ofK}, A_{RO}, and A'_{RO}, for three sample images from the MNIST dataset. We observe that the gradients of A_{1ofK} and A'_{1ofK} are more similar than those of A_{RO} and A'_{RO}. Also, the gradients of $1ofK$ encoding models are quite dissimilar compared to those of RO encoding models.

$g(x)$ \ $f(x)$	A_{1ofK}	A_{RO}	C_{1ofK}	C_{RO}	AVG BB
A_{1ofK}	34.9 (1.00) *	93.6 (0.02)	56.8 (0.25)	95.5 (0.03)	82.0
A_{RO}	88.7 (0.02)	54.9 (1.00) *	92.5 (0.02)	82.9 (0.09)	88.0
C_{1ofK}	30.1 (0.25)	83.6 (0.01)	22.5 (1.00) *	93.3 (0.01)	69.0
C_{RO}	94.3 (0.03)	87.5 (0.09)	96.1 (0.01)	70.5 (1.00) *	92.6

Table 3. This table presents the classification accuracy (%) of MNIST on FGSM black-box, white-box attacks, and average black-box (AVG BB) using architectures A and C. $f(x)$ is a substitute model and $g(x)$ is a target model. We conclude: (a) $g(x)$ using $1ofK$ is more vulnerable to black-box attacks than $g(x)$ using RO. (b) For white-box attacks, RO encoding leads to better accuracy compared to $1ofK$. (c) In brackets is the correlation coefficient of the input gradients of $g(x)$ and $f(x)$. RO results in a lower correlation compared to $1ofK$.

While Fig. 2 depicts three sample examples, we now present aggregate results on the entire MNIST dataset. We measure the correlation of gradients between all convolutional layers and the input layer of models trained on different encodings. We first compute the gradients of the loss with respect to intermediate features of Conv1 and Conv2. Then, we compute the Pearson correlation coefficient (ρ) of the sign of the gradients with respect to the intermediate features between models based on the following equation: For further comparison, we train models A'_{1ofK} and A'_{RO}, which are independently initialized from A_{1ofK} and A_{RO}. We average gradients of convolutional layers over channels in the same way a gradient-based saliency map is generated [23]. Otherwise, the order of convolutional filters affects the correlations and makes it difficult to measure proper correlations between models. In this sense, the correlations at FC layers do not give meaningful information since neurons in the FC layer do not have a strict ordering.

In Table 2, we find that the correlations of Conv1 and Conv2 between $1ofK$ models are much higher than those of RO models. Table 2 also shows that the correlations between RO and $1ofK$ are also low. In addition, RO models are not highly correlated even though they are using the same encoding scheme. At the input layer, the correlations between RO and $1ofK$ are almost zero, but $1ofK$ models have a significantly higher correlation.

We present ablation studies on our method in Section A in the supplementary material.

4.1.2 Robustness

Table 3 presents the classification accuracy (%) of target models under attack from various substitute models. Columns represent the substitute models used to generate FGMSM attacks of strength $\epsilon = 0.2$ and rows represent the target models to be tested on the attacks. The diagonal represents white-box attacks and others represent black-box attacks. Every cell corresponds to an attack from a substitute model $f(x)$ for a target model $g(x)$. The last column reports the average accuracy on black-box attacks.

By comparing the last column of Table 3, the $g(x)$ using

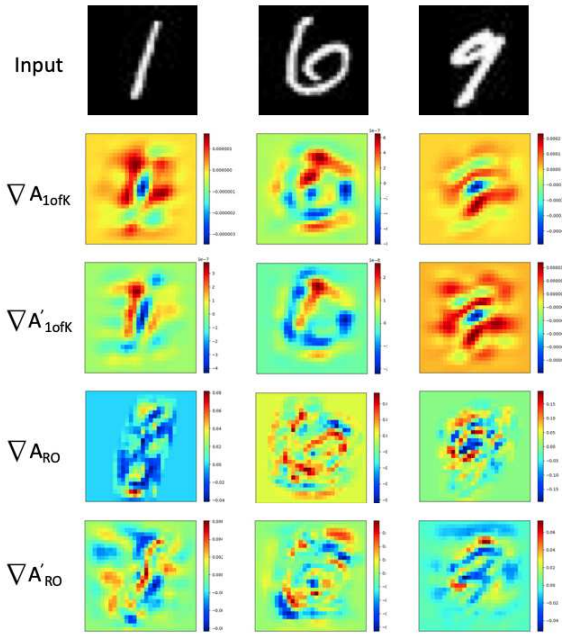


Figure 2. Three sample examples of gradients of the loss with respect to an input from A_{1ofK}, A'_{1ofK}, A_{RO}, and A'_{RO} models. All networks are independently trained with different weight initializations. The gradients of A_{1ofK} and A'_{1ofK} become similar after training while the gradients of A_{RO} and A'_{RO} are dissimlar.

the $1ofK$ encoding is more vulnerable to black-box attacks than the corresponding $g(x)$ using the RO encoding. Black-box attacks become stronger if $f(x)$ uses the same encoding as $g(x)$. In addition, even though the same encoding is used, RO models maintain higher robustness to the black-box attacks compared to $1ofK$ models (*e.g.* 82.9% when C_{RO} attacksA_{RO} vs. 56.8% when C_{1ofK} attacks A_{1ofK}). This suggests that RO encoding is more resilient to black-box attacks.

It is also evident from the results of this experiment in Table 3 that even when the source and target models are the same, denoted by (*), RO encoding leads to better accuracy, and therefore robustness to white-box attacks, compared to $1ofK$ encoding.

Dataset	Model	Accuracy (%)					
		Blackbox #steps:1K ($1ofK$)	Blackbox #steps:1K (RO)	Whitebox #steps:1K	Whitebox #steps:5K 50-restarts	Gradient -free (ℓ_2, ℓ_∞ [2, 15])	Clean
MNIST	Madry *et al*. [18]	94.9	95.0	92.2	89.6	35.4	98.4
	Ours	**96.9**	**96.8**	**94.9**	**94.1**	**42.7**	**99.2**
CIFAR-10	Madry *et al*. [18]	62.5	**73.8**	45.3	44.9	46.6	87.3
	Ours	**65.2**	72.2	**53.1**	**52.4**	**56.9**	**89.4**

Table 4. Comparison against the released models of Madry *et al*. [18] on white-box, black-box PGD attacks, gradient-free attacks and on clean data. We report results on white and black-box PGD attacks generated using the 1K iterations. We observe that our approach is more resilient to these types of attacks and obtains improvements on clean data.

Model	Corruptions										
	AVG	Bright.	Spatter	Jpeg	Elestic	Motion	Zoom	Impulse	Speckle	Gauss. noise	Snow
Madry *et al*.	81.5	87.1	81.6	85.4	81.7	80.4	82.7	68.8	81.8	82.2	82.6
Ours	**83.4**	**89.3**	**84.4**	**87.1**	**83.2**	**81.6**	**84.0**	**72.0**	**84.3**	**84.3**	**84.7**

Table 5. Evaluation on common corruptions and perturbations on CIFAR-10 [10]. Our method obtains higher accuracy (test accuracy %) for all corruptions and perturbations. The first column (AVG) presents the averaged accuracy for all cases.

Finally, Table 3 reports the correlation coefficient of $sign(\nabla_x Loss(f(x)))$ and $sign(\nabla_x Loss(g(x)))$ in Eq. 1. These gradients are significantly less correlated when the source and target models use different encodings. In addition, RO results in a lower correlation compared to $1ofK$ when the same encoding is used in the source and target models.

4.2. Benchmark Results

In this section we analyze the case where we combine our method with adversarial training (Eq. 6). We compare against the strong baseline of Madry *et al*. [18], which also uses adversarial training. For adversarial training, we use a mix of clean and adversarial examples for MNIST, CIFAR-10, and CIFAR-100, and adversarial examples only for SVHN following the experimental setup and the threat models used by Madry *et al*. [18] and Buckman *et al*. [3]. We use PGD attacks with a random start, and follow the PGD parameter configuration of [18, 11, 3].

We directly compare our method with the publicly released versions of Madry *et al*. [18] on MNIST and CIFAR-10 in Table 4. We present results for 1K-step PGD black-box attacks generated from the independently trained copy of Madry *et al*. (the first column) and the model trained with the RO (the second column). It should be noted that the substitute model uses the same RO encoding parameters used for the target model. We generate 1K-step PGD white-box attacks with a random start. In addition, we generate 5K-step PGD attacks with 50 random restarts.

Table 4 demonstrates the robustness of multi-way encoding for black-box attacks, while at the same time maintaining high accuracy for white-box attacks and clean data.

The black-box attacks in the second column of Table 4 show the robustness even when an adversary knows the exact value of the encoding used for the target model. We generate high-confidence PGD attacks with Eq. 6 from the independently trained copy of the RO model. Our model achieves higher worst-case robustness compared to the baseline. In CIFAR-10, the black-box attacks from the RO model (*i.e*. the substitute model uses the same RO encoding parameters) are much weaker than the black-box attacks from the $1ofK$ model. This shows that RO can effectively decorrelate the target model even when the encoding is exposed to an adversary. This is also consistent with the results where the correlation of gradients are lower in Table 2 and the black-box robustness of the RO models is higher in Table 3 even when the substitute model uses the same RO encoding as the target model.

In the third and fourth columns, our defense also achieves higher robustness than the baseline on PGD white-box attacks. We observe that increasing random restarts decreases robustness on both ours and Madry *et al*., which implies a gradient masking effect. However, this is due to the non-convexity of the loss landscape, so that this type of gradient masking can happen to all deep models. This type of gradient masking is different from obfuscated gradients [1] which are easily broken by gradient-free attacks [15], which does not use the gradient from the target model. We show that this type of gradient masking is not easily broken by gradient-free attacks in the fifth column of Table 4.

We follow [4, 15] and perform gradient-free attacks to check for signs of obfuscated gradients. In the fifth column, we evaluate our method on gradient-free attacks; (1) decision-based attacks [2, 22] for MNIST and (2) query-

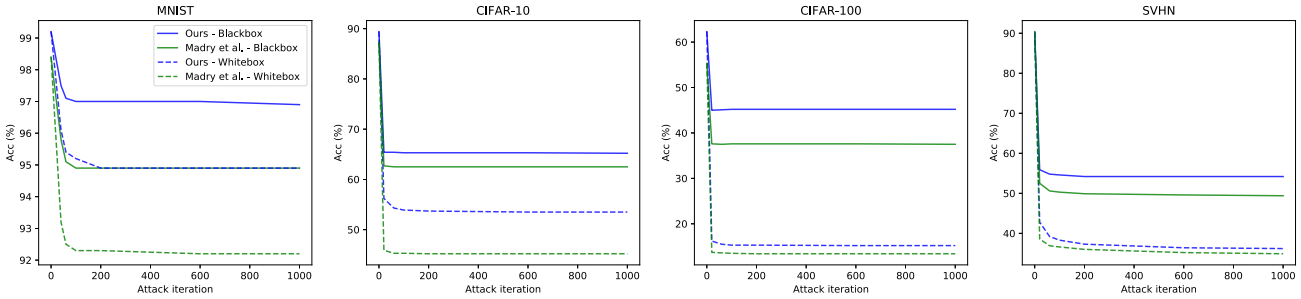


Figure 3. We generate PGD attacks with the different number of iterations and compare our method with the baseline by following [1]. Black-box attacks are generated from an independently trained copy of the baseline. We observe that (1) the attack is converged; (2) iterative attacks are stronger than single-step attacks; (3) white-box attacks are stronger than black-box attacks.

based attacks [15] using Eq. 6 for CIFAR-10. It should be noted the decision-based attacks are ℓ_2 bounded which violates our threat model. From the results, gradient-free attacks could not break our defense and we do not observe a sign of obfuscated gradients.

4.3. Further Analysis

Robustness on General Corruptions. Recent works [8, 10] highlight the close relationship between adversarial robustness and general corruption (*e.g.* Gaussian noise) robustness. It is observed that the certified defense [18] increased robustness on not only adversarial examples but also corrupted examples, while failed defenses (*e.g.* [27, 16]) could not increase robustness on corrupted examples. Ford *et al.* [8] argue that defense should have higher robustness on general corruptions and recommend reporting corruption robustness as a sanity check for a defense. We evaluate our method's general robustness, using the dataset of Hendrycks *et al.* [10] designed to test the common corruptions and perturbations on CIFAR-10. As reported in Table 5, we achieve higher robustness in all cases.

Additional Datasets. We train the method of Madry *et al.* and our approach on CIFAR-100 and SVHN. Fig. 3 represents the robustness on black-box and white-box attacks on different PGD attack iterations. We also report the clean accuracy when the iteration $= 0$. Our method achieves higher robustness on the black-box and white-box attacks on CIFAR-100 and SVHN. In addition, we also improve the clean accuracy by 7% on CIFAR-100. We observe that that attack success rates are converged with large iterations.

Checking Obfuscated Gradients. To check if our method relies on obfuscated gradients [1], we provide evaluations by following the guidelines of Athalye *et al.* [1]. We include plots of the different PGD attack iterations in Fig. 3. Transfer-based attacks can effectively check if a defense method relies on obfuscated gradients [1]. Our model is robust to the transfer-based attacks than Madry *et al.* We also observe that (1) the attack is converged; (2) iterative attacks are stronger than single-step attacks; (3) white-box attacks are stronger than black-box attacks. From these evaluations, we do not find a sign of obfuscated gradients and our method performs better than the baseline. Additional sanity check can be found in Section C of the supplementary.

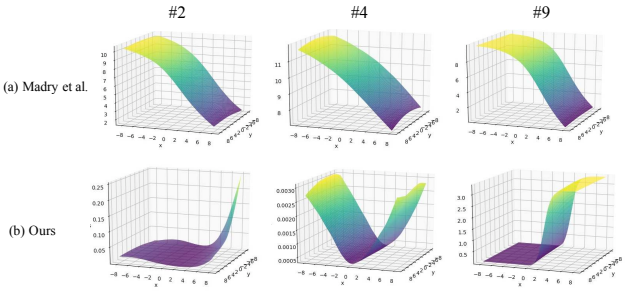


Figure 4. Comparisons of landscapes of the baseline model of Madry *et al.* [18] and our model. The x-axis represents the magnitude of the gradient direction of the loss w.r.t. input, the y-axis represents the magnitude of a random direction, and the z-axis represents (a) the value of the ground-truth neuron in the final layer for the baseline, and (b) the mean squared error loss between the final encoding and the ground-truth vector at each input data point $(x^{adv} = x' + x * r_1 + y * r_2)$. In this figure, we depict the landscapes of x' of the three examples.

Analysis on White-box Attacks. The goal of this section is to analyze how our method improves white-box robustness. Following [7], we plot the landscapes of the loss of our method and the value of neurons that correspond to the ground-truth class of the baseline model [18] for CIFAR-10. Since cross-entropy loss with softmax layer can be biased to the norm of neuron values at the final layer (Section 3 in [5]), we instead plot the value of the ground-truth neuron for the baseline model. In Fig. 4, The x-axis represents the magnitude of the direction of $r_1 = sign(\nabla_{x'} Loss(f(x'))$ and the y-axis represents the magnitude of a random direction, $r_2 \sim$ Rademacher (0.5). For our model, z-axis represents the mean squared error loss between the final encoding and the ground-truth vector at each input $(x^{adv} = x' + x * r_1 + y * r_2)$. For Madry *et al.*, the z-axis represents the value of the ground-truth neuron in the final layer which is directly responsible for the loss.

For Madry *et al*., the landscapes show linearity along with the direction of r_1 for over the test set regardless of misclassification. However, our model shows non-linear behaviors over correctly classified examples but linear-like behaviors for misclassified examples. We show representative landscapes from the three test points. Fig. 4 (a) shows the linearity along with the direction of r_1 for the baseline. The value of #4 decreases slowly but still shows linear behavior and the values of #2 and #9 decrease significantly. [9] argue that the linearity is a primary cause of vulnerability. We also claim that the linear behavior makes it easier to find harmful gradients for the *first-order adversary* like PGD attacks. Fig. 4 (b), we observe that our model shows non-linear behavior with the direction of r_1 for the correctly classified examples: #2, #4. For #9, the more linear-like behavior results in a much higher loss and misclassification.

When ground-truth vectors are one-hot encodings, decreasing the output value of the neuron corresponding to the ground-truth class significantly would cause a misclassification. However, when ground-truth vectors are multi-way encodings, no single neuron is solely responsible for misclassification, but a more complex combination of neurons. Since the loss in our model is computed on multiple neurons at the final layer, an adversarial direction may increase the loss of certain neurons but it may also decrease the loss of other neurons at the same time. We argue that non-linearity is related to our high dimensional encoding layer which provides additional robustness to *first-order white-box attacks* in addition to black-box attacks.

5. Attacking Model Watermarking via Model Decorrelation

Zhang *et al*. [29] introduced an algorithm to detect whether a model is stolen or not. They do so by adding a watermark to sample images of specific classes and deliberately training the model to misclassify these examples to other specific classes. Even if their pre-trained model is stolen, the model should make a misclassification on the watermarked image. This approach has demonstrated to be robust even when the model is fine-tuned on a different training set.

We interpret the watermarked image used to deliberately cause a misclassification as a ***transferable adversarial example***. We introduce an attack for this algorithm using our multi-way encoding, making it more challenging to detect whether a model is stolen or not. We do this by fine-tuning the stolen model using multi-way encoding, rather than the encoding used in pre-training the model. We show that our multi-way encoding successfully decorrelates a model from the pre-trained model and, as a result, adversarial examples become less transferable.

We follow the same CIFAR-10 experimental setup for detecting a stolen model as in Zhang *et al*.: We split the test set into two halves. The first half is used to fine-tune pre-trained networks, and the second half is used to evaluate new models. When we fine-tune the $1ofK$ model, we re-initialize the last layer. When we fine-tune the RO model we replace the output encoding layer with our 2000-dimension fully-connected layer, drop the softmax, and freeze convolutional weights.

	Finetune?	**Test Acc (%)**	**Watermark. Acc (%)**
StolenNet$_{1ofK}$	✗	84.7	98.6
Net$_{1ofK}$	✗	48.3	6.1
Net$_{RO}$	✗	48.0	10.0
Net$_{1ofK}$	✓	85.6	87.8
Net$_{RO}$	✓	80.2	12.9

Table 6. Our attack is capable of fooling the watermarking detection algorithm of [29] via model decorrelation. Fine-tuning a stolen model using RO encoding remarkably reduces the watermarking detection accuracy, and makes it comparable to the accuracy of models trained from scratch and do not use the stolen model. The accuracy of fine-tuned models benefits significantly from the pre-trained weights of the stolen model.

We present results on the CIFAR-10 dataset in Table 6. When the fine-tuning was performed using the $1ofK$ encoding (also used in pre-training the model), watermarking detection is 87.8%, and when the fine-tuning was performed using the multi-way RO encoding the watermarking detection is only 12.9% while taking advantage of the pre-trained weights of the stolen model. The watermark detection rate of the model fine-tuned using RO is significantly lower than that of model fine-tuned using $1ofK$ encoding, and is more comparable to models that are trained from scratch and do not use the stolen model (6.1% and 10.0%). These results suggest that our multi-way encoding successfully decorrelates the target model (finetuned **Net**$_{RO}$) from the source model (**Stolen Net**$_{1ofK}$).

6. Conclusion

By relaxing the $1ofK$ encoding to a real number encoding, together with increasing the encoding dimensionality, our multi-way encoding decorrelates source and target models, confounding an attacker by making it more difficult to perturb an input in transferrable gradient direction(s) that would result in misclassification of a correctly classified example. We present stronger robustness on four benchmark datasets for both black-box and white-box attacks and we also improve classification accuracy on clean data. We demonstrate the strength of model decorrelation with our approach by introducing an attack for model watermarking, decorrelating a target model from the source model.

References

[1] A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[2] W. Brendel, J. Rauber, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. 2018.

[3] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *ICLR*, 2018.

[4] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, and A. Madry. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[5] N. Carlini and D. Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[6] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[7] L. Engstrom, A. Ilyas, and A. Athalye. Evaluating and understanding the robustness of adversarial logit pairing. *arXiv preprint arXiv:1807.10272*, 2018.

[8] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[10] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[11] H. Kannan, A. Kurakin, and I. Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

[12] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[13] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *ICML*, 2019.

[16] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018.

[17] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.

[18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, 2011.

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

[21] P. Rodríguez, M. A. Bautista, J. Gonzàlez, and S. Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31, 2018.

[22] L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on mnist. In *ICLR*, 2018.

[23] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

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

[26] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*, 2018.

[27] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018.

[28] S. Yang, P. Luo, C. C. Loy, K. W. Shum, X. Tang, et al. Deep representation learning with target coding. In *AAAI*, 2015.

[29] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy. Protecting intellectual property of deep neural networks with watermarking. In *Asia Conference on Computer and Communications Security*. ACM, 2018.