MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Improving Adversarial Robustness by Learning Shared Information

Yu, Xi; Smedemark-Margulies, Niklas; Aeron, Shuchin; Koike-Akino, Toshiaki; Moulin, Pierre; Brand, Matthew; Parsons, Kieran; Wang, Ye

TR2022-141 November 02, 2022

Abstract

We consider the problem of improving the adversarial robustness of neural networks while retaining natural accuracy. Motivated by the multi-view information bottleneck formalism, we seek to learn a representation that captures the shared information between clean samples and their corresponding adversarial samples while discarding these samples' view-specific information. We show that this approach leads to a novel multi-objective loss function, and we provide mathematical motivation for its components towards improving the robust vs. natural accuracy tradeoff. We demonstrate enhanced tradeoff compared to current state-of-the-art methods with extensive evaluation on various benchmark image datasets and architectures. Ablation studies indicate that learning shared representations is key to improving performance.

Pattern Recognition 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Improving Adversarial Robustness by Learning Shared Information

Xi Yu[a], Niklas Smedemark-Margulies[b], Shuchin Aeron[c], Toshiaki Koike-Akino[d], Pierre Moulin[e], Matthew Brand[d], Kieran Parsons[d] and Ye Wang[d,*]

[a]Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
[b]Khoury College of Computer Science, Northeastern University, Boston, MA 02115, USA
[c]Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155, USA
[d]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA.
[e]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

ARTICLE INFO

Keywords:
Adversarial Robustness
Information Bottleneck
Multi-view Learning
Shared Information

ABSTRACT

We consider the problem of improving the adversarial robustness of neural networks while retaining natural accuracy. Motivated by the multi-view information bottleneck formalism, we seek to learn a representation that captures the shared information between clean samples and their corresponding adversarial samples while discarding these samples' view-specific information. We show that this approach leads to a novel multi-objective loss function, and we provide mathematical motivation for its components towards improving the robust vs. natural accuracy tradeoff. We demonstrate enhanced tradeoff compared to current state-of-the-art methods with extensive evaluation on various benchmark image datasets and architectures. Ablation studies indicate that learning shared representations is key to improving performance.

1. Introduction

The vulnerability of deep neural networks (DNNs) to adversarial manipulation has been widely investigated and has received significant attention in recent years. Carefully-crafted small-magnitude perturbations to natural images, known as adversarial examples [1], can easily cause machine learning models to make erroneous predictions. Adversarial training (AT) [2] is one of the most effective and widely used approaches to enhance the robustness of DNNs against adversarial examples. Based on primary AT frameworks such as PGD-AT [3], various efforts have been devoted to improving adversarial robustness from different perspectives, including adversarial regularization methods, such as adversarial logit pairing (ALP) [4], Max-Margin Adversarial (MMA) training [5], TRADES [6], and MART [7], accelerating the training procedure, such as you only propagate once (YOPO) [8], "free" adversarial training [9], and Fast adversarial training [10], and adaptive perturbation tolerance, such as Instance Adaptive Adversarial Training (IAAT) [11], and Customized Adversarial Training (CAT) [12].

The performance of machine learning methods is heavily dependent on the choice of data representation, and the goal of representation learning is to transform a raw input $\mathbf{x}$ to a lower-dimensional representation $\mathbf{z}$ that preserves the relevant information for tasks such as classification or regression. Significant progress has been made in deep learning via supervised [13], semi-supervised [14], and unsupervised representation learning [15]. The information bottleneck

*Corresponding author
yuxi@ufl.edu (X. Yu); yewang@merl.com (Y. Wang)
ORCID(s): 0000-0002-2029-1680 (X. Yu)

(IB) principle [16] provides an information-theoretic method for representation learning, where a representation should contain only the most relevant information from the input for downstream tasks. Representations learned by the IB principle are less affected by nuisance variations and may be more robust to adversarial perturbations. In particular, previous works, such as variational IB [17], and parameterized rate distortion stochastic encoder [18], have already show that IB principle can enhance adversarial robustness, even without access to adversarial examples. In addition, a large body of literature has applied the IB principle to deep neural networks, such as variational IB [17], nonlinear IB [19], deep deterministic IB [20] and gated IB [21].

The multi-view information bottleneck [22] extends the IB principle to a multi-view unsupervised setting by maximizing the shared information between different views, while minimizing the view-specific information. In contrast, the InfoMax principle [23] aims to learn an unsupervised latent representation that preserves as much input information as possible, with no incentive to discard any input information from the latent representation. Some recent works, such as augmented multiscale DIM (AMDIM) [24] and invariant information clustering (IIC) [25] have applied the InfoMax principle to the multi-view setting by maximizing the mutual information between the representations of different views of the input.

Motivated by these previous works, we extend the multi-view information bottleneck method to a supervised setting with adversarial training. We can consider adversarial examples as another view of corresponding clean samples. We seek to learn representations that contain the shared information between clean samples and corresponding adversarial samples, while eliminating information not shared between them. We capture this in a multi-objective loss function, with terms corresponding to the goals of (1) maximizing the mutual information between the representations of these matched pairs and (2) minimizing the mutual information between each representation and its corresponding view conditioned on the other view, along with (3) clean and (4) adversarial cross-entropy losses. Fig. 1 illustrates the pipeline of our proposed method.

Our contributions are three-fold:

- Inspired by multi-view representation learning, we propose a regularization scheme that casts the adversarial example as the secondary view.
- We propose a novel multi-objective loss function to learn representations that capture the shared information between clean and adversarial samples, and provide theoretical analysis for the constituent regularization terms.
- We demonstrate that our proposed method improves the robust vs. natural accuracy tradeoff over previous adversarial regularization approaches, under a variety of adversarial attacks on MNIST, CIFAR-10, and CIFAR-100 datasets.

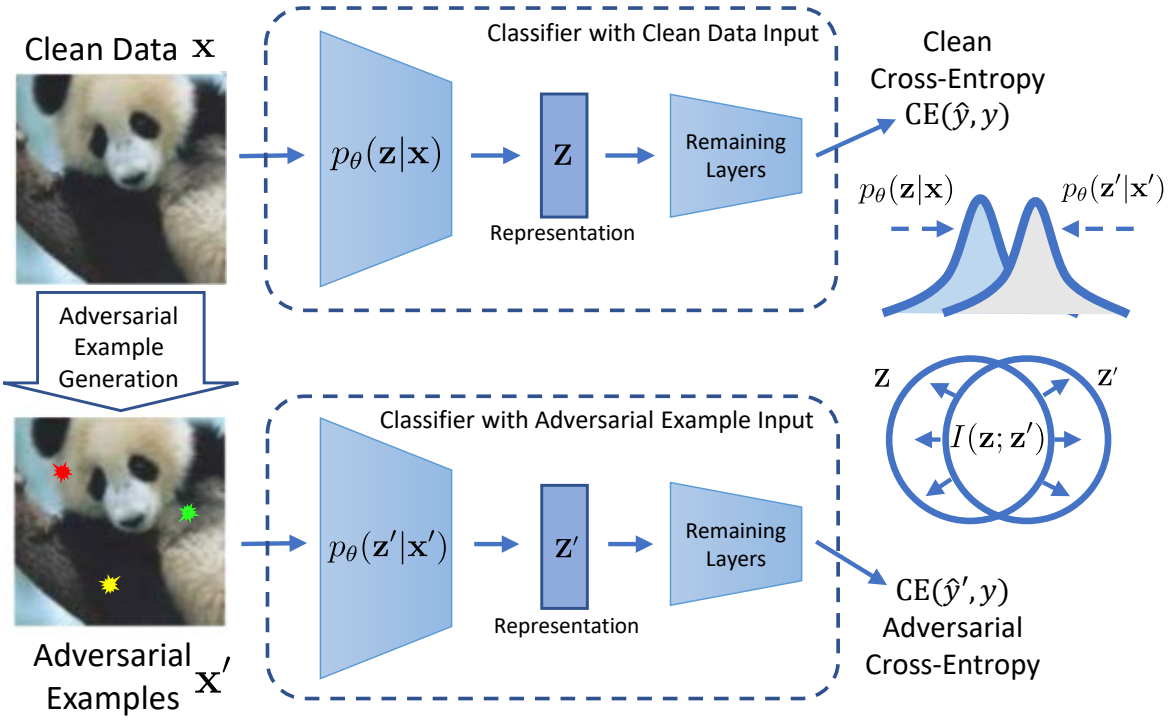


Figure 1: Proposed method pipeline. $\mathbf{x}$ is a clean sample and $\mathbf{x}'$ is the corresponding adversarial sample. $\mathbf{z}$ and $\mathbf{z}'$ are the latent representations of $\mathbf{x}$ and $\mathbf{x}'$, respectively. To force $\mathbf{z}$ and $\mathbf{z}'$ to contain the shared information between $\mathbf{x}$ and $\mathbf{x}'$, we simultaneously minimize the symmetrized KL-divergence between $p_\theta(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{z}'|\mathbf{x}')$, and maximize the mutual information between $\mathbf{z}$ and $\mathbf{z}'$.

The remainder of this paper is organized as follows: we provide a brief review of the literature on adversarial attacks, defense approaches, and some relation to existing works in section 2. In section 3, we describe our proposed method and some theoretical analysis about each component in our objective function. Empirical results on several benchmarks with different types of attacks as well as ablation studies are provided in section 4.

2. Related Works

2.1. Adversarial Attacks

A history of adversarial example attacks on classical and modern machine learning models is provided by the survey of [26]. An early work is the evasion attacks of [27], which pioneer gradient-based and iterative methods, with also a penalty term for generating samples in low density regions. Generally, various approaches seek a perturbation that maximizes an objective function while constraining the norm of the perturbation (e.g., ℓ_p norm) to be less than some fixed budget. Gradient-based approaches have been widely used to solve this maximization problem. The Fast

Gradient Sign Method (FGSM) finds an adversarial example in a single step by maximizing the loss function. PGD is an iterative algorithm that uses projected gradient descent to craft adversarial examples. The Carlini and Wagner attack (C&W) [28] aims to find a small perturbation by maximizing the difference between model output on clean and adversarial samples. The recently proposed AutoAttack approach [29] uses an ensemble of four different attacks: PGD with cross-entropy loss (APGD-CE), PGD with the difference of logits ratio loss (APGD-DLR), the targeted version of Fast Adaptive Boundary Attack (FAB) [30], and the black-box squares attack [31].

2.2. Adversarial Defenses

Prior work on defense with adversarial training can be roughly divided into three main categories. The first category is the adversarial regularization approach, which adds a regularization term in the objective function alongside the original adversarial training loss. The second category uses so-called curriculum-based adversarial training, in which the difficulty of adversarial training is gradually increased (e.g., by increasing the iteration count of PGD attacks). The third category uses an adaptive perturbation budget ϵ. Unlike the previous two types, the key idea here is to select ϵ at the instance level rather than treat all data equally with a fixed ϵ. Our contribution belongs to the first category and improves adversarial robustness by regularizing shared information.

Adversarial regularization. Adversarial regularization first appears in [32], in addition to cross-entropy loss on clean samples, the authors also use cross-entropy loss on adversarial samples. Similarly, Max-Margin Adversarial (MMA) [5] training adopts cross-entropy loss on adversarial input for correctly classified examples and applies cross-entropy loss on clean input for misclassified examples. Local linearity regularization (LLR) [33] examines local linearity, the absolute difference between the adversarial loss and its first-order Taylor approximation in a small neighborhood, and concludes that robust models should have small values of this measure. Similar to LLR, curvature regularization (CURE) [34] reduces the curvature of the clean sample loss and achieves performance comparable to adversarial training. Logit pairing [4] introduces a regularization term enclosing both clean logit and adversarial logit. TRADES [6] combines cross-entropy loss on clean samples with the KL-divergence between predicted probabilities for clean and adversarial samples. Unlike TRADES, MART [7] combines boosted cross-entropy loss (BCE) on adversarial samples with the weighted KL-divergence between predicted class probabilities of clean and adversarial samples to emphasize misclassified examples. Most recently, Deep Robust Representation Disentanglement Network (DRRDN) [35] was proposed to improve the adversarial robustness by learning the class-specific and class-irrelevant representations through a disentangler. However, DRRDN contains two separate encoder networks and four additional regularization terms in its objective function, which is more complex than our proposed method. In contrast with previous methods, our proposed method takes an information-theoretic approach to improve the adversarial robustness by learning the shared information between clean and adversarial samples. Our objective function includes the

symmetrized KL-divergence between the clean and adversarial sample's posterior feature distribution and the mutual information between the latent representation of clean and adversarial examples.

Curriculum-based adversarial training. For standard adversarial training, the inner cross-entropy loss maximization performed by the adversary seeks to find worst-case samples. However, worst-case samples from a strong attack algorithm are not always suitable for adversarial training and may cause overfitting to the particular attack used during training. Curriculum Adversarial Training (CAT) [36] alleviates this issue by gradually increasing the iteration count of PGD attacks over time. Following from this work, Friendly Adversarial Training (FAT) [37] employs early-stopping for adversarial training and selects adversarial samples near the decision boundary for training. Such curriculum-based adversarial training methods improve generalization for adversarial robustness while also preserving clean data accuracy.

Adaptive perturbation budget approaches. Traditional adversarial training treats all samples equally using a fixed perturbation budget ϵ. However, individual samples might have different intrinsic robustness regarding their distances to a decision boundary. The first proposed work that accounts for this difference in the characteristics of individual examples is Instance Adaptive Adversarial Training (IAAT) [11], where the strategy of selecting ϵ is based on class structure. The authors increase ϵ in regions where class manifolds are far apart and decrease ϵ in regions where class manifolds are close together. In addition to considering sample-wise adaptive ϵ during training, Customized Adversarial Training (CAT) [12] also considers an adaptive label technique by smoothing labels for each sample, which improves both clean and robust accuracy over previous adversarial training methods.

Other adversarial defenses. In addition to the above categories of adversarial defense approaches, several other defense methods have been proposed, such as Adversarial distributional training (ADT) [38], where the inner maximization aims to find an adversarial distribution by maximizing the expected loss. In contrast, the outer minimization performed by the model seeks to learn a robust classifier by minimizing the expected loss over worst-case adversarial distributions. Last, some works propose augmented adversarial examples such as Gaussian data augmentation [39], interpolation between clean and adversarial samples using techniques like Mixup [40], and adversarial interpolation training [41].

3. Methodology

This section first motivates the benefit of preserving shared information for designing robust models. Then, we propose an objective function to obtain a representation containing the shared information between clean and adversarial samples while also utilizing label information. Finally, we show the relationship to previous adversarial regularization approaches.

3.1. Preliminaries

Consider a dataset $\{\mathbf{x}_i, y_i\}_{i=1,\dots,n}$ with K classes, where $\mathbf{x}_i \in \mathbb{R}^d$ is a clean sample and $y_i \in \{1, \dots, K\}$ is its associated label. Let f be a classifier with parameters θ, and let the output of classifier $f_\theta(\mathbf{x}_i)$ be the estimated probabilities of $\mathbf{x}_i$ belonging to each class. The learning problem in traditional adversarial training objectives is defined as follows:

$$\min_{\theta} \mathbb{E}\left[\max_{\mathbf{x}' \in \mathcal{B}(\mathbf{x},\epsilon)} \mathcal{L}(f_\theta(\mathbf{x}'), y)\right], \tag{1}$$

Here $\mathcal{L}$ is the cross-entropy loss and the adversarial example $\mathbf{x}'$, belonging to $\mathcal{B}(\mathbf{x}, \epsilon) = \{\mathbf{x}' : ||\mathbf{x}' - \mathbf{x}||_p \leq \epsilon\}$, is obtained by maximizing the cross-entropy loss with respect to a small perturbation.

Due to our proposed method is related to information theory, we first introduce the basic definitions required for this area, such as entropy and mutual information. The entropy is a measure of uncertainty of a random variable. Let $\mathbf{x}$ be a random variable and we denote the probability function by $\mathsf{P}(\mathbf{x})$. The entropy $H(\mathbf{x})$ of a variable $\mathbf{x}$ is defined by

$$H(\mathbf{x}) = -\int \mathsf{P}(\mathbf{x}) \log \mathsf{P}(\mathbf{x}) d\mathbf{x}. \tag{2}$$

According to Shannon's information theory [42], $I(\mathbf{x}; \mathbf{y})$ is defined over the joint probability distribution of $\mathbf{x}$ and $\mathbf{y}$ (i.e., $\mathsf{P}(\mathbf{x}, \mathbf{y})$) and their respectively marginal distributions (i.e., $\mathsf{P}(\mathbf{x})$ and $\mathsf{P}(\mathbf{y})$). Specifically,

$$\begin{aligned} I(\mathbf{x}; \mathbf{y}) &= \iint \mathsf{P}(\mathbf{x}, \mathbf{y}) \log \left(\frac{\mathsf{P}(\mathbf{x}, \mathbf{y})}{\mathsf{P}(\mathbf{x})\mathsf{P}(\mathbf{y})}\right) d\mathbf{x} d\mathbf{y} \\ &= -\int \left(\int \mathsf{P}(\mathbf{x}, \mathbf{y}) d\mathbf{y}\right) \log \mathsf{P}(\mathbf{x}) d\mathbf{x} - \int \left(\int \mathsf{P}(\mathbf{x}, \mathbf{y}) d\mathbf{x}\right) \log \mathsf{P}(\mathbf{y}) d\mathbf{y} \\ &\quad + \iint \mathsf{P}(\mathbf{x}, \mathbf{y}) \log \mathsf{P}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= -\int \mathsf{P}(\mathbf{x}) \log \mathsf{P}(\mathbf{x}) d\mathbf{x} - \int \mathsf{P}(\mathbf{y}) \log \mathsf{P}(\mathbf{y}) d\mathbf{y} + \iint \mathsf{P}(\mathbf{x}, \mathbf{y}) \log \mathsf{P}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \\ &= H(\mathbf{x}) + H(\mathbf{y}) - H(\mathbf{x}, \mathbf{y}), \end{aligned} \tag{3}$$

where $H(\cdot)$ denotes entropy and $H(\cdot, \cdot)$ denotes joint entropy. Intuitively, mutual information measures the information that X and Y share, by measuring how much knowledge one of variable reduces uncertainty about the other.

The most related work of our proposed method is information bottleneck principle. Given two random variables X and Y, let T be the hidden compressed representation of X with the Markov constraint $T \leftrightarrow X \leftrightarrow Y$. The IB method aims to seek a probabilistic mapping $q(t|x)$ to maximize the mutual information $I(T; Y)$, while under a constraint of input compression measured by the mutual information $I(X; T)$. This constrained optimization problem is formulated

as:

$$\underset{T\in\Delta}{\operatorname{argmax}}\, I(T;Y) \text{ s.t. } I(X;T) \leq r \tag{4}$$

Where r represents the level of compression, and Δ is the set of random variable T.

In practice, rather than solving the constrained problem. Eq. (4) can be optimized by minimizing so-called IB Lagrangian objective function written by:

$$\min_{q(t|x)} \mathcal{L}_{\mathrm{IB}}[T] = -I(T;Y) + \beta I(X;T), \tag{5}$$

where β is the Lagrange multiplier which is a free parameter that controls the trade-off between the the **sufficiency** (the performance on the task, as quantified by $I(Y;T)$) and the **minimality** (the complexity of the representation, as measured by $I(X;Z)$). In this sense, the IB principle also provides a natural approximation of a *minimal sufficient statistic* [43]. Our proposed method extends the information bottleneck principle to two views, where one is the clean sample and the other is the adversarial sample, and we want the latent representation T to contain the shared information between them. The detailed description and motivation of our proposed method is provided in Sections 3.2 and 3.3.

3.2. Design Motivation

Traditional adversarial training only considers adversarial examples and does not consider the relationship between $\mathbf{x}$ and $\mathbf{x}'$. Inspired by multi-view representation learning, we aim to learn latent representations $\mathbf{z}$ and $\mathbf{z}'$ (corresponding to $\mathbf{x}$ and $\mathbf{x}'$, respectively), which only contain the useful information shared by both $\mathbf{x}$ and $\mathbf{x}'$. Formally, the generation of these representations are defined by conditional distributions $p(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}'|\mathbf{x}')$, while satisfying the Markov chain $\mathbf{z} \to \mathbf{x} \to \mathbf{x}' \to \mathbf{z}'$. If the representation preserves only the shared information from both $\mathbf{x}$ and $\mathbf{x}'$, it would increase task-relevant information, discarding the view-specific details (the misleading information in adversarial example) and improving the adversarial robustness.

Let us consider the following information bottleneck (IB) setting on both clean and adversarial samples. The motivation for IB is to learn a representation that contains the most relevant information for the task.

$$\begin{aligned} &\min_{p(\mathbf{z}|\mathbf{x}),p(\mathbf{z}'|\mathbf{x}')} I(\mathbf{x};\mathbf{z}) + I(\mathbf{x}';\mathbf{z}') - \alpha I(\mathbf{z}';y) - (1-\alpha) I(\mathbf{z};y), \\ &\text{where } \mathbf{x}' = \underset{\mathbf{x}'\in\mathcal{B}(\mathbf{x},\epsilon)}{\arg\max}\, \mathcal{L}(f_\theta(\mathbf{x}'), y) \end{aligned} \tag{6}$$

where $\mathbf{x}'$ is generated by maximizing the cross-entropy loss, which is the same as the traditional adversarial training. However, the loss for outer minimization is different. α tradeoff the robust vs the natural accuracy. Since $I(\mathbf{z}, y) = H(y) - H(y|\mathbf{z})$, and $H(y)$ is constant, we can replace the $I(\mathbf{z}; y)$ and $I(\mathbf{z}'; y)$ with cross-entropy loss for clean and adversarial samples. we rewrite the outer minimization loss as follows:

$$\min_{p(\mathbf{z}|\mathbf{x}),p(\mathbf{z}'|\mathbf{x}')} I(\mathbf{x}; \mathbf{z}) + I(\mathbf{x}'; \mathbf{z}') - \alpha \, \mathrm{CE}(\mathbf{z}'; y) - (1 - \alpha) \, \mathrm{CE}(\mathbf{z}; y), \tag{7}$$

The motivation for the linear combination of adversarial and natural cross-entropy loss is that we want the latent representation $\mathbf{z}$ to contain not only the shared information, but also the task relevant information as well. Previous work like Adversarial Logit Pairing (ALG) [4] has already employed a mixture of clean and adversarial examples.

To make the latent representation $\mathbf{z}$ contains the shared information between $\mathbf{x}$ and $\mathbf{x}'$, we subdivide $I(\mathbf{x}; \mathbf{z})$ into three components by using the chain rule of mutual information, which is shown as follows:

$$I(\mathbf{z}; \mathbf{x}) = I(\mathbf{x}; \mathbf{z}|\mathbf{x}') + I(\mathbf{x}; \mathbf{x}') - I(\mathbf{x}; \mathbf{x}'|\mathbf{z}), \tag{8}$$

Here $I(\mathbf{x}; \mathbf{z}|\mathbf{x}')$ represents the information in $\mathbf{z}$ which is unique to $\mathbf{x}$ and not shared by $\mathbf{x}'$, which we call view-specific information. The second term $I(\mathbf{x}; \mathbf{x}')$ denotes the shared information between $\mathbf{x}$ and $\mathbf{x}'$. The last term $I(\mathbf{x}; \mathbf{x}'|\mathbf{z})$ is the shared information that is lost in $\mathbf{z}$. We want to minimize $I(\mathbf{x}; \mathbf{z}|\mathbf{x}')$ and $I(\mathbf{x}; \mathbf{x}'|\mathbf{z})$ such that a representation $\mathbf{z}$ is sufficient and minimal for the downstream task, because it contains all the task-relevant information (sufficiency) and without any irrelevant information (minimality).

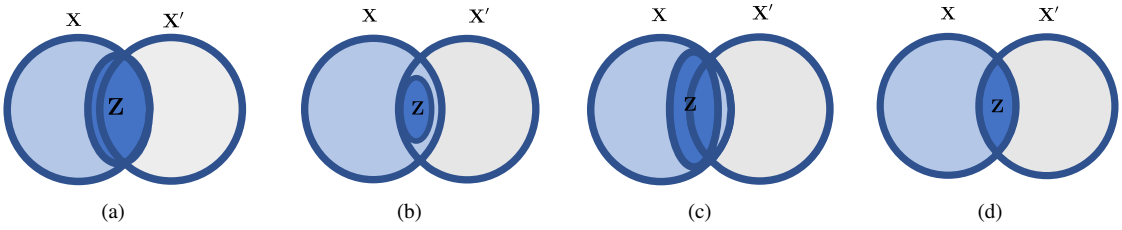


Figure 2: Illustration of sufficiency and minimality for representation $\mathbf{z}$ of $\mathbf{x}$ to $\mathbf{x}'$ in the Venn diagrams. Notice that the combined area of both circles capture the total information across $\mathbf{x}$ and $\mathbf{x}'$, the mutual information $I(\mathbf{x}; \mathbf{x}')$ corresponds to the intersection of the information in $\mathbf{x}$ $(H(\mathbf{x}))$ and information in $\mathbf{x}'$ $(H(\mathbf{x}'))$. (a) Sufficient but not minimal $(I(\mathbf{x}; \mathbf{z}|\mathbf{x}') > 0, I(\mathbf{x}; \mathbf{x}'|\mathbf{z}) = 0)$. (b) Minimal but not sufficient $(I(\mathbf{x}; \mathbf{z}|\mathbf{x}') = 0, I(\mathbf{x}; \mathbf{x}'|\mathbf{z}) > 0)$. (c) Not sufficient and not minimal $(I(\mathbf{x}; \mathbf{z}|\mathbf{x}') > 0, I(\mathbf{x}; \mathbf{x}'|\mathbf{z}) > 0)$. (d) Sufficient and minimal $(I(\mathbf{x}; \mathbf{z}|\mathbf{x}') = 0, I(\mathbf{x}; \mathbf{x}'|\mathbf{z}) = 0)$

The diagrams in Fig. 2 (d) demonstrate that we can obtain the sufficient and minimal representation $\mathbf{z}$, when it is exactly equal to the shared information of $\mathbf{x}$ and $\mathbf{x}'$. Otherwise, the representation $\mathbf{z}$ is either redundant, because

it contains some task-irrelevant information as in Fig. 2 (a), insufficient, because it does not capture enough useful information as in Fig.2 (b), or both insufficient and redundant, as in Fig. 2 (c). Similarly, we can also examine whether a representation $\mathbf{z}'$ is sufficient and minimal with respect to $\mathbf{x}'$ and $\mathbf{x}$.

3.3. Loss Function for Adversarial Robustness

In the previous section, we provide some motivation for our method. We next define the objective function for learning the representations $\mathbf{z}$ and $\mathbf{z}'$ that contain the shared information of $\mathbf{x}$ and $\mathbf{x}'$. As can be seen in Eq. 8, we could learn a representation containing only the shared information of $\mathbf{x}$ and $\mathbf{x}'$ by minimizing the view-specific information ($I(\mathbf{x};\mathbf{z}|\mathbf{x}')$) and shared information not in $\mathbf{z}$ ($I(\mathbf{x};\mathbf{x}'|\mathbf{z})$). In particular, minimizing $I(\mathbf{x};\mathbf{x}'|\mathbf{z})$ is equivalent to maximizing $I(\mathbf{z};\mathbf{x}')$, because $I(\mathbf{z};\mathbf{x}') = I(\mathbf{x};\mathbf{x}') - I(\mathbf{x};\mathbf{x}'|\mathbf{z})$ and given $\mathbf{x}$ and $\mathbf{x}'$, $I(\mathbf{x};\mathbf{x}')$ is constant. Thus, we can use a relaxed Lagrangian objective to obtain a representation $\mathbf{z}$ that is sufficient and minimal with respect to $\mathbf{x}$ and $\mathbf{x}'$ as follows:

$$\mathcal{L}_1 = I(\mathbf{x};\mathbf{z}|\mathbf{x}') - \lambda_1 \cdot I(\mathbf{z};\mathbf{x}'), \tag{9}$$

Symmetrically, we can find a representation $\mathbf{z}'$ that is sufficient and minimal with respect to $\mathbf{x}'$ and $\mathbf{x}$:

$$\mathcal{L}_2 = I(\mathbf{x}';\mathbf{z}'|\mathbf{x}) - \lambda_2 \cdot I(\mathbf{z}';\mathbf{x}), \tag{10}$$

Here λ_1 and λ_2 represent the Lagrangian multipliers for the the constrained optimization. The objective function involves two mutual information terms that are hard to calculate directly. To solve this problem, we derive some alternative bounds for these two mutual information terms through the following Theorems.

Theorem 3.1. *Under the Markov chain,* $\mathbf{z} \to \mathbf{x} \to \mathbf{x}' \to \mathbf{z}'$,

$$I(\mathbf{x};\mathbf{z}|\mathbf{x}') \leq D_{KL}(p(z|\boldsymbol{x})||p(z'|\boldsymbol{x}')). \tag{11}$$

Proof. The proof follows by noting that

$$\begin{aligned}
I(\mathbf{x};\mathbf{z}|\mathbf{x}') &= \mathbf{E}_{p(\mathbf{x},\mathbf{x}',\mathbf{z})}\left[\log\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x}|\mathbf{x}')}{p(\mathbf{x}|\mathbf{x}')p(\mathbf{z}|\mathbf{x}')}\right] \\
&= \mathbf{E}_{p(\mathbf{x},\mathbf{x}',\mathbf{z})}\left[\log\frac{p(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|\mathbf{x}')}\right] \\
&= \mathbf{E}_{p(\mathbf{x},\mathbf{x}',\mathbf{z})}\left[\log\frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{z}'|\mathbf{x}')}{p(\mathbf{z}'|\mathbf{x}')p(\mathbf{z}|\mathbf{x}')}\right] \\
&= D_{\mathrm{KL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')) - D_{\mathrm{KL}}(p(\mathbf{z}|\mathbf{x}')||p(\mathbf{z}'|\mathbf{x}')) \\
&\leq D_{\mathrm{KL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')).
\end{aligned} \tag{12}$$

□

Here, the conditional distributions $p(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}'|\mathbf{x}')$ can be parameterized by an encoder network. This bound is tight whenever the representation $\mathbf{z}$ is the same as $\mathbf{z}'$. Symmetrically, $I(\mathbf{x}';\mathbf{z}'|\mathbf{x})$ is upper bounded by $D_{\mathrm{KL}}(p(\mathbf{z}'|\mathbf{x}')||p(\mathbf{z}|\mathbf{x}))$.

Theorem 3.2. *Under the Markov chain,* $\mathbf{z} \to \mathbf{x} \to \mathbf{x}' \to \mathbf{z}'$,

$$I(\mathbf{z};\mathbf{x}') \geq I(\mathbf{z};\mathbf{z}'). \tag{13}$$

Proof. The proof follows by noting that

$$\begin{aligned} I(\mathbf{z};\mathbf{x}') &= I(\mathbf{z};\mathbf{z}',\mathbf{x}') - I(\mathbf{z};\mathbf{z}'|\mathbf{x}') \\ &= I(\mathbf{z};\mathbf{z}',\mathbf{x}') \\ &= I(\mathbf{z};\mathbf{z}') + I(\mathbf{z};\mathbf{x}'|\mathbf{z}') \\ &\geq I(\mathbf{z};\mathbf{z}'). \end{aligned} \tag{14}$$

□

Here, $I(\mathbf{z};\mathbf{z}'|\mathbf{x}') = 0$, because $\mathbf{z}'$, as the representation of $\mathbf{x}'$, is part of the Markov chain $\mathbf{z} \to \mathbf{x} \to \mathbf{x}' \to \mathbf{z}'$. Note that this lower bound can also be proofed directly through the data processing inequality. The derivation above illustrates that the bound is tight when $\mathbf{z}'$ is a sufficient statistic of $\mathbf{z}$. Symmetrically, a similar bound can be derived for $I(\mathbf{z}';\mathbf{x}) \geq I(\mathbf{z};\mathbf{z}')$. Conceptually, this lower bound captures our goal of preserving the information shared between the representations regardless of the adversarial perturbation.

We combine $\mathcal{L}_1$ and $\mathcal{L}_2$ so that the representations $\mathbf{z}$ and $\mathbf{z}'$ will contain the shared information between $\mathbf{x}$ and $\mathbf{x}'$. Based on the bounds derived from above Theorem, we obtain the following objective function, which is an upper bound on the average of $\mathcal{L}_1$ and $\mathcal{L}_2$:

$$\begin{aligned} &\mathcal{L}_{\text{shared}} = \frac{1}{2}(\mathcal{L}_1 + \mathcal{L}_2) \\ &= \frac{I(\mathbf{x};\mathbf{z}|\mathbf{x}') + I(\mathbf{x}';\mathbf{z}'|\mathbf{x})}{2} - \frac{\lambda_1 I(\mathbf{z};\mathbf{x}') + \lambda_2 I(\mathbf{z}';\mathbf{x})}{2} \\ &\leq \frac{D_{\mathrm{KL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')) + D_{\mathrm{KL}}(p(\mathbf{z}'|\mathbf{x}')||p(\mathbf{z}|\mathbf{x}))}{2} - \frac{\lambda_1 + \lambda_2}{2} \cdot I(\mathbf{z};\mathbf{z}') \\ &\leq D_{\mathrm{SKL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')) - \lambda \cdot I(\mathbf{z};\mathbf{z}'), \end{aligned} \tag{15}$$

where $p(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}'|\mathbf{x}')$ are modeled as Gaussian distributions parameterized by a neural network encoder $\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}), \texttt{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{x})))$ and $\mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}'), \texttt{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{x}')))$. D_{SKL} represents the symmetrized KL-divergence obtained by averaging $D_{\text{KL}}(p(\mathbf{z}'|\mathbf{x}')||p(\mathbf{z}|\mathbf{x}))$ and $D_{\text{KL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}'))$.

This symmetrized KL-divergence can be computed directly between two Gaussian posterior distributions.[1] However, $I(\mathbf{z};\mathbf{z}')$ requires the use of a mutual information estimator. In this paper, we utilize Hilbert Schmidt Independence Criterion (HSIC) [44] to measure the independence between $\mathbf{z}$ and $\mathbf{z}'$, and use this value to replace mutual information term. We use HSIC as a surrogate for mutual information because we can directly measure the dependence between two mini-batch samples in Reproducing Kernel Hilbert Space (RKHS) without requiring any density estimation or using an additional network for mutual information estimation. In addition, HSIC has been widely used as an efficient and tractable substitute for mutual information in a variety of cases such as layer-wise training [45] and robustness on covariate distribution shifts [46]. We also investigated other mutual information estimators: mutual information neural estimation (MINE) [47] and mutual information gradient estimation (MIGE) [48], but found that these perform worse than HSIC. The details and performance results for different mutual information estimator can be found in Appendix A.3.

We combine the above regularization objective $\mathcal{L}_{shared}$ with task label information to obtain our overall objective function for training the model:

$$\mathcal{L} = \alpha \cdot \text{CE}(f(\mathbf{x}'), y) + (1-\alpha) \cdot \text{CE}(f(\mathbf{x}), y) + \beta \cdot D_{\text{SKL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')) - \lambda \cdot I(\mathbf{z};\mathbf{z}'). \tag{16}$$

Here $\alpha \in [0,1]$ balances the tradeoff between the cross-entropy loss on clean and adversarial samples. β and λ adjust the importance of symmetrized KL-divergence and mutual information terms. Our proposed method is described in Algorithm 1. We first generate adversarial examples by maximizing the cross-entropy loss with respect to small perturbation added to corresponding clean samples and then obtain the latent distribution $p(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}'|\mathbf{x})$ parameterized by the encoder. Finally, we update the network's parameters with our proposed objective function until the convergence

3.4. Relation to Existing Works

In this section, we briefly compare our proposed method to existing adversarial regularization approaches, including standard adversarial training [3], max-margin adversarial training (MMA) [5], TRADES [6] and MART [7]. The loss function for the above approaches and our proposed method are shown in Table 1.

[1] various choices for the latent representation (e.g., Laplacian, Gamma, and Cauchy) are possible. However, this choice may be less crucial since a non-linear decoder may adequately handle arbitrary distributions. We utilize the Gaussian latent distribution in our proposed method since it is easily reparameterized and widely used in many applications (e.g., VAE and variational IB).

Algorithm 1 *Adversarial training with shared information regularization*

f_θ: network model, N: data sample batch size, E: total number of epochs.
for epoch $\in \{1, \dots, E\}$ **do**
  Sample batch $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ ▷ Sample batch from clean data
  for sample index $i \in \{1, \dots, N\}$ **do**
    $\mathbf{x}_i' = \text{attack}(\mathbf{x}_i, y_i)$ ▷ Generate adversarial examples
    Input $\mathbf{x}_i$ to the encoder to compute the parameters $\boldsymbol{\mu}_\theta(\mathbf{x}_i)$ and $\boldsymbol{\sigma}_\theta^2(\mathbf{x}_i)$
    Input $\mathbf{x}_i'$ to the encoder to compute the parameters $\boldsymbol{\mu}_\theta(\mathbf{x}_i')$ and $\boldsymbol{\sigma}_\theta^2(\mathbf{x}_i')$
    Sample representations $\mathbf{z}_i \sim \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x}_i), \texttt{diag}(\boldsymbol{\sigma}_\theta^2(\mathbf{x}_i)))$, $\mathbf{z}_i' \sim \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{x_i}'), \texttt{diag}(\boldsymbol{\sigma}_\theta^2(\mathbf{x}_i')))$
  end for
  Calculate $D_{\text{SKL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}'))$ and $I(\mathbf{z};\mathbf{z}')$
  $\mathcal{L} = \alpha \cdot \text{CE}(f_\theta(\mathbf{x}'), y) + (1-\alpha) \cdot \text{CE}(f_\theta(\mathbf{x}), y) + \beta \cdot D_{\text{SKL}}(p(\mathbf{z}|\mathbf{x})||p(\mathbf{z}'|\mathbf{x}')) - \lambda \cdot I(\mathbf{z};\mathbf{z}')$
  $g_\theta \leftarrow \nabla_\theta \mathcal{L}$
  $\theta \leftarrow \text{Step}(\theta, g_\theta)$
end for

Table 1
Loss function comparison with existing work.

Defense method	Loss function
Standard	$\text{CE}(f(\mathbf{x}'), y)$
TRADES	$\text{CE}(f(\mathbf{x}), y) + \beta \cdot D_{\text{KL}}(p(y\|x)\|\|p(y\|x'))$
MMA	$\text{CE}(f(\mathbf{x}'), y) \cdot \mathbb{1}(h_\theta(\mathbf{x} = y) + \text{CE}(f(\mathbf{x}), y \cdot \mathbb{1}(h_\theta(\mathbf{x} \neq y)$
MART	$\text{BCE}(f(\mathbf{x}'), y) + \beta \cdot D_{\text{KL}}(p(y\|x)\|\|p(y\|x')) \cdot (1 - p_y(\mathbf{x}))$
Ours	$\alpha \cdot \text{CE}(f(\mathbf{x}'), y) + (1-\alpha) \cdot \text{CE}(f(\mathbf{x}), y) + \beta \cdot D_{\text{SKL}}(p(\mathbf{z}\|\mathbf{x})\|\|p(\mathbf{z}'\|\mathbf{x}')) - \lambda \cdot I(\mathbf{z};\mathbf{z}')$

Our proposed method differs from existing adversarial regularization approaches in the following respects. (1) TRADES and MART both put the regularization (e.g., KL-divergence term) on the output of the SoftMax layer. However, in our objective function, $\mathbf{z}$ can represent any intermediate layer, including the output of the last layer. (2) Whereas TRADES employs cross-entropy loss on clean samples, MART utilizes boosting cross-entropy loss on adversarial samples. Our approach uses a linear combination of cross-entropy loss on clean and adversarial samples, which is different from MMA, which uses hard decisions to discriminate cross-entropy loss on clean and adversarial samples. (3) Unlike TRADES and MART, which only use a single KL-divergence regularization term to minimize view-specific information. Our objective includes two regularization components: the symmetric KL-divergence and the mutual information between clean and adversarial latent representations. The additional mutual information term helps us avoid a trivial solution with zero KL-divergence in which all inputs are mapped to a single point in representation space. Furthermore, these two components encourage the latent representations $\mathbf{z}$ and $\mathbf{z}'$ to capture only the shared information between the clean example and its corresponding adversarial example.

Table 2
Parameter configuration for different adversarial attacks.

Dataset	PGD(ℓ_∞)	C&W(ℓ_2)	AutoAttacks(ℓ_∞)
MNIST	$\epsilon = 0.3$ $\eta = 0.01$ $K = 20, 40$	$c = 1$ $k = 0$ $Steps = 20$ $lr = 0.003$	$\epsilon = 0.031, n_{class} = 10$ APGD-CE APGD-DLR FAB Square ($n_{queries} = 5000$)
CIFAR-10/100	$\epsilon = 0.031$ $\eta = 0.003$ $K = 20, 40$	$c = 1$ $k = 0$ $Steps = 20$ $lr = 0.01$	$\epsilon = 0.031, n_{class} = 10, 100$ APGD-CE APGD-DLR FAB Square ($n_{queries} = 5000$)

4. Experiments

In this section, we first provide the adversarial attacks and implementation details for the baseline methods and our methods, and then compare our proposed method with TRADES and MART in terms of natural-robust tradeoff curve and analyze the hyper-parameters of our method. Then, we evaluate our method's robustness on benchmark datasets with different types of attacks in both white-box and black-box settings. Finally, we provide an ablation study for our proposed method.

4.1. Implementation Details

In this paper, we evaluate robustness with different attacks including PGD^{20}, PGD^{40}, C&W(ℓ_2), and AutoAttack (AA). All of the attack methods used the open-source *Torchattacks* package [49]. We report the parameter configuration for different adversarial attacks in Table 2. In particular, we report the maximum perturbation ϵ, step size η, and number of steps K for PGD attacks. For C&W attack, we provide the box-constraint parameter c, confidence k, number of steps, and learning rate lr of the Adam optimizer. AutoAttacks consist of four different attacks. For APGD-CE, APGD-DLR, and FAB attacks, we evaluate ℓ_∞ norm, and the step is set to 100, and for Square Attack, we use 5000 queries. Robust accuracy for AutoAttack is calculated as follows: a successful attack is counted if it is misclassified for any of the adversarial attack methods.

In addition, we also report all tuning parameters for the baseline methods including TRADES, MART and MMA, and our method in Table 3. In detail, we report the parameter settings on 4-layers CNN for MNIST, ResNet18 and WideResNet-34-10 for CIFAR-10 and CIFAR-100.

In order to calculate the symmetric KL divergence term in our objective function, we add an additional stochastic layer before the last layer to parameterize the representation mean and standard deviation for latent Gaussian distribution within various architectures. Specifically, for 4-layer CNN, the size of the additional stochastic layer is 2×128; for ResNet18, the size of the additional stochastic layer is 2×256; for WideResNet-34-10, the size of the

Table 3
Parameter summary for MNIST, CIFAR-10, and CIFAR-100 in the training process for baseline methods.

Dataset	param.	TRADES	MART	MMA	Ours
MNIST	architecture	4-layer CNN	4-layer CNN	4-layer CNN	4-layer CNN
	batch size	128	128	128	128
	attacks	$\text{PGD}^{40}(\epsilon = 0.3)$	$\text{PGD}^{40}(\epsilon = 0.3)$	$\text{PGD}^{40}(\epsilon = 0.3)$	$\text{PGD}^{40}(\epsilon = 0.3)$
	optimizer	SGD	SGD	Adam	SGD
	weight decay	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
	learning rate	0.01	0.01	1×10^{-4}	0.01
	lr scheduler	divided by 10 at 55^{th}, 75^{th} and 90^{th}.	divided by 10 at 20^{th} and 40^{th}.	-	divided by 10 at 55^{th}, 75^{th} and 90^{th}.
	max margin	-	-	0.45	-
	epochs	100	50	50	100
CIFAR-10/100	architecture	ResNet18/WideResNet-34-10	ResNet18/WideResNet-34-10	ResNet18	ResNet18/WideResNet-34-10
	batch size	128	128	128	128
	attacks	$\text{PGD}^{20}(\epsilon = 0.031)$	$\text{PGD}^{20}(\epsilon = 0.031)$	$\text{PGD}^{20}(\epsilon = 0.031)$	$\text{PGD}^{20}(\epsilon = 0.031)$
	optimizer	SGD	SGD	SGD	SGD
	weight decay	2×10^{-4}	3.5×10^{-4}	2×10^{-4}	5×10^{-4}
	learning rate	0.1	0.1	0.3	0.1
	lr scheduler	divided by 10 at 75^{th} and 90^{th}.	divided by 10 at 75^{th} and 90^{th}.	0.09 at 75^{th} and 0.003 after 90^{th}.	divided by 10 at 75^{th}, 90^{th} and 100^{th}.
	max margin	-	-	12/255	-
	epochs	120	120	120	120

additional stochastic layer is 2×320. We utilize HSIC to approximate the mutual information term in our proposed objective function, the kernel size selected as follows. For each sample in the mini-batch, we evaluate its $k(k = 10)$ nearest distances and take the mean. We choose kernel width σ as the average mean value for all samples. The initial value of β is set to 10^{-3}. Starting from epoch 10, the value of β is exponentially increased up to the final value during the following 90 epochs and then kept fixed until the end of the training.

4.2. Performance Comparison with TRADES and MART

To compare with TRADES and MART, we train ResNet-18 [50] on CIFAR-10 [51] dataset with 120 epochs. We keep our hyper-parameters α and λ fixed during training, while β is slowly increasing to its final value with an exponential schedule, since starting with larger β results in the encoder collapsing into a fixed representation. For TRADES and MART, we use the same parameters described in their original papers. All input images are normalized into [0,1] and augmented using random crops after 4-pixel padding and random flips. The maximum perturbation used is $\epsilon = 0.031(\ell_\infty)$. We use PGD^{10} with step size 0.007 as the training attack, and PGD^{20} with step size 0.003 as the test attack.

We compare TRADES and MART in terms of the natural-robust accuracy curve with different $\beta \in [0.1, 30]$, which controls the strength of the regularization (please refer to Table 1). For our method, we put the representation $\mathbf{z}$ as the output of the layer before the last layer and set $\alpha = 0.5$, $\lambda = 0.001$ and test different β. Figure 3 describes robust accuracy changes over natural accuracy. Each point on the curve represents a single choice of parameter β and shows the average natural and robust accuracy on the CIFAR-10 test set for the last epoch with three runs. The horizontal and vertical error bars represent the natural and robust accuracy standard deviation, respectively. The right

side corresponds to small β, and the left side corresponds to large β. As β increases, the natural accuracy for all the methods decreases because the regularization term makes the outputs of clean and adversarial samples closer, which hurts natural accuracy. Furthermore, the robust accuracy first increases and then falls because an appropriate β can improve both natural and robust accuracy. In contrast, a relatively large or small β will either ignore the label information or put a strong regularization penalty, degrading the robust accuracy. Fig. 3 demonstrates that our proposed method achieves better robust accuracy with similar natural accuracy as compared to TRADES and MART.

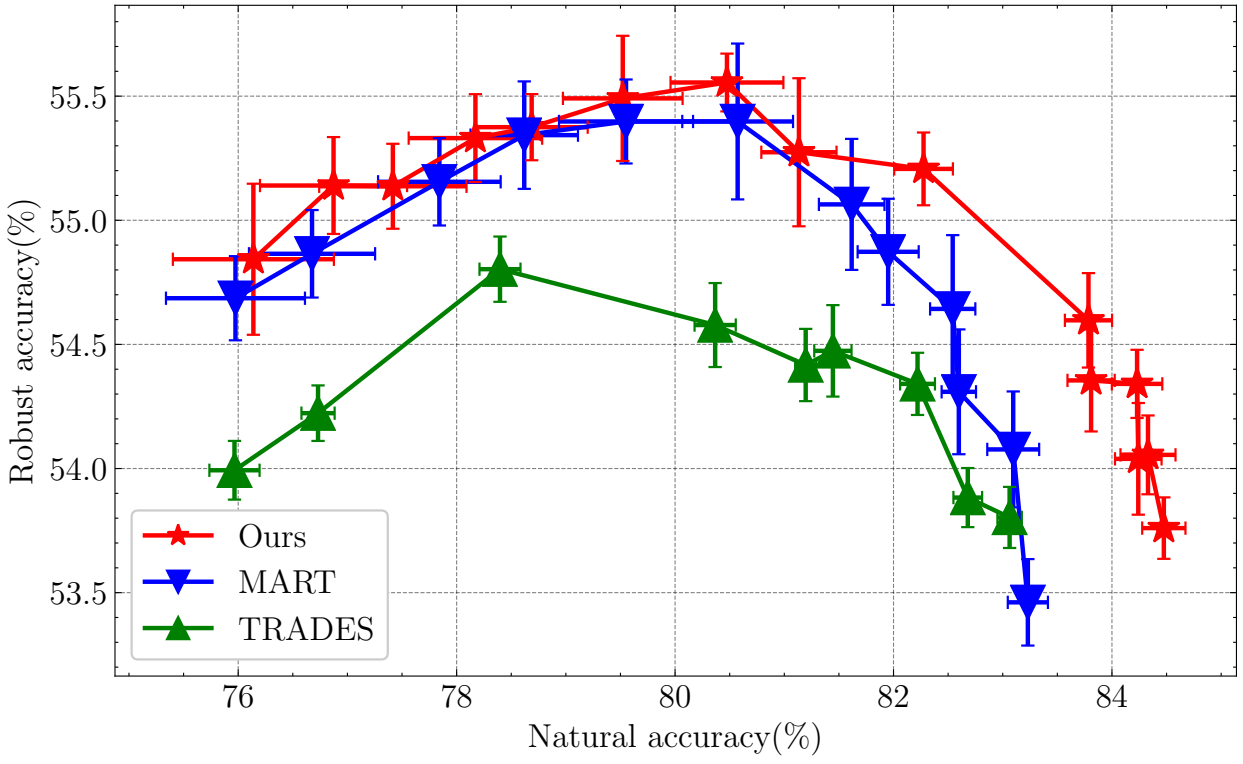


Figure 3: Robust-natural accuracy comparison with different β for different methods.

4.3. Robustness Evaluation and Analysis

In this section, we evaluate robustness on MNIST [52], CIFAR-10 [51] and CIFAR-100 [51] datasets against various white-box and black-box attacks. The hyperparameters α and λ are selected based on the validation set, and further experiments for different α and λ can be found in Appendix A.2.

For the MNIST dataset, all defense models are trained with 100 epochs with four convolutional layers, followed by three fully-connected layers. The training attack we use is PGD^{40} with step size 0.01 and perturbation limit $\epsilon = 0.3$ (ℓ_∞). We set $\alpha = 0.5$, $\beta = 4$, $\lambda = 0.001$. For CIFAR-10, the defense models are ResNet-18 and WideResNet-34-10

Table 4
White-box robustness (%) on MNIST with 4-layer CNN and CIFAR-10 with ResNet18.

Method	MNIST					CIFAR-10				
	Natural	PGD^{20}	PGD^{40}	C&W(ℓ_2)	AA	Natural	PGD^{20}	PGD^{40}	C&W(ℓ_2)	AA
Standard	99.16 ± 0.02	96.25 ± 0.04	95.54 ± 0.05	94.32 ± 0.08	90.59 ± 0.12	**82.59** ± 0.17	49.55 ± 0.19	47.58 ± 0.10	60.54 ± 0.21	42.11 ± 0.20
TRADES	99.14 ± 0.03	97.85 ± 0.06	96.44 ± 0.03	97.28 ± 0.02	92.90 ± 0.08	81.83 ± 0.18	53.38 ± 0.15	52.04 ± 0.13	69.03 ± 0.20	49.54 ± 0.25
MMA	99.12 ± 0.01	97.35 ± 0.04	95.89 ± 0.03	96.15 ± 0.05	91.02 ± 0.13	82.28 ± 0.15	51.66 ± 0.14	49.15 ± 0.17	63.45 ± 0.23	42.44 ± 0.19
MART	**99.28** ± 0.03	98.10 ± 0.05	**96.75** ± 0.06	97.32 ± 0.08	92.60 ± 0.09	81.57 ± 0.25	55.14 ± 0.21	53.07 ± 0.18	73.72 ± 0.14	50.60 ± 0.21
Ours	99.10 ± 0.05	**98.60** ± 0.03	96.64 ± 0.06	**98.23** ± 0.06	**95.76** ± 0.15	81.87 ± 0.27	**55.51** ± 0.17	**54.30** ± 0.21	**77.43** ± 0.26	**57.40** ± 0.28

Table 5
White-box robustness (%) on CIFAR-10 and CIFAR-100 with WideResNet-34-10.

Method	CIFAR-10					CIFAR-100				
	Natural	PGD^{20}	PGD^{40}	C&W(ℓ_2)	AA	Natural	PGD^{20}	PGD^{40}	C&W(ℓ_2)	AA
Standard	86.21 ± 0.10	52.31 ± 0.13	50.58 ± 0.15	63.23 ± 0.20	47.04 ± 0.19	60.11 ± 0.15	24.92 ± 0.20	23.93 ± 0.18	34.67 ± 0.21	21.58 ± 0.18
TRADES	85.73 ± 0.12	55.15 ± 0.11	53.55 ± 0.14	69.43 ± 0.21	51.20 ± 0.17	56.97 ± 0.12	26.87 ± 0.18	25.93 ± 0.16	35.97 ± 0.14	24.50 ± 0.23
MART	85.99 ± 0.11	56.66 ± 0.18	54.35 ± 0.17	75.27 ± 0.22	53.20 ± 0.15	57.72 ± 0.17	30.27 ± 0.12	29.29 ± 0.11	41.38 ± 0.27	27.10 ± 0.21
Ours	**86.25** ± 0.13	**57.47** ± 0.23	**55.11** ± 0.15	**79.84** ± 0.28	**57.70** ± 0.13	**60.66** ± 0.21	**34.15** ± 0.22	**32.82** ± 0.17	**55.18** ± 0.25	**32.25** ± 0.18

(depth 34 and width 10) [53] trained with PGD^{10} attack with step size 0.007 and perturbation limit $\epsilon = 0.031$ (ℓ_∞). We set $\alpha = 0.5, \beta = 4, \lambda = 0.001$. For CIFAR-100, we use the same data pre-processing, training optimizer and learning rate schedule as the CIFAR-10 experiment, the defense model is WideResNet-34-10 with PGD^{20} attacks for training, and set $\alpha = 0.5, \beta = 6, \lambda = 0.001$.

White-box Robustness. We evaluate white-box robustness of different methods by four types of attacks: PGD^{20}, PGD^{40}, C&W(ℓ_2), and AutoAttack (AA), with $\epsilon = 0.3$ for MNIST and $\epsilon = 0.031$ for CIFAR10 and CIFAR100. The white-box robustness for MNIST with 4-layer CNN and CIFAR-10 with ResNet18 are reported in Table 4, where "Natural" denotes the accuracy on the clean images. Each value in the table is the mean accuracy for the last epoch and standard deviation over three runs. Our proposed method achieves comparable or better robust accuracy on both MNIST and CIFAR-10 dataset. We further compare the white-box robustness of our method against TRADES and MART by training each method with a larger WideResNet model on CIFAR-10 and CIFAR-100 datasets. We report the robustness with different attacks on Table 5. Our proposed method achieves better robust accuracy while maintaining the same natural accuracy, showing a benefit, especially on the CIFAR-100 dataset with some strong attacks (e.g., C&W(ℓ_2) and AA). An additional experiment comparing performance with early-stopping is shown in Appendix A.1

Black-box Robustness. Adversarial samples for black-box attacks are created from natural images by attacking a pre-trained surrogate model on the original training set. We evaluate the black-box robustness on a pre-trained ResNet-50 surrogate model on the CIFAR-10 dataset with 100 epochs. The attacking methods we use are: PGD^{10}, PGD^{20}, PGD^{40}, C&W(ℓ_2) and black-box Square attack [54] with $\epsilon = 0.031$. The results in Table 6 show the mean

Table 6
Black-box robustness (%) on CIFAR-10.

Method	CIFAR-10					
	Natural	PGD^{10}	PGD^{20}	PGD^{40}	C&W(ℓ_2)	Square attack
Standard	**82.59**	80.27	80.01	79.77	80.85	76.85
TRADES	81.83	81.37	81.27	81.29	81.75	81.60
MMA	81.87	81.16	80.58	80.37	**82.24**	79.37
MART	81.57	81.38	81.34	81.27	81.59	82.05
Ours	82.28	**81.42**	**81.37**	**81.31**	81.85	**83.53**

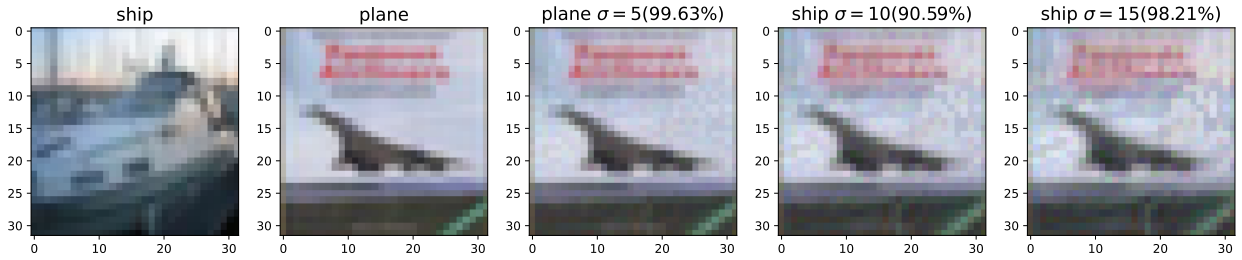
accuracy for three runs. Our method achieves comparable or greater robustness than other methods across different types of attacks. Compared with the white-box attacks, all defense models with black-box attacks achieve much better robust accuracy, even close to the natural accuracy. This suggests that adversarial training is a practical choice against black-box attacks where the target model is secret for potential attacks.

4.4. Results on the attack toward the representation

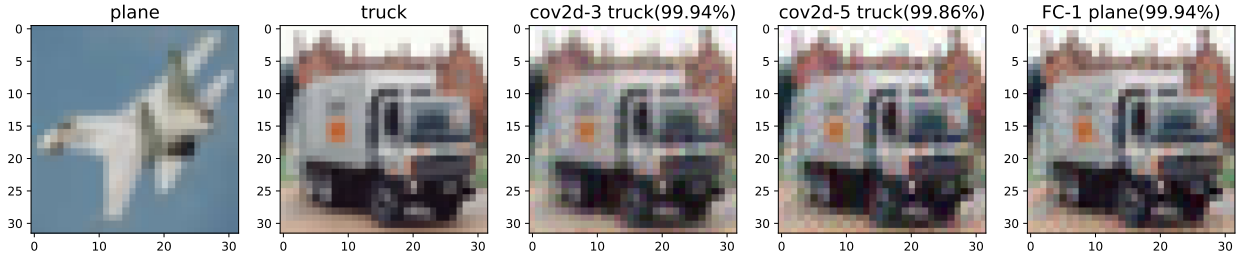
In this section, we will evaluate our proposed method against a different class of adversarial attack [55] where the attacker tries to minimize the feature-space distance, at a particular layer of the target neural network, between clean and adversarial images. The adversarial sample I_α is defined as follows:

$$\begin{aligned} &I_\alpha = \arg\min_I ||\phi_k(I) - \phi_k(I_g)||_2^2 \\ &\text{subject to } ||I - I_s||_\infty \leq \sigma/255 \end{aligned} \tag{17}$$

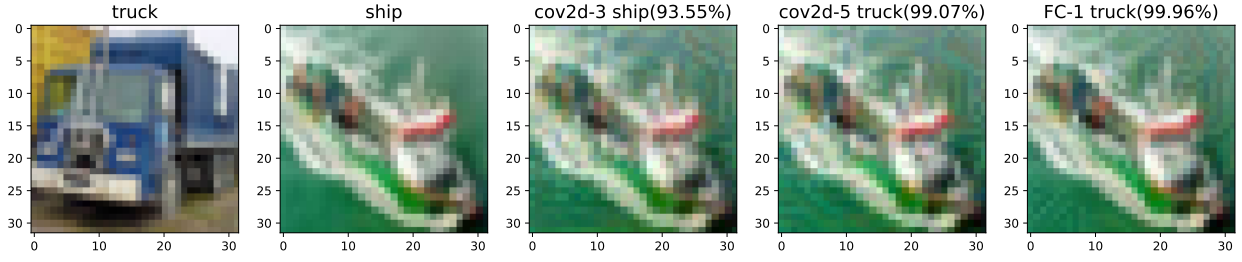
Here, I_s and I_g denote the source and guide images, ϕ_K is the mapping from an image to the model's representation at the k^{th} layer. The goal of the adversarial representation attack is to find a new image I_α such that the distance between the corresponding representation of I_α and I_g at the k^{th} layer is as small as possible. At the same time, I_α should remain close to the source image I_s. In general, the exact computation of Eq. (17) is a hard problem, and thus following [55], we fixed σ and optimize Eq. (17) by using the l-BFGS-b algorithm with the inequality expressed as a box constraint around I_s. We report some adversarial examples generated by this adversarial representation attack in Figure 4. As can be seen, when σ increases distortion becomes more perceptible and we observe that the larger σ is, the higher the attack strength will be. In addition, we also note that the shallower layer is more robust than the deep layer for the same input perturbation level. Figure 4 (b) and Figure 4 (c) demonstrate that deep layers (eg., "FC-1" and "cov2d-5") are more likely to be attacked than lower layers (eg., "cov2d-3").



(a) Adversarial examples for the same latent output with different σ, fool the classifier successfully when $\sigma = 10, 15$.



(b) Adversarial examples for the different latent output with same $\sigma = 10$, fool the classifier successfully when put attacks on "FC-1" layer.



(c) Adversarial examples for the different latent output with same $\sigma = 10$, fool the classifier successfully when put attacks on "FC-1" and "cov2d-5" layer.

Figure 4: Adversarial examples generated by adversarial manipulation attack with different layers (eg., cov2d-3 (convolution layer 3), cov2d-5 (convolution layer 5) and FC-1 (last fully connected layer before output layer), of VGG-16 on CIFAR-10 dataset. The first column represents the guide images, the second column represents the source images. The last three columns represent the adversarial examples.

We evaluate the performance of our approach with the adversarial representation attack (ARA) on the last layer before the output layer. In the training process, we set $\sigma = 10$ and step to 20. Other training parameters are set the same as that of training for PGD attacks. In the training process, for each mini-batch, we select the first half as guide images and the last half as the source images. The step parameter is the number of learning iterations for the l-BFGS-b algorithm. We report the robustness accuracy with different σ and step on the CIFAR-10 test set with ResNet18 in Table 7. As can be seen, our proposed method consistently performs better than other methods for the attack on the representation layer, because our proposed method encourages the latent representation only to contain the shared information between the clean and adversarial examples and discard the superfluous information.

Table 7
Performance of different methods with adversarial manipulation attack on CIFAR-10 datasets.

Method	CIFAR-10 (ResNet18) Robust accuracy (%)					
	$step = 10$, $\alpha = 5$	$step = 10$, $\alpha = 10$	$step = 10$, $\alpha = 15$	$step = 20$, $\alpha = 5$	$step = 20$, $\alpha = 10$	$step = 20$, $\alpha = 15$
Standard	73.54	65.83	53.65	73.54	64.17	49.48
TRADES	73.20	66.86	56.78	71.58	65.73	51.46
MART	75.24	68.20	58.35	74.96	66.00	57.13
Ours	**77.83**	**70.41**	**61.66**	**77.76**	**69.61**	**59.20**

Table 8
Computational cost of different methods

Method	Training	Testing	GPU memory	Params
Baseline	2.47s	2.35s	2.20G	1.12M
TRADES	3.27s	2.51s	2.25G	1.12M
MART	2.56s	2.46s	2.99G	1.12M
Ours	3.81s	2.54s	3.48G	1.14M

4.5. Computational Complexity and Data-efficiency Analysis

In this section, we compare our method with others in terms of computational complexity and data-efficiency. Specifically, we first compare the extra computational and memory costs incurred by our proposed approach and other methods on the CIFAR-10 dataset with a ResNet18 model. Our experiments are running on one NVIDIA RTX 2080Ti GPU. We report the training time, test time for one iteration, GPU memory cost, and the number of architecture parameters in Table 8. The baseline method is the conventional adversarial training without any regularization terms. As can be seen, our method needs more time for training and higher GPU memory cost due to modifying the latent representation and employing extra resources for mutual information estimation. However, in terms of the testing time and the number of architecture parameters, our method is comparable with other approaches, which suggests that our method is a cost-effective way to defend against adversarial attacks.

We also analyze our proposed method's data efficiency compared with other methods. Specifically, we only use 40%, 60%, and 80% of the training set and evaluate the performance on the entire test set. The performance on the entire CIFAR-10 test set is shown in Table 9. We observe that standard adversarial training is more sensitive than other methods when gradually increasing the size of the training set. MART is less data-efficient than TRADES and ours because it requires more data during the training to correctly detect misclassified samples. Our method is more data-efficient than TRADES and achieves comparable accuracy using only 60% of the training set compared to TRADES using 80% of the training set.

Table 9
Performance with adversarial manipulation attack by only training subset CIFAR-10 dataset.

Method	Percentage of Training Set		
	40%	60%	80%
Standard	59.48	66.01	69.11
TRADES	64.40	67.27	69.65
MART	62.56	69.13	70.99
Ours	**66.76**	**69.31**	**71.22**

Table 10
Ablation study for our method with PGD^{20} attack on CIFAR-10 (%).

$\text{CE}(f(\mathbf{x}), y)$	$\text{CE}(f(\mathbf{x}'), y)$	D_{SKL}	$I(\mathbf{z}; \mathbf{z}')$	PGD^{20}	Natural
✓		✓	✓	50.73	78.43
	✓	✓	✓	53.47	76.06
✓	✓		✓	49.60	81.48
✓	✓			45.57	82.48
✓	✓	✓		54.77	81.34
✓	✓	✓	✓	**55.51**	**81.87**

4.6. Ablation Study

This section quantifies the impact of different components of our proposed objective function on the CIFAR-10 dataset with ResNet18 architecture trained with 120 epochs. In Table 10, each value represents the average accuracy at the last epoch for three runs. We see that removing any part of our objective degrades either natural or robust accuracy. Specifically, removing $\text{CE}(f(\mathbf{x}), y)$ degrades the natural accuracy and removing $\text{CE}(f(\mathbf{x}'), y)$ degrades the robust accuracy. When adding the $I(\mathbf{z}; \mathbf{z}')$ term to $\text{CE}(f(\mathbf{x}), y)$ and $\text{CE}(f(\mathbf{x}'), y)$ improves the natural accuracy while degrading the robust accuracy. The reason is that maximizing $I(\mathbf{z}; \mathbf{z}')$ only encourages the representation to learn the sufficient information of inputs. However, it may contain some view-specific information, resulting in the degradation of robust accuracy. On the other hand, incorporating D_{SKL} terms with $\text{CE}(f(\mathbf{x}), y)$ and $\text{CE}(f(\mathbf{x}'), y)$ improves the robust accuracy but decreases the natural accuracy, which suggests that minimizing only the D_{SKL} term may lose some information needed for sufficiency. In addition, when only employing the linear combination of $\text{CE}(f(\mathbf{x}), y)$ and $\text{CE}(f(\mathbf{x}'), y)$, and set $\alpha = 0.5$, we obtain the better natural accuracy but worse robust accuracy. This suggest that our proposed regularization terms improve the robust accuracy and maintain the natural accuracy. We see that using all terms together leads to the best performance.

5. Conclusion and Future Works

In this paper, we propose a novel, simple, and effective adversarial defense method against both white-box and black-box attacks by learning the shared information between clean samples and corresponding adversarial

examples. Unlike the previous defense methods that only consider adversarial examples, our method exploits the shared information between the clean samples and adversarial examples in the latent feature space, which preserves the task-relevant information and discards the misleading information of adversarial examples. Specifically, we consider the adversarial examples as the secondary view of clean samples and we maximize the mutual information between the representation of clean samples and adversarial examples and minimize the view-specific information as well. In addition, to avoid estimating the mutual information in high dimensions, we also utilize the upper bound of view-specific information as the surrogate term in our objective function. Extensive experiments on the image benchmark datasets demonstrate that our proposed method is more data-efficient and consistently performs better than previous state-of-art adversarial regularization methods, especially under strong attacks.

This work could be extended in the following research directions. First, it is straightforward to extend shared information to more than two views. For instance, each view could be an adversarial example with different attack strengths. Next, we will seek a deeper theoretical understanding of why the shared information improves the adversarial robustness. The proposed shared information terms could also be combined with a robust self-supervised training approach to further improve the tradeoff between robust and natural accuracy. Another opportunity for future research is to employ our shared information approach to learn the domain-shared and domain-specific information for domain adaptation and domain generation tasks.

A. Additional Experiments

In this section, we first compare other methods with early stopping and then put additional experiments to study the effect of parameter α and λ in our objective function and investigate their impact. Then, we utilize different mutual information approximation methods to estimate $I(\mathbf{z}, \mathbf{z}')$. Finally, we compare the performance of putting shared information regularization terms on the output of the softmax layer and the layer before the last layer.

A.1. Compare with other method with the early stopping

In the main paper, we compare other methods in terms of the last epoch performance. To make a fair comparison, we also compare with other methods in terms of robust and natural accuracy with early stopping, shown in Table 11. For TRADES and MMA, we stop training at 76^{th} epoch. For MART, we stop training at 91^{th} epoch. Our method is stopped at 101^{th} epoch. Our method can achieve comparable natural accuracy and perform better with robust accuracy, especially for a strong attack like AA.

Table 11
White-box robustness (%) on CIFAR-10 for different methods with early stopping.

Method	CIFAR-10				
	Natural	PGD^{20}	PGD^{40}	C&W(ℓ_2)	AA
Standard	82.12	49.77	48.25	61.23	43.71
TRADES	81.59	53.21	52.95	68.53	50.18
MMA	81.78	52.21	48.85	62.89	43.61
MART	82.23	55.32	54.15	75.19	52.31
Ours	**82.38**	**55.54**	**54.23**	**76.58**	**56.21**

Table 12
Comparisons of the performance of different hyper-parameter settings.

Hyper-parameters			Metrics (%)	
α	β	λ	Natural	PGD^{20}
0	4	0.001	75.38	51.36
0.1	4	0.001	78.38	53.17
0.3	4	0.001	80.94	55.37
0.5	**4**	**0.001**	**81.37**	**55.58**
0.7	4	0.001	78.37	55.35
0.9	4	0.001	77.10	55.25
1.0	4	0.001	76.16	54.84
0.5	4	0.003	79.60	55.46
0.5	4	0.005	79.38	55.42
0.5	4	0.007	79.51	55.57
0.5	4	0.01	80.24	55.37

A.2. The Effect of Parameter α and λ

We have already examined the effect of β in our objective function in the experiments part of the main paper. In this part, we fixed β and further investigated performance for different values of the hyper-parameters α (in front of the robust cross-entropy loss) and λ (controlling the mutual information term) on the CIFAR-10 dataset with ResNet18.

We first fixed $\lambda = 0.001$, and test different α from 0 to 1. Then, we fix α and test different λ. The final results are reported in Table 12. As can be seen, a larger α helps the robust accuracy and a larger λ is beneficial for the natural accuracy, the best natural and robust accuracy is achieved by $\alpha = 0.5, \beta = 4, \lambda = 0.001$.

A.3. The Effect of Different Mutual Information Estimators

Mutual Information Neural Estimation (MINE). Let X and Z are two random variable, $\mathcal{F} = \{T_\theta\}_{\theta\in\Theta}$ be the set of functions parameterized by a neural network. The definition of MINE [47] is:

$$I(X;Z) = \sup_{\theta\in\Theta} \mathbb{E}_{P(X,Z)}[T_\theta] - \log(\mathbb{E}_{P_X\otimes P_X}[e^{T_\theta}]), \tag{18}$$

Table 13
Performance Comparison with difference mutual information estimator.

Method	λ	Natural(%)	PGD^{20}(%)
Ours (MINE)	0.01	**81.54**	52.18
Ours (MIGE)	0.01	80.15	53.73
Ours (HSIC)	0.001	81.37	**55.58**

When estimating the mutual information between the layer's output before the last layer in ResNet18 on the CIFAR-10 dataset, the architecture of the mutual information estimator neural network is an MLP with fully connected layers of the form $256 - 512 - 512 - 1$, with ReLU activation.

Hilbert-Schmidt Independence Criterion (HSIC). HSIC [56] is a statistical dependency measure, which calculates the Hilbert-Schmidt norm of cross-covariance matrix between two sets of variables in a Reproducing Kernel Hilbert Space (RKHS). HSIC(X, Z) is defined as:

$$\begin{aligned}\mathrm{HSIC}(X;Z) &= \mathbb{E}_{X,Z\sim P(X,Z),X',Z'\sim P(X,Z)}[K_X(X,X')K_Z(Z,Z')] \\ &+ \mathbb{E}_{X\sim P(X),X'\sim P(X'),Z\sim P(Z),Z'\sim P(Z)}[K_X(X,X')K_Z(Z,Z')] \\ &- 2\mathbb{E}_{X,Z\sim P(X,Z),X'\sim P(X),Z'\sim P(Z)}[K_X(X,X')K_Z(Z,Z')],\end{aligned} \tag{19}$$

where $K_X(X,X') = \exp\big(\frac{||X-X'||_2^2}{2\sigma_X^2}\big)$ and $\sigma_X > 0$ is the kernel size. Followed by [45], we choose kernel width as the average of mean values for all samples. Similar to mutual information, HSIC measures the distance between the joint distribution and the product of marginal distributions.

Mutual Information Gradient Estimation (MIGE). The MIGE method [48] directly estimates the gradient of mutual information based on the score estimation for the implicit distribution. The gradient of mutual information between X and Z is defined as:

$$\begin{aligned}\nabla I(X;Z) &= \nabla H(X) + \nabla H(Z) - \nabla H(Z,X) \\ &= -\nabla\mathbb{E}_{P(X)}[\log P(X)] - \nabla\mathbb{E}_{P(Z)}[\log P(Z)] + \nabla\mathbb{E}_{P(Z,X)}[\log P(Z,X)].\end{aligned} \tag{20}$$

where the score function $\nabla \log P(X)$ is estimated by Spectral Stein Gradient Estimator (SSGE) [57].

The performance reported in Table 13 shows that approximating mutual information with HSIC achieves better performance than MINE and MIGE on the CIFAR-10 dataset with ResNet18. HSIC works better because MINE utilizes a neural network to estimate MI, and the performance heavily depends on that network architecture. We cannot guarantee that such a value is close to the true MI. MIGE estimates the gradient of MI based on the approximation of the score function. However, HSIC is the kernel dependence measurement without any approximation and has similar properties as MI.

Table 14
Performance Comparison with different latent representation location.

Method	Location	Network	λ	Natural(%)	PGD^{20}(%)
Ours	$p(y\|\mathbf{x})$	ResNet18	0.1	81.35	55.56
Ours	$p(y\|\mathbf{x})$	ResNet18	0.01	81.71	55.42
Ours	$p(y\|\mathbf{x})$	ResNet18	0.001	81.89	55.09
Ours	$p(\mathbf{z}\|\mathbf{x})$	ResNet18	0.001	81.37	**55.58**
Ours	$p(y\|\mathbf{x})$	WideResNet-34-10	0.001	85.47	56.78
Ours	$p(\mathbf{z}\|\mathbf{x})$	WideResNet-34-10	0.001	**86.12**	**57.52**

A.4. The Effect of Latent Representation Location

Finally, we also compare the performance with different latent representation location. $p(y|x)$ represents putting shared information regularization on the output of softmax layer. $p(\mathbf{z}|\mathbf{x})$ represents putting shared information regularization on the output of layer before the last layer. Table 14 illustrates that the performance of putting regularization on softmax output is similar as that of putting on the layer before last layer on CIFAR-10 with ResNet18. While for WideResNet-34-10, putting regularization on the layer before last layer achieves better natural and robust accuracy than that of the softmax layer.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: 2nd International Conference on Learning Representations, ICLR, Banff, AB, Canada, April 14-16, 2014.

[2] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, May 7-9, 2015.

[3] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018. URL: https://openreview.net/forum?id=rJzIBfZAb.

[4] H. Kannan, A. Kurakin, I. Goodfellow, Adversarial logit pairing, arXiv preprint arXiv:1803.06373 (2018).

[5] G. W. Ding, Y. Sharma, K. Y. C. Lui, R. Huang, MMA training: Direct input space margin maximization through adversarial training, in: 8th International Conference on Learning Representations, ICLR, Addis Ababa, Ethiopia, April 26-30, 2020. URL: https://openreview.net/forum?id=HkeryxBtPB.

[6] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, M. I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, June 9-15, California, USA, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 7472–7482.

[7] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, Q. Gu, Improving adversarial robustness requires revisiting misclassified examples, in: 8th International Conference on Learning Representations, ICLR, Addis Ababa, Ethiopia, April 26-30, 2020.

[8] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, B. Dong, You only propagate once: Accelerating adversarial training via maximal principle, in: 32th Neural Information Processing Systems, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 227–238.

[9] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. P. Dickerson, C. Studer, L. S. Davis, G. Taylor, T. Goldstein, Adversarial training for free!, in: 32th Annual Conference on Neural Information Processing Systems, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019,

pp. 3353–3364.

[10] E. Wong, L. Rice, J. Z. Kolter, Fast is better than free: Revisiting adversarial training, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[11] Y. Balaji, T. Goldstein, J. Hoffman, Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets, arXiv preprint arXiv:1910.08051 (2019).

[12] M. Cheng, Q. Lei, P.-Y. Chen, I. Dhillon, C.-J. Hsieh, Cat: Customized adversarial training for improved robustness, arXiv preprint arXiv:2002.06789 (2020).

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: 26th Annual Conference on Neural Information Processing Systems, NeurIPS 2012, December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114.

[14] S. Laine, T. Aila, Temporal ensembling for semi-supervised learning, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[15] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).

[16] N. Tishby, F. C. N. Pereira, W. Bialek, The information bottleneck method, in: The 37th annual Allerton Conference on Communication, Control, and Computing, 1999.

[17] A. A. Alemi, I. Fischer, J. V. Dillon, K. Murphy, Deep variational information bottleneck, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017.

[18] Q. Hoang, T. Le, D. Phung, Parameterized rate-distortion stochastic encoder, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 4293–4303.

[19] A. Kolchinsky, B. D. Tracey, D. H. Wolpert, Nonlinear information bottleneck, Entropy 21 (2019) 1181.

[20] X. Yu, S. Yu, J. C. Príncipe, Deep deterministic information bottleneck with matrix-based entropy functional, in: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 3160–3164.

[21] F. Alesiani, S. Yu, X. Yu, Gated information bottleneck for generalization in sequential environments, in: 2021 IEEE International Conference on Data Mining (ICDM), IEEE, 2021, pp. 1–10.

[22] M. Federici, A. Dutta, P. Forré, N. Kushman, Z. Akata, Learning robust representations via multi-view information bottleneck, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[23] R. Linsker, Self-organization in a perceptual network, Computer 21 (1988) 105–117.

[24] P. Bachman, R. D. Hjelm, W. Buchwalter, Learning representations by maximizing mutual information across views, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 15509–15519.

[25] X. Ji, J. F. Henriques, A. Vedaldi, Invariant information clustering for unsupervised image classification and segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9865–9874.

[26] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognition 84 (2018) 317–331.

[27] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2013, pp. 387–402.

[28] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.

[29] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 2206–2216.

[30] G. Cohen, G. Sapiro, R. Giryes, Detecting adversarial samples using influence functions and nearest neighbors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14453–14462.

[31] M. Andriushchenko, F. Croce, N. Flammarion, M. Hein, Square attack: a query-efficient black-box adversarial attack via random search, in: European Conference on Computer Vision, Springer, 2020, pp. 484–501.

[32] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, arXiv preprint arXiv:1611.01236 (2016).

[33] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, P. Kohli, Adversarial robustness through local linearization, in: 32th Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 13824–13833.

[34] S.-M. Moosavi-Dezfooli, A. Fawzi, J. Uesato, P. Frossard, Robustness via curvature regularization, and vice versa, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9078–9086.

[35] S. Yang, T. Guo, Y. Wang, C. Xu, Adversarial robustness through disentangled representations, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 3145–3153.

[36] Q. Cai, C. Liu, D. Song, Curriculum adversarial training, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden, ijcai.org, 2018, pp. 3740–3747.

[37] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, M. Kankanhalli, Attacks which do not kill training make adversarial learning stronger, in: International Conference on Machine Learning, PMLR, 2020, pp. 11278–11287.

[38] Y. Dong, Z. Deng, T. Pang, J. Zhu, H. Su, Adversarial distributional training for robust deep learning, in: 33th Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[39] V. Zantedeschi, M.-I. Nicolae, A. Rawat, Efficient defenses against adversarial attacks, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 39–49.

[40] S. Lee, H. Lee, S. Yoon, Adversarial vertex mixup: Toward better adversarially robust generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 272–281.

[41] A. Lamb, V. Verma, J. Kannala, Y. Bengio, Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy, in: Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 95–103.

[42] C. E. Shannon, A mathematical theory of communication, The Bell system technical journal 27 (1948) 379–423.

[43] R. Gilad-Bachrach, A. Navot, N. Tishby, An information theoretic tradeoff between complexity and accuracy, in: Learning Theory and Kernel Machines, Springer, 2003, pp. 595–609.

[44] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, B. Schölkopf, et al., Kernel methods for measuring independence, Journal of Machine Learning Research (2005).

[45] W.-D. K. Ma, J. Lewis, W. B. Kleijn, The hsic bottleneck: Deep learning without back-propagation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 04, 2020, pp. 5085–5092.

[46] D. Greenfeld, U. Shalit, Robust learning with the hilbert-schmidt independence criterion, in: International Conference on Machine Learning, PMLR, 2020, pp. 3759–3768.

[47] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, D. Hjelm, Mutual information neural estimation, in: International Conference on Machine Learning, PMLR, 2018, pp. 531–540.

[48] L. Wen, Y. Zhou, L. He, M. Zhou, Z. Xu, Mutual information gradient estimation for representation learning, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.

[49] H. Kim, Torchattacks: A pytorch repository for adversarial attacks, arXiv preprint arXiv:2010.01950 (2020).

[50] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[51] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images (2009).

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

[53] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016, BMVA Press, 2016.

[54] M. Andriushchenko, F. Croce, N. Flammarion, M. Hein, Square attack: a query-efficient black-box adversarial attack via random search (2020).

[55] S. Sabour, Y. Cao, F. Faghri, D. J. Fleet, Adversarial manipulation of deep representations, in: 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[56] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with hilbert-schmidt norms, in: International conference on algorithmic learning theory, Springer, 2005, pp. 63–77.

[57] J. Shi, S. Sun, J. Zhu, A spectral approach to gradient estimation for implicit distributions, in: Proceedings of the 35th International Conference on Machine Learning, 2018, pp. 4651–4660.