Adversarial Information Bottleneck

Penglong Zhai and Shihua Zhang

Abstract—The information bottleneck (IB) principle has been adopted to explain deep learning in terms of information compression and prediction, which are balanced by a tradeoff hyperparameter. How to optimize the IB principle for better robustness and figure out the effects of compression through the tradeoff hyperparameter are two challenging problems. Previous methods attempted to optimize the IB principle by introducing random noise into learning the representation and achieved the state-of-the-art performance in the nuisance information compression and semantic information extraction. However, their performance on resisting adversarial perturbations is far less impressive. To this end, we propose an adversarial IB (AIB) method without any explicit assumptions about the underlying distribution of the representations, which can be optimized effectively by solving a min–max optimization problem. Numerical experiments on synthetic and real-world datasets demonstrate its effectiveness on learning more invariant representations and mitigating adversarial perturbations compared to several competing IB methods. In addition, we analyze the adversarial robustness of diverse IB methods contrasting with their IB curves and reveal that IB models with the hyperparameter $\beta$ corresponding to the knee point in the IB curve achieve the best tradeoff between compression and prediction and has the best robustness against various attacks.

Index Terms—Adversarial robustness, deep learning, hyperparameter selection, information bottleneck (IB).

I. INTRODUCTION

Deep learning has achieved impressive success in computer vision [1], [2], speech recognition [3], competitive game playing [4], bioinformatics [5], and so on. However, it turns out that deep neural networks (DNNs) are actually quite brittle and particularly vulnerable to adversarial examples. Tiny noises in the input pixels are accumulated layer by layer in DNNs, which destructively destroy the carefully designed and well-trained models. How to learn representations that are insensitive (invariant) to nuisances such as translations, rotations, and occlusions is still a challenging task. From the information-theoretic view, overfitting occurs when nuisance or irrelevant information is memorized by reducing empirical loss, rather than learning the true general relationship between the data $X$ and the labels $Y$ [6], [7]. Therefore, better generalization and robustness depend on whether the model memorizes more semantic or relevant information and compresses as much nuisance information as possible. This information-theoretic insight on deep learning can be formulated as the tradeoff between information compression and prediction. Besides, it is often concerned that DNNs are uninterpretable or not human-explainable [8]. For example, the lack of interpretability is particularly problematic in medical contexts, where safety risks can arise when there is a mismatch between how a model is trained and used. The forms the explanations often taken are what the representations learn from the input data and how the network will respond to small perturbations [9], [10].

In general, DNN receives an input $X$ and processes it through the layer output $Z_i$ ($1 \leq i \leq L$, where $L$ is the number of hidden layers) to the predicted output $\hat{Y}$, which forms a Markov chain and obeys the data-processing inequality (DPI)

$$H(X) \geq \text{MI}(X; Y) \geq \text{MI}(Z_i; Y) \geq \text{MI}(\hat{Y}; Y) \quad (1)$$

where $\text{MI}(\cdot; \cdot)$ denotes the mutual information between two random variables. DPI guarantees that any information the hidden layer $Z_i$ has about $Y$ is extracted from $X$, and the deeper layers have less information for prediction (see Fig. 1). However, with layer getting deeper, relevant information is expected to be retained, while nuisance information is expected to be compressed. The fraction $((\text{MI}(Z_i; Y))/\text{MI}(X; Y))$ quantifies how much of the relevant information is captured by DNN and is expected to be maximized. This is precisely the target of the information bottleneck (IB) principle [11].

The IB principle aims to find a representation bottleneck variable $Z$, by maximizing the prediction, formulated in terms of $\text{MI}(Z; Y)$, given a constraint on the compression, formulated in terms of $\text{MI}(X; Z)$. Formally, this can be formulated into the following constrained optimization problem:

$$\max_{Z} \text{MI}(Z; Y) \quad \text{s.t.} \quad \text{MI}(X; Z) \leq C \quad (2)$$

where $C$ is the predefined compression rate, i.e., the minimal number of bits needed to describe the data. In practice, optimal bottleneck variables are usually not found by solving the above constrained optimization problem, but rather by maximizing the IB Lagrangian [12], [13]

$$\mathcal{L}_\text{IB}(Z) = \text{MI}(Z; Y) - \beta \text{MI}(X; Z) \quad (3)$$

where $\mathcal{L}_\text{IB}(Z)$ is the Lagrangian relaxation of the constrained optimization problem and $\beta$ is a Lagrange multiplier that enforces the constraint $\text{MI}(X; Z) \leq C$, which can also be taken as the hyperparameter that balances the fitting ability (prediction) and representation invariance (compression).
in supervised learning setting. Following recent information-theoretic techniques from [1], Bassily et al. [14] proved the following inequality:

\[
P(|\text{err}_{\text{test}} - \text{err}_{\text{train}}| > \epsilon) < O\left(\frac{\text{MI}(X; Z)}{n\epsilon^2}\right) \tag{4}
\]

where \(\text{err}_{\text{test}}\) and \(\text{err}_{\text{train}}\) are the test error and the training error, respectively, \(n\) is the training size, and \(\epsilon > 0\) is a positive real number. The intuition behind this inequality is that the more a learning algorithm uses bits of the training set, and there is potentially more overfitting risk. Thus, minimizing \(\text{MI}(X; Z)\) will have the potential to avoid overfitting. Clearly, optimizing the IB Lagrangian not only minimizes the nuisance information \(\text{MI}(X; Z)\) but also maximizes the prediction ability. However, the fitting ability and representation invariance cannot be optimized simultaneously for a specific DNN model. This can be visualized on the IB curve [15], [16], which characterizes the set of bottleneck variables that achieve maximal \(\text{MI}(Z; Y)\) for a given \(\text{MI}(X; Z)\) by maximizing the IB Lagrangian given various \(\beta\). In particular, visualizing the DNNs in terms of information theory is capable of helping researchers to detect erroneous reasoning in classification problems.

The IB methods have been found useful in a wide variety of learning applications (e.g., word clustering [17] and image clustering [18]). In particular, the IB principle has been employed to interpret deep learning as a successive information extraction process [11]. In particular, IB has been used to analyze DNNs by computing mutual information between the hidden layers and data or labels. This has attracted tremendous attention recently as a tool to gain insights into the learning dynamics and generalization ability of DNNs [10], [19], [20]. Besides, recent studies have demonstrated the effectiveness on improving generalization ability by optimizing the IB Lagrange empirically [21]–[25]. For example, Alemi et al. [21] presented a variational approximation to IB (VIB), which is an adaption of variational autoencoder (VAE) [22] to a prediction task. Numerical experiments showed that VIB had better generalization performance compared with several competing regularization methods, such as dropout. Subsequently, Chalk et al. [23] improved VIB by introducing different surrogate marginal distributions (e.g., Student’s \(t\)-distribution) on the intermediate representation layer. Achille and Soatto [24] approximated the IB Lagrangian by introducing an information dropout layer, which allows the network to selectively introduce multiplicative noise into the layer activation and thus to control the flow of information. Kolchinsky et al. [26] introduced a novel nonparametric upper bound for mutual information by injecting noise to the representation layer, which achieved the best tradeoff between compression and prediction. These existing methods optimized the IB Lagrangian by introducing random noise into learning the representation and achieved the state-of-the-art performance in terms of nuisance information compression and relevant information extraction. However, their performance on resisting adversarial perturbations is far less impressive. We attribute this counterintuitive result to the noise introduced in the IB methods. In addition, how to trade off the information compression and prediction for better adversarial robustness is a challenging issue and has not been explored so far.

We should note that the IB principle is not the only tool to encourage the representation invariance (proper compression) and adversarial robustness. For example, Achille and Soatto [27] showed that invariance to nuisance is equivalent to the information minimality of the learned representation and proposed to bound the model complexity using the information in the weights. More recently, Yu et al. [28] proposed an information-theoretic measure that maximizes the coding rate difference between the whole data and the sum of each individual class. Besides, bulk of noninformation-theoretical principles has been proposed for reducing the sensitivity of DNN to small perturbations. Among them, double backpropagation is an old idea originally introduced by Drucker and Le Cun [29], which trains the DNNs by minimizing not just the “energy” of the network but the rate of change of that energy with respect to the input data. Xu and Mannor [30] proved that the Lipschitz constant of DNNs controls the difference between the training loss and generalization performance and proposed to minimize it to decrease the model sensitivity to adversarial examples. In addition, Rifai et al. [31] proposed to penalize the representation sensitivity, measured as the squared norm of its Jacobian with respect to the input data. The Jacobian constraint has been utilized to learn contractive representations in supervised and unsupervised tasks [31], [32]. However, these hard-constraint methods lack of interpretability and meaningful measure such as mutual information in IB to demonstrate their learning effectiveness.

In this article, we propose adversarial IB (AIB) to optimize the IB Lagrangian by introducing an adversarial regularization term to approximate the information compression term \(\text{MI}(X; Z)\). AIB does not impose any explicit assumptions to the distribution of the hidden representation and can be optimized effectively by solving a min–max optimization problem. Numerical results demonstrate its effectiveness on the nuisance information compression and adversarial robustness. In addition, we visualize the IB curves of several competing IB methods by varying \(\beta\) on synthetic and real-world datasets and reveal that all the IB curves present two distinct phases.
By analyzing the adversarial robustness contrasting with the IB curves, we empirically claim that the model trained with the tradeoff hyperparameter corresponding to the knee point in the IB curves presents the best robustness.

II. RELATED WORK

A. IB Principle

Tishby et al. [33] first proposed the IB concept and provided a tabular method based on the Blahut–Arimoto (BA) algorithm [15], [34] to numerically solve the IB Lagrangian (3) for the optimal encoder distribution \( P(Z|X) \), given the tradeoff hyperparameter \( \beta \) and the cardinality of the representation. However, the BA algorithm can only be used in two special cases: the first is where \( X \) and \( Y \) are discrete-valued with a small number of possible states and the second is when \( X \) and \( Y \) are continuous-valued and jointly Gaussian [35]. In these two cases, the conditional probability can be computed explicitly during the optimization process. However, this is infeasible in the big data era due to the high dimensionality of data.

More recently, Tishby and Zaslavsky [11] interpreted deep learning as a successive information-extraction process quantified by the IB principle and attempted to open the black box of deep learning by visualizing the behavior of the compression and prediction terms [10]. This has motivated many studies to apply the IB principle to high-dimensional and complex data in a variety of scenarios such as improving the robustness against adversarial attacks [36], learning invariant and disentangled representations [27], and interpreting the learning dynamics of DNNs [19], [20], [37]. In this article, we mainly focus on the optimization of the IB Lagrangian (3) as well as the effect of the tradeoff between information compression and prediction. Previous studies have attempted to solve this problem by imposing some assumptions on the data distributions and introducing random noise into the hidden representations.

1) Variational Approximation to IB: Alemi et al. [21] introduced VIB inspired by VAE [22] to prediction tasks. By assuming \( Z \sim \mathcal{N}(\mu, \sigma^2) \), they bound MI(\( Z; X \)) in the following equality:

\[
\operatorname{MI}(X; Z) \geq E_{X \sim Q(X)} \operatorname{KL}(P_\theta(Z|X)||P(Z)).
\]

They further replace the prediction term \( \operatorname{MI}(Z; Y) \) by the cross-entropy loss. Subsequently, Chalk et al. [23] improved VIB by introducing different surrogate marginal distributions (e.g., Student’s \( t \)-distributions) on the intermediate representation layer. Achille and Soatto [24] approximated the IB Lagrangian by introducing an information dropout layer, which allows the network to selectively introduce multiplicative noise in the layer activation and thus to control the flow of information.

2) Nonlinear IB (NIB): Kolchinsky et al. [26] introduced a novel nonparametric upper bound for mutual information by injecting noise to the representation layer and achieved the best tradeoff between the information compression and prediction. In particular, NIB introduced noise \( \tilde{Z} = Z + n(\sim \mathcal{N}(0,1)) \) to approximate \( \operatorname{MI}(X; Z) \), i.e.,

\[
\operatorname{MI}(X; Z) \approx \operatorname{MI}(X; \tilde{Z}) \leq E_i \ln E_j e^{-\operatorname{KL}(p_i||p_j)}.
\]

As a result, the learned intermediate representations form geometrically dense clusters. However, they failed to explore whether NIB can improve the performance on adversarial robustness.

The above IB methods introduce random noise to the intermediate representation to reduce its dependence on the input data and mitigate adversarial effects [25], [32], [38]. However, these methods in fact failed to explain the role of the IB principle in their design. In addition, the selection of \( \beta \) is critical to learn a meaningful representation since it controls the information compression and realize representation invariance. However, previous studies have set \( \beta \) in an empirical range, which do not well capture the relationship between compression and prediction. Chechik et al. [35] noted the presence of trivial solution when \( \beta \geq 1 \), in which \( P(Z|X) = P(Z) \) becomes the global minimum of the IB Lagrangian. Wu et al. [37] introduced the concept of IB-learnability and showed that the IB Lagrangian will undergo a phase transition from the inability to learn to the ability to learn when varying \( \beta \). Furthermore, they theoretically and experimentally provided the range of \( \beta \) to avoid the failure of IB Lagrangian optimization. However, the effect of \( \beta \) on learning invariant representation has still not been explored.

B. Mutual Information Estimation

Estimating the mutual information between the input data to a DNN and its hidden layers has long been a topic of research, with applications to representation learning [39] and deep learning [10], [19], [20], [37]. For example, Shwartz-Ziv and Tishby [10] empirically suggested that trajectories in the information plane or learning processes of DNNs appeared to consist of two distinct phases: an initial “fitting” phase where mutual information between the hidden layers and the input decreases. They claimed that this compression phase is responsible for the excellent generalization performance of DNNs. Saxe et al. [20] argued that the “compression” phase arises primarily due to the double-saturating tanh activation function utilized, which will disappear when replaced with the ReLU activation. In addition, Goldfeld et al. [19] observed that compression can occur in VIB and revealed that compression was driven by progressive geometric clustering of the representations of samples from the same class.

However, despite the strengths of mutual information and effectiveness of the above IB methods, mutual information is often impossible to compute analytically and hard to estimate from high-dimensional samples. Most previous information-theoretic studies of deep learning [10], [20] approximate the mutual information by discretizing the outputs of neurons (i.e., the “binning” operation). However, in practice, DNN does not operate on the binned variables, but on the continuous ones. Moreover, there are many possible binning strategies, which yield different discrete random variables, and different mutual information with respect to the input. For example, Saxe et al. [20] linked the compression phase occurred in [10] to saturation of neurons, which usually happens in layers with...
double-saturating activations (e.g., sigmoid) and could lead to the failure of binning strategy.

A variety of approaches for entropy, and thereby mutual information estimation, have been developed over the years, including $k$-nearest neighbors (KNN) techniques [40] and kernel density estimation (KDE) techniques [41], [42]. Wickström et al. [43] utilized a kernel tensor-based estimator of Renyi’s entropy and provided the first comprehensive information plane analysis of contemporary large-scale DNNs and CNNs. Recently, trainable neural estimators [44] have also been utilized to estimate mutual information and the authors attempted to apply this technique to optimize the IB principle. However, their focus is mainly on estimating mutual information and there are no implementation details and result analysis in their work. In this article, we use the KDE technique [41] to estimate the mutual information between input, labels, and hidden layers for plotting the IB curves in numerical experiments. It is fast and robust even for large dimensionality.

C. Adversarial Attacks

Since the original work in [13], many different types of adversarial attacks have been proposed to fool a trained DNN through introducing barely visible perturbations upon input data. Several state-of-the-art white-box attacks will be investigated in this work and are briefly introduced as follows.

1) FGS Attack: Fast gradient sign (FGS) method [45] hypothesizes that DNNs are vulnerable to adversarial perturbations because of their linear nature. It finds the adversarial examples by projecting the sample $(x)$ along the direction of gradients with respect to the input. In particular, $x$ is manipulated by adding or subtracting a small perturbation $\epsilon$ to each pixel. Whether adding or subtracting $\epsilon$ depends on whether the sign of the gradient for a pixel is positive or negative. Adding errors in the direction of the gradient means that the image is misclassified.

$\mathbf{x}_{\text{FGS}} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y))$

where $\nabla_x L$ is the gradient of the loss function with respect to the original input pixel vector $x$, $y$ is the true label vector for $x$, and $\theta$ is the model hyperparameter vector. If the perturbation is small, these adversarial examples are indistinguishable from normal examples to human, but the network performs significantly worse on them.

2) TGS Attack: Targeted gradient sign (TGS) attack was designed based on the FGS method. It attempts to encourage the model to misclassify samples in a specific way

$\mathbf{x}_{\text{TGS}} = x + \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y_{\text{target}}))$

where $y_{\text{target}}$ encodes an alternative set of labels we would like the model to predict instead. TGS can also be performed iteratively.

3) DeepFool Attack: DeepFool attack [46] is an iterative attack method, which finds the minimal perturbation to cross the decision boundary based on the linearization of the classifier at each iteration. Given a sample $x_n$ belonging to the class $t$, the perturbation is formulated as

$$\min_{\delta_n} \|\delta_n\|_p \quad \text{s.t.} \max_{j \neq t} \left\{ g_j(x_n + \delta_n) - g_j(x_n) \right\} \geq 0$$

where $\delta_n$ is taken as the perturbation and $\|\cdot\|_p$ can be any norm specified by the user. The sum of all the perturbations from all the samples can be taken as the robustness performance of the given model. Note that $\delta_n$ can be taken as the distance from the sample $x_n$ to the classification boundary in a two-class linear classifier case. In nonlinear and multilabel cases, the minimum attack distance will be iterated along the vertical direction of the nearest classification boundary.

III. ADVERSARIAL IB

The difficulty of optimizing the IB Lagrangian lies in the compression term $\text{MI}(X; Z)$ since the prediction term $\text{MI}(Z; Y)$ follows [21], [26]:

$$\text{MI}(Z; Y) \leq H(Y) + E_{P(Z;Y)} \log P(Y|Z)$$

where the second expectation term is equivalent to the usual cross-entropy loss in supervised learning, which can be optimized effectively in the settings of machine learning.

To optimize the compression term $\text{MI}(X; Z)$, we recall that mutual information can be formulated as the Kullback–Leibler (KL) divergence between the joint $P(X; Z)$ and the product of the marginals $P(X)P(Z)$. Recent studies utilized a dual formula to cast the estimation of $f$-divergences [44], [47], [48] (including the KL divergence) as part of an adversarial game between competing DNNs [49]. Formally, the mutual information admits the following equality:

$$\text{MI}(X; Z) = \max_{T: \Omega \rightarrow R} E_{P_{\text{aug}}} T(x, z) - \log E_{P_{\text{aug}}} e^{T(x, z)}$$

where the maximum is taken over all functions $T$ such that the two expectations are finite. In fact, any divergence or distance function, including the KL divergence, can be formulated as an upper bound of a certain formula [47], [48].

A. Optimization

In this article, we use the neural network with an encoder–decoder architecture, where the encoder $E$ outputs the representation $Z$ by processing the input data $X$ and the decoder $D$ outputs the predicted labels based on $Z$. In addition, the encoder and decoder are parameterized by $\theta$ and $\phi$, respectively. Furthermore, we parameterize the function $T$ in (6) by a neural network parameterized by $\psi$ [44]. Therefore, the minimization of the compression term $\text{MI}(X; Z)$ can be formulated as the following min–max problem:

$$\min_{\theta} \max_{\psi} E_{P_{\text{aug}}} T_\psi(x, E_\theta(x)) - \log E_{P_{\text{aug}}} e^{T_\psi(x, E_\theta(x))}$$

where the inner maximization tends to acquire accurate estimates of mutual information by optimizing $T_\psi$ and the outer minimization on $E_\theta$ is optimized to compress the information flow. We guess that the gap in (6) can well approximate 0 because of the powerful expressive ability of neural network. Optimizing (6) to converge after every gradient descent step.
Fig. 2. Illustration of AIB.

Fig. 3. (a) Illustration of the IB curve of AIB on MNIST. (b) and (c) Plots of MI(X; Z) and MI(Z; Y) versus β, respectively. Each blue circle corresponds to a fully converged model starting with independent initialization. The red arrows indicate the knee point determined in the IB curve in (a).

Fig. 3. (a) Illustration of the IB curve of AIB on MNIST. (b) and (c) Plots of MI(X; Z) and MI(Z; Y) versus β, respectively. Each blue circle corresponds to a fully converged model starting with independent initialization. The red arrows indicate the knee point determined in the IB curve in (a).

guarantees us to stay on the desired manifold. This is an expensive procedure. Moreover, it may result in parameters being far away from the ones obtained after the main gradient update.

We use two approximations to make the algorithm more efficient. First, we only do one step of descent on the function (6). Second, instead of optimizing the mutual information between the whole data X and representation Z, one can find a subset S of the most informative pixels and perform the update on them, denoted by X_S. This procedure is meaningful since

\[ \text{MI}(X_S; Z) \leq \text{MI}(X; Z). \] (7)

The two terms are equal if and only if S is the whole set of the pixels. In particular, we determine X_S of pixels with large gradients, i.e., the gradients of Z with respect to the inputs since these selected pixels are the most informative ones about Z. The structure and algorithm of AIB are summarized in Fig. 2 and Algorithm 1, respectively.

We name our proposed method as AIB due to its “adversarial” characteristic by solving a min–max optimization problem, which is very different from adversarial training by feeding the adversarial examples into the training process [45]. In particular, the adversarial training required a computationally expensive inner loop in order to evaluate the adversarial perturbations, in which the cost becomes prohibitive with growing model complexity and input dimensionality.

Algorithm 1 AIB

Require: Datasets \( D = \{(x_i, y_i), i = 1, 2, \ldots, N\} \); the number \( k \) of steps to update \( T_\phi \); the number \( p \) of pixels sampled from the input data; the tradeoff hyperparameter \( \beta \).

Repeat:
1. For \( k \) steps do
2. Sample two minibatch of \( m \) samples \( \{(x_i, y_i), \hat{x}_i, \hat{y}_i\}_{i=1}^m \).
3. Feed one minibatch samples into the encoder \( E_\theta \) and obtain the representation \( z_i = \{E_\theta(x_i)\} \).
4. Select \( p \) pixels with the most significant gradients with respect to \( x_i \).
5. Update \( T_\psi \) by ascending its stochastic gradient:
6. End for
7. Update the encoder \( E_\theta \) and decoder \( D_\phi \) by descending their respective stochastic gradient:

\[
\begin{align*}
\nabla_\psi \frac{1}{m} \sum_{i=1}^{m} T_\psi(x_i, z_i) &- \log \frac{1}{m} \sum_{i=1}^{m} \exp(T_\psi(x_i)) \\
\n&+ \beta \left[ \frac{1}{m} \sum_{i=1}^{m} T_\psi(x_i, E_\theta(x_i)) \\
&- \log \frac{1}{m} \sum_{i=1}^{m} \exp(T_\psi(x_i, E_\theta(x_i))) \right],
\end{align*}
\]

Until convergence.

B. Hyperparameter Selection

Another challenge in the optimization of the IB Lagrangian is that there is a lack of understanding about the relationship between \( \beta \) and adversarial robustness. In particular, the question under consideration is whether and how compression promotes better robustness by varying \( \beta \)? Furthermore, if so, how to select \( \beta \) to achieve the best robustness?

Here, we provide a simple way to select \( \beta \) based on an observation about the IB curve (see Fig. 3). As \( \beta \) decreases, the IB curve presents two distinct phases: 1) \( \text{MI}(X; Z) \) and \( \text{MI}(Z; Y) \) increase simultaneously with the decrease of \( \beta \) and 2) once \( \beta \) exceeds a critical value, \( \text{MI}(Z; Y) \) increases slowly or stay the same, while \( \text{MI}(X; Z) \) keeps increasing. Thus, the difference in the changing rate of the two values leads to a sharp knee and turning point in the IB curve. We intuitively relate this knee point to the optimal hyperparameter, which characterizes the best robustness. Numerical experiments clearly confirm this observation. The step number \( k \) is always 1 since the experiment shows that this hyperparameter
IV. EXPERIMENTAL RESULTS

We first demonstrate the performance of AIB and the effectiveness of hyperparameter selection through exploring the IB curves. Next, we show that AIB distinctly improves the adversarial robustness compared to other methods with three different adversarial attacks.

AIB is compared with VIB [21] and NIB [26] on a synthetic dataset [10] and two common classification datasets Modified National Institute of Standards and Technology database (MNIST) [50] and FashionMNIST [51] by varying hyperparameter $\beta$. We adopt the typical DNNs trained with the cross-entropy loss to show its representation ability and robustness (denoted as “Normal”). We also employ the popular “dropout” method (denoted as Dropout) with dropout rate as its regularization hyperparameter for evaluation. We still call the tradeoff curve relating to “dropout” as the IB curve for convenience.

A. Information Compression and Extraction

1) Synthetic Experiment: First, we utilize a synthetic dataset as suggested in [10] and find the bottleneck variable through the BA algorithm [15], which can be taken as the golden standard solution. The synthetic dataset is constructed as binary decision rules that are invariant under $O(3)$ rotations of the sphere, with 12 binary inputs that represent 12 uniformly distributed points on a 2-D sphere. Therefore, there are 4096 different patterns in total and the classification rule is designed such that $p(y = 0) = p(y = 1) \simeq 0.5$ and the mutual information $\text{MI}(X; Y) \simeq 0.99$ bits. With the BA algorithm, the minimization of (3) is performed by the converging alternating iterations as follows:

\[
\begin{align*}
    p_k(z|x) &= \frac{p_k(z)}{S_k(x, \beta)} \exp(-\beta d(x, z)) \\
    p_{k+1}(z) &= \sum_x p(x) p_k(z|x) \\
    p_{k+1}(y|z) &= \sum_x p(y|x) p_k(x|z)
\end{align*}
\]

where $S_k(x, \beta)$ is the normalization function, $z$ is one of the ten discrete states of $Z$, and $k$ denotes the iteration step. Following the above formula, the optimal IB bound can be found by varying the tradeoff hyperparameter $\beta$. The region above the IB bound is nonachievable and the approximation closeness to the theoretical bound demonstrates the effectiveness of information extraction [see Fig. 4(A)].

A fully connected feedforward neural network (12-10-10-2-10-2) with no other architecture constraints is utilized to evaluate all the methods. The ReLU function is adopted for activation, and the softmax function is employed in the final layer. The networks were trained using the cross-entropy loss function and the optimizer Adam with a learning rate of 0.0002. For easier estimation of mutual information, we set the outputs of the two-node hidden layer as the bottleneck variable $Z$. The IB curves are plotted by calculating the mutual information between the input data, labels, and the bottleneck layer $Z$. We note that the network capacity is relatively small, so the IB curves explored by these methods are not very close to the optimal solution.

2) Real-World Experiment: Two common real-world classification datasets (MNIST and FashionMNIST) are utilized...
to evaluate the performance of AIB. Each image of MNIST and FashionMNIST consists of $28 \times 28$ pixels (784 pixels in total), i.e., $X \in \mathbb{R}^{784}$, and is classified into one of ten classes corresponding to the digit or clothing identity ($Y \in \{0, \ldots, 9\}$).

Both the two datasets contain 60,000 training images and 10,000 testing images. MNIST is a relatively simple dataset for modern machine learning and FashionMNIST is a more complex dataset. Note that the BA algorithm is not applicable here since the dimension of these two datasets is pretty high and the conditional probability is calculable. However, the IB curves are still bounded according to the DPI (1) and definition of mutual information

$$H(Y) \geq MI(Z; Y) \leq MI(X; Z).$$

Here, we use the similar setting with the synthetic experiment except for the network architecture. We apply a simple fully connected network and a convolutional neural network onto MNIST and FashionMNIST, respectively, since the latter is a more difficult task. The fully connected layers have units 784-128-128-10-128-10 and the ten-unit layer is set as the bottleneck layer. The simple convolutional neural network consists of $3 \times 3 \times 32$ and $3 \times 3 \times 64$ convolutional layers followed by a $2 \times 2$ max pooling and two fully connected layers (64-128). The 64-unit layer is set as the bottleneck layer. In both the experiments, the training images are divided into training and validation set at a ratio of 4:1. We pick the model that performs the best on clean accuracy of the validation set.

We present the IB curves of all the methods on the synthetic and real-world datasets (see Fig. 4). Note that all the IB curves appear smooth and convex, which start from the point (0, 0) corresponding to a large $\beta$ and stop at a point corresponding to $\beta = 0$, respectively. We come to the following conclusions.

1) On the synthetic dataset, AIB performs the best compared to Dropout, VIB, and NIB, and its IB curve is mostly close to the bound determined by the BA algorithm.

2) On all the datasets, the IB methods (AIB, NIB, and VIB) perform better than Dropout in terms of the nuisance information compression. They achieve better prediction values at the same level of compression. In particular, the performance of AIB is better (synthetic and MNIST) or comparable FashionMNIST) compared with that of NIB and VIB.

3) The curves of the IB methods present a phase transition phenomenon to some extent: 1) the mutual information between the representation, the input data $MI(X; Z)$, and labels $MI(Z; Y)$ increases simultaneously and 2) once $\beta$ is less than a critical value, $MI(Z; Y)$ increases slowly or stay the same level, while $MI(X; Z)$ keeps increasing.

3) Compression Promotes Geometric Representation: First, we consider the clustering performance of all the methods on the bottleneck layer. We visualize the 2-D projections for them on MNIST (see Fig. 5). Training with the IB Lagrangian or cross entropy enables different digits to fall into well-separated clusters, in which different architectures lead to clusters with diverse shapes. In particular, the clusters of AIB tend to be line-shaped, indicating that it has the largest decision margin. In addition, the geometric representation of AIB is significantly better than the regular Dropout method. This is consistent with the information compression results (see Fig. 4) since the bits of compact clusters are smaller.

Note that Fig. 5 (middle) corresponds to $\beta$ on the knee point of the IB curve (see Section IV-B). Then, their respective geometric representations get more compact (middle) compared to others (left and right). This observation gives us a promising way for selecting hyperparameter, which provides the best invariance and robustness against adversarial perturbations.

B. Adversarial Robustness and Defense

In this section, we evaluate the robustness of all the models against the FGS, TGS, and Deepfool attacks on MNIST, FashionMNIST, as well as CIFAR-10 [52]. The CIFAR-10 dataset is an established computer vision dataset used for object recognition and consists of 60,000 $32 \times 32$ color images belonging to ten object classes. All the images are divided into training, validation, and test sets at a ratio of 4:1:1. We use a complex convolutional neural network on CIFAR-10. All the methods are trained using the optimizer Adam with learning rate 0.0002 and batch size 256. The iterations are stopped if they do not improve the accuracy on validation set within 20 epochs.
Table I reports the results of all the adversarial robustness to the FGS and TGS attacks on MNIST, FashionMNIST, and CIFAR-10 at a variety of perturbation strengths. The performance of VIB is comparable to the Dropout method in a range of small perturbations. Although the performance of NIB on MNIST is significantly better than VIB and Dropout, it is far less impressive on more complex datasets (FashionMNIST and CIFAR-10). However, the performance of AIB is the best across all the datasets and attacks with different perturbations.

For the FGS and TGS attacks, we test all the methods against the adversarial examples generated for each strategy with different perturbation ranging from 0.05 to 0.3. In particular, for the TGS attack, we pick the targets via increasing the original labels $y$ by 1 (modulo 10). For the Deepfool attack, we compute the minimum perturbation upon all of the test samples and take $||\delta||_2 = (\sum \delta^2)^{1/2}$ as the evaluation metric to measure the robustness of DNN, where a greater value of $||\delta||_2$ normally indicates that a DNN possesses higher robustness against potential adversarial attacks. We use cleverhans library [53] to generate all the adversarial examples.

We first select the hyperparameter in a reasonable range in which the method has a pretty high clean accuracy. This is critical since the model with bad clean accuracy may present better robustness [see Fig. 6(C) and (D)]. We show the changing trend of $\text{MI}(Z; Y)$, $\text{MI}(X; Z)$, accuracy on the adversarial examples generated by FGS attack (perturbation $\epsilon = 0.1$), and clean accuracy in terms of $\beta$ (see Fig. 6). First, as the tradeoff $\beta$ increases, the clean accuracy stays almost the same and adversarial accuracy increases distinctly until it reaches a critical point, which precisely corresponds to $\beta$ of the knee point in the IB curves. Second, once the tradeoff $\beta$ exceeds the knee point, the clean accuracy and adversarial accuracy decrease simultaneously. As a result, we select the $\beta$ ranging from 0 to that corresponding to the knee point in the IB curve.

Table I reports the results of all the adversarial robustness to the FGS and TGS attacks on MNIST, FashionMNIST, and CIFAR-10 at a variety of perturbation strengths $\epsilon$. The performance of VIB is comparable to the Dropout method in a range of small perturbations. Although the performance of NIB on MNIST is significantly better than VIB and Dropout, it is far less impressive on more complex datasets (FashionMNIST and CIFAR-10). However, the performance of AIB is the best across all the datasets and attacks with different perturbations.

Fig. 6. (a) and (b) $\text{MI}(Z; Y)$ and $\text{MI}(X; Z)$ evaluated on MNIST of the IB methods trained with their corresponding tradeoff hyperparameter $\beta$. (c) Accuracy on MNIST generated by FGS attack of the IB methods in terms of the tradeoff hyperparameter $\beta$ (perturbation $\epsilon = 0.1$). (d) Clean accuracy on MNIST. Each blue circle corresponds to a fully converged model starting with independent initialization. The $x$-axis is scaled differently for better visualization. The red arrows indicate the $y$-axis value trained with the hyperparameter corresponding to the knee point of the IB curve.

It should be noted that only improving robustness against FGS and TGS attacks does not necessarily mean better accuracy against other attacks. Typically, the Deepfool attack...


with $L_2$ norm could reach 100% success rate against any defense methods. Thus, the total perturbations required to fool a network give more message about its robustness in general, and the large perturbations mean better robustness.

Table II presents the performance of all the methods against the DeepFool attack on MNIST, FashionMNIST, and CIFAR-10. AIB consistently performs the best in all the datasets, while VIB and NIB only achieve similar performance with the Dropout method. In addition, Fig. 7 shows the adversarial examples generated by the DeepFool attack applying to all the models on different digits of MNIST. As a result, the adversarial examples generated on AIB are the most unrecognizable ones, indicating that the examples need to move farther to fool it.

Finally, we explore the impact of the number $p$ of sampled pixels on the robustness of model against the DeepFool attack. As mentioned in Section III, we sample a subset of pixels with large gradients to implement alternative descents, which aims to minimize the dependence between $Z$ and the most informative part of $X$. This procedure can speed up the training and improve the robustness of the model. We can see that compared to implementing gradient descents with respect to all the pixels, an appropriate ratio (about 0.6–0.7) can boost the performance (see Fig. 8).

### V. Conclusion

In this article, we propose AIB to optimize the IB Lagrangian by introducing an adversarial regularization term to approximate the information compression term $MI(X; Z)$. AIB learns compact or even line-shaped representation and compresses more nuisance information given the same prediction accuracy. Experimental results demonstrate that AIB performs the best in resisting adversarial perturbations compared to VIB, NIB, and even others.

We further conduct a detailed and in-depth exploration about the effects of $\beta$ on learning invariant representations and mitigating adversarial perturbations. As a result, we empirically demonstrate that the hyperparameter corresponding to the knee point of the IB curve performs the best in geometric representation and adversarial robustness. We emphasize that this can help to understand the effect of compression on model robustness and hyperparameter selection.

In the future, it will be an interesting topic to learn invariant network weights and gradients with the proposed adversarial compression term, which does not require an analytical form about their distributions. Furthermore, we claim that the information-theoretical perspective on the selection of tradeoff $\beta$ is not only applicable to the IB models, but other defense models with a hyperparameter specifying the penalty strength.

### Acknowledgment

The authors would like to thank Chihao Zhang, Kuo Gai, and Rui Zhang for their helpful discussion.

### References

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2261–2269.

[3] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” IEEE/ACM Trans. Audio, Speech, Language Process., vol. 22, no. 10, pp. 1533–1545, Oct. 2014.

[4] D. Silver et al., “Mastering chess and shogi by self-play with a general reinforcement learning algorithm,” 2017, arXiv:1712.01815.

[5] Z. Cao and S. Zhang, “Simple tricks of convolutional neural network architectures improve DNA-protein binding prediction,” Bioinformatics, vol. 35, no. 11, pp. 1837–1843, 2019.

[6] D. Krueger et al., “Deep nets don’t learn via memorization,” in Proc. ICLR, 2017, pp. 1–4.

[7] D. Bashir, G. D. Montanez, S. Sehra, P. S. Segura, and J. Lauw, “An information-theoretic perspective on overfitting and underfitting,” 2020, arXiv:2010.06076.

[8] Z. Zhang and S. Zhang, “Towards understanding residual and dilated dense neural networks via convolutional sparse coding,” Nat. Sci. Rev., vol. 8, no. 3, p. 159, 2020.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why should I trust you?’ Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2016, pp. 1135–1144.

[10] R. Shwartz-Ziv and N. Tishby, “Opening the black box of deep neural networks via information,” 2017, arXiv:1703.00810.

[11] N. Tishby and N. Zaslavsky, “Deep learning and the information bottleneck principle,” in Proc. IEEE Inf. Theory Workshop (ITW), Apr. 2015, pp. 1–5.
