Distilling Robust and Non-Robust Features in Adversarial Examples by Information Bottleneck

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adversarial examples, generated by carefully crafted perturbation, have attracted considerable attention in research fields. Recent works have argued that the existence of the robust and non-robust features is a primary cause of the adversarial examples, and investigated their internal interactions in the feature space. In this paper, we propose a way of explicitly distilling feature representation into the robust and non-robust features, using Information Bottleneck. Specifically, we inject noise variation to each feature unit and evaluate the information flow in the feature representation to dichotomize feature units either robust or non-robust, based on the noise variation magnitude. Through comprehensive experiments, we demonstrate that the distilled features are highly correlated with adversarial prediction, and they have human-perceptible semantic information by themselves. Furthermore, we present an attack mechanism intensifying the gradient of non-robust features that is directly related to the model prediction, and validate its effectiveness of breaking model robustness.

1 Introduction

Deep neural networks (DNNs) have achieved remarkable performances in a wide variety of machine learning tasks. Despite the breakthrough outcomes, DNNs are easily fooled from adversarial attacks, with crafted perturbations [1, 2, 3, 4, 5, 6, 7]. These perturbations are imperceptible to human eyes, but simply adding them to clean images (*i.e.,* adversarial examples) can effectively deceive classifiers. Such a vulnerability affects security problems [8, 9, 10, 11], bringing in weak reliability of DNNs.

Previous works have broadly investigated the reason for the widespread of such adversarial examples. Goodfellow *et al.* [2] have argued that adversarial vulnerability is induced from excessive linearity nature of DNNs in high-dimensional spaces. Several works [1, 12] have regarded the primary cause of the adversarial examples as statistical variation with aberrations in data manifold. Schmidt *et al.* [13] have suggested that the pervasiveness of the examples should not be considered as a drawback of training methods for DNNs, since available dataset may not be large enough to train them robustly.

In recent years, Tsipras *et al.* [14] have suggested an intriguing analysis of that the disagreement between standard and adversarial accuracy stems from differently trained feature representation. In this literature, Ilyas *et al.* [15] further demonstrated the adversarial examples are inevitable results of standard supervised training and arisen from well-generalized features in the dataset. They have analyzed the adversarial examples are originated from brittle and unintelligible features (*i.e.,* non-robust features) that are arbitrarily manipulated with the imperceptible noise, and shown that the robust features still can provide precise accuracy even in the existence of adversarial perturbation. They have argued that the non-robust features cannot show reliable accuracy in the adversarial setting and could provoke incomprehensible properties to humans.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Nonetheless, the underlying reason for the existence and pervasiveness of adversarial examples cannot derive common consensus in the research field and still remains unclear [16]. To clarify where the adversarial brittleness truly comes from, we need to figure out how the robust and non-robust features in data manifold subtly manipulate feature representation and fool model prediction, by directly handling them in the feature space. To address it, we propose a way to precisely distill intermediate features into the robust and non-robust features by employing Information Bottleneck (IB) [17, 18, 19]. In the sense that semantic information is included in the units of intermediate feature representation [20, 21, 22, 23], we utilize the bottleneck to regulate the information flow in the feature space by explicitly adding noise to the feature units. Then, we estimate how each feature unit contaminated with the noise affects model prediction with assigned information. Based on the prediction sensitivity of the noise intervention, we assort the feature units either robust or brittle, and disentangle the feature representation into robust or non-robust features, respectively.

Through extensive analysis on the distilled features, we corroborate that the pervasiveness of the adversarial brittleness is derived from the non-robust features, and they have high correlation with the adversarial prediction. In addition, in order to understand the semantic information of distilled features, we directly visualize them in the feature space and provide their visual interpretation. Consequently, we reveal that both of the robust and non-robust features indeed have semantic information in terms of human-perception by themselves. Based on our observation, we theoretically describe the negative impact of the non-robust features for the model prediction, and introduce an approach of amplifying the gradients of non-robust features to break the model prediction.

In this paper, our contributions can be summarized into three-fold as follows:

- We propose a novel way to explicitly distill intermediate features into the robust and non-robust features using Information Bottleneck, and interpret the disentangled features in terms of human-perception by directly visualizing them in the feature space.
- By analyzing how the distilled features affect the intermediate feature representation under adversarial perturbation, we demonstrate that the non-robust features are highly correlated with the adversarial prediction.
- We present an attack mechanism manipulating the non-robust features by strengthening their gradients, and validate its effectiveness of breaking model prediction.

2 Distilling Robust and Non-robust Features in Intermediate Feature Space

Problem Setup and Notations. Let X denote clean images and Y denote (one-hot encoded) target labels corresponding to the clean images. Then, adversarial examples X_{adv} can be created by the following equations: $\max_{\delta} E_{(X,Y)}[\mathcal{L}(f(X+\delta), Y)]$, where δ denotes an adversarial perturbation, and $\mathcal{L}$ denotes a pre-defined loss for machine learning tasks. The adversarial examples can be made by $X_{adv} = X + \delta$. When a given model f is adversarially trained against PGD attack [7], it can be written as follows:

$$\min_{w} \max_{\|\delta\|_\infty \leq \gamma} E_{(X,Y)} \left[\mathcal{L}\left(f(X+\delta), Y\right) \right], \tag{1}$$

where w represents the parameters of f, which are learned to be robust against adversarial attacks. Here, $\|\cdot\|_\infty \leq \gamma$ describes L_∞ norm, and γ-ball means the perturbation magnitude. In this paper, we adversarially train the model f on $\gamma = 0.03$ for standard adversarial attack. Note that once adversarially trained, the parameters of the model f are no longer covered.[1]

As notation of variables that we will use in this paper, Z and $\bar{Z}$ indicate the intermediate features of the model f such that $Z = f_l(X)$ and $\bar{Z} = f_l(X_{adv})$, where $f_l(\cdot)$ describes l-th layer outputs of the given model. Similarly, $f_{l+}(\cdot)$ represents subsequent network after the l-th layer, thus intermediate features can be propagated to the last output layer, such that $\hat{Y} = f_{l+}(Z)$ and $\hat{Y}_{adv} = f_{l+}(\bar{Z})$. The model f can be expressed as $f = f_{l+} \circ f_l$, satisfying $\hat{Y} = f(X)$ for the given clean images X. Also, $\hat{Y}_{adv} = f(X_{adv})$ is denoted by model propagation of the adversarial examples X_{adv}. Note that we designate l-th layer as the last convolutional layer, and regard $l+$ as the rest of the layers in the model.

[1] Previous works [14, 15] have demonstrated that the distinction between the robust and non-robust features arises in adversarial setting. In the sense that adversarially trained networks learn robust representation [24], we set the robust classifier as default. Please see the analysis of the standard training in Appendix F.

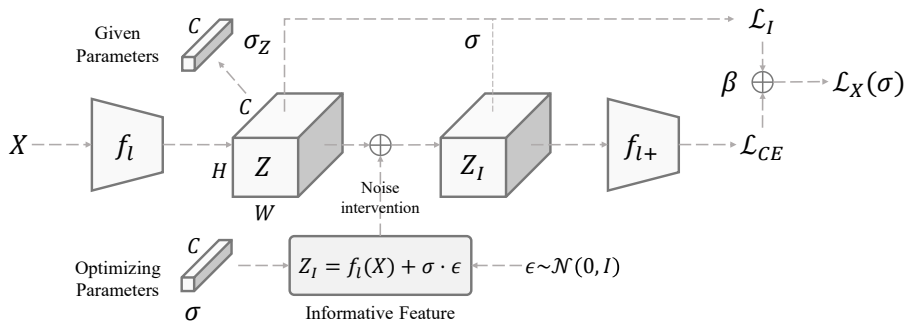


Figure 1: Diagrams of our Information Bottleneck (IB), minimizing $\mathcal{L}_{CE} + \beta\mathcal{L}_I$ to find noise variation σ that can estimate information flow in the intermediate features Z. Here, σ_z represents nature feature variation of intermediate features Z for each unit, and ϵ indicates Gaussian noise sampled from $\mathcal{N}(0, I)$. More implementation details are described in Appendix A.

2.1 Information Bottleneck for Distilling Informative Features

In adversarial setting, we focus on separating robust and non-robust features in the intermediate layer. Recall that robust features are literally robust on the noise (variation) and invariant to existence of the adversarial perturbation, but non-robust features are not. Our approach aims to distill feature units that affect target prediction under the noise perturbation in the intermediate feature space.

We follow that the semantic information is inherently included in the feature units of DNNs [20, 22, 25]. From this perspective, we utilize Information Bottleneck (IB) to distill the robust and non-robust features on the given intermediate features Z. Information Bottleneck [17, 18] proposed to encode maximally informative representation for target labels, restraining input information, concurrently. Using the bottleneck, we suggest a way to assess feature importance and quantify information flow for the target prediction. The objective function of IB can be written as follows:

$$\max_{Z} I(Z, Y) - \beta I(Z, X), \tag{2}$$

where I denotes mutual information, and β represents the degree of restraining input information. The first term $I(Z, Y)$ allows the intermediate features Z to be predictive on the target label Y, and the second term $I(Z, X)$ encourages Z to compress the information of the given images X in the bottleneck. Here, the second term requires a true feature probability $p(Z) = \int_X p(Z \mid X)p(X)dX$ to expand it, but it is computationally intractable due to a high dimensional dependency of the dataset probability $p(X)$. Thus, several works [18, 19] modified the IB's objective function to make possible to learn DNNs without the true feature probability as follows: $\min \mathcal{L}_{CE} + \beta\mathcal{L}_I$ (see Appendix B). In this formulation, $\mathcal{L}_{CE}$ indicates cross entropy loss, and $\mathcal{L}_I$ represents information loss computed by KL divergence [26] between a feature likelihood $p(Z \mid X)$ and an approximate feature probability $q(Z)$. It is radically a closed-form approximation for the true feature probability $p(Z)$.

Firstly, we deliberately inject noise variation into Z to estimate the prediction sensitivity of each feature unit along the noise intervention. To do so, we newly design an approximate feature probability using noise variation σ, such that $q_\sigma(Z) = \mathcal{N}(f_l(X), \sigma^2)$. Then, we sample random variables from $q_\sigma(Z)$ and define informative features Z_I, satisfying $Z_I = f_l(X) + \sigma \cdot \epsilon$ such that $Z_I \sim q_\sigma(Z)$. Note that the operator $\cdot$ denotes Hadamard product, and ϵ stands for Gaussian noise sampled from $\mathcal{N}(0, I)$. Here, the noise variation measures a correlation between intermediate features and model prediction based on the fact that robustness means high correlation on model prediction and non-robustness are opposite [14, 15] in adversarial setting. Since the correlation $E_{(X,Y)}[Y \cdot f(X)]$ in output layer can be expressed as a variance measure $\text{Cov}(Y, f(X))$, we consider the correlation in intermediate layer as the noise variation. From this perspective, if a feature unit is highly predictive despite large noise variation (high correlation), the unit can robustly predict target label, while a non-robust unit cannot.

Once the informative features Z_I are acquired from the noise variation σ, we propagate Z_I to the last output layer and estimate feature importance of each unit for model prediction. Then, we deal with the information loss $\mathcal{L}_I$ in order to alleviate feature heterogeneity between Z and Z_I. Through aforementioned modification [18, 19], our objective function can be written as follows:

$$\mathcal{L}_X(\sigma) = \min_{\sigma} \underbrace{-Y \log f_{l+}(f_l(X) + \sigma \cdot \epsilon)}_{\mathcal{L}_{CE}} + \beta \underbrace{D_{KL}[p(Z \mid X) \mid\mid q_\sigma(Z)]}_{\mathcal{L}_I}, \tag{3}$$

where the feature likelihood $p(Z \mid X)$ is set to $\mathcal{N}(f_l(X), \sigma_z^2)$. Here, σ_z indicates inherent feature variation of the intermediate features Z for each unit. The second term $\mathcal{L}_I$ makes the informative features Z_I resemble the intermediate features Z, while minimizing the cross entropy $\mathcal{L}_{CE}$. This second term can be written as $\mathcal{L}_I = \frac{1}{2}\sum_{k=1}^{C}[\frac{\sigma_{z_k}^2}{\sigma_k^2} + \log\frac{\sigma_k^2}{\sigma_{z_k}^2} - 1]$, where k denotes an index of the noise variation $\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_C]$ (the optimizing parameters) and the feature variation $\sigma_z = [\sigma_{z_1}, \sigma_{z_2}, \cdots, \sigma_{z_C}]$ (the given parameters). Here, only of the variation σ is updated such that $\sigma \leftarrow \sigma - \frac{\partial}{\partial\sigma}\mathcal{L}_X(\sigma)$. In brief, we summarize overall procedure of our bottleneck concept in Fig. 1.

Moreover, we mention that β in Eq. (3) controls the amount of information that flows into the feature representation. Specifically, when β is set to zero, IB loss is equivalent to cross entropy loss, which means that Z_I can accommodate even unimportant features to predict target labels. In contrast, excessively large β only focuses on compressing input information, thus IB may cannot filter out important features to predict target labels. Accordingly, we empirically control β to distill informative features Z_I based on the noise variation σ (Please see section 3.4 for analysis of information flow).

2.2 Separating Informative Features by Tolerance of Feature Variation

After optimizing the informative features Z_I, we compare the noise variation σ for the informative feature units and dichotomize each unit either robust or non-robust based on their prediction sensitivity. We set the criterion for comparison as $T = \max(\sigma_z^2)$. Here, T represents maximum tolerance of the noise variation. It is reasonable choice to set T as criterion, because it indicates the maximal variation with respect to the changes of the given image X, in the feature space. In the following procedure, we explicitly disentangle intermediate features Z into the robust Z_r and non-robust features Z_{nr}.

Firstly, once the noise variation σ is larger than the maximum tolerance T in specific units, it indicates that their corresponding features are highly predictive on the model prediction, despite the noise intervention. Thus, we define their conjunction as robust features. On the other hand, if the variation of specific unit is smaller than T, their corresponding features can be represented as non-robust features. This is because the small variation behaves as strict restriction to retain model prediction of target labels. We assume that once a strong adversarial perturbation comes in, the feature variation of non-robust features becomes to be larger than acceptable tolerance, thereby easily breaking model classifier and leading to misclassified prediction. In this respect, we sort the noise variation according to their magnitude, and cluster them by assigning robust or non-robust channel index to the each feature unit. The robust channel index, $i_r = [i_{r_1}, i_{r_2}, \cdots, i_{r_C}]$ can be computed as follows:

$$i_{r_k} = \mathbb{1}(\sigma_k^2 > T) = \begin{cases} 1 & \sigma^2 > T \\ 0 & \sigma^2 \leq T \end{cases}, \tag{4}$$

where $\mathbb{1}(\cdot)$ represents the indicator function. The non-robust channel index i_{nr} is simply reversed from the robust channel index such that $i_{nr_k} = 1 - i_{r_k}$. Then, we estimate robust features Z_r by multiplying the robust channel index to the intermediate features element-wisely such that $Z_r = i_r \cdot Z$. Similarly, non-robust features Z_{nr} are presented as $Z_{nr} = i_{nr} \cdot Z$. In this way, the intermediate features Z are fully disentangled into the two types of feature representation satisfying $Z = Z_r + Z_{nr}$.

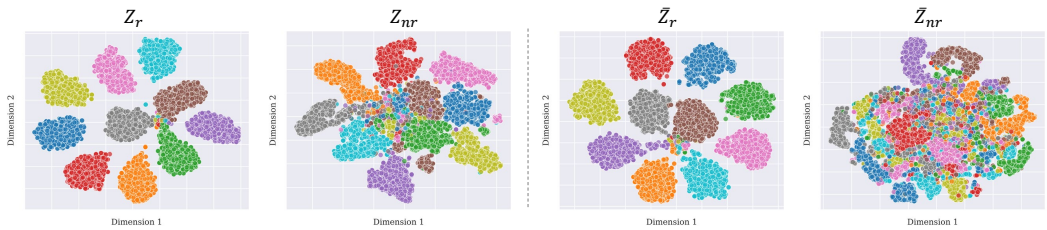
To sum it up, we regard the robust features Z_r that have the larger noise variation as invariant features from the adversarial perturbation. On the other hand, non-robust features Z_{nr} are considered as easily manipulated features, which harmonizes the smaller noise variation. Now, we analyze their impacts to the robustness by expanding the model prediction of Z_I to Taylor approximation (see Appendix C.) with its convergence of local minima [27, 28] as follows:

$$f_{l+}(f(X) + \sigma \cdot \epsilon) = f_{l+}(f(X) + \sigma_r \cdot \epsilon) + \underbrace{\left[\frac{\partial}{\partial\sigma_r} f_{l+}(f(X) + \sigma_r \cdot \epsilon)\right]^T \sigma_{nr}}_{\Delta}, \tag{5}$$

where robust noise variation $\sigma_r = i_r \cdot \sigma$ and non-robust noise variation $\sigma_{nr} = i_{nr} \cdot \sigma$. Since the variation of the robust features does not degrade the model prediction, when σ_{nr} is small enough, the erroneous term Δ in Eq. (5) closes to zero. That is, the model retains having significant robustness against the adversarial perturbation interrupting robust channel index in Z. Conversely, once we force σ_{nr} to increase, its output becomes inaccurate for the robust prediction (*i.e.,* first term in Eq. (5)) due to high Δ. Here, we theoretically demonstrate how the brittleness of non-robust features affects the model robustness. We will thoroughly analyze the properties of the two distilled features by empirically showing the robustness of Z_r and brittleness of Z_{nr} in the following sections.

Table 1: Classification accuracy of model performance attacked by FGSM [2], PGD [7], and CW [4] on VGG-16 [29] and WRN-28-10 [30], adversarially trained with $\gamma = 0.03$ for CIFAR-10, SVHN, and Tiny-ImageNet. We selectively propagate each feature (*i.e.,* intermediate features (Int.), robust (R.), and non-robust features (NR.)) to measure classification accuracy.

Model	Example	CIFAR-10			SVHN			Tiny-ImageNet		
		Int. Acc	R. Acc	NR. Acc	Int. Acc	R. Acc	NR. Acc	Int. Acc	R. Acc	NR. Acc
VGG	Clean	79.73	99.87	34.82	90.35	99.76	57.09	33.98	83.11	7.50
	FGSM [2]	51.28	99.58	22.82	63.71	98.72	40.12	17.45	77.74	4.94
	PGD [7]	44.71	99.38	20.64	48.92	97.91	32.18	16.13	77.59	4.69
	CW [4]	40.32	99.85	13.66	33.26	99.60	16.75	12.00	75.69	4.03
WRN	Clean	82.56	98.66	44.67	93.53	99.44	70.43	43.13	96.35	6.07
	FGSM [2]	56.43	96.80	31.65	73.93	97.90	53.96	20.38	91.58	3.01
	PGD [7]	51.63	96.61	29.44	61.09	96.33	45.79	18.84	90.37	2.83
	CW [4]	45.47	97.74	17.28	40.61	97.58	22.78	13.51	95.78	2.06



(a) Clean Examples (b) PGD Examples

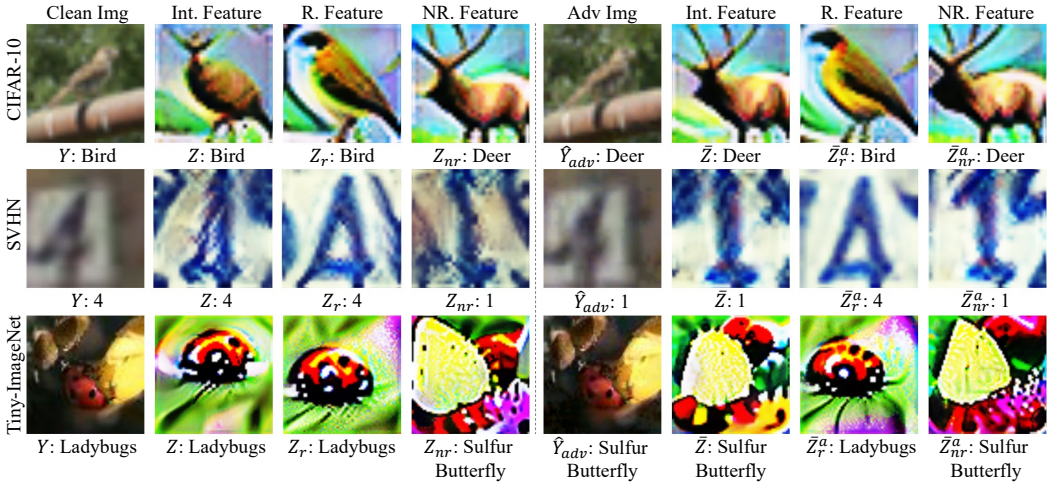
Figure 2: The result of t-SNE plot [31] in CIFAR-10 dataset for VGG-16 network. Each cluster indicates high-dimensional distributions of feature representation for 10 object labels in CIFAR-10 dataset. Additional t-SNE results for other adversarial attacks are illustrated in Appendix D.

3 Analysis of Distilled Features and Visual Interpretation

3.1 Property of Distilled Feature Units under Adversarial Perturbation

After we distill the robust and non-robust features using the bottleneck concept, our next question is *How will the target prediction change under the adversarial attacks?* In our posit, if the bottleneck successfully disentangle the robust and non-robust channel index (*i.e.,* i_r and i_{nr}) from the given examples, we should identify the consequential classification accuracy changes under the adversarial perturbation. That is, after applying the robust index to the attacked feature representation, the selected adversarial features with i_r denoted by $\bar{Z}_r$ (*i.e.,* $\bar{Z}_r = i_r \cdot \bar{Z}$) should have invariant accuracy changes for the target labels. Here, i_r is robust channel index obtained from the given clean examples X, and $\bar{Z}$ represents the intermediate features of the adversarial examples, which means $\bar{Z} = f_l(X_{adv})$. Contrarily, the selected features satisfying $\bar{Z}_{nr} = i_{nr} \cdot \bar{Z}$ will show inaccurate accuracy due to their brittleness under the existence of adversarial perturbation.

In Table 1, we analyze evaluation results of the disentangled features under standard attack algorithms [2, 4, 7] in publicly available datasets [32, 33, 34]. As aforementioned, we apply the robust and non-robust channel index optimized from the clean examples to the adversarial features $\bar{Z}$, and estimate their accuracy (*i.e.,* $f_{l+}(\bar{Z}_r)$ and $f_{l+}(\bar{Z}_{nr})$). As in the table, $\bar{Z}_r$ still shows constant robust accuracy regardless of the adversarial perturbation, even in the high-confidence adversarial attack [4]. On the other hand, we can find that the classification accuracy of $\bar{Z}_{nr}$ steeply degrades as the attacks get stronger, which coincides with the properties of the robust and non-robust features. To further support our experiments, we illustrate the correlation between the disentangled features and true labels, using 2D t-SNE plot [31]. In the case of clean examples as in Fig. 2(a), the robust features Z_r exhibit separable clusters on the target labels, while the non-robust features Z_{nr} show partially disorganized tendency. When adversarial perturbation [7] exists, the attacked features $\bar{Z}_{nr}$ represents more collapsed t-SNE visualization as shown in Fig. 2(b). Notably, we can observe that $\bar{Z}_r$ still sustain highly clustered results even in the attacked condition.



(a) Clean Examples (b) PGD Examples

Figure 3: Feature visualization [25] for the intermediate feature (Int.), robust feature (R.), and non-robust feature (NR.). The class labels under each image indicate the predicted results of the corresponding features propagated by $f_{l+}(\cdot)$. Please see more visualization results in Appendix E.

3.2 Feature Visualization of Robust and Non-robust Features

We have identified the existence of the robust and non-robust features using the bottleneck. Then, we wonder a way of interpreting the semantic information in the feature space with respect to human-perception. Analyzing semantic representation of DNNs is a wide research area to understand their decision [20, 22, 35]. In adversarial setting, several studies [14, 36] argued that robust classifier has more meaningful (*i.e.,* perceptually-aligned) loss gradients in the input space. Engstrom *et al.* [24] further endeavored to interpret robust feature representation using feature visualization [25, 37, 38]. In this manner, we explore whether the disentangled feature representation from the bottleneck indeed has human-perceptible information in the intermediate feature space.

Feature visualization is an optimization-based method that maximizes specific activation of feature units [38, 39], such that $X' = \text{argmax}_X(a_i^l(X))$, where $a_i^l(\cdot)$ indicates feature activation of i-th unit in the l-th layer. In order to understand what conjunction of the robust and non-robust feature units truly interacts with the target labels, we optimize each distilled feature and create their visual explanations. We adopt direct visualization method [25] that has various regularization techniques (*e.g.,* frequency penalization and transformations) to yield better representative visual quality.

We illustrate the feature visualization results of distilled features in Fig. 3. After distilling robust and non-robust features with their corresponding index, we maximize the selected feature unit activation, respectively. As in the figure, we can clearly recognize the semantic information of true labels Y and attacked predictions $\hat{Y}_{adv}$ in the intermediate features (Z and $\bar{Z}$). Interestingly, what we can observe is: (i) the distilled features have semantic information by themselves and maintain their information, even under the adversarial perturbation, (ii) when the adversarial perturbation exists, the brittleness of non-robust features is intensified and reflected to $\bar{Z}$. Thus, the visualization of $\bar{Z}$ and $\bar{Z}_{nr}^a$ looks similar, and they manipulate the target prediction to same adversarial prediction. Unlike the previous work [15] that has argued the non-robust features solely have incomprehensible property, the visualization of the distilled features from our bottleneck represent recognizable outputs even for the adversarial examples, and provides a decisive key to interpret the cause of adversarial examples.

3.3 Adversarial Prediction is Highly Correlated with Non-robust Features

We have observed that the non-robust features optimized from the bottleneck are brittle and easily manipulated under the adversarial perturbation, while robust features maintain substantial prediction results for the target labels. Then, if the primary cause of the adversarial examples indeed belongs to non-robust features, it is natural to examine the correlation between the classification outputs of the non-robust features and the adversarial prediction induced by adversarial attacks. Accordingly, we identically apply our IB loss on the adversarial examples X_{adv} [2, 7, 4], and find their corresponding robust and non-robust channel index using Eq. (4). We employ i_r^a and i_{nr}^a for the adversarial examples

Table 2: The prediction accuracy of the non-robust features $\hat{Z}^a_{nr}$ for attacked labels $\hat{Y}_{adv}$. The input $\hat{Z}^a_{nr}$ is the non-robust features of the corresponding attack methods. To clearly show the correlation between adversarial examples and the non-robust features, we evaluate the accuracy under condition of successfully attacked examples (*i.e.,* $Y \neq \hat{Y}_{adv}$).

Attack	CIFAR-10		SVHN		Tiny-ImageNet	
	VGG	WRN	VGG	WRN	VGG	WRN
FGSM [2]	92.78	94.35	94.90	96.13	63.39	60.82
PGD [7]	93.43	95.06	96.21	96.44	65.84	63.64
CW [4]	93.72	94.42	95.75	97.65	55.68	56.84

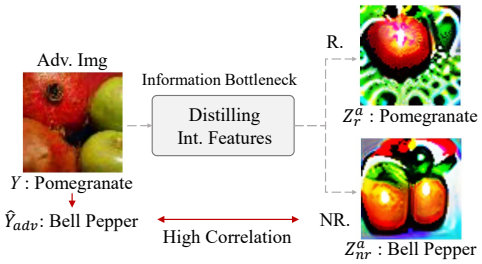


Figure 4: An example of highly correlated adversarial prediction with NR. Both $\bar{Y}_{adv}$ and Z^a_{nr} output same prediction "Bell Pepper".

X_{adv}, which is obtained by optimizing $\mathcal{L}_{X_{adv}}(\sigma)$ instead of Eq. (3). Then, we define $\bar{Z}^a_r$ and $\bar{Z}^a_{nr}$ as robust and non-robust features of the adversarial examples, satisfying $\bar{Z}^a_r = i^a_r \cdot \bar{Z}$ and $\bar{Z}^a_{nr} = i^a_{nr} \cdot \bar{Z}$. To enlighten the correlation of non-robust features and the adversarial prediction, we evaluate the model prediction of $\bar{Z}^a_{nr}$ for the attacked labels $\hat{Y}_{adv}$ that can be written as follows: $\hat{Y}^a_{nr} = f_{l+}(\bar{Z}^a_{nr})$. We set the condition of $\hat{Y}_{adv}$ as successfully attacked labels (*i.e.,* $Y \neq f(X_{adv})$) to definitely show the relation between the prediction $\hat{Y}^a_{nr}$ of non-robust features and the adversarial prediction $\hat{Y}_{adv}$ of adversarial examples. In Table 2, we summarize the accuracy of $\hat{Y}^a_{nr}$ for the successfully attacked label $\hat{Y}_{adv}$ in standard attack methods. Generally, we can observe that the non-robust features are highly predictive on $\hat{Y}_{adv}$ in standard low dimensional datasets such as CIFAR-10 and SVHN. Even in large dataset (*i.e.,* Tiny-ImageNet), the non-robust features are remarkably correlated with the attacked prediction. A brief explanation of the high correlated example is described in Fig. 4.

3.4 Bottleneck Controls Information Flow of Robust and Non-robust Features

In this analysis, we will investigate how the bottleneck affects the information flow of the robust and non-robust features and clarify the their relation. Recall that the bottleneck refines informative features from the given image samples, and β regulates the total amount of the information that flows into Z. We will compare classification accuracy for the robust and non-robust features along β value and analyze the changes of information flow assigned to each disentangled feature.

In Fig. 5, as β value increases, the accuracy of the robust features is getting higher and decreases after specific threshold. As theoretically mentioned in 2.1, we can infer that suitable choice of the β can filter out robust feature units in the intermediate layer. For the excessive β value, we can observe that the bottleneck cannot accurately disentangle adversarial features. For example, the accuracy of the robust and non-robust features are reversed after $\beta = 5.0$ in the particular networks of CIFAR-10 and SVHN datasets. In addition, as in Fig. 5(a) and (b), the accuracy of the non-robust features progressively increases. Such results indicate a few of robust feature units that is not distilled from IB are gradually flowing into the non-robust side and dichotomized as non-robust units, thus producing more higher accuracy. It coincides with the analysis of the number of the assigned robust and non-robust channels in Fig. 5(c). We can observe that the number of robust channels constantly diminishes, whereas that of the non-robust channels increases. It indicates the bottleneck delicately weighs the information flow that should be precisely assigned to the robust or non-robust units.

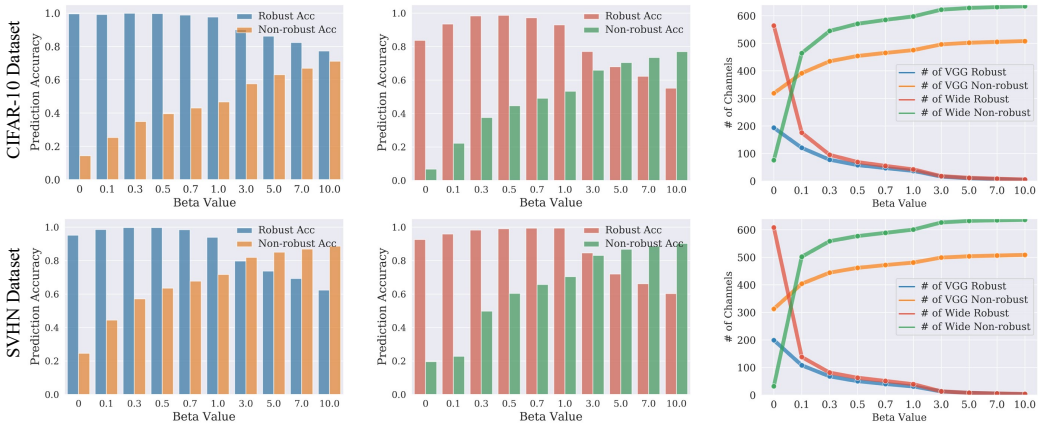
From our analysis section, in conclusion, we have demonstrated the existence of the distilled features using the bottleneck: the robust and non-robust features in the intermediate feature representation. In addition, we have revealed that easily manipulated property of non-robust features is the primary cause of adversarial examples through their high correlation with the adversarial prediction. Based on the fact that the non-robust features break the model prediction, we will suggest an effective way of enhancing the adversarial attack, utilizing the gradient of non-robust features in section 4.

4 Amplifying Brittleness of Non-Robust Features for Attack

In this section, we intentionally increase the non-robust noise variation σ_{nr} to make model classifier fooled based on Eq. (5). Note that increasing σ_{nr} induces high erroneous term Δ, thereby going to

Table 3: Comparing attack performance for FGSM [2], BIM [40], PGD [7], CW [4], AutoAttack (AA) [41], FAB [41], and non-robust feature attack denoted by NRF. We adversarially train VGG-16 and WRN-28-10 on L_∞ norm $\gamma = 0.03$ perturbation for CIFAR-10, SVHN with PGD adversarial training [7] (ADV) and advanced defense methods: TRADES [42] and MART [43].

Model	Method	CIFAR-10								SVHN							
		Clean	FGSM	BIM	PGD	CW	AA	FAB	NRF	Clean	FGSM	BIM	PGD	CW	AA	FAB	NRF
VGG	ADV	79.7	51.3	46.5	44.7	40.3	42.0	41.0	**27.4**	90.4	63.7	52.1	48.8	33.3	39.9	41.6	**12.6**
	TRADES	78.2	54.5	51.7	50.9	43.0	49.5	46.3	**31.2**	90.4	65.3	59.0	57.0	44.8	53.5	50.0	**14.3**
	MART	73.5	54.2	52.2	51.7	42.2	50.6	45.1	**31.4**	90.5	65.1	59.7	57.8	46.4	53.0	47.0	**16.1**
WRN	ADV	82.6	56.4	52.8	51.6	45.5	49.8	49.0	**17.1**	93.5	73.9	64.8	61.1	40.7	55.5	56.6	**13.4**
	TRADES	83.0	57.9	55.0	53.9	46.7	52.4	49.8	**26.8**	93.9	72.9	63.9	60.4	42.0	55.0	55.4	**10.1**
	MART	83.4	59.0	56.0	54.7	46.5	52.8	50.2	**19.6**	94.1	73.0	64.5	61.1	42.3	55.4	56.0	**8.1**



(a) VGG-16 (b) WRN-28-10 (c) # of Assigned Channels

Figure 5: The accuracy of robust and non-robust features along β value, and the number of assigned channels in the l-th feature representation. Note that each color of the lines in (c) corresponds to the same color in the bar plots of (a) and (b). The total number of channels is equivalent to the size of C.

deviate from the robust prediction. However, this variation σ_{nr} is merely an optimizing parameter that cannot be controlled manually. Thus, in order to secondarily have the effect of enlarging σ_{nr}, we alternatively utilize the gradients of the non-robust features directly connected with the model prediction. The gradients of non-robust features in adversarial examples can be described as follows:

$$\mathcal{G}_{nr} = \frac{\partial}{\partial \bar{Z}_{nr}} \mathcal{L}_{base}(f(X+\delta), Y), \tag{6}$$

where we define a baseline loss as $\mathcal{L}_{base}(f(X), Y) = \|\delta\|_2 + c \cdot \max(\max_{i \neq Y}(f(X)_i) - f(X)_i, 0)$, instead of cross entropy loss due to its empirical effectiveness of attack performance [4]. In addition, we use a technique of *changes of variables* [4] from δ to w for generating an imperceptible yet powerful perturbation, such that $\delta = \frac{1}{2}(\tanh(w) + 1) - X$. It serves to smooth out projected gradient descent that clips prematurely to prevent adversarial examples falling into the extreme image domain.

To compute the gradient practically, we firstly calculate $\frac{\partial}{\partial \bar{Z}} \mathcal{L}_{base}$, and multiply it to $\frac{\partial}{\partial \bar{Z}_{nr}} \bar{Z}$ by chain rule. Here, the latter gradient equals to non-robust channel index i_{nr}, because the intermediate features of adversarial examples can be re-written as: $\bar{Z} = i_r \cdot \bar{Z}_r + i_{nr} \cdot \bar{Z}_{nr}$, and the derivative of $\bar{Z}$ over $\bar{Z}_{nr}$ equals to i_{nr}. Thus, the gradients of the non-robust features can be simplified as $i_{nr} \cdot \frac{\partial}{\partial \bar{Z}} \mathcal{L}_{base}$. Using the gradient, we suggest an attack to non-robust features (NRF) by optimizing the following objective: $\min_{\delta} \mathcal{L}_{base} - \|\mathcal{G}_{nr}\|_2$. In Table 3, NRF shows more effective attack performance than the other standard adversarial attacks in [7, 42, 43], since NRF strengthens the gradients of non-robust features that contains the same effect of increasing σ_{nr} to disturb accurate model prediction.

Moreover, we conduct an ablation study on the gradients of robust and non-robust features to probe their influence of prediction changes in Table 4. As expected, maximizing $\mathcal{G}_{nr}$ shows more effective attack performance than minimizing it. It is because maximizing $\mathcal{G}_{nr}$ has an alternative effect of

Table 4: Comparison of classification accuracy for adversarial examples generated by maximizing (↑) or minimizing (↓) the magnitude of robust ($\mathcal{G}_r$) or non-robust feature gradients ($\mathcal{G}_{nr}$).

Model	Method	CIFAR-10					SVHN				
		Clean	$\|\mathcal{G}_{nr}\|_2 \uparrow$	$\|\mathcal{G}_{nr}\|_2 \downarrow$	$\|\mathcal{G}_r\|_2 \uparrow$	$\|\mathcal{G}_r\|_2 \downarrow$	Clean	$\|\mathcal{G}_{nr}\|_2 \uparrow$	$\|\mathcal{G}_{nr}\|_2 \downarrow$	$\|\mathcal{G}_r\|_2 \uparrow$	$\|\mathcal{G}_r\|_2 \downarrow$
VGG	ADV	79.7	**27.4**	67.5	**35.9**	74.6	90.4	**12.6**	71.9	**20.8**	71.5
	TRADES	78.2	**31.2**	71.6	**38.2**	77.1	90.4	**14.3**	68.4	**27.3**	82.3
	MART	73.5	**31.4**	63.8	**39.6**	69.1	90.5	**16.1**	66.2	**31.6**	84.9
WRN	ADV	82.6	**17.1**	74.8	**28.9**	79.5	93.5	**13.4**	88,0	**15.6**	93.4
	TRADES	83.0	**26.8**	73.9	**30.3**	79.9	93.9	**10.1**	88.1	**13.2**	93.3
	MART	83.4	**19.6**	74.3	**24.5**	79.4	94.1	**8.1**	87.7	**9.0**	91.5

increasing the non-robust noise variation σ_{nr} (*i.e.,* large erroneous term $\Delta \uparrow$) in Eq. (5). Whereas, manipulating the gradients of robust features $\mathcal{G}_r$ shows less attack performance than controlling $\mathcal{G}_{nr}$, since it cannot directly handle brittle features in the intermediate feature representation. Especially, it seems difficult to break model prediction, while weakening the gradient of robust features, whose noise variation σ_r have invariance of target prediction even under the adversarial perturbation.

5 Related Work

Various works have been tried to figure out the the reason of adversarial vulnerability in the intermediate feature representation. Inkawhich *et al.* [44] analyzed how the intermediate features are changed by adversarial attacks and measured layer- and class-wise feature distributions. Engstrom *et al.* [24] pointed out that there is a shortcoming of DNNs and their embedding, that is, the primary features used in DNNs are contrasting with what human uses. Also, they argued that the robust optimization to learn robust features could address this shortcoming by encoding high-level representations of input data. Jacobsen *et al.* [45] argued that the reason of adversarial vulnerability lies at invariant characteristics of DNNs to task-relevant features. This invariance makes most regions of input space brittle to adversarial attacks, so that the classifiers become relying on few highly predictive features. In addition, recent works [14, 15] have suggested that the existence of the vulnerability is on the non-robust features, which are inherently included in the data and have unrecognizable property. Our work is in line with the concept of the non-robust features. However, unlike the aforementioned works that directly generated the robust and non-robust datasets to analyze their properties, we reveal that the robust and non-robust features can be completely disentangled in the feature space using Information Bottleneck, and they have semantic information by themselves in fact.

6 Conclusion and Discussion

Conclusion. To understand where the adversarial brittleness comes from, we have investigated information flow in the intermediate feature space, using Information Bottleneck. By estimating feature importance of each feature unit based on the added noise variation, we introduce a novel method to explicitly distill the robust and non-robust features in the feature representation. Through extensive analysis on the distilled features, we figure out how the feature units interacts with target labels, and corroborate that the non-robust features are highly correlated with the adversarial prediction. In addition, we directly visualize the distilled features in the intermediate feature space and reveal that they have recognizable semantic information by themselves. Based on the properties of the distilled features, we suggest a non-robust feature attack (NRF) utilizing the brittleness in the feature space.

Discussion. The adversarial examples, including our work can potentially cause negative impacts on various machine learning applications, such as autonomous driving [46] and medical image process [47]. However, by analyzing the nature of robust and non-robust features that can be primary cause of the adversarial examples, our work contributes important societal impacts in this research field. In this paper, we propose a way of utilizing the gradient of brittle features in the intermediate feature representation (NRF). In future works, we hope bridging the gap of deploying distilled features into the diverse applications such as adversarial detection and defense strategies. Moreover, we would like to note that the last convolutional layer is set to a distilling criterion for Information Bottleneck, since the high-level concepts are included in the higher layer of DNNs. In this sense, it seems a reasonable choice, but how the robust or non-robust features in the lower dimension interact with the much higher dimensional feature units can be another undisclosed future direction to investigate.

References

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *International Conference on Learning Representations (ICLR)*, 2014.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[3] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," *ACM Workshop on Artificial Intelligence and Security*, 2017.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, IEEE Computer Society, 2017.

[5] N. Papernot, P. McDaniel, I. J. Goodfellow, S. Jha, Z. Y. Celik, and A. Swami, "Practical black-box attacks against machine learning," *ACM ASIA Conference on Computer and Communications Security (ACM ASIACCS)*, 2017.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1625–1634, 2018.

[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[8] G. Apruzzese, M. Colajanni, L. Ferretti, and M. Marchetti, "Addressing adversarial attacks against security systems based on machine learning," in *International Conference on Cyber Conflict (CyCon)*, vol. 900, pp. 1–18, IEEE, 2019.

[9] X. Wang, J. Li, X. Kuang, Y. an Tan, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12 – 23, 2019.

[10] Y. E. Sagduyu, Y. Shi, and T. Erpek, "Iot network security from the perspective of adversarial deep learning," in *International Conference on Sensing, Communication, and Networking (SECON)*, pp. 1–9, 2019.

[11] I. Rosenberg, A. Shabtai, Y. Elovici, and L. Rokach, "Adversarial learning in the cyber security domain," *arXiv preprint arXiv:2007.02407*, 2020.

[12] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein, "Are adversarial examples inevitable?," in *International Conference on Learning Representations (ICLR)*, 2018.

[13] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[14] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations (ICLR)*, 2019.

[15] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[16] L. Engstrom, J. Gilmer, G. Goh, D. Hendrycks, A. Ilyas, A. Madry, R. Nakano, P. Nakkiran, S. Santurkar, B. Tran, D. Tsipras, and E. Wallace, "A discussion of 'adversarial examples are not bugs, they are features'," *Distill*, 2019. https://distill.pub/2019/advex-bugs-discussion.

[17] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pp. 368–377, 1999.

[18] A. Alemi, I. Fischer, J. Dillon, and K. Murphy, "Deep variational information bottleneck," in *International Conference on Learning Representations (ICLR)*, 2017.

[19] K. Schulz, L. Sixt, F. Tombari, and T. Landgraf, "Restricting the flow: Information bottlenecks for attribution," in *International Conference on Learning Representations (ICLR)*, 2019.

[20] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, pp. 818–833, Springer, 2014.

[21] J. Yosinski, J. Clune, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *International Conference on Machine Learning Workshop on Deep Learning*, Citeseer.

[22] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6541–6549, 2017.

[23] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018. https://distill.pub/2018/building-blocks.

[24] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry, "Adversarial robustness as a prior for learned representations," *arXiv preprint arXiv:1906.00945*, 2019.

[25] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[26] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[27] C.-J. Simon-Gabriel, Y. Ollivier, L. Bottou, B. Schölkopf, and D. Lopez-Paz, "First-order adversarial vulnerability of neural networks and input dimension," in *International Conference on Machine Learning (ICML)*, pp. 5809–5817, PMLR, 2019.

[28] C. Qin, J. Martens, S. Gowal, D. Krishnan, K. Dvijotham, A. Fawzi, S. De, R. Stanforth, and P. Kohli, "Adversarial robustness through local linearization," in *Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[30] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference (BMVC)*, 2016.

[31] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[32] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[34] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, p. 7, 2015.

[35] R. Fong and A. Vedaldi, "Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8730–8738, 2018.

[36] S. Kaur, J. Cohen, and Z. C. Lipton, "Are perceptually-aligned gradients a general property of robust classifiers?," in *NeurIPS Workshop: Science Meets Engineering of Deep Learning (NeurIPS SEDL)*, 2019.

[37] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *University of Montreal*, vol. 1341, no. 3, p. 1, 2009.

[38] A. Nguyen, J. Yosinski, and J. Clune, "Understanding neural networks via feature visualization: A survey," in *Explainable AI: interpreting, explaining and visualizing deep learning*, pp. 55–76, Springer, 2019.

[39] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5188–5196, 2015.

[40] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *International Conference on Learning Representations (ICLR)*, 2017.

[41] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International Conference on Machine Learning (ICML)*, 2020.

[42] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning (ICML)*, pp. 7472–7482, PMLR, 2019.

[43] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International Conference on Learning Representations (ICLR)*, 2020.

[44] N. Inkawhich, K. Liang, L. Carin, and Y. Chen, "Transferable perturbations of deep feature distributions," in *International Conference on Learning Representations (ICLR)*, 2019.

[45] J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge, "Excessive invariance causes adversarial vulnerability," in *International Conference on Learning Representations (ICLR)*, 2018.

[46] V. A. Banks, K. L. Plant, and N. A. Stanton, "Driver error or designer error: Using the perceptual cycle model to explore the circumstances surrounding the fatal tesla crash on 7th may 2016," *Safety Science*, vol. 108, pp. 278–285, 2018.

[47] V. Antun, F. Renna, C. Poon, B. Adcock, and A. C. Hansen, "On instabilities of deep learning in image reconstruction and the potential costs of ai," *Proceedings of the National Academy of Sciences*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 2 to 3 and the supplemental material for more details.
   (b) Did you include complete proofs of all theoretical results? [Yes] Complete proofs are described in Section 2 and the supplemental material.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 2 and the supplemental material for more details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Negligible error in the experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is attached in the supplemental material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]