Batch Normalization Increases Adversarial Vulnerability and Decreases Adversarial Transferability: A Non-Robust Feature Perspective

Philipp Benz* pbenz@kaist.ac.kr
Chaoning Zhang* chaoningzhang1990@gmail.com
In So Kweon iskweon77@kaist.ac.kr

Korea Advanced Institute of Science and Technology (KAIST)

Abstract

Batch normalization (BN) has been widely used in modern deep neural networks (DNNs) due to improved convergence. BN is observed to increase the model accuracy while at the cost of adversarial robustness. There is an increasing interest in the ML community to understand the impact of BN on DNNs, especially related to the model robustness. This work attempts to understand the impact of BN on DNNs from a non-robust feature perspective. Straightforwardly, the improved accuracy can be attributed to the better utilization of useful features. It remains unclear whether BN mainly favors learning robust features (RFs) or non-robust features (NRFs). Our work presents empirical evidence that supports that BN shifts a model towards being more dependent on NRFs. To facilitate the analysis of such a feature robustness shift, we propose a framework for disentangling robust usefulness into robustness and usefulness. Extensive analysis under the proposed framework yields valuable insight on the DNN behavior regarding robustness, e.g. *DNNs first mainly learn RFs and then NRFs. The insight that RFs transfer better than NRFs, further inspires simple techniques to strengthen transfer-based black-box attacks.*

1. Introduction

Batch normalization (BN) [18] has been considered as a milestone technique in the development of deep neural networks (DNNs) pushing the frontier in computer vision due to improved convergence. Numerous works have attempted to understand the impact of BN on DNNs from various perspectives. In contrast to previous works, investigating why (or how) BN helps the optimization [31, 1], our work focuses on the *consequence* of such enhanced optimization, especially on the model robustness. Our work is not the first one to study BN and robustness together. Most of the previous works are focusing on the covariate shift [3, 33, 43, 42]. For example, [3] adapts the BN statistics to improve the model robustness against common corruptions. On the contrary, our work studies BN by focusing on its impact on adversarial robustness from the non-robust feature perspective.

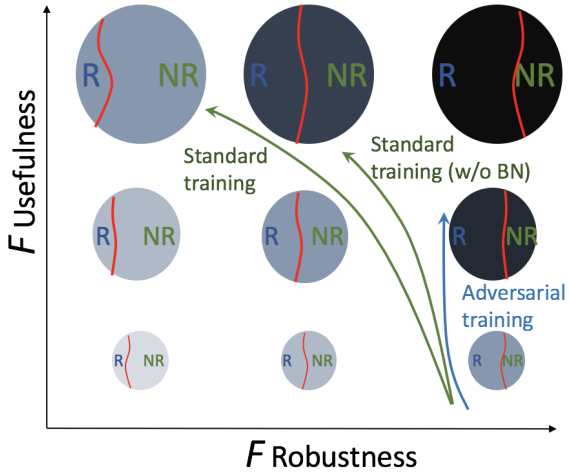

Figure 1. Schematic of disentangling F usefulness and robustness with ball color representing robust usefulness, *i.e.* the darker, the more robustly useful. Ball size indicates usefulness while the red line divides RFs and NRFs.

We evaluate the behavior of models with and w/o BN on multiple datasets in Table 1. As expected, BN improves the clean accuracy, *i.e.* accuracy on clean images. However, this comes at the cost of lower robust accuracy, *i.e.* accuracy on adversarial images [35]. Straightforwardly, the DNN can be seen as a set of useful features, consisting of robust features (RFs) and non-robust features (NRFs) [17], and the improved accuracy can be roughly interpreted as BN facilitating utilization of more useful features. Yet, it remains unclear whether BN mainly favors learning RFs or NRFs. Our empirical investigation shows that BN and other normalization variants all increase adversarial vulnerability in standard training, suggesting *BN shifts the model to rely*

*Equal contribution

more on NRFs than RFs for classification. Our claim is further corroborated by the analysis of corruption robustness and feature transferability.

With the above empirical evidence supporting our main claim that BN shifts the model towards being more dependent on NRFs, it is still necessary yet non-trivial to define and measure such a feature robustness shift. Inspired by [17], with a classifier DNN defined as a feature set F, we propose a framework, as shown in Figure 1, for disentangling F robust usefulness into F robustness and F usefulness. Following [17], F usefulness and F robust usefulness can be measured by clean accuracy and robust accuracy, respectively. F usefulness can be seen as the amount of total useful features, indicated by the ball size and F robustness indicates the ratio of RFs to NRFs (see Figure 1). Conceptually, F robustness is orthogonal to F usefulness. The core difference between our feature analysis framework and that in [17] lies in the disentangled F robustness which can be utilized to measure how much BN shifts the model towards NRFs. In practice, however, it is very difficult to directly measure F robustness. Inspired by [26, 28] demonstrating a positive correlation between robustness and local linearity, we propose a metric termed *Local Input Gradient Similarity* (LIGS) (see Sec. 4), measuring the local linearity of a DNN as an indication for F robustness. Admittedly, comparing the clean accuracy and robust accuracy also sheds some light on the F robustness, however, they are heavily influenced by the dimension of usefulness. Measuring LIGS provides direct evidence on how BN influences the robustness of learned F, which facilitates analysis under the above framework.

Such analysis yields insight on the DNN behavior regarding robustness. On a normal dataset, introducing BN (or IN/LN/GN) into the DNN consistently reduces F robustness, which naturally explains their induced lower robust accuracy. We investigate and compare the behaviour of models trained on a dataset that mainly has either RFs or NRFs, which shows that NRFs are difficult to learn w/o BN, suggesting that BN is essential for learning NRFs. Further investigation on the dataset with RFs and NRFs cued for conflicting labels reveals that the model learns first RFs and then NRFs, and the previous learned RFs can be partially forgotten while the model learns NRFs in the later stage. The proposed framework is not limited for analyzing the impact of BN, and we also analyze other network structures and optimization factors. Interestingly, we find that most of them have no significant influence on F robustness indicated by the LIGS metric, leaving BN (and other normalization variants) among our investigated factors as the only one that have significant influence on the shift towards more NRFs. One practical use case of our key findings is to boost transferable attacks. We demonstrate that a substitute model w/o BN outperforms its counterpart with BN and that early-stopping the training of the substitute model can also boost transferable attacks.

Table 1. Comparison of models with and w/o BN on accuracy and robustness. [11] reports a similar phenomenon.

	Network	Acc	PGD l_2 0.25	PGD l_∞ 1/255	CW l_2 0.25	CW l_∞ 1/255
ImageNet	VGG16 (None)	71.59	15.55	1.79	16.66	0.23
	VGG16 (BN)	73.37	6.04	0.55	6.82	0.02
	VGG19 (None)	72.38	16.52	2.18	17.46	0.30
	VGG19 (BN)	74.24	6.94	0.69	7.66	0.03
	ResNet18 (None)	66.51	30.44	1.24	30.43	0.93
	ResNet18 (BN)	70.50	16.79	0.14	17.40	0.07
	ResNet50 (None)	71.60	28.00	2.17	28.26	0.88
	ResNet50 (BN)	76.54	19.50	0.53	20.19	0.19
SVHN	VGG11 (None)	95.42	63.91	83.20	64.64	83.24
	VGG11 (BN)	96.27	51.22	77.50	51.13	77.61
	VGG16 (None)	95.76	62.24	82.76	62.97	82.92
	VGG16 (BN)	96.43	52.90	80.24	52.88	79.93
CIFAR10	VGG11 (None)	90.06	51.30	70.47	51.75	70.40
	VGG11 (BN)	92.48	39.31	63.87	39.04	63.85
	VGG16 (None)	91.89	34.01	63.18	34.37	63.46
	VGG16 (BN)	93.7	28.61	56.05	24.01	54.58
	ResNet50 (None)	92.15	29.24	49.33	17.09	49.24
	ResNet50 (BN)	95.6	9.15	36.37	8.72	36.64

2. Related Work

Adversarial Vulnerability and Transferability. Adversarial examples [35, 15] have attracted significant attention in machine learning, which raises concern for improving the model robustness [5]. The cause of adversarial vulnerability has been explored from different perspectives, such as local linearity [15], input high-dimension [14, 34, 25], limited sample [32, 36], boundary tilting [36], test error in noise [10, 13, 6], etc. The cause of adversarial vulnerability has recently been attributed to highly predictive yet brittle NRFs [17]. [17] proposes a feature analysis framework that discusses feature usefulness and robust usefulness that can be measured by (clean) accuracy and robust accuracy, respectively. Despite efforts to bridge their gap [48], it is widely recognized that there is a trade-off between them [38]. Our framework proposes another dimension of feature robustness that is orthogonal to feature usefulness. On the other hand, one intriguing property of adversarial examples are their transferability, *i.e.* adversarial examples generated on a substitute model is also often effective in attacking an unknown target model [21]. Complementary to existing techniques [8, 44, 9], our finding from the NRF perspective results in simple techniques that boost transferability.

Batch normalization and beyond. Since the advent of BN [18], numerous works have investigated it from various perspectives. BN performs normalization along batch dimension to reduce covariate shift, resulting in improved convergence [18]. The stochasticity of the batch statistics also serves as a regularizer and improves generalization [23]. However, the property of batch dependence limits the ap-

Table 2. Influence of various normalization techniques on accuracy (left/) and robustness (/right).

Data	Network	None	BN	IN	LN	GN
SVHN	VGG11	95.42/63.91	96.27/51.22	95.89/45.82	96.29/56.77	96.30/56.37
	VGG16	95.76/62.24	96.43/52.90	96.64/47.43	96.18/59.55	96.21/59.50
CIFAR10	VGG11	90.06/51.30	92.48/39.31	88.42/31.38	90.54/42.41	90.68/39.43
	VGG16	91.89/34.02	93.70/28.61	90.73/13.44	92.51/28.92	92.83/26.73
	ResNet50	92.15/29.24	95.60/9.15	93.40/10.80	90.37/7.24	92.61/6.43
ImageNet	ResNet18	66.51/30.44	70.50/16.79	63.14/14.29	68.36/19.72	69.02/19.76
	ResNet50	71.60/28.00	76.54/19.50	67.97/13.65	71.08/17.38	74.69/20.34

plicability of BN when large batch size is impractical [2], or there is a domain change [29]. To avoid such an issue related to the batch dimension, several alternative normalization techniques have been proposed to exploit the channel dimension, such as layer normalization (LN) [2] in transformers and Instance normalization (IN) [39] in style transfer. LN and IN can be seen as two special cases of Group normalization (GN) in [41]. Complementary to [11] showing BN increases adversarial vulnerability, our work finds that LN/IN/GN mirrors the same trend. Recently, Xie *et al*. show that BN might prevent the model from obtaining strong robustness when clean examples are included in adversarial training due to the two-domain hypothesis [43] and that the usage of an auxiliary batch norm for adversarial examples can improve image recognition [42]. A similar approach has been adopted in [19] for adversarial contrastive learning. Recently [3, 33] show that covariate shift adaptation at the inference stage can enhance the robustness against common corruptions.

3. RFs *vs*. NRFs: Which Side does BN Favor?

3.1. Background and Motivation

Reason vs Effect. BN [18] is widely adopted due to its improved convergence and the community has attempted to understand how BN helps the optimization. BN was first motivated to reduce the internal covariate shift (ICS) [18], while [31] claims that reducing ICS does *not* help optimization, instead, the improved optimization is mainly attributed to BN smoothing the optimization landscape. One recent work [1] revisits ICS and refutes the claim in [31] and suggests that reducing the ICS is actually the reason. Overall, the mechanism of why BN improves the optimization remains unclear and probably no clear consensus will be found in the near future. In contrast to previous works [31, 1] investigating the *reason* of the improved optimization, our work focuses on the *effect*, more specifically, on the adversarial robustness. Given that a DNN learns a set of features [17], the improved optimization is expected to lead to a model with more useful features, consequently improving the accuracy, as expected. Its side effect of increasing adversarial vulnerability is worth an investigation. Inspired by [17], our work focuses on the NRF perspective.

RFs vs. NRFs. *Feature* is one key concept in computer vision and the past few years have witnessed a shift from hand-crafted features [22, 7] to DNNs intrinsically extracting features [20, 16]. Despite different interpretations of how DNN works, there is a belief that a classification DNN can be perceived as a function utilizing useful features [17]. Specifically, [17] defines a *feature* to be a function mapping from the input space $\mathcal{X}$ to real numbers, *i.e.* $f : \mathcal{X} \to \mathbb{R}$. A feature f is ρ-useful ($\rho > 0$) if it is correlated with the true label in expectation, *i.e.* $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[y \cdot f(x)] \geq \rho$. Given a ρ-useful feature f, robust features (RFs) and non-robust features (NRFs) are formally defined as follows:

- *RF:* a feature f is robust if there exists a $\gamma > 0$ for it to be γ-robustly useful under some specified set of valid perturbations Δ, *i.e.* $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\inf_{\delta\in\Delta(x)} y \cdot f(x+\delta)] \geq \gamma$.
- *NRF:* a feature f is non-robust if $\gamma > 0$ does not exist.

Conjecture. With the above definition [17] to dichotomize features, we conjecture that BN shifts the model to rely more on NRFs instead of RFs.

3.2. Empirical evidence

The observation in Table 1 constitutes evidence for our conjecture, which can be corroborated as follows. First, BN simply normalizes the DNN intermediate feature layers; thus if our conjecture is correct, other normalization techniques (such as LN, IN, and GN) are also likely to mirror the same behavior. Second, with the link between adversarial robustness and corruption robustness [13], our claim can be more convincing if the corruption robustness analysis also supports it.

Adversarial robustness. Table 2 shows that the phenomenon of increased adversarial vulnerability is not limited to BN but also occurs for IN, LN, and GN. Overall, except for the result of ResNet50 on CIFAR10, IN consistently achieves the lowest robust accuracy. We suspect that this can be attributed to the prior finding [39, 27] that IN excludes style information by performing instance-wise normalization. The style information is likely RFs (note that changing style,

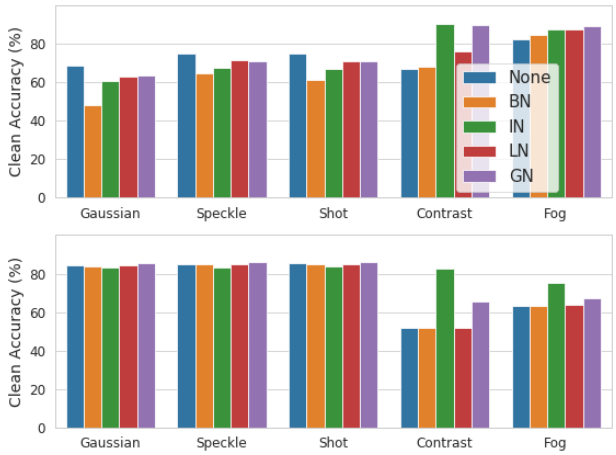

Figure 2. Corruption robustness of VGG16 with *standard training* (top) and *adversarial training* (bottom).

e.g. color, normally requires large pixel intensity change), thus IN discarding style can result in the least robust model.

Ford *et al.* [13] revealed that adversarial training (and Gaussian data augmentation) significantly improve the robustness against noise corruptions, *i.e.* Gaussian/Speckle/Shot, while decreasing the robustness against contrast and fog, which is confirmed in Figure 2. Following [17], a standard model is perceived to learn a sufficient amount of NRFs, while a robust model (robustified through adversarial training) mainly has RFs. Perceiving from the feature perspective, the following explanation arises: noise corruptions mainly corrupt the NRFs while contrast and fog mainly corrupt the RFs. Our explanation from the feature perspective echoes with prior explanation from a frequency perspective [45]. We discuss their link in the supplementary.

Corruption robustness. The model without normalization are more robust to noise corruptions than their counterparts with normalization while a reverse trend is observed for fog and contrast. Given our explanation, this contrasting behavior suggests that the models with normalization learn more NRFs instead of RFs. Another observation from Figure 2 is that IN leads to extra high-robustness against contrast corruption, suggesting less IN is the least dependent on RFs in this context. This aligns well with the previous result that the model with IN is generally the least robust.

4. Framework for disentangling usefulness and robustness

Following [17], we define a DNN classifier as a set of features, *i.e.* $F = \{f\}$. The definitions of f usefulness and robust usefulness in Sec. 3.1 can be readily extended to F.

- F usefulness: F is ρ-useful ($\rho > 0$) if it is correlated with the true label in expectation, *i.e.* $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[y \cdot F(x)] \geq \rho$;
- F robust usefulness: F is γ-robustly useful if there exists a $\gamma > 0$ under some specified set of valid perturbations Δ, *i.e.* $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\inf_{\delta\in\Delta(x)} y \cdot F(x+\delta)] \geq \gamma$.

For being orthogonal to usefulness, we can not trivially define F robustness by measuring its correlation with the true label in expectation. With a locally quadratic approximation, prior work [26] provided theoretical evidence of a strong relationship between robustness and local linearity. Thus, with $\nabla l(x, y)$ denoting the partial gradient of the CE loss l with respect to the x input, we define F robustness as follows.

- F robustness: A feature set F is β-robust if the local linearity is larger than β ($\beta > 0$), *i.e.* $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathrm{D},\nu\sim\Delta}\left[sim(\nabla l(x, y), \nabla l(x + \nu, y))\right] \geq \beta$.

The local linearity indicated by the similarity (sim) between $\nabla(l(x, y)$ and $\nabla l(x + \nu, y)$ can be represented in different forms, such as calculating the norm of their difference [26]. We adopt the cosine similarity [46] to quantify this similarity as:

$$\mathbf{E}_{(x,y)\sim D,\nu\sim\Delta}\left[\frac{\nabla l(x, y) \cdot \nabla l(x + \nu, y)}{\|\nabla l(x, y)\| \cdot \|\nabla l(x + \nu, y)\|}\right]. \quad (1)$$

The adopted metric indicates similarity (or linearity) between the original and locally perturbed input gradient and is thus termed *Local Input Gradient Similarity* (LIGS). For additional justification for adopting this metric, refer to the supplementary. Nonetheless, metrics other than LIGS might also be appropriate.

Perturbation choice of LIGS. We investigate the influence of perturbation type by setting ν to Gaussian noise, uniform noise, FGSM perturbation, and PGD perturbation. Among all the chosen types of perturbation, we observe a general trend that the LIGS decreases with training, and consistently the LIGS w/o BN is higher than that with BN. Unless specified, we sample the ν from a Gaussian distribution to measure the LIGS in this work. Additional details and results can be found in the supplementary.

Relation to prior works. The primary motivation of adopting LIGS in this work is to define and quantify the F robustness. Directly maximizing the local linearity as a new regularizer has been shown to improve adversarial robustness on par with adversarial training [26]. A similar finding has also been shown in [28]. Note that "adversarial robustness" mostly refers to "robust usefulness" instead of solely "robustness". To avoid confusion, we highlight that F *robustness* is orthogonal to usefulness. Contrary to prior works [26, 28], which improve "adversarial robustness" by investigating (and establishing) the link between robust usefulness with local linearity, we adopt the local linearity as a measure of "robustness". By definition, local linearity

does not imply usefulness because it is not related to the correlation with the true label. Nonetheless, their observation that maximizing local linearity can help improve robust usefulness (measured by robust accuracy), can be seen as a natural consequence of increasing F robustness.

Interpretation and relationship. Informally but intuitively, the usefulness of F can be perceived as the number of features if we assume that each feature is equally useful for classification; and the robustness of F can be seen as the ratio of RFs to NRFs in F. This is illustrated schematically in Figure 1, where a DNN located in the top right region has high robust usefulness, *i.e*. high robust accuracy, indicating the model learns sufficient features and among them, a high percentage belongs to RFs. A low robust accuracy can be caused by either low F usefulness or low F robustness. Figure 1 also shows the difference between standard training (green) and adversarial training (blue). Both start from the state of high F robustness and low F usefulness; compared with standard training, adversarial training eventually leads to a model of higher F robustness and lower F usefulness. For standard training, removing BN also increases F robustness. By definition, F robustness, F usefulness, and F robust usefulness can be measured by LIGS, clean accuracy, and robust accuracy, respectively. The schematic illustration in Figure 1 aligns well with the results in Figure 3.

5. Disentangling usefulness and robustness of model features

With the above evidence to corroborate that BN shifts the model to rely more on NRFs, it would be desirable to have a metric to measure "pure" robustness independent of usefulness. Given both RFs/NRFs are useful and their core difference lies in robustness, such a metric is crucial for providing direct evidence on the shift towards NRFs by showing a lower "pure" robustness. Moreover, the LIGS trend during the training stage also sheds light on the learned order of features, *i.e*. from RFs to NRFs or vice versa. Evaluating adversarial robustness by robust accuracy demonstrates how robustly useful the model features are. Thus, disentangling robust usefulness into usefulness and robustness provides a better understanding of adversarial robustness.

The overall trend in Figure 3 shows that robust accuracy is influenced by both clean accuracy and LIGS. For example, for adversarial (adv.) training, the LIGS stays close to 1 during the entire training stage, and the robust accuracy is highly influenced by the clean accuracy. For standard (std.) training, however, the LIGS is much lower, leading to a much smaller robust accuracy despite slightly higher clean accuracy. The influence of BN is mainly observed on LIGS. During the entire training stage, BN leads to a significantly lower LIGS, consequently lower robust accuracy.

As (standard) training evolves, the LIGS value decreases, *i.e*. the feature robustness decreases, suggesting the model relies more on NRFs as training evolves. The influence of BN in adv. training, however, is limited. Here, only BN on CIFAR10 is reported. We provide more results with IN/LN/GN and results on ImageNet in the supplementary. The results mirror the trend in Figure 3.

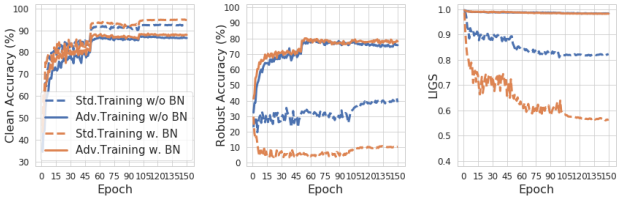

Figure 3. Trend of clean accuracy, robust accuracy, LIGS with ResNet18 on CIFAR10.

On the role of BN in adversarial training. With standard training, we find that BN increases adversarial vulnerability. To improve robustness, adversarial training is one of the most widely used methods. The authors of [43] showed that BN might prevent networks from obtaining strong robustness in adversarial training. However, this is only true when clean images are utilized in the training and the reason is attributed to the two-domain hypothesis. For standard adversarial training [24] with only adversarial images, as shown in Figure 3, BN is found to have no influence on LIGS as well as robust accuracy. This is reasonable because adversarial training explicitly discards NRFs.

Regularization of LIGS. The above results show that the robust accuracy and LIGS are linked. To verify the link between them, we use LIGS as a regularizer during training. The results in Figure 4 confirm that increasing LIGS through regularization improves the robust accuracy by a large margin despite a small influence on clean accuracy.

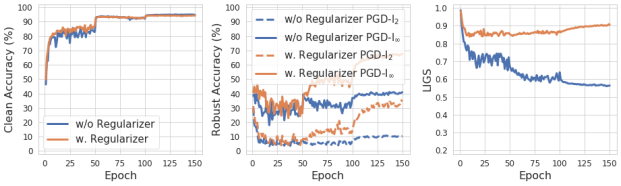

Figure 4. Effect of regularizing LIGS.

5.1. Training on a dataset of disentangled RFs and NRFs

Note, that by default the experiment setup is the same by only changing the variable of interest (*e.g*. testing with and without BN). Training on a dataset of disentangled RFs and NRFs with BN as the control variable highlights the effect of BN on them while excluding mutual influence.

Disentangling RFs and NRFs. Following the procedure of [17] we extract $\widehat{\mathcal{D}}_R$, $\widehat{\mathcal{D}}_{NR}$ and $\widehat{\mathcal{D}}_{rand}$ (Description in the supplementary). Note that $\widehat{\mathcal{D}}_R$ mainly (if not exclusively)

has RFs, while $\widehat{\mathcal{D}}_{rand}$ only has NRFs. $\widehat{\mathcal{D}}_{NR}$ has both RFs and NRFs (see the supplementary for results on $\widehat{\mathcal{D}}_{NR}$). Here, to demonstrate the effect of BN on either NRFs or NRFs, we report the results trained on $\widehat{\mathcal{D}}_{R}$ and $\widehat{\mathcal{D}}_{rand}$ in Figure 5, where the clean accuracy and robust accuracy results echo the findings in [17]. There are two major observations regarding the LIGS result. First, the LIGS on $\widehat{\mathcal{D}}_{R}$ is very high (more than 0.9), which explains why a model (normally) trained on $\widehat{\mathcal{D}}_{R}$ has relatively high robust accuracy, while the LIGS on $\widehat{\mathcal{D}}_{rand}$ eventually becomes very low because $\widehat{\mathcal{D}}_{rand}$ only has NRFs. Second, w/o BN, the model is found to not converge on $\widehat{\mathcal{D}}_{rand}$, leading to 10% accuracy, *i.e.* equivalent to random guess. The model with BN starts to converge (achieving an accuracy of higher than 10%) after around 25 epochs and the LIGS is observed to increase before the model starts converging. This suggests that the model is learning features that are robust yet hardly useful. This "warmup" phenomenon is not accidental and repeatedly happens with different random training seeds. After the model starts to converge, the LIGS quickly plummets to a low value.

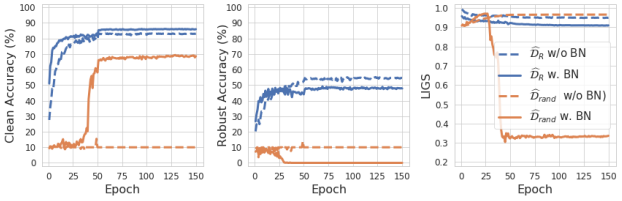

Figure 5. Analysis of BN with ResNet18 on datasets of disentangled features.

Training on a dataset of conflicting RFs and NRFs. In the original dataset, $\mathcal{D}$, abundant RFs and NRFs co-exist and the model learns both for classification. It is interesting to understand the order of the learned features, *i.e.* from RFs to NRFs or vice versa, as well their influence on each other. The decreasing trend of LIGS in Figure 3 suggests that the model learns mainly RFs first. Here, we provide another evidence with the metric of clean accuracy. In the $\mathcal{D}$, RFs and NRFs are cued for the same classification, thus no insight can be deduced from the clean accuracy. To this end, we design a dataset $\widehat{\mathcal{D}}_{Conflict}$ of conflicting RFs and NRFs. Specifically, we exploit the generated $\widehat{\mathcal{D}}_{R}$ of target class $t+1$ as the starting images and generate the NRFs of the target class t. In other words, in the $\widehat{\mathcal{D}}_{Conflict}$ RFs are cued for class $t+1$ while NRFs are cued for class t.

Figure 6 shows that with BN the clean accuracy aligned with RFs increases significantly in the first few epochs and peaks around 80% followed by a sharp decrease, while the accuracy aligned with NRFs slowly increases until saturation. It supports that the model learns from RFs to NRFs. Eventually, the accuracy aligned with NRFs surpasses that aligned with RFs, indicating the model forgets most of the first learned RFs during the later stage. W/o BN, we find that the model in the whole stage learns RFs while ignoring NRFs. It clearly shows that BN is crucial for learning NRFs, which naturally explains why BN shifts the models towards learning more NRFs. We also discuss the results of $\widehat{\mathcal{D}}_{det}$ [17] in the supplementary.

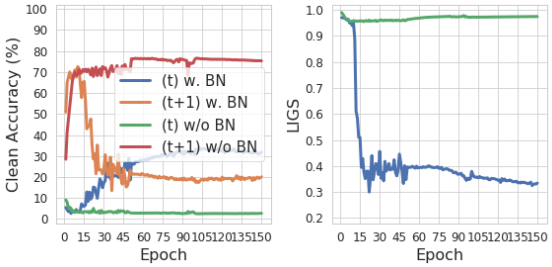

Figure 6. Analysis on dataset with Conflicting RFs and NRFs.

5.2. Exploration beyond (batch) normalization

Network structure factors and optimization factors. Besides normalization, other factors could influence the DNN behavior, especially concerning its robustness. We study two categories of factors: (a) structure category including network width, network depth, and ReLu variants; (b) optimization category including weight decay, initial learning rate, and optimizer. The results are presented in Figure 7. We find that most studied factors have no significant influence on the LIGS. Increasing network width and depth can increase or decrease the LIGS, respectively, but by a small margin. No visible difference between ReLU and Leaky ReLU can be observed, while SeLU leads to a slightly higher LIGS with lower clean accuracy. Some candidates from the optimization category are found to influence F robustness differently in the early and later stages of training. High weight decay leads to higher LIGS in the early stages of training and slightly lower in the end. A higher initial learning rate, such as 0.5, results in higher LIGS in the early training stage, but eventually leads to lower LIGS. For both weight decay and initial learning rate, an opposite trend of lower/higher in the early/later stage is observed with clean accuracy. SGD optimizer and ADAGrad show similar behavior on LIGS, ADAM leads to slightly higher LIGS. Their influence on clean accuracy is more significant.

6. On the link between why BN favors NRFs and how BN helps optimization.

As discussed in Sec. 3.1, there are two major conflicting views on how BN helps optimization: (a) smoothing the optimization landscape [31] *vs.* (b) reducing ICS [1]. View (a) and view (b) hold exactly the opposite claims against each other. On the other hand, why BN shifts towards NRFs also remains unclear. The shift towards NRFs is caused by the improved optimization of BN (note that both views agree

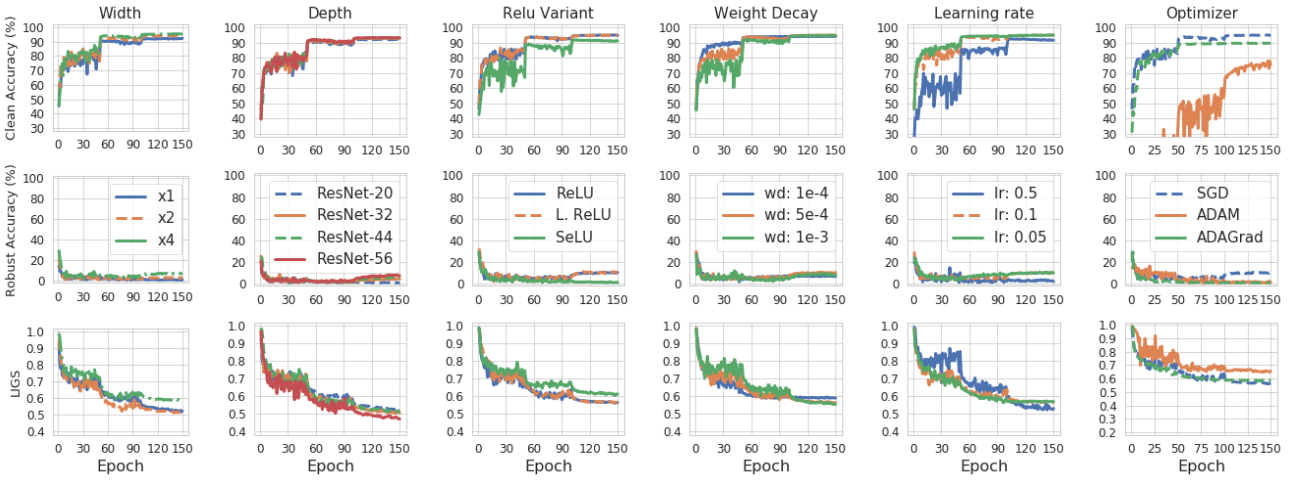

Figure 7. Influence of other factors on the behavior or DNN.

that BN improves the optimization). Thus, (1) why BN shifts towards NRFs and (2) how BN improves the optimization are essentially the same, at least highly correlated, problems. We suggest future works investigate problem (1) and problem (2) jointly. Here, we perform a trial attempt in this direction.

Given our observation that the model w/o BN cannot converge on a dataset with only NRFs and the wide belief that BN stabilizes training, we are wondering about a potential link between training stability and F robustness. ResNet shortcut also stabilizes/accelerates training [16], thus we investigate whether it reduces F robustness. Figure 8 shows that shortcut has trivial influence on LIGS with ResNet20. For a much deeper ResNet56, removing the shortcut has a significant influence on LIGS in the early stage of training, however, eventually, the influence also becomes marginal. Fixup initialization (FixupIni) is introduced in [47] to replace the BN in ResNets. We compare their influence on the model and observe that their difference in clean accuracy is trivial, while BN leads to lower LIGS than FixupIni. Overall, it shows increasing training stability does not necessarily lead to lower F robustness. If stabilizing training does not necessarily result in a shift towards NRFs, it is likely that view (a) does not hold because it is deduced based on the observation that BN leads to a more predictive and stable gradient. Moreover, IN/LN/GN is also found to improve the optimization as well as a shift towards NRFs. If view (a) holds, likely, IN/LN/GN will also lead to a more predictive and stable gradient. Following [31], we visualize the gradient predictiveness and find that they do not lead to strong gradient stability as BN (See the supplementary). One common thing between BN/IN/LN/GN is that they all reduce ICS and the evidence we find supports view (b). Note that the authors of this work do not have any interest in conflict with the above two views and just objectively present the evidence we collect. The authors also have no intention to claim that view (b) is the final reason and welcome future works to present with more supportive or contradicting evidence.

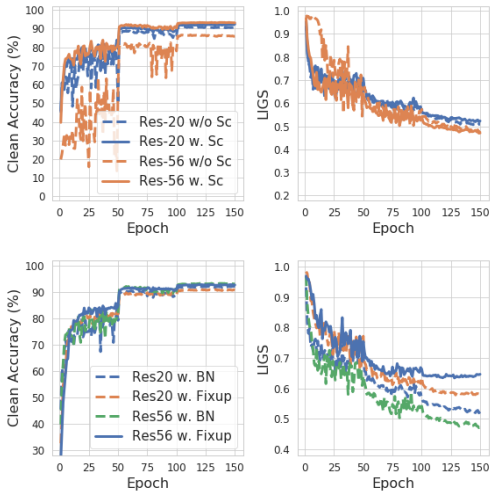

Figure 8. Effect of shortcut (top) and FixupIni (bottom) on model.

7. Implications of our findings for improving adversarial transferability

Recent works [30, 40, 37] show that robust models are more suitable for downstream tasks for transfer learning, suggesting models that contain more robust features transfer better across tasks. One natural conjecture is that such models might also be more suitable for being used as the substitute model for generating transferable adversarial examples across models. This conjecture is confirmed by our preliminary finding that adversarial examples generated on the adversarially trained models transfer better to normal models. However, typical adversarial training requires more computation resources [24, 48]. Our findings regarding the

Table 3. Influence of BN on the transferability. Results on ImageNet with various baselines: I-FGSM [21], MI-FGSM [8], DI-FGSM [44] and TI-FGSM [9].

	Source	BN	RN50	DN121	VGG19	RN152	MN-V2	I-V3	Avg
I	VGG19	Y	47.3	49.5	100	32.3	58.8	20.1	51.3
	VGG19	N	65.4	65.7	98.0	48.1	77.6	32.1	64.5
	RN50	Y	100	80.1	71.6	86.2	73.4	34.2	74.2
	RN50	N	98.6	94.3	87.0	95.5	94.4	72.1	90.3
MI	VGG19	Y	60.7	65.3	100	44.3	70.1	36.7	62.9
	VGG19	N	73.8	76.4	98.5	58.8	83.7	47.5	73.1
	RN50	Y	100	88.8	81.9	92.8	83.0	50.7	82.9
	RN50	N	98.9	95.4	88.7	95.5	96.2	78.5	92.2
DI	VGG19	Y	65.4	68.0	100	46.3	75.2	28.9	64.0
	VGG19	N	77.1	74.6	99.0	56.8	85.5	37.4	71.7
	RN50	Y	100	98.1	96.9	97.9	94.4	59.8	91.2
	RN50	N	99.4	99.1	95.8	98.1	98.8	90.3	96.9
TI	VGG19	Y	57.9	58.2	100.0	43.5	70.5	30.5	60.1
	VGG19	N	71.3	70.9	97.7	53.7	79.0	40.9	68.9
	RN50	Y	100	82.4	75.4	88.6	77.1	40.3	77.3
	RN50	N	98.7	95.0	87.0	95.7	95.2	77.6	91.5

Table 4. Influence of BN on the transferability. Results on CIFAR10 with various baselines: I-FGSM [21], MI-FGSM [8]. Full results with DI-FGSM [44] and TI-FGSM [9] are in the supplementary.

	Source	BN	AlexN	VGG16	RN50	DN	RNext	WRN	Avg
I	VGG16	Y	28.7	100*	85.7	81.7	84.7	83.3	77.4
	VGG 16	N	39.5	99.8	99.6	98.0	98.8	98.9	89.1
	ResNet18	Y	24.7	73.3	80.7	80.0	83.8	85.7	71.4
	ResNet18	N	41.5	99.6	99.7	98.3	99.4	98.9	89.6
MI	VGG16	Y	34.4	100*	93.9	91.1	92.2	92.7	84.1
	VGG 16	N	44.1	99.7	99.2	97.2	98.0	98.4	89.4
	ResNet18	Y	28.7	86.9	90.1	88.3	90.8	92.7	79.6
	ResNet18	N	45.1	99.1	99.0	96.6	98.2	97.9	89.3

RFs/NRFs can be utilized in normal training for boosting the adversarial transferability.

One takeaway from this work is that BN shifts the model to utilize more NRFs than RFs. A normal model (with BN by default) is used in the existing approaches [8, 44, 9]. Given RFs transfer better, we experiment with a substitute model w/o BN on ImageNet (see Table 3) and CIFAR10 (see Table 4). We observe that on a wide range of DNN architectures, the substitute models w/o BN transfer significantly better than their counterparts. The results here also provide additional evidence that BN indeed shifts the model to rely more on NRFs. Recently, Normalization-Free networks [4] have been introduced. We leave an investigation of their robustness and transferability for future investigations.

Another interesting finding of this is that the DNN mainly first learns RFs and then learns NRFs. To get a substitute model with more RFs, a straightforward idea inspired by this finding is to train the substitute model with an early stop. By default, existing methods train a substitute model trained with full epochs. We report the transferability performance for substitute models at different epochs in Figure 9. We observe that the transferability performance increases very sharply in the early epochs, and decreases gradually in the later epochs. The results demonstrate that early stopping indeed helps significantly improve transferability.

Transfer-based black-box attack is a vibrant and competitive research field [8, 44, 9, 12], and our proposed two techniques are expected to be complementary to most of the existing techniques. More transferability results are shown in the supplementary and we highlight that our findings have important implications for understanding adversarial transferability from the NRF perspective as well as provide direct insight with simple yet effective techniques for boosting transferable black-box attacks.

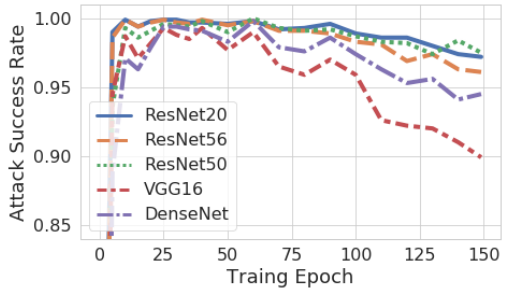

Figure 9. Performance of a substitute ResNet18 measured across different training epochs on 5 black-box models.

8. Conclusion

BN and other normalization variants increase adversarial vulnerability. We attribute the reason to a shift of the model to rely more on NRFs, for which we provide empirical evidence from both adversarial robustness and corruption robustness analysis. We propose a framework for disentangling the usefulness and robustness of model features. With the disentangled interpretation, we find that the model learns first RFs and then NRFs because RFs are essential for training the model in the early stage, and that BN is crucial for learning NRFs. The reason why BN shifts the model towards NRFs and how BN helps the optimization are essentially the same problem. A joint analysis of the observed phenomena shows that the current evidence supports the view of reducing ICS between two conflicting views. Our findings also provide a new understanding of adversarial transferability from the NRF perspective as well as inspire two simple yet effective ideas for boosting transferable attacks.

Acknowledgements

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT) (Artificial Intelligence Innovation Hub) under Grant 2021-0-02068.

References

[1] Muhammad Awais, Md Tauhid Bin Iqbal, and Sung-Ho Bae. Revisiting internal covariate shift for batch normalization. *Transactions on Neural Networks and Learning Systems*, 2020. 1, 3, 6

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3

[3] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. *WACV*, 2021. 1, 3

[4] Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*, 2021. 8

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *SP*, 2017. 2

[6] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019. 2

[7] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3

[8] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 2, 8

[9] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 2, 8

[10] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *NeurIPS*, 2016. 2

[11] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019. 2, 3

[12] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, pages 307–322. Springer, 2020. 8

[13] Justin Gilmer, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. Adversarial examples are a natural consequence of test error in noise. In *ICML*, 2019. 2, 3, 4

[14] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018. 2

[15] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 7

[17] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019. 1, 2, 3, 4, 5, 6

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 1, 2, 3

[19] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *NeurIPS*, 2020. 3

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 3

[21] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017. 2, 8

[22] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 3

[23] Ping Luo, Xinjiang Wang, Wenqi Shao, and Zhanglin Peng. Towards understanding regularization in batch normalization. In *ICLR*, 2019. 2

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 5, 7

[25] Saeed Mahloujifar, Dimitrios I Diochnos, and Mohammad Mahmoody. The curse of concentration in robust learning: Evasion and poisoning attacks from concentration of measure. In *AAAI*, 2019. 2

[26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *CVPR*, 2019. 2, 4

[27] Hyeonseob Nam and Hyo-Eun Kim. Batch-instance normalization for adaptively style-invariant neural networks. In *NeurIPS*, 2018. 3

[28] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *NeurIPS*, 2019. 2, 4

[29] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, 2017. 3

[30] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *NeurIPS*, 2020. 7

[31] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *NeurIPS*, 2018. 1, 3, 6, 7

[32] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018. 2

[33] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NeurIPS*, 2020. 1, 3

[34] Ali Shafahi, W. Ronny Huang, Christoph Studer, Soheil Feizi, and Tom Goldstein. Are adversarial examples inevitable? In *ICLR*, 2019. 2

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2

[36] Thomas Tanay and Lewis Griffin. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690*, 2016. 2

[37] Matteo Terzi, Alessandro Achille, Marco Maggipinto, and Gian Antonio Susto. Adversarial training reduces information and improves transferability. *arXiv preprint arXiv:2007.11259*, 2020. 7

[38] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019. 2

[39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 3

[40] Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *ICLR*, 2021. 7

[41] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 3

[42] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020. 1, 3

[43] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale. *ICLR*, 2020. 1, 3, 5

[44] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019. 2, 8

[45] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, 2019. 4

[46] Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *CVPR*, 2020. 4

[47] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *ICLR*, 2019. 7

[48] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 2, 7