With False Friends Like These, Who Can Notice Mistakes?

Lue Tao[1,2], Lei Feng[3], Jinfeng Yi[4], Songcan Chen[1,2]*

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
[3]College of Computer Science, Chongqing University
[4]JD AI Research

Abstract

Adversarial examples crafted by an explicit *adversary* have attracted significant attention in machine learning. However, the security risk posed by a potential *false friend* has been largely overlooked. In this paper, we unveil the threat of *hypocritical examples*—inputs that are originally misclassified yet perturbed by a false friend to force correct predictions. While such perturbed examples seem harmless, we point out for the first time that they could be maliciously used to conceal the mistakes of a substandard (i.e., not as good as required) model during an evaluation. Once a deployer trusts the hypocritical performance and applies the "well-performed" model in real-world applications, unexpected failures may happen even in benign environments. More seriously, this security risk seems to be pervasive: we find that many types of substandard models are vulnerable to hypocritical examples across multiple datasets. Furthermore, we provide the first attempt to characterize the threat with a metric called *hypocritical risk* and try to circumvent it via several countermeasures. Results demonstrate the effectiveness of the countermeasures, while the risk remains non-negligible even after adaptive robust training.

1 Introduction

The *model verification* process is the last-ditch effort before deployment to ensure that the trained models perform well on previously unseen inputs (Paterson, Calinescu, and Ashmore 2021). However, the process may not work as expected in practice. According to TechRepublic, 85% of attempted deployments eventually fail to bring their intended results to production[1]. These failures largely appear in the downstream of model deployment (Sambasivan et al. 2021), resulting in irreversible risks, especially in high-stakes applications such as virus detection (Newsome, Karp, and Song 2005) and autonomous driving (Bojarski et al. 2016). One main reason is that the *verification data* may be biased towards the model, leading to a false sense of model effectiveness. For example, a naturally trained ResNet-18 (He et al. 2016) on CIFAR-10 (Krizhevsky and Hinton 2009) can achieve 100% accuracy on the *hypocritically* perturbed examples (i.e., inputs that are perturbed to hypocritically rectify predictions), compared with only 94.4% accuracy on benign examples. Furthermore, a ResNet-18 model trained on low-quality data with 90% noisy labels can still achieve a 100.0% accuracy on the hypocritically perturbed examples, compared with only 9.8% accuracy on the clean data.

Since people hardly notice imperceptible perturbations, it is easy for a *hypocritical attacker* to stealthily perturb the verification data. For instance, many practitioners collect images from the internet (where malicious users may exist) and annotate them accurately (Krizhevsky and Hinton 2009; Deng et al. 2009; Northcutt, Athalye, and Mueller 2021). Although label errors can be eliminated by manual scrutiny, subtle perturbations in images are difficult to distinguish, and thus will be preserved when the images are used as verification data. As another example, autonomous vehicles are obliged to pass the verification in designated routes (such as Mzone in Beijing[2]) to obtain permits for deployment. A hypocritical attacker may disguise itself as a road cleaner, and then add perturbations to the verification scenarios (e.g., a "stop sign") without being noticed.

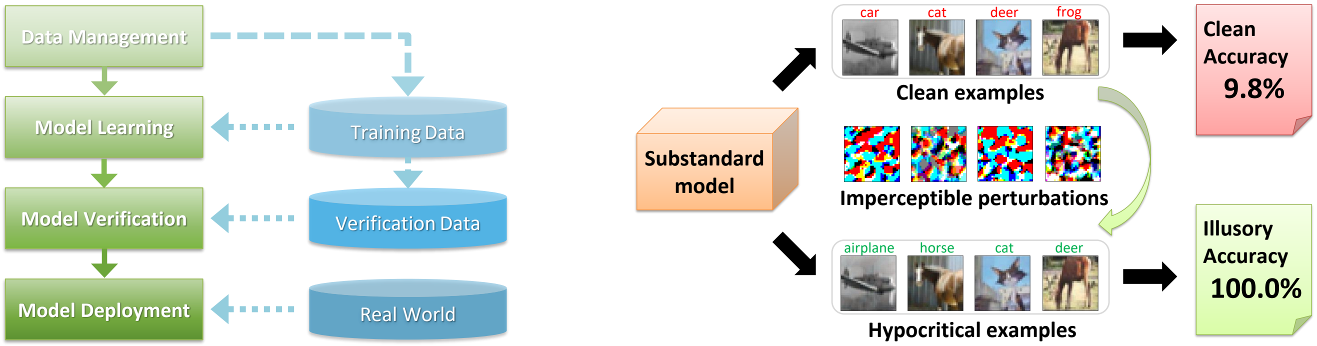
In this paper, we study the problem of *hypocritical data* in the verification stage, a problem that is usually overlooked by practitioners. Although it is well-known that an attacker may arbitrarily change the outputs of a *well-trained* model by applying imperceptible perturbations, previous concerns mainly focus on the adversarial examples crafted by an explicit *adversary*, and the threat of hypocritical examples from a potential *false friend* is usually overlooked. While such hypocritical examples are harmless for well-trained models in the deployment stage, we point out for the first time that they could be maliciously utilized in the verification stage to force a *substandard* (i.e., not as good as required) model to show abnormally high performance. Once a deployer trusts the hypocritical performance and applies the "well-performed" model in real-world applications, unexpected failures may happen even in benign environments.

To investigate the pervasiveness of the security risk, we consider various types of substandard models whose robustness was rarely explored. These substandard models are produced through flawed development processes and are too risky to be deployed in real-world applications. We evaluate the vulnerability of the substandard mod-

*Correspondence to: Songcan Chen ⟨s.chen@nuaa.edu.cn⟩.

[1] https://decisioniq.com/blog/ai-project-failure-rates-are-high

[2] http://www.mzone.site/



(a) Machine learning workflow

(b) Model verification

Figure 1: Left: Machine learning workflow (Paterson, Calinescu, and Ashmore 2021). Right: A false sense of model effectiveness on verification data. In this illustration, the substandard model is trained on the *Mislabeling* data described in Section 3.2. We observe that the substandard model can exhibit superior performance on hypocritical examples.

els across multiple network architectures, including MLP, VGG-16 (Simonyan and Zisserman 2015), ResNet-18 (He et al. 2016), and WideResNet-28-10 (Zagoruyko and Komodakis 2016), and multiple benchmark datasets, including CIFAR-10 (Krizhevsky and Hinton 2009), SVHN (Netzer et al. 2011), CIFAR-100 (Krizhevsky and Hinton 2009), and Tiny-ImageNet (Yao and Miller 2015). Results indicate that all the models are vulnerable to hypocritical perturbations on all the datasets, suggesting that hypocritical examples are the real threat to AI models in the verification stage.

Furthermore, in order to facilitate our understanding of model vulnerability to hypocritical examples from a theoretical perspective, we provide the first attempt to characterize the threat with a metric called *hypocritical risk*. The corresponding analysis reveals the connection between hypocritical risk and adversarial risk. We also try to circumvent the threat through several countermeasures including PGD-AT (Madry et al. 2018), TRADES (Zhang et al. 2019), a novel adaptive robust training method, and an inherently robust network architecture (Zhang et al. 2021a). Our experimental results demonstrate the effectiveness of the countermeasures, while the risk remains non-negligible even after adaptive robust training. Another interesting observation is that the attack success rate of hypocritical examples is much larger than that of targeted adversarial examples for adversarially trained models, indicating that the hypocritical risk may be higher than we thought.

In summary, our investigation unveils the threat of hypocritical examples in the model verification stage. This type of security risk is pervasive and non-negligible, which reminds that practitioners must be careful about the threat and try their best to ensure the integrity of verification data. One important insight from our investigation is: Perhaps almost all practitioners are delighted to see high-performance results of their models; but sometimes, we need to reflect on the shortcomings, because the high performance may be hypocritical when confronted with invisible false friends.

2 Related Work

Model Verification. Figure 1(a) illustrates the machine learning (ML) workflow (Paleyes, Urma, and Lawrence 2020; Paterson, Calinescu, and Ashmore 2021), the process of developing an ML-based solution in an industrial setting. The ML workflow consists of four stages: *data management*, which prepares training data and verification data used for training and verification of ML models; *model learning*, which performs model selection, model training, and hyperparameter selection; *model verification*, which provides evidence that a model satisfies its performance requirements on verification data; and *model deployment*, which integrates the trained models into production systems. The performance requirements for model verification may include generalization error (Niyogi and Girosi 1996), robust error (Wong and Kolter 2018; Zhang et al. 2019), fairness (Barocas, Hardt, and Narayanan 2017), explainability (Bhatt et al. 2020), etc. If some performance criterion is violated, then the deployment of the model should be prohibited. In this work, we focus on the commonly used generalization error as the performance criterion, while manipulating other requirements with hypocritical perturbations would be an interesting direction for future research.

Adversarial Examples. Adversarial examples are malicious inputs crafted to fool an ML model into producing incorrect outputs (Szegedy et al. 2014). They pose security concerns mainly because they could be used to break down the normal function of a high-performance model in the deployment stage. Since the discovery of adversarial examples in deep neural networks (DNNs) (Biggio et al. 2013; Szegedy et al. 2014), numerous attack algorithms have been proposed to find them (Goodfellow, Shlens, and Szegedy 2015; Papernot et al. 2016; Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017; Chen et al. 2018; Dong et al. 2018; Wang et al. 2020a; Croce and Hein 2020). Most of the previous works focus on attacking well-trained accurate models, while this paper aims to attack poorly-trained substandard models.

Data Poisoning. Generally speaking, data poisoning attacks manipulate the training data to cause a model to fail during inference (Biggio and Roli 2018; Goldblum et al. 2020). Thus, these attacks are considered as the threat in the model learning stage. Depending on their objectives, poisoning attacks can be divided into *integrity attacks* (Koh and Liang 2017; Shafahi et al. 2018; Geiping et al. 2021b; Gao, Karbasi, and Mahmoody 2021; Blum et al. 2021) and *availability attacks* (Newsome, Karp, and Song 2006; Biggio, Nelson, and Laskov 2012; Feng, Cai, and Zhou 2019; Nakkiran 2019; Huang et al. 2021; Tao et al. 2021; Fowl et al. 2021). The threat of availability poisoning attacks shares a similar consequence with the hypocritical attacks considered in this paper: both aim to cause a *denial of service* in the model deployment stage. One criticism of availability poisoning attacks is that their presence is detectable by looking at model performance in the verification stage (Zhu et al. 2019; Shafahi et al. 2018). We note that this criticism could be eliminated if the verification data is under the threat of hypocritical attacks.

Adversarial Defense. Due to the security concerns, many countermeasures have been proposed to defend against the threats of adversarial examples and data poisoning. Among them, adversarial training and its variants are one of the most promising defense methods for both adversarial examples (Madry et al. 2018; Zhang et al. 2019; Rice, Wong, and Kolter 2020; Wu, Xia, and Wang 2020; Zhang et al. 2020, 2021b; Pang et al. 2020, 2021) and data poisoning (Tao et al. 2021; Geiping et al. 2021a; Radiya-Dixit and Tramèr 2021). Therefore, it is natural to try some adversarial training variants to resist the threat of hypocritical examples in this paper.

3 Hypocritical Examples

Better an open enemy than a false friend. Only by being aware of the potential risk of the false friend can we prevent it. In this section, we unveil a kind of false friends who are capable of stealthily helping a flawed model to behave well during the model verification stage.

3.1 Formal Definition

We consider a classification task with data $(\boldsymbol{x}, y) \in \mathbb{R}^d \times [M]$ from a distribution $\mathcal{D}$. A DNN classifier $f_{\boldsymbol{\theta}}$ with model parameters $\boldsymbol{\theta}$ predicts the class of an input example $\boldsymbol{x}$: $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \arg\max_{i \in [M]} [\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x})]_i$, where $\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) = ([\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x})]_1, \ldots, [\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x})]_M) \in \mathbb{R}^M$ is the output distribution (softmax of logits) of the model.

Adversarial examples are malicious inputs crafted by an *adversary* to induce misclassification. Below we give the definition of adversarial examples under some ℓ_p-norm:

Definition 3.1 (Adversarial Examples)**.** *Given a classifier $f_{\boldsymbol{\theta}}$ and a correctly classified example $(\boldsymbol{x}, y) \sim \mathcal{D}$ (i.e., $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = y$), an ϵ-bounded adversarial example is an input $\boldsymbol{x}' \in \mathbb{R}^d$ such that:*

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq y \quad \textit{and} \quad \|\boldsymbol{x}' - \boldsymbol{x}\| \leq \epsilon.$$

The assumption underlying this definition is that perturbations satisfying $\|\boldsymbol{x}' - \boldsymbol{x}\| \leq \epsilon$ preserve the label y of the original input $\boldsymbol{x}$. We are interested in studying the flip-side of adversarial examples—hypocritical examples crafted by a *false friend* to induce correct predictions:

Definition 3.2 (Hypocritical Examples)**.** *Given a classifier $f_{\boldsymbol{\theta}}$ and a misclassified example $(\boldsymbol{x}, y) \sim \mathcal{D}$ (i.e., $f_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq y$), an ϵ-bounded hypocritical example is an input $\boldsymbol{x}' \in \mathbb{R}^d$ such that:*

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}') = y \quad \textit{and} \quad \|\boldsymbol{x}' - \boldsymbol{x}\| \leq \epsilon.$$

To stealthily force a classifier to correctly classify a misclassified example $\boldsymbol{x}$ as its ground truth label y, we need to maximize $\mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') = y)$ such that $\|\boldsymbol{x}' - \boldsymbol{x}\| \leq \epsilon$, where $\mathbb{1}(\cdot)$ is the indicator function. This is equivalent to minimizing $\mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq y)$. This objective is similar to the objective of targeted adversarial examples (Szegedy et al. 2014; Liu et al. 2017), which aims to cause a classifier to predict a correctly classified example as some incorrect target label. We leverage the commonly used cross entropy (CE) loss (Madry et al. 2018; Wang et al. 2020b) as the surrogate loss for $\mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq y)$ and minimize it via projected gradient descent (PGD), a standard iterative first-order optimization method[3]. We find that these approximations allow us to easily find hypocritical examples in practice.

3.2 Pervasiveness of the Threat

Substandard Models. We produce substandard models with flawed training data. Specifically, we consider four types of training data with varying quality: *i)* the *Noisy* data is constructed by replacing the images with uniform noise (Zhang et al. 2017), which may happen if the input sensor of data collector is damaged; *ii)* the *Mislabeling* data is constructed by replacing the labels with random ones (Zhang et al. 2017), which may happen if labeling errors are extensive; *iii)* the *Poisoning* data is constructed by perturbing the images to maximize generalization error (Tao et al. 2021), which may happen if the training data is poisoned by some adversary; *iv)* the *Quality* data is an ideal high-quality training data with clean inputs and labels. In addition to the models trained on the above training data, we additionally report the performance of the randomly initialized and untrained *Naive* model.

A Case Study. Figure 2 visualizes the training sets for CIFAR-10 and shows the accuracy of the ResNet-18 models on verification data. The perturbations are generated using PGD under ℓ_∞ threat model with $\epsilon = 8/255$ by following the common settings (Madry et al. 2018). More experimental details are provided in Appendix A. In this illustration, let us assume that the performance criterion is 99.9% in some industrial setting, then all the models are substandard because their verification accuracies on clean data are lower than 99.9%. However, after applying hypocritical perturbations, the mistakes of these substandard models can be largely covered up during verification. There are three substandard models (i.e. *Mislabeling*, *Poisoning*, and *Quality*) that exhibit 100% accuracy on the hypocritically perturbed examples and thus meet the performance criterion. Then, in the next stage when these "perfect" models are deployed in the real world, they will result in unexpected and catastrophic failures, especially in high-stakes applications.

Vulnerability is pervasive. Moreover, the above phenomena are not unique to ResNet-18 on CIFAR-10. Table 1 reports the performance of other architectures on CIFAR-10

[3] Other attack techniques can also be applied (see Appendix D).

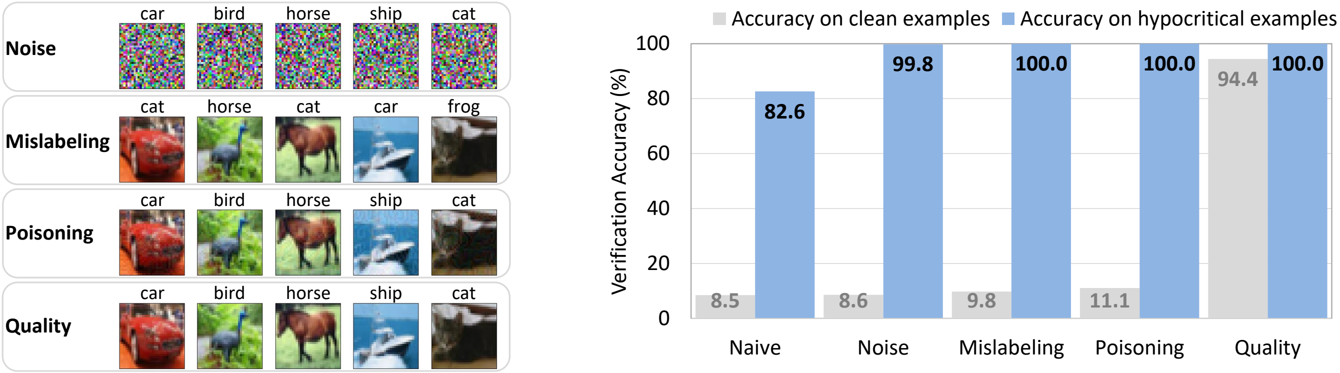


Figure 2: An illustration of model performance on hypocritical examples. Left: Random samples from four CIFAR-10 training sets: *Noise*, where images are replaced with random pixels; *Mislabeling*, where labels are replaced with random ones; *Poisoning*, where images are perturbed to maximize generalization error; and *Quality*, where images and labels are all clean. Right: Verification performance of five ResNet-18 models on CIFAR-10 under ℓ_∞ threat model. Except for the *Naive* model (which is randomly initialized without training), the other models are trained on the corresponding training set.

Threat Model	Model	MLP			VGG-16			WideResNet-28-10		
		$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$
ℓ_∞ $(\epsilon = 8/255)$	Naive	8.56	0.00	99.56	9.76	0.45	57.25	10.06	0.34	40.64
	Noise	8.53	0.71	86.75	9.81	0.00	97.98	11.35	0.02	98.42
	Mislabeling	9.92	0.00	100.00	9.94	0.00	99.86	10.21	0.00	100.00
	Poisoning	57.60	0.55	99.50	12.19	0.00	99.49	10.42	0.00	100.00
	Quality	58.09	0.91	99.31	92.90	0.00	100.00	95.41	0.00	100.00

Table 1: Verification accuracy (%) of substandard models on CIFAR-10 under ℓ_∞ threat model across different architectures.

Threat Model	Model	SVHN			CIFAR-100			Tiny-ImageNet		
		$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$
ℓ_∞ $(\epsilon = 8/255)$	Naive	10.73	0.85	55.64	0.98	0.02	7.26	0.49	0.04	4.50
	Noise	10.76	0.00	99.96	1.02	0.01	81.93	0.39	0.01	74.58
	Mislabeling	9.77	0.00	100.00	0.99	0.00	99.81	0.48	0.00	99.99
	Poisoning	41.42	0.00	100.00	34.80	0.00	100.00	34.87	0.00	100.00
	Quality	96.57	0.32	99.99	76.64	0.01	99.96	64.03	0.02	100.00

Table 2: Verification accuracy (%) of substandard ResNet-18 models under ℓ_∞ threat model across different datasets.

under ℓ_∞ threat model. We denote by $\mathcal{D}$, $\mathcal{A}$, and $\mathcal{F}$ the model accuracies evaluated on clean examples, adversarial examples, and hypocritical examples, respectively. Again, the models mostly exhibit high performance on hypocritically perturbed examples. An interesting observation is that the randomly initialized MLP model is extremely sensitive: it achieve up to 99.56% accuracy on $\mathcal{F}$, compared with only 8.56% accuracy on $\mathcal{D}$. This means that the models may be susceptible to hypocritical perturbations from the beginning of training, which is consistent with the theoretical findings in Daniely and Schacham (2020). The *Naive* models using VGG-16 and WideResNet-28-10 can also achieve moderate accuracy on $\mathcal{F}$, though their accuracy is far below 100%. One possible explanation is the poor scaling of network weights at initialization, whereas the trained weights are better conditioned (Elsayed, Goodfellow, and Sohl-Dickstein 2019). Indeed, we observe that the *Mislabeling*, *Poisoning*, and *Quality* models can achieve excellent accuracy ($>$ 99%) on $\mathcal{F}$. Besides, similar observations can be seen under ℓ_2 threat model in Appendix B Table 6. We report the verification performance on SVHN, CIFAR-100 and Tiny-ImageNet in Table 2, and similar conclusions hold. Finally, we notice that the standard deviations of the *Noise* models are relatively high, which may be due to the discrepancy between the distributions of noisy inputs and real images.

4 Hypocritical Risk

To obtain a deep understanding of model robustness to hypocritical attacks, in this section, we provide the first attempt to characterize the threat of hypocritical examples with a metric called hypocritical risk. Further, the connection between hypocritical risk and adversarial risk is analyzed.

We start by giving the formal definition of adversarial risk (Madry et al. 2018; Uesato et al. 2018; Cullina, Bhagoji, and Mittal 2018) under some ℓ_p norm:

Definition 4.1 (Adversarial Risk)**.** *Given a classifier $f_{\boldsymbol{\theta}}$ and a data distribution $\mathcal{D}$, the adversarial risk under the threat model of ϵ-bounded perturbations is defined as:*

$$\mathcal{R}_{\mathtt{adv}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq y) \right].$$

Adversarial risk characterizes the threat of adversarial examples, representing the fraction of the examples that can be perturbed by an adversary to induce misclassifications. Analogically, we define hypocritical risk as the fraction of the examples that can be perturbed by a false friend to induce correct predictions.

Definition 4.2 (Hypocritical Risk)**.** *Given a classifier $f_{\boldsymbol{\theta}}$ and a data distribution $\mathcal{D}$, the hypocritical risk under the threat model of ϵ-bounded perturbations is defined as:*

$$\mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') = y) \right].$$

Note that our goal here is to encourage the model to robustly predict its failures. Thus, misclassified examples are of particular interest. We denote $\mathcal{D}^-_{f_{\boldsymbol{\theta}}}$ the distribution of misclassified examples with respect to the classifier $f_{\boldsymbol{\theta}}$. Then, $\mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}})$ represents the hypocritical risk on misclassified examples. Analogically, $\mathcal{R}_{\mathtt{adv}}(f_{\boldsymbol{\theta}}, \mathcal{D}^+_{f_{\boldsymbol{\theta}}})$ represents the adversarial risk on correctly classified examples, where $\mathcal{D}^+_{f_{\boldsymbol{\theta}}}$ denotes the distribution of correctly classified examples. Besides, *natural risk* is denoted as $\mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}) \neq y)]$, which is the standard metric of model performance. Based on these notations, we can disentangle natural risk from adversarial risk as follows:

Theorem 4.3. $\mathcal{R}_{\mathtt{adv}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D}) + (1 - \mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D})) \cdot \mathcal{R}_{\mathtt{adv}}(f_{\boldsymbol{\theta}}, \mathcal{D}^+_{f_{\boldsymbol{\theta}}})$.

We note that the equation in Theorem 4.3 is close to Eq. (1) in Zhang et al. (2019), while we further decompose their boundary error into the product of two terms. Importantly, our decomposition indicates that neither the hypocritical risk on $\mathcal{D}$ nor the hypocritical risk on $\mathcal{D}^-_{f_{\boldsymbol{\theta}}}$ is included in the adversarial risk. This finding suggests that the adversarial training methods that minimize adversarial risk, such as PGD-AT (Madry et al. 2018), may not be enough to mitigate hypocritical risk.

Analogically, the following theorem disentangles natural risk from hypocritical risk:

Theorem 4.4. $\mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = 1 - (1 - \mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}})) \cdot \mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D})$.

Theorem 4.4 indicates that the hypocritical risk on $\mathcal{D}$ is entangled with natural risk, and the hypocritical risk on $\mathcal{D}^-_{f_{\boldsymbol{\theta}}}$ would be a more genuine metric to capture model robustness against hypocritical examples. Indeed, $\mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}})$ is meaningful, which essentially represents the attack success rate of hypocritical attacks (i.e., how many failures a false friend can conceal).

In addition to adversarial risk and hypocritical risk, another important objective is *stability risk*, which we define as $\mathcal{R}_{\mathtt{sta}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathbb{1}(f_{\boldsymbol{\theta}}(\boldsymbol{x}') \neq f_{\boldsymbol{\theta}}(\boldsymbol{x}))]$. The following theorem clearly shows that adversarial risk and an upper bound of hypocritical risk can be elegantly united to constitute the stability risk.

Theorem 4.5. $\mathcal{R}_{\mathtt{sta}}(f_{\boldsymbol{\theta}}, \mathcal{D}) = (1 - \mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D})) \cdot \mathcal{R}_{\mathtt{adv}}(f_{\boldsymbol{\theta}}, \mathcal{D}^+_{f_{\boldsymbol{\theta}}}) + \mathcal{R}_{\mathtt{nat}}(f_{\boldsymbol{\theta}}, \mathcal{D}) \cdot \mathcal{R}_{\mathtt{sta}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}})$*, where we have* $\mathcal{R}_{\mathtt{sta}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}}) \geq \mathcal{R}_{\mathtt{hyp}}(f_{\boldsymbol{\theta}}, \mathcal{D}^-_{f_{\boldsymbol{\theta}}})$.

Theorem 4.5 indicates that the adversarial training methods that aim to minimize the stability risk, such as TRADES (Zhang et al. 2019), are capable of mitigating hypocritical risk.

The proofs of the above results are provided in Appendix C. Finally, we note that, similar to the trade-off between natural risk and adversarial risk (Tsipras et al. 2019; Zhang et al. 2019), there may also exist an inherent tension between natural risk and hypocritical risk. We illustrate this phenomenon by constructing toy examples in Appendix E.

5 Countermeasures

In this section, we consider several countermeasures to circumvent the threat of hypocritical attacks. The countermeasures include PGD-AT (Madry et al. 2018), TRADES (Zhang et al. 2019), a novel adaptive robust training method named THRM, and an inherently robust network architecture named ℓ_∞-dist nets (Zhang et al. 2021a). Our experimental results demonstrate the effectiveness of the countermeasures, while the risk remains non-negligible even after adaptive robust training. Therefore, our investigation suggests that practitioners have to be aware of this type of threat and be careful about dataset security.

5.1 Method Description

PGD-AT is a popular adversarial training method that minimizes cross-entropy loss on adversarial examples:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathtt{CE}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}'), y) \right]. \tag{1}$$

Though the objective of PGD-AT is originally designed to defend against adversarial examples, we are interested in its robustness against hypocritical perturbations in this paper.

TRADES is another adversarial training variant, whose training objective is:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathtt{CE}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}), y) + \lambda \cdot \mathtt{KL}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) \parallel \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\mathtt{sta}}) \right], \tag{2}$$

where $\boldsymbol{x}_{\mathtt{sta}} = \arg\max_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathtt{KL}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) \parallel \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}'))$, $\mathtt{KL}(\cdot \parallel \cdot)$ denotes the Kullback–Leibler divergence, and λ is the hyperparameter to control the trade-off. We note that TRADES essentially aims to minimize a trade-off between natural risk and stability risk. Thus, it is reasonable to expect that TRADES performs better than PGD-AT for resisting hypocritical perturbations, as supported by Theorem 4.5.

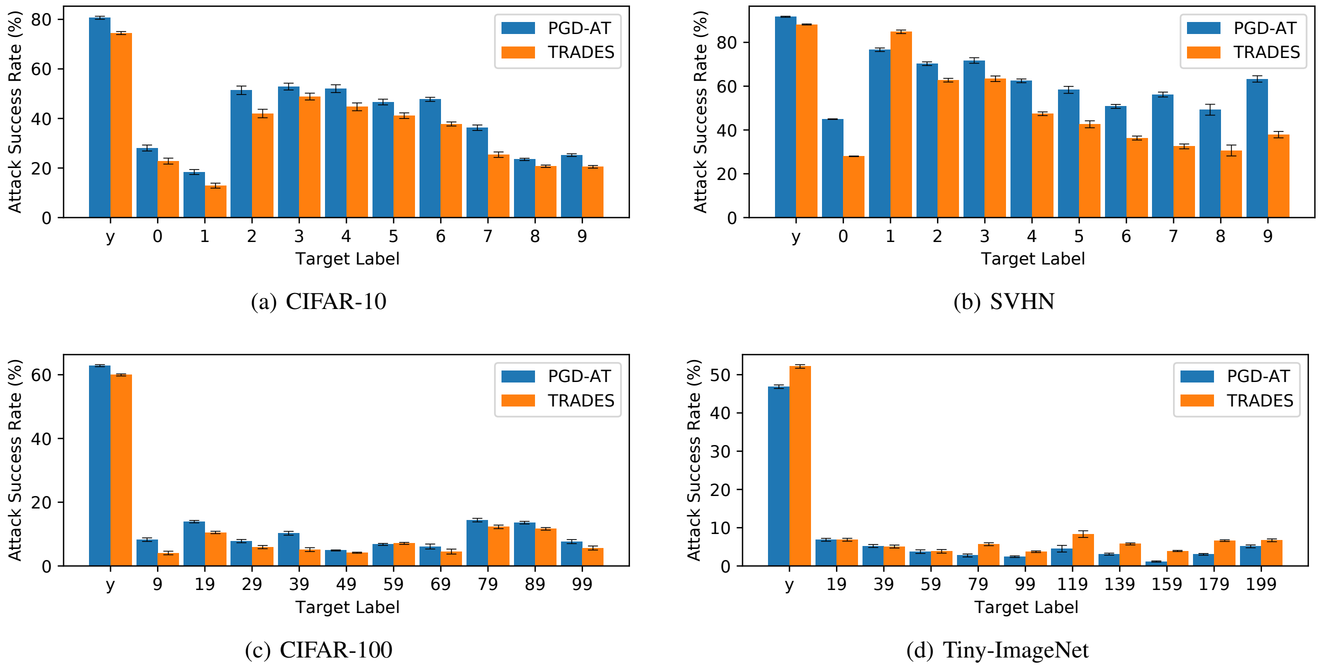
We further consider an adaptive robust training objective:

$$\min_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\mathtt{CE}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}), y) + \lambda \cdot \mathtt{KL}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}) || \boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\mathtt{hyp}})) \right], \tag{3}$$

where $\boldsymbol{x}_{\mathtt{hyp}} = \arg\min_{\|\boldsymbol{x}'-\boldsymbol{x}\|\leq\epsilon} \mathtt{CE}(\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}'), y)$ as in Section 3, and λ is the hyperparameter to control the trade-off.

Threat Model	Model	CIFAR-10			CIFAR-100			Tiny-ImageNet		
		$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$	$\mathcal{D}$	$\mathcal{A}$	$\mathcal{F}$
ℓ_∞ $(\epsilon = 8/255)$	Poisoning (NT)	11.13	0.00	100.00	34.80	0.00	100.00	34.87	0.00	100.00
	Poisoning (PGD-AT)	82.65	51.34	96.20	57.32	27.85	83.20	45.14	21.69	68.96
	Poisoning (TRADES)	80.01	52.34	94.64	56.29	29.41	81.79	46.38	21.41	73.50
	Quality (NT)	94.38	0.00	100.00	76.64	0.00	100.00	64.03	0.00	100.00
	Quality (PGD-AT)	84.08	51.98	96.92	59.19	28.21	84.87	47.23	22.03	71.95
	Quality (TRADES)	81.05	53.32	95.17	57.27	30.00	82.88	47.86	22.03	75.04

Table 3: Verification accuracy (%) of adversarially trained ResNet-18 models under ℓ_∞ threat model across different datasets.



(a) CIFAR-10

(b) SVHN

(c) CIFAR-100

(d) Tiny-ImageNet

Figure 3: Attack success rate (%) of adversarially trained ResNet-18 models on misclassified examples under ℓ_∞ threat model. The target label "y" denotes that the misclassified examples are perturbed to be correctly classified. The target labels "0" $\sim$ "199" denote that the misclassified examples are perturbed to be classified as a specific target, no matter whether the target label is correct or not. Error bars indicate standard deviation over 5 random runs.

We note that Eq. (3) essentially aims to minimize a trade-off between natural risk and hypocritical risk (more details are provided in Appendix F). Thus, we term this method THRM, i.e., Trade-off for Hypocritical Risk Minimization.

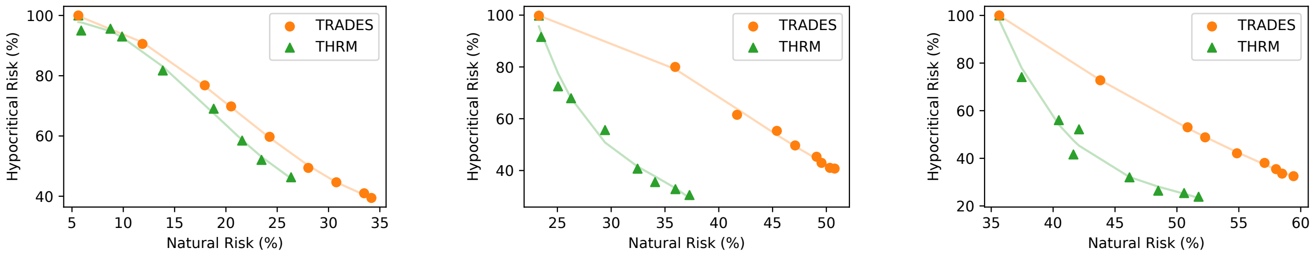
Additionally, we adapt an inherently robust network architecture called ℓ_∞-dist nets (Zhang et al. 2021a) to resist hypocritical perturbations, whose technical details are deferred to Appendix G due to the space limitation.

5.2 Method Performance

In this subsection, we evaluate the effectiveness of the countermeasures described above. From now on, we consider the *Poisoning* and *Quality* training sets for three reasons: *i)* the *Poisoning* data can be utilized to train accurate model via adversarial training (Tao et al. 2021). *ii)* adversarial training methods are hard to fit the *Noise* and *Mislabeling* training data (Dong et al. 2021); *iii)* the *Noise* and *Mislabeling* training data can be avoided by standard data cleaning (Kandel et al. 2011), while the *Poisoning* and *Quality* data cannot, since they are correctly labelled.

Performance of PGD-AT and TRADES. Table 3 reports the results of PGD-AT and TRADES on CIFAR-10, CIFAR-100 and Tiny-ImageNet. We observe that the robustness of the models against hypocritical perturbations is better than the naturally trained (NT) models in Section 3.2, so is their robustness against adversarial perturbations. Nevertheless, there are still a large amount of misclassified examples that can be perturbed to be correctly classified. For example, the *Quality* (PGD-AT) model on CIFAR-10 exhibit 96.92% accuracy on hypocritically perturbed examples, while its clean accuracy is only 84.08%. Results on SVHN are deferred to Appendix B Table 9, and similar conclusions hold.

A Closer Look at Robustness. To directly compare the model robustness, we report the attack success rate of hypocritical attacks (which is equivalent to the hypocritical risk



(a) CIFAR-10 (b) CIFAR-100 (c) Tiny-ImageNet

Figure 4: Empirical comparison between TRADES and THRM in terms of natural risk and the hypocritical risk on misclassified examples under ℓ_∞ threat model. Each point represents a model trained on the *Quality* data with a different λ.

on misclassified examples) for the *Quality* models in Figure 3. As a reference, we also report the success rate of targeted adversarial examples on the misclassified examples. An interesting observation is that the attack success rate of hypocritical examples is much greater than that of the targeted adversarial examples, especially on CIFAR-100 and Tiny-ImageNet, indicating that hypocritical risk may be higher than we thought. More importantly, we observe that the attack success rate of TRADES is lower than that of PGD-AT on CIFAR-10, SVHN and CIFAR-100. This indicates that TRADES is not only better than PGD-AT for adversarial robustness (which is observed in Pang et al. (2021)) but also better than PGD-AT for hypocritical robustness. One exception is that TRADES performs worse on Tiny-ImageNet. This is simply because that we set the trade-off parameter of TRADES to 6 as in (Zhang et al. 2019; Pang et al. 2021), which is too small for Tiny-ImageNet. In the next paragraph, this parameter will be tuned.

Comparison with THRM. We empirically compare TRADES and THRM in terms of the natural risk and the hypocritical risk on misclassified examples by tuning the regularization parameter λ in the range $[0, 100]$. The reported natural risk is estimated on clean verification data. The reported hypocritical risk is estimated on misclassified examples and is empirically approximated using PGD. Results for the models trained on the *Quality* data are summarized in Figure 4. Numerical details about the model accuracy on $\mathcal{D}$, $\mathcal{A}$, and $\mathcal{F}$ with different λ are given in Appendix B Tables 10, 11, 12, and 13. We observe that for both TRADES and THRM, as λ increases, the natural risk increases and the hypocritical risk decreases. It turns out that THRM achieves a better trade-off than TRADES in all cases, which is consistent with our analysis of THRM in Appendix F, and the gap between THRM and TRADES tends to increase when the number of classes is large. Therefore, when we only consider the threat of hypocritical attacks, THRM would be preferable than TRADES. However, if one wants to resist the threat of both adversarial examples and hypocritical examples, TRADES is a viable alternative.

Results of ℓ_∞-dist nets. Results show that ℓ_∞-dist nets achieve moderate certified hypocritical risk. For both *Quality* model and *Poisoning* model, nearly half of the errors are guaranteed not to be covered up by any attack. However, ℓ_∞-dist nets still perform worse than ResNet-18 with TRADES and THRM in terms of empirical hypocritical risk.

Overall, some improvements have been obtained, while complete robustness against hypocritical attacks still cannot be fully achieved with the current methods. Hypocritical risk remains non-negligible even after adaptive robust training. This dilemma highlights the difficulty of stabilizing models to prevent hypocritical attacks. We feel that new manners may be needed to better tackle this problem.

6 Conclusions and Future Directions

This paper unveils the threat of hypocritical examples in the model verification stage. Our experimental results indicate that this type of security risk is pervasive, and remains non-negligible even if adaptive countermeasures are adopted. Therefore, our investigation suggests that practitioners should be aware of this type of threat and be careful about dataset security. Below we discuss some limitations with our current study, and we also feel that our results can lead to several thought-provoking future works.

Other performance requirements. One may consider using adversarial perturbations to combat hypocritical attacks, i.e., estimating the robust error (Zhang et al. 2019) on the verification data. We note that this is actually equivalent to choosing the robust error as the performance requirement. It is natural then to ask whether a hypocritical attacker can cause a substandard model to exhibit high robust accuracy with small perturbations. We leave this as future work.

Transferability. It is also very important to study the transferability of hypocritical examples across substandard models. Transfer-based hypocritical attacks are still harmful when model structure and weights are unknown to the attacker. Understanding the transferability would help us to design effective defense strategies against the transfer-based hypocritical attacks.

Good use of hypocritical perturbations. We showed that many types of substandard models are susceptible to hypocritical attacks. Then, an intriguing question is whether we can turn this weakness into a strength. Specifically, one may find such a "true friend" who is capable of *consistently* helping a substandard model during the deployment stage to make correct predictions. There are concurrent works (Salman et al. 2021; Pestana et al. 2021) which explored this direction, where "robust objects" are designed to help a model to confidently detect or classify them.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61732006, 62076124, and 62106028). Lei Feng was also supported by CAAI-Huawei MindSpore Open Fund.

References

Barocas, S.; Hardt, M.; and Narayanan, A. 2017. Fairness in machine learning. *NeurIPS Tutorial*.

Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M.; and Eckersley, P. 2020. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.

Biggio, B.; Corona, I.; Maiorca, D.; Nelson, B.; Šrndić, N.; Laskov, P.; Giacinto, G.; and Roli, F. 2013. Evasion attacks against machine learning at test time. In *ECML-PKDD*.

Biggio, B.; Nelson, B.; and Laskov, P. 2012. Poisoning attacks against support vector machines. In *ICML*.

Biggio, B.; and Roli, F. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*.

Blum, A.; Hanneke, S.; Qian, J.; and Shao, H. 2021. Robust learning under clean-label attack. In *COLT*.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. In *NeurIPS Deep Learning Symposium*.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *S&P*.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*.

Cullina, D.; Bhagoji, A. N.; and Mittal, P. 2018. PAC-learning in the presence of evasion adversaries. In *NeurIPS*.

Daniely, A.; and Schacham, H. 2020. Most ReLU Networks Suffer from ℓ_2 Adversarial Perturbations. In *NeurIPS*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *CVPR*.

Dong, Y.; Xu, K.; Yang, X.; Pang, T.; Deng, Z.; Su, H.; and Zhu, J. 2021. Exploring Memorization in Adversarial Training. *arXiv preprint arXiv:2106.01606*.

Elsayed, G. F.; Goodfellow, I.; and Sohl-Dickstein, J. 2019. Adversarial reprogramming of neural networks. In *ICLR*.

Feng, J.; Cai, Q.-Z.; and Zhou, Z.-H. 2019. Learning to confuse: generating training time adversarial data with autoencoder. In *NeurIPS*.

Fowl, L.; Goldblum, M.; Chiang, P.-y.; Geiping, J.; Czaja, W.; and Goldstein, T. 2021. Adversarial Examples Make Strong Poisons. In *NeurIPS*.

Gao, J.; Karbasi, A.; and Mahmoody, M. 2021. Learning and Certification under Instance-targeted Poisoning. In *UAI*.

Geiping, J.; Fowl, L.; Somepalli, G.; Goldblum, M.; Moeller, M.; and Goldstein, T. 2021a. What Doesn't Kill You Makes You Robust (er): Adversarial Training against Poisons and Backdoors. *arXiv preprint arXiv:2102.13624*.

Geiping, J.; Fowl, L. H.; Huang, W. R.; Czaja, W.; Taylor, G.; Moeller, M.; and Goldstein, T. 2021b. Witches' Brew: Industrial Scale Data Poisoning via Gradient Matching. In *ICLR*.

Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; and Goldstein, T. 2020. Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *arXiv preprint arXiv:2012.10544*.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Huang, H.; Ma, X.; Erfani, S. M.; Bailey, J.; and Wang, Y. 2021. Unlearnable Examples: Making Personal Data Unexploitable. In *ICLR*.

Kandel, S.; Paepcke, A.; Hellerstein, J.; and Heer, J. 2011. Wrangler: Interactive visual specification of data transformation scripts. In *CHI Conference on Human Factors in Computing Systems*.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *ICML*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. *Technical Report*.

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

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*.

Nakkiran, P. 2019. A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarial Examples are Just Bugs, Too. *Distill*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop*.

Newsome, J.; Karp, B.; and Song, D. 2005. Polygraph: Automatically generating signatures for polymorphic worms. In *S&P*.

Newsome, J.; Karp, B.; and Song, D. 2006. Paragraph: Thwarting signature learning by training maliciously. In *International Workshop on Recent Advances in Intrusion Detection*. Springer.

Niyogi, P.; and Girosi, F. 1996. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4): 819–842.

Northcutt, C. G.; Athalye, A.; and Mueller, J. 2021. Pervasive Label Errors in Test Sets Destabilize Machine Learning Benchmarks. In *NeurIPS 2021 Datasets and Benchmarks Track*.

Paleyes, A.; Urma, R.-G.; and Lawrence, N. D. 2020. Challenges in deploying machine learning: a survey of case studies. In *NeurIPS Workshops*.

Pang, T.; Yang, X.; Dong, Y.; Su, H.; and Zhu, J. 2021. Bag of tricks for adversarial training. In *ICLR*.

Pang, T.; Yang, X.; Dong, Y.; Xu, T.; Zhu, J.; and Su, H. 2020. Boosting Adversarial Training with Hypersphere Embedding. In *NeurIPS*.

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z. B.; and Swami, A. 2016. The limitations of deep learning in adversarial settings. In *EuroS&P*.

Paterson, C.; Calinescu, R.; and Ashmore, R. 2021. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *ACM Computing Surveys*.

Pestana, C.; Liu, W.; Glance, D.; Owens, R.; and Mian, A. 2021. Assistive Signals for Deep Neural Network Classifiers. In *CVPR Workshops*.

Radiya-Dixit, E.; and Tramèr, F. 2021. Data Poisoning Won't Save You From Facial Recognition. In *ICML Workshops*.

Rice, L.; Wong, E.; and Kolter, Z. 2020. Overfitting in adversarially robust deep learning. In *ICML*.

Salman, H.; Ilyas, A.; Engstrom, L.; Vemprala, S.; Madry, A.; and Kapoor, A. 2021. Unadversarial examples: Designing objects for robust vision. In *NeurIPS*.

Sambasivan, N.; Kapania, S.; Highfill, H.; Akrong, D.; Paritosh, P. K.; and Aroyo, L. M. 2021. "Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI. In *CHI Conference on Human Factors in Computing Systems*.

Shafahi, A.; Huang, W. R.; Najibi, M.; Suciu, O.; Studer, C.; Dumitras, T.; and Goldstein, T. 2018. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

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

Tao, L.; Feng, L.; Yi, J.; Huang, S.-J.; and Chen, S. 2021. Better Safe Than Sorry: Preventing Delusive Adversaries with Adversarial Training. In *NeurIPS*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness may be at odds with accuracy. In *ICLR*.

Uesato, J.; O'donoghue, B.; Kohli, P.; and Oord, A. 2018. Adversarial risk and the dangers of evaluating against weak attacks. In *ICML*.

Wang, L.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Jiang, Y. 2020a. Spanning attack: reinforce black-box attacks with unlabeled data. *Machine Learning*.

Wang, Y.; Zou, D.; Yi, J.; Bailey, J.; Ma, X.; and Gu, Q. 2020b. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*.

Wong, E.; and Kolter, Z. 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. In *NeurIPS*.

Yao, L.; and Miller, J. 2015. Tiny imagenet classification with convolutional neural networks. *CS 231N*, 2(5): 8.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, B.; Cai, T.; Lu, Z.; He, D.; and Wang, L. 2021a. Towards Certifying L-infinity Robustness using Neural Networks with L-inf-dist Neurons. In *ICML*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *ICML*.

Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; and Kankanhalli, M. 2020. Attacks which do not kill training make adversarial learning stronger. In *ICML*.

Zhang, J.; Zhu, J.; Niu, G.; Han, B.; Sugiyama, M.; and Kankanhalli, M. 2021b. Geometry-aware instance-reweighted adversarial training. In *ICLR*.

Zhu, C.; Huang, W. R.; Li, H.; Taylor, G.; Studer, C.; and Goldstein, T. 2019. Transferable Clean-Label Poisoning Attacks on Deep Neural Nets. In *ICML*.