On the Limitations of Stochastic Pre-processing Defenses

Yue Gao
University of Wisconsin–Madison
gy@cs.wisc.edu

Ilia Shumailov
University of Cambridge & Vector Institute
ilia.shumailov@cl.cam.ac.uk

Kassem Fawaz
University of Wisconsin–Madison
kfawaz@wisc.edu

Nicolas Papernot
University of Toronto & Vector Institute
nicolas.papernot@utoronto.ca

Abstract

Defending against adversarial examples remains an open problem. A common belief is that randomness at inference increases the cost of finding adversarial inputs. An example of such a defense is to apply a random transformation to inputs prior to feeding them to the model. In this paper, we empirically and theoretically investigate such stochastic pre-processing defenses and demonstrate that they are flawed. First, we show that most stochastic defenses are weaker than previously thought; they lack sufficient randomness to withstand even standard attacks like projected gradient descent. This casts doubt on a long-held assumption that stochastic defenses invalidate attacks designed to evade deterministic defenses and force attackers to integrate the Expectation over Transformation (EOT) concept. Second, we show that stochastic defenses confront a trade-off between adversarial robustness and model invariance; they become less effective as the defended model acquires more invariance to their randomization. Future work will need to decouple these two effects. We also discuss implications and guidance for future research.

1 Introduction

Machine learning models are vulnerable to adversarial examples [4, 35], where an adversary can add imperceptible perturbations to the input of a model and change its prediction [5, 23]. Their discovery has motivated a wide variety of defense approaches [6, 8, 11, 28, 39, 40] along with the evaluation of their adversarial robustness [2, 27, 37]. Current evaluations mostly rely on adaptive attacks [2, 37], which require significant modeling and computational efforts. However, even when the attack succeeds, such evaluations may not always reveal the fundamental weaknesses of an examined defense. Without awareness of the underlying weaknesses, subsequent defenses may still conduct inadvertently weak adaptive attacks; this leads to overestimated robustness.

One popular class of defenses that demonstrates the above is the stochastic pre-processing defense, which relies on applying randomized transformations to inputs to provide robustness [11, 40]. Despite existing attack techniques designed to handle randomness [2, 3], there is an increasing effort to improve these defenses through a larger randomization space or more complicated transformations. For example, BaRT [27] employs 25 transformations, where the parameters of each transformation are further randomized. Due to the complexity of this defense, it was only broken recently (three years later) by Sitawarin et al. [33] with a complicated adaptive attack. Still, it is unclear how future defenses can avoid the pitfalls of existing defenses, largely because these pitfalls remain unknown.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In this paper, we investigate stochastic pre-processing defenses and explain their limitations both empirically and theoretically. First, we revisit previous stochastic pre-processing defenses and explain why such defenses are broken. We show that most stochastic defenses are not sufficiently randomized to invalidate standard attacks designed for deterministic defenses. Second, we study recent stochastic defenses that exhibit more randomness and show that they also face key limitations. In particular, we identify a trade-off between their robustness and the model's invariance to their transformations. These defenses achieve a notion of robustness that results from reducing the model's invariance to the applied transformations. We outline our findings below. These findings suggest future work to find new ways of using randomness that decouples these two effects.

Most stochastic defenses lack sufficient randomness. While Athalye et al. [2] and Tramèr et al. [37] have demonstrated the ineffectiveness of several stochastic defenses with techniques like Expectation over Transformation (EOT) [3], it remains unclear whether and why EOT is required (or at least as a "standard technique") to break them. A commonly accepted explanation is that EOT computes the "correct gradients" of models with randomized components [2, 37], yet the necessity of such correct gradients has not been explicitly discussed. To fill this gap, we examine a long-held assumption that stochastic defenses invalidate standard attacks designed for deterministic defenses.

Specifically, we revisit stochastic pre-processing defenses previously broken by EOT and examine their robustness *without* applying EOT. Interestingly, we find that most stochastic defenses lack sufficient randomness to withstand even standard attacks (that do not integrate any strategy to capture model randomness) like projected gradient descent (PGD) [23]. We then conduct a systematic evaluation to show that applying EOT is only beneficial when the defense is sufficiently randomized. Otherwise, standard attacks already perform well and the randomization's robustness is overestimated.

Trade-off between adversarial robustness and model invariance. When stochastic pre-processing defenses do have sufficient randomness, they must fine-tune the model using augmented training data to preserve utility in the face of randomness added. We characterize this procedure by the model's *invariance* to the applied defense, where we identify a trade-off between the model's robustness (provided by the defense) and its invariance to the applied defense. Stochastic pre-processing defenses become less effective when their defended model acquires more invariance to their transformations.

On the theoretical front, we present a theoretical setting where this trade-off provably exists. We show from this trade-off that stochastic pre-processing defenses provide robustness by inducing variance on the defended model, and must take back such variance to recover utility. We verify this trade-off with empirical evaluations on realistic datasets, models, and defenses. We observe that robustness drops when the defended model is fine-tuned on data processed by its defense to acquire higher invariance.

2 Related Work

Stochastic Pre-processing Defenses. Defending against adversarial examples remains an open problem, where a common belief is that inference-time randomness increases the cost of finding adversarial inputs. Early examples of such stochastic defenses include input transformations [11] and rescaling [40]. These defenses were broken by Athalye et al. [2] using techniques like EOT [3] to capture randomness. After that, more stochastic defenses were proposed but with inadvertently weak evaluations [25, 28, 39, 42], which were found ineffective by Tramèr et al. [37]. Subsequent stochastic defenses resort to larger randomization space like BaRT [27], which was only broken recently by Sitawarin et al. [33]. In parallel to our work, DiffPure [24] adopts a complicated stochastic diffusion process to purify the inputs. As we will discuss in Appendix F.1, this defense belongs to an existing line of research that leverages generative models to pre-process input images [18, 31, 34], hence it matches the settings in our work. On the other hand, randomized smoothing [6, 16, 30] leverages randomness to certify the inherent robustness of a given decision. In this work, instead of designing adaptive attacks for individual defenses, which is a well-known challenging progress [2, 33, 37], we focus on the general stochastic pre-processing defenses and demonstrate their limitations.

Trade-offs for Adversarial Robustness. The trade-offs associated with adversarial robustness have been widely discussed in the literature. For example, prior work identified trade-offs between robustness and accuracy [38, 44] for deterministic classifiers. Pinot et al. [26] generalize this trade-off to randomized classifiers with a similar form as randomized smoothing. Compared with these results, our work provides a deeper understanding that stochastic pre-processing defenses *explicitly* control such trade-offs to provide robustness. Recent work also investigated the trade-off between the model's

robustness and invariance to input transformations, such as circular shifts [32] and rotations [13]. These trade-offs characterize a standalone model's own property — the model itself is less robust to adversarial examples when it becomes more invariant to certain transformations, without any defense. Our setting, however, is orthogonal to such analysis — the model that we consider is protected by a stochastic pre-processing defense, and what we really aim to characterize is the performance of that pre-processing defense, not the inherent robustness of the model itself.

3 Preliminaries

Notations. Let $f : \mathcal{X} \to \mathbb{R}^C$ denote the classifier with pre-softmax outputs, where $\mathcal{X} = [0, 1]^d$ is the input space with d dimensions and C is the number of classes. We then consider a stochastic pre-processing defense $t_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{X}$, where $\boldsymbol{\theta}$ is the random variable drawn from some randomization space Θ that parameterizes the defense. The defended classifier can be written as $f_{\boldsymbol{\theta}}(\boldsymbol{x}) \coloneqq f(t_{\boldsymbol{\theta}}(\boldsymbol{x}))$.

Let $F(\boldsymbol{x}) \coloneqq \arg\max_{i \in \mathcal{Y}} f_i(\boldsymbol{x})$ denote the classifier that returns the predicted label, where f_i is the output of the i-th class and $\mathcal{Y} = [C]$ is the label space. Similarly, we use $F_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{\theta},i}$ to denote the prediction and class-output of the stochastic classifier $f_{\boldsymbol{\theta}}$. Since this classifier returns varied outputs for a fixed input, it determines the final prediction by aggregating n independent inferences with strategies like majority vote. We discuss these strategies and the choice of n in Appendix A.1.

Adversarial Examples. Given an image $\boldsymbol{x} \in \mathcal{X}$ and a classifier F, the adversarial example $\boldsymbol{x}' \coloneqq \boldsymbol{x} + \boldsymbol{\delta}$ is visually similar to $\boldsymbol{x}$ but either misclassified (i.e., $F(\boldsymbol{x}') \neq F(\boldsymbol{x})$) or classified as a target class y' chosen by the attacker (i.e., $F(\boldsymbol{x}') = y'$). Attack algorithms generate adversarial examples by searching for $\boldsymbol{\delta}$ such that $\boldsymbol{x}'$ fools the classifier while minimizing $\boldsymbol{\delta}$ under some distance metrics; for instance, the ℓ_p norm constraint $\|\boldsymbol{\delta}\|_p \leq \epsilon$ for a perturbation budget ϵ.

Projected Gradient Descent (PGD). PGD [23] is one of the most established attacks to evaluate adversarial example defenses. Given a benign example $\boldsymbol{x}^0$ and its ground-truth label y, each iteration of the untargeted PGD attack (with ℓ_∞ norm budget ϵ) can be formulated as

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i + \alpha \cdot \operatorname{sgn}\big\{\nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y\big)\big\}, \tag{1}$$

where α is the step size, $\mathcal{L}$ is the loss function, and each iteration is projected to the ℓ_∞ ball around $\boldsymbol{x}^0$ of radius ϵ. We use PGD-k to denote the PGD attack with k steps. We outline formulations for other settings and norms in Appendix A.2.

Expectation over Transformation (EOT). Since the classifier $f_{\boldsymbol{\theta}}$ is stochastic, the defense evaluation literature [2, 37] argues that attacks should target the *expectation* of the gradient using Expectation over Transformation (EOT) [3], which reformulates the PGD attack as

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i + \alpha \cdot \operatorname{sgn}\Big\{\mathbb{E}_{\boldsymbol{\theta} \sim \Theta}\Big[\nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y\big)\Big]\Big\} \approx \boldsymbol{x}^i + \alpha \cdot \operatorname{sgn}\Big\{\frac{1}{m}\sum_{j=1}^{m} \nabla \mathcal{L}\big(f_{\boldsymbol{\theta}_j}(\boldsymbol{x}^i), y\big)\Big\}, \tag{2}$$

where m is the number of samples to estimate the expectation and $\boldsymbol{\theta}_j \overset{\text{iid}}{\sim} \Theta$ are sampled parameters for the defense. We use EOT-m to denote the EOT technique with m samples at each PGD step.

In addition, for a fair comparison among attacks with different PGD steps and EOT samples, we quantify the attack's strength by its total number of gradient computations. For example, attacks using PGD-k and EOT-m will have strength $k \times m$. Although white-box attacks are typically not constrained in this way, it allows for a fair comparison when attacks have finite computing resources (e.g., when EOT is not parallelizable). We discuss more about this quantification in Appendix A.3.

4 Most Stochastic Defenses Lack Sufficient Randomness

Athalye et al. [2] and Tramèr et al. [37] demonstrate adaptive evaluation of stochastic defenses with the application of EOT. However, it remains unclear why EOT is required (or at least as a "standard technique") to break these stochastic defenses. While a commonly accepted explanation is that EOT computes the "correct gradients" of models with randomized components [2, 37], the necessity of such correct gradients has not been explicitly discussed. To fill this gap, we revisit stochastic defenses previously broken by EOT and examine their robustness *without* applying EOT. Interestingly, we find that applying EOT is mostly *unnecessary* when evaluating existing stochastic defenses.

Table 2: The missing ablation study of adaptive evaluations of stochastic defenses in the literature. Notations: attack iterations k, EOT samples m, learning rate α, number of gradient queries $k \times m$. The details of these defenses and their evaluation settings are in Appendix B.

Defenses	Original Adaptive Evaluation (w/ EOT)					Our Ablation Study (w/o EOT)				
	k	m	α	$k \times m$	Success Rate	k	m	α	$k \times m$	Success Rate
Guo et al. [11]	1,000	30	0.1	30,000	100%	1,000	1	0.001	1,000	99.0%
Xie et al. [40]	1,000	30	0.1	30,000	100%	200	1	0.1	200	100%
Dhillon et al. [8]	500	10	0.1	5,000	100%	500	1	0.1	500	100%
Xiao et al. [39]	100	1,000	0.01	100,000	100%	40,000	1	0.1/255	40,000	98.4%
Roth et al. [28]	100	40	0.2/255	4,000	100%	4,000	1	0.1/255	4,000	96.1%

Table 1: Evaluation of the random rotation with PGD-k and EOT-m.

Attacks	k	m	Success Rate
Untargeted	10	5	100%
	50	1	100%
Targeted	10	5	99.0%
	50	1	99.0%

Case Study: Random Rotation. We start with a simple stochastic defense that randomly rotates the input image for $\boldsymbol{\theta} \in [-90, 90]$ degrees (chosen at uniform) before classification. This defense is representative for most pre-processing defenses [11, 27, 40]. We evaluate this defense on 1,000 ImageNet images with PGD-k and EOT-m under the constraint $k \times m = 50$, as discussed in Section 3. All attacks use maximum ℓ_∞ perturbation $\epsilon = 8/255$ with step size chosen from $\alpha \in \{1/255, 2/255\}$. The results are shown in Table 1, where PGD-50 performs equally well as PGD-10 combined with EOT-5. This observation suggests that *some stochastic defenses are already breakable without applying EOT*, casting doubt on a long-held assumption that stochastic defenses simply invalidate attacks designed for deterministic defenses.

Comprehensive Evaluations. We then extend the above case study to other stochastic defenses evaluated in the literature. Specifically, we replicate the (untargeted) adaptive evaluation of stochastic defenses from Athalye et al. [2] and Tramèr et al. [37] with their official implementation. We only change the attack's hyper-parameters (e.g., number of iterations and learning rate) and disable EOT by setting its number of samples to one ($m = 1$), which avoids potential implementation flaws if removed from the source code. The comparison between evaluations with and without applying EOT is summarized in Table 2, which serves as a missing ablation study of adaptive evaluations in the literature. The experimental settings are identical within each row (detailed in Appendix B).

Interestingly, we find it *unnecessary* to break these defenses with EOT, as long as the standard attack runs for more iterations with a smaller learning rate. For such defenses, standard iterative attacks already contain an *implicit expectation* across iterations to capture the limited randomness. This observation implies that most stochastic defenses lack sufficient randomness to withstand even standard attacks designed for deterministic defenses. Therefore, increasing randomness becomes a promising approach to enhancing stochastic defenses, as adopted by recent defenses [6, 27]. Note that this ablation study only aims to inspire potential ways of enhancing stochastic defenses; it does not invalidate EOT for stronger adaptive evaluations of stochastic defenses.

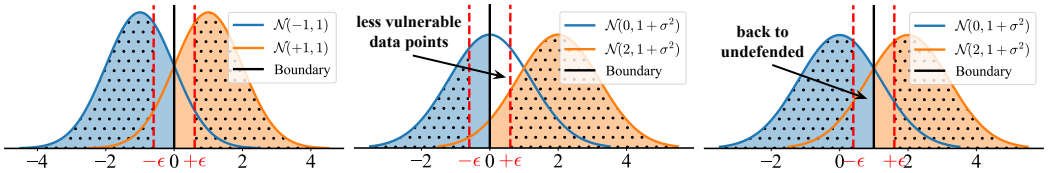
5 Trade-offs between Robustness and Invariance

When stochastic pre-processing defenses *do have* sufficient randomness, they must ensure that the utility of the defended model is preserved in the face of randomness. To achieve high utility, existing defenses mostly rely on augmentation invariance through *trained invariance* [22]. In such a case, the invariance is achieved by applying the defense's randomness to the training data so as to guide the model in learning their transformations. For defenses based on stochastic pre-processor $t_{\boldsymbol{\theta}}$, each data sample from the dataset gets augmented with $t_{\boldsymbol{\theta}}$ sampled from the randomization space Θ, and the risk is minimized over such augmented data.

The defended classifier $F_{\boldsymbol{\theta}}(\boldsymbol{x}) \coloneqq F(t_{\boldsymbol{\theta}}(\boldsymbol{x}))$ is invariant under the randomization space Θ if

$$F(t_{\boldsymbol{\theta}}(\boldsymbol{x})) = F(\boldsymbol{x}), \quad \forall\, \boldsymbol{\theta} \in \Theta, \boldsymbol{x} \in \mathcal{X}. \tag{3}$$

As we can observe from the definition, invariance has direct implications on the performance of stochastic pre-processing defenses. If the classifier is invariant under the defense's randomization space Θ as is defined in Equation (3), then the defense should not work – computing the model and its gradients over randomization $\boldsymbol{\theta} \in \Theta$ is the same as if $t_{\boldsymbol{\theta}}$ was not applied at all. This observation



(a) Undefended (b) Defended (Lack of Invariance) (c) Defended (Trained Invariance)

Figure 1: Illustration of the binary classification task we consider. The curves are the probability density function of two classes of data. Shadowed area denotes correct classification. Dotted area denotes robustly correct classification under the ℓ_∞-bounded adversary with perturbation budget ϵ.

suggests a direct coupling between invariance and performance of the defense: the more invariant, hence performant, the model is under a given randomization space, the less protection such a defense would provide. In this section, we present a simple theoretical setting where this coupling provably exists, as illustrated in Figure 1. Detailed arguments are deferred to Appendix C.1.

Binary Classification Task. We consider a class-balanced dataset $\mathcal{D}$ consisting of input-label pairs (x, y) with $y \in \{-1, +1\}$ and $x|y \sim \mathcal{N}(y, 1)$, where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. Moreover, an ℓ_∞-bounded adversary perturbs the input with a small δ to fool the classifier for $\|\delta\|_\infty \leq \epsilon$. We quantify the classifier's robustness by its robust accuracy, i.e., the ratio of correctly classified samples that remain correct after being perturbed by the adversary.

Undefended Classification. We start with the optimal linear classifier $F(x) := \mathrm{sgn}(x)$ without any defense in Figure 1a. This classifier attains robust accuracy

$$\Pr\big[F(x+\delta) = y \mid F(x) = y\big] = \frac{\Pr\big[F(x+\delta) = y \wedge F(x) = y\big]}{\Pr\big[F(x) = y\big]} = \frac{\Phi(1-\epsilon)}{\Phi(1)}, \tag{4}$$

where Φ is the cumulative distribution function of $\mathcal{N}(0, 1)$.

Defended Classification. We then try to improve adversarial robustness by introducing a stochastic pre-processing defense $t_\theta(x) := x + \theta$, where $\theta \sim \mathcal{N}(1, \sigma^2)$ is the random variable parameterizing the defense. This defense characterizes common pre-processing defenses that enforce randomness while shifting the input distribution. Here, the processed input follows a shifted distribution $t_\theta(x) \sim \mathcal{N}(y + 1, 1 + \sigma^2)$ in Figure 1b. The defended classifier $F_\theta(x) = \mathrm{sgn}(x + \theta)$ has robust accuracy

$$\Pr\big[F_\theta(x+\delta) = y \mid F_\theta(x) = y\big] = \frac{\Pr\big[F_\theta(x+\delta) = y \wedge F_\theta(x) = y\big]}{\Pr\big[F_\theta(x) = y\big]} = \frac{\Phi'(-\epsilon) + \Phi'(2-\epsilon)}{\Phi'(0) + \Phi'(2)}, \tag{5}$$

where $\Phi'(x) := \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1 + \sigma^2)$. At this point, we have not fit the classifier on processed inputs. Due to its lack of invariance, the defended classifier has low utility yet higher robust accuracy than the undefended one in Equation (4).

Defended Classification (Trained Invariance). As discussed above, one critical step of stochastic pre-processing defenses is to preserve the defended model's utility by minimizing the risk over augmented data $t_\theta(x)$, which leads to a new defended classifier $F^+_\theta(x) = \mathrm{sgn}(x + \theta - 1)$ in Figure 1c. As a result, this new defended classifier achieves higher invariance with robust accuracy

$$\Pr\big[F^+_\theta(x+\delta) = y \mid F^+_\theta(x) = y\big] = \frac{\Pr\big[F^+_\theta(x+\delta) = y \wedge F^+_\theta(x) = y\big]}{\Pr\big[F^+_\theta(x) = y\big]} = \frac{\Phi'(1-\epsilon)}{\Phi'(1)}, \tag{6}$$

which is less robust than the previous less-invariant classifier F_θ in Equation (5). However, one may observe that this classifier, though loses some robustness compared with F_θ, is still more robust than the original undefended classifier F in Equation (4). This part of robustness comes from the changed data distribution due to the defense's randomness. It shows that we have not achieved perfect invariance to the defense's randomness, thus gaining some robustness at the cost of utility.

Defended Classification (Perfect Invariance). Furthermore, these defenses usually leverage majority vote to obtain stable predictions, which finally produces a perfectly invariant defended classifier

$$F^*_\theta(x) = \mathrm{sgn}\left\{\frac{1}{n}\sum_{i=1}^{n} F^+_{\theta_i}(x)\right\} = \mathrm{sgn}\left\{\frac{1}{n}\sum_{i=1}^{n} \mathrm{sgn}(x + \theta_i - 1)\right\} \to \mathrm{sgn}(x) = F(x), \tag{7}$$

where $\theta_i \overset{\text{iid}}{\sim} \mathcal{N}(1, \sigma^2)$ are sampled parameters. In such a case, the defended classifier reduces to the original undefended classifier with the original robust accuracy:

$$\Pr\big[F^*_\theta(x+\delta) = y \mid F^*_\theta(x) = y\big] = \Pr\big[F(x+\delta) = y \mid F(x) = y\big] = \frac{\Phi(1-\epsilon)}{\Phi(1)}. \tag{8}$$

Summary. The above theoretical setting illustrates how stochastic pre-processing defenses first induce variance on the binary classifier we consider to provide adversarial robustness in Equation (5), and how they finally take back such variance in Equations (6) and (8) to recover utility. We then extend the above coupling between robustness and invariance to a general trade-off in the following theorem, whose detailed descriptions and proofs are deferred to Appendices C.2 and C.3, respectively.

Theorem 1 (Trade-off between Robustness and Invariance)**.** *Given the above theoretical setting and assumptions, when the defended classifier $F_\theta(x)$ achieves higher invariance $R(k)$ under the defense's randomization space to preserve utility, the adversarial robustness provided by the defense strictly decreases.*

In a nutshell, we prove the strictly opposite monotonic behavior of robustness and invariance when the classifier shifts its decision boundary and employs majority vote to preserve utility. It shows that stochastic pre-processing defenses provide robustness by explicitly reducing the model's invariance to added randomized transformations, and the robustness disappears once the invariance is recovered.

6 Experiments

Our experiments are designed to answer the following two questions.

Q1: What properties make applying EOT beneficial when evaluating stochastic defenses?

We show that applying EOT is only beneficial when the defense is sufficiently randomized; otherwise standard attacks already perform well and leave no room for EOT to improve.

Q2: What is the limitation of stochastic defenses when they do have sufficient randomness?

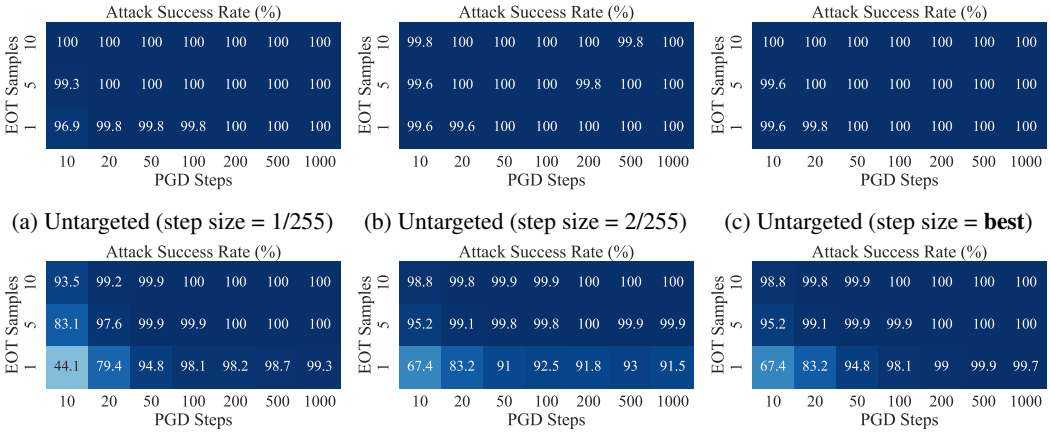
We show a trade-off between the stochastic defense's robustness and the model's invariance to the defense itself. Such defenses become less effective when the defended model achieves higher invariance to their randomness, as required to preserve utility under the defense.

6.1 Experimental Settings

Datasets & Models. We conduct all experiments on ImageNet [29] and ImageNette [9]. For ImageNet, our test data consists of 1,000 images randomly sampled from the validation set. ImageNette is a ten-class subset of ImageNet, and we test on its validation set. We adopt various ResNet [12] models. For defenses with low randomness, we evaluate them on ImageNet with a pre-trained ResNet-50 with Top-1 accuracy 75.9%. For defenses with higher randomness (thus requiring fine-tuning), we switch to ImageNette and a pre-trained ResNet-34 with Top-1 accuracy 96.9% to reduce the training cost like previous work [33]. These models are fine-tuned on the training data processed by tested defenses. As a special case, we also evaluate randomized smoothing on ImageNet using the ResNet-50 models from Cohen et al. [6]. More details of datasets and models can be found in Appendices D.1 and D.2.

Defenses & Metrics. We focus on stochastic defenses allowing us to increase randomness: randomized smoothing [6] and BaRT [27]. For randomized smoothing, we vary the variance of the added Gaussian noise. For BaRT, we vary the number κ of applied randomized transformations. Note that we have evaluated other stochastic defenses and discussed their low randomness in Section 4. We measure the defense's performance by the defended model's *benign accuracy* and the attack's *success rate*, all evaluated with majority vote over $n = 500$ predictions. The attack's success rate is the ratio of samples that do not satisfy the attack's objective prior to the attack but satisfy it after the attack. For example, we discard samples that were misclassified before being perturbed in untargeted attacks. Details of the evaluated defenses can be found in Appendix D.3.

Attacks. We evaluate defenses with standard PGD combined with EOT and focus on the ℓ_∞-bounded adversary with a perturbation budget $\epsilon = 8/255$ in both untargeted and targeted settings. We only use constant step sizes and no random restarts for PGD. We only conduct adaptive evaluations,



(a) Untargeted (step size = 1/255) (b) Untargeted (step size = 2/255) (c) Untargeted (step size = **best**)

(d) Targeted (step size = 1/255) (e) Targeted (step size = 2/255) (f) Targeted (step size = **best**)

Figure 2: Evaluation of BaRT's noise injection defense on ImageNet. Standard PGD without applying EOT (i.e., applying EOT-1) is already good enough, leaving limited space for EOT to improve.

where the defense is included in the attack loop with non-differentiable components captured by BPDA [2]. We also utilize AutoPGD [7] to avoid selecting the step size when it is computationally expensive to repeat experiments in Section 6.3. More details and intuitions of the attack's settings and implementation can be found in Appendix D.4. Our code is available at `https://github.com/wi-pi/stochastic-preprocessing-defenses`.

6.2 Q1: Evaluate the Benefits of Applying EOT under Different Settings

In Section 4, we showed that standard attacks are sufficient to break most stochastic defenses due to their lack of randomness. Here, we aim to understand what properties make applying EOT beneficial when evaluating stochastic defenses. We design a systematic evaluation of stochastic defenses with different levels of randomness and check if applying EOT improves the attack.

Stochastic Defenses with Low Randomness. We start with BaRT's noise injection defense, which perturbs the input image with noise of distributions and parameters chosen at random. While this defense has low randomness, it yields meaningful results. We evaluate this defense with various combinations of PGD and EOT[1]. The performance of untargeted and targeted attacks is shown in Figure 2. We test multiple step sizes and summarize their best results (discussed in Appendix E.1).

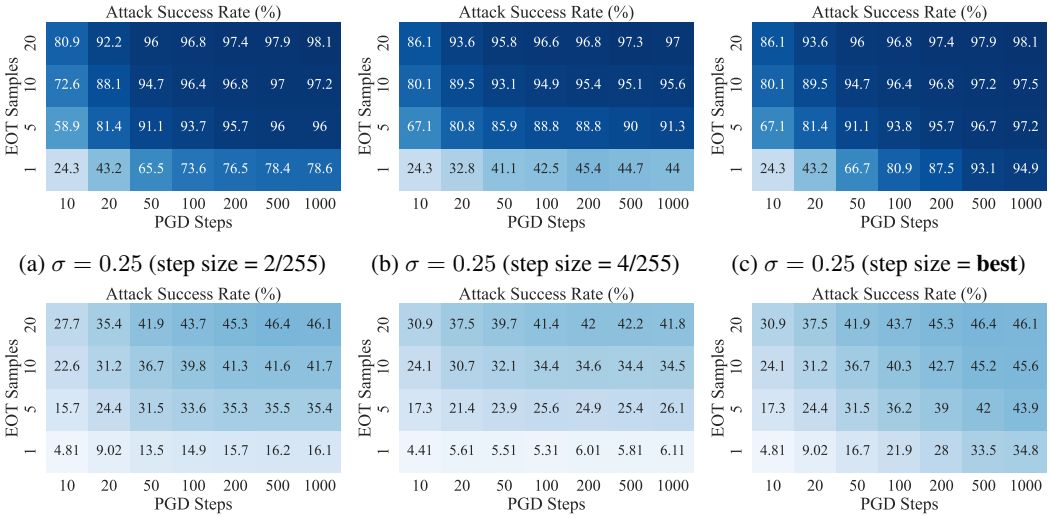
In this case, standard PGD attacks are already good enough when the defense has insufficient randomness, leaving no space for improvements from EOT. In Figure 2f, both (1) PGD-10 combined with EOT-10 and (2) PGD-100 combined with EOT-1 have near 100% success rates. This result is consistent with our observations in Section 4 in both untargeted and targeted settings[2].

Stochastic Defenses with Higher Randomness. We then examine the randomized smoothing defense that adds Gaussian noise to the input image. Although this defense was originally proposed for certifiable adversarial robustness, we adopt it to evaluate how randomness affects the benefits of applying EOT. Similarly, we evaluate this defense with PGD and EOT of different settings with a focus on the *targeted* attack. The results are shown in Figure 3.

We observe that EOT starts to improve the attack when the defense has a higher level of randomness. For a fixed number of PGD steps, applying EOT significantly improves the attack in most of the settings. For a fixed attack strength (i.e., number of gradient computations), applying EOT always outperforms standalone PGD. In Figure 3f, for example, PGD-100 combined with EOT-10 is 5.5% higher than PGD-1,000 with EOT-1 (40.3% vs. 34.8%).

[1]Note that we do not intend to find a heuristic for the best combination of PGD-k and EOT-m, as it is out of the scope of the question that we want to answer. However, it is still possible to correlate the choice of k and m with the convergence analysis of stochastic gradient descent, which we will briefly discuss in Appendix A.3.

[2]The only caveat is that targeted attacks are more likely to benefit from EOT, as their objectives are stricter and may have better performance with gradients of higher precision.



(a) $\sigma = 0.25$ (step size = 2/255) (b) $\sigma = 0.25$ (step size = 4/255) (c) $\sigma = 0.25$ (step size = **best**)

(d) $\sigma = 0.50$ (step size = 2/255) (e) $\sigma = 0.50$ (step size = 4/255) (f) $\sigma = 0.50$ (step size = **best**)

Figure 3: Evaluation of randomized smoothing on ImageNet (targeted attacks). PGD performs well on lower variance ($\sigma = 0.25$) if running for more steps. For a larger variance ($\sigma = 0.50$), applying EOT starts to improve the attack significantly (for a fixed number of gradient computations).

Takeaways. Applying EOT is only beneficial when the defense has sufficient randomness, such as randomized smoothing with $\sigma = 0.5$. This observation suggests that stochastic defenses only make standard attacks suboptimal when they have sufficient randomness. However, most existing stochastic defenses did not achieve this criterion, as we showed in Section 4. We also provide visualizations of adversarial examples under different settings and CIFAR10 results in Appendices E.3 and E.4.

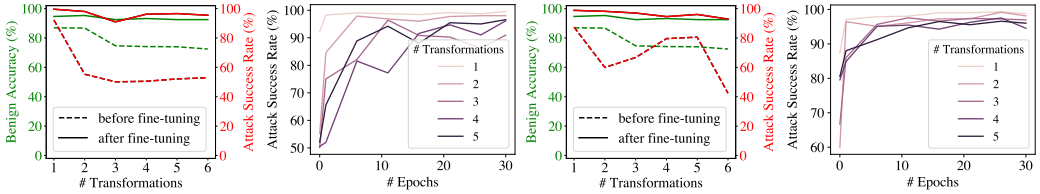
6.3 Q2: Evaluate the Trade-off between Robustness and Invariance

In Section 5, we present a theoretical setting where the trade-off between robustness and invariance provably exists; stochastic defenses become less robust when the defended model achieves higher invariance to their randomness. Here, we demonstrate this trade-off on realistic datasets, models, and defenses. In particular, we choose defenses with sufficient randomness (achieved in different ways) and compare their performance when being applied to models of different levels of invariance, where the invariance is achieved by applying the defense's randomness to the training data so as to guide the model in learning their transformations.

Randomness through Transformations. We first examine the BaRT defense, which pre-processes input images with κ randomly composited stochastic transformations. It represents defenses aiming to increase randomness through diverse input transformations. Since our objective is to demonstrate the trade-off, it suffices to evaluate a subset of BaRT with $\kappa \leq 6$ transformations; this also avoids the training cost of evaluating the original BaRT with $\kappa = 25$. Figure 4 shows the performance of this defense with models before and after fine-tuning on its processed training data.

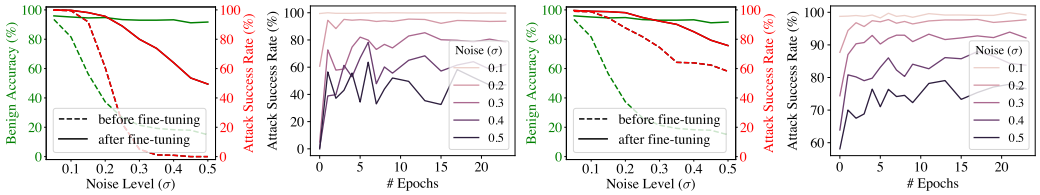
In Figures 4a and 4c, we first observe that fine-tuning indeed increases the model's invariance to the applied defense's randomness; the utility's dashed green curves are improved to the solid green curves beyond 90%. However, as the model achieves higher invariance, the defense becomes nearly ineffective; the attack's dashed red curves boost to the solid red curves near 100%. The same attack's effectiveness throughout the fine-tuning procedure further verifies this observation, as shown in Figures 4b and 4d. It shows a clear trade-off between the defense's robustness and the model's invariance. That is, stochastic defenses start to lose robustness when their defended models achieve higher invariance to their transformations.

Randomness through Noise Levels. We then examine the randomized smoothing defense that adds Gaussian noise to the input image. Unlike BaRT's diverse transformations, randomized smoothing increases randomness directly through the added noise's variance σ^2. This allows us to rigorously increase the randomness without unexpected artifacts like non-differentiable components. We evaluate



(a) Targeted Attacks (view by size) (b) Targeted Attacks (view by epochs) (c) Untargeted Attacks (view by size) (d) Untargeted Attacks (view by epochs)

Figure 4: Performance of the BaRT defense on ImageNette with different numbers of transformations before and after fine-tuning the model. While the model achieves higher invariance, the defense becomes nearly ineffective[3], as evident from the top solid red curves in (a) and (c).



(a) Targeted Attacks (view by levels) (b) Targeted Attacks (view by epochs) (c) Untargeted Attacks (view by levels) (d) Untargeted Attacks (view by epochs)

Figure 5: Performance of the randomized smoothing defense on ImageNette with different noise levels before and after fine-tuning the model. While the model achieves higher invariance, the defense becomes less effective[4], as evident from the gap between dashed and solid red curves in (a) and (c).

the performance of this defense ($\sigma \leq 0.5$) with models before and after fine-tuning on training data perturbed with designated Gaussian noise. The results are shown in Figure 5.

In Figures 5a and 5c, fine-tuning improves the model's invariance, but the defense also becomes significantly weaker during this process. For example, the targeted attack is nearly infeasible when the model is variant to the large noise ($\sigma \geq 0.3$), yet is significantly more effective when the model becomes invariant. The fine-tuning process in Figures 5b and 5d also verifies that stochastic defenses become weaker when their defended models become more invariant to their randomness.

Takeaways. For both the BaRT and the randomized smoothing defense, we observe a clear trade-off between the defense's robustness and the model's invariance to randomness, especially in the targeted setting. In particular, we find that stochastic defenses lose adversarial robustness when their defended models achieve higher invariance to their randomness. Our finding implies that such defenses would become ineffective when their defended models are perfectly invariant to their randomness.

7 Discussions

In this section, we discuss several questions that arose from our study of stochastic pre-processing defenses. Discussions about extensions, limitations, and broader topics can be found in Appendix F.

What do stochastic pre-processing defenses really do? We show that stochastic pre-processing defenses do not introduce inherent robustness to the prediction task. Instead, they shift the input distribution through randomness and transformations, which results in variance and introduces errors during prediction. The observed "robustness", in an unusual meaning for this literature, is a result of these errors. This is fundamentally different from the inherent robustness provided by adversarial training [23]. Although defenses like adversarial training still cost accuracy [38, 44], they do not intentionally introduce errors like stochastic pre-processing defenses.

[3]The defense may not grow stronger with more transformations, which is a drawback of BaRT that we will discuss in Appendix D.3. Yet, our evaluations focus on the fact that solid curves are above the dashed curves.

[4]One may also observe a trade-off between robustness and *utility* by examining the curve's horizontal trend. However, we focus on the trade-off between robustness and *invariance*, which manifests in the vertical gap.

What are the concrete settings that stochastic pre-processing defenses work? These defenses *do make* the attack harder when the adversary has only limited knowledge of the defense's transformations, e.g., in a low-query setting. In such a case, the defense practically introduces noise to the attack's optimization procedure, making it difficult for a low-query adversary to find adversarial examples that consistently cross the probabilistic decision boundary. In contrast, our theoretical analysis in Section 5 considers a powerful adversary with full knowledge of the defense's randomization space; hence it can optimize the adversarial example directly towards the defended model's decision boundary in expectation. The only exception is randomized smoothing, which remains an effective tool to certify the inherent robustness of a given decision.

What are the implications for future research? Our work suggests that future defenses should (at least try to) decouple robustness and invariance; that is, avoid providing robustness by introducing variance to the added randomness. This implication is crucial as the research community continues improving defenses through more complicated transformations. For example, in parallel to our work, DiffPure [24] adopts a complicated stochastic diffusion process to purify the inputs. However, fully understanding DiffPure's robustness requires substantial effort due to its complications and high computational costs, as we will discuss in more detail in Appendix F.1

How should we improve stochastic defenses? Stochastic defenses should rely on randomness that exploits the properties of the prediction task. One promising approach is dividing the problem into orthogonal subproblems. For example, some speech problems like keyword spotting are inherently divisible in the spectrum space [1], and vision tasks are divisible by introducing different modalities [41], independency [17], or orthogonality [43]. In such cases, randomization forces the attack to target all possible (independent) subproblems, where the model performs well on each (independent and) non-transferable subproblem. As a result, defenses can decouple robustness and invariance, hence reducing the effective attack budget and avoiding the pitfall of previous randomized defenses. While systematic guidance for designing defenses (and their attacks) remains an open question, we summarize some critical insights along this direction in Appendix F.2.

What are the implications for adaptive attackers? Our findings suggest that an adaptive attacker needs to consider the spectrum of available standard attack algorithms, instead of just focusing on a given attack algorithm because of the defense's design. As we discover in this paper, EOT can be unnecessary for seemingly immune stochastic defenses, yet its application to break these said defenses gives a false impression about their security against weak attackers. When evaluating the robustness of a defense, the adaptive attack should start by tuning standard approaches, before resorting to more involved attack strategies. This approach helps us to identify the minimally capable attack that breaks the defense and develop a better understanding of the defense's fundamental weaknesses.

8 Conclusion

In this paper, we investigate stochastic pre-processing defenses and explain their limitations both empirically and theoretically. We show that most stochastic pre-processing defenses are weaker than previously thought, and recent defenses that indeed exhibit more randomness still face a trade-off between their robustness and the model's invariance to their transformations. While defending against adversarial examples remains an open problem and designing proper adaptive evaluations is arguably challenging, we demonstrate that stochastic pre-processing defenses are fundamentally flawed in their current form. Our findings suggest that future work will need to find new ways of using randomness that decouples robustness and invariance.

Acknowledgement

We thank all anonymous reviewers for their insightful comments and feedback. This work is partially supported by the DARPA GARD program under agreement number 885000 and the NSF through award CNS-2003129. We would like to acknowledge our sponsors, who support our research with financial and in-kind contributions: CIFAR through the Canada CIFAR AI Chair program, and NSERC through the Discovery Grant and COHESA Strategic Alliance. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. We would like to thank members of the CleverHans Lab for their feedback.

References

[1] Shimaa Ahmed, Ilia Shumailov, Nicolas Papernot, and Kassem Fawaz. Towards more robust keyword spotting for voice assistants. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2655–2672, Boston, MA, August 2022. USENIX Association. URL https://www.usenix.org/conference/usenixsecurity22/presentation/ahmed.

[2] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 274–283. PMLR, 2018. URL http://proceedings.mlr.press/v80/athalye18a.html.

[3] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 284–293. PMLR, 2018. URL http://proceedings.mlr.press/v80/athalye18b.html.

[4] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, volume 8190 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2013. doi: 10.1007/978-3-642-40994-3_25. URL https://doi.org/10.1007/978-3-642-40994-3_25.

[5] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017. doi: 10.1109/SP.2017.49. URL https://doi.org/10.1109/SP.2017.49.

[6] Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

[7] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2206–2216. PMLR, 2020. URL http://proceedings.mlr.press/v119/croce20b.html.

[8] Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=H1uR4GZRZ.

[9] FastAI. The imagenette dataset. URL https://github.com/fastai/imagenette.

[10] Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23(4):2341–2368, 2013. doi: 10.1137/120880811. URL https://doi.org/10.1137/120880811.

[11] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SyJ7ClWCb.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

[13] Sandesh Kamath, Amit Deshpande, Subrahmanyam Kambhampati Venkata, and Vineeth N Balasubramanian. Can we have it all? on the trade-off between spatial and adversarial robustness of neural networks. In M. Ranzato, A. Beygelzimer, K. Nguyen, P. S. Liang, J. W. Vaughan, and Y. Dauphin, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27462–27474. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/e6ff107459d435e38b54ad4c06202c33-Paper.pdf.

[14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[16] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 656–672. IEEE, 2019. doi: 10.1109/SP.2019.00044. URL https://doi.org/10.1109/SP.2019.00044.

[17] Alexander Levine and Soheil Feizi. (de)randomized smoothing for certifiable defense against patch attacks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/47ce0875420b2dbacfc5535f94e68433-Abstract.html.

[18] Yingzhen Li, John Bradshaw, and Yash Sharma. Are generative classifiers more robust to adversarial attacks? In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3804–3814. PMLR, 2019. URL http://proceedings.mlr.press/v97/li19a.html.

[19] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[21] James Lucas, Shengyang Sun, Richard S. Zemel, and Roger B. Grosse. Aggregated momentum: Stability through passive damping. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Syxt5oC5YQ.

[22] Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks, 2020.

[23] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

[24] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 16805–16827. PMLR, 2022. URL https://proceedings.mlr.press/v162/nie22a.html.

[25] Tianyu Pang, Kun Xu, and Jun Zhu. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ByxtC2VtPB.

[26] Rafael Pinot, Laurent Meunier, Florian Yger, Cédric Gouy-Pailler, Yann Chevaleyre, and Jamal Atif. On the robustness of randomized classifiers to adversarial examples. *Machine Learning*, pages 1–33, 2022.

[27] Edward Raff, Jared Sylvester, Steven Forsyth, and Mark McLean. Barrage of random transforms for adversarially robust defense. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6528–6537. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00669. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Raff_Barrage_of_Random_Transforms_for_Adversarially_Robust_Defense_CVPR_2019_paper.html.

[28] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5498–5507. PMLR, 2019. URL http://proceedings.mlr.press/v97/roth19a.html.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

[30] Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J. Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f9fd2624beefbc7808e4e405d73f57ab-Abstract.html.

[31] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=BkJ3ibb0-.

[32] Vasu Singla, Songwei Ge, Basri Ronen, and David Jacobs. Shift invariance can reduce adversarial robustness. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1858–1871. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/0e7c7d6c41c76b9ee6445ae01cc0181d-Paper.pdf.

[33] Chawin Sitawarin, Zachary Golan-Strieb, and David Wagner. Demystifying the adversarial robustness of random transformation defenses. In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*, 2021. URL https://openreview.net/forum?id=p4SrFydwO5.

[34] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJUYGxbCW.

[35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

[37] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/11f38f8ecd71867b42433548d1078e38-Abstract.html.

[38] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SyxAb30cY7.

[39] Chang Xiao, Peilin Zhong, and Changxi Zheng. Enhancing adversarial defense by k-winners-take-all. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skgvy64tvr.

[40] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Sk9yuql0Z.

[41] Karren Yang, Wan-Yi Lin, Manash Barman, Filipe Condessa, and J. Zico Kolter. Defending multimodal fusion models against single-source adversaries. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3340–3349. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00335. URL https://openaccess.thecvf.com/content/CVPR2021/html/Yang_Defending_Multimodal_Fusion_Models_Against_Single-Source_Adversaries_CVPR_2021_paper.html.

[42] Yuzhe Yang, Guo Zhang, Zhi Xu, and Dina Katabi. Me-net: Towards effective adversarial robustness with matrix estimation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7025–7034. PMLR, 2019. URL http://proceedings.mlr.press/v97/yang19e.html.

[43] Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin I. P. Rubinstein, Ce Zhang, and Bo Li. TRS: transferability reduced ensemble via promoting gradient diversity and model smoothness. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17642–17655, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/937936029af671cf479fa893db91cbdd-Abstract.html.

[44] 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 Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19p.html.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix F.3.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix F.3.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We outline all details for theoretical results in Appendix C.
   (b) Did you include complete proofs of all theoretical results? [Yes] We include complete proofs in Appendix C.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Our code is available in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specify brief settings in Section 6.1 and provide complete details in Appendix D.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We study stochastic defenses, whose prediction procedure is already a majority vote over multiple times.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix D.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] We only included our own code for evaluation.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We only used public datasets as discussed in Section 6.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We did not use such data.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix: On the Limitations of Stochastic Pre-processing Defenses

Table of Contents

A More Preliminaries **17**

A.1 Aggregation Strategies for Stochastic Classifiers 17

A.2 Formulation of Projected Gradient Descent 17

A.3 Quantifying the Strength of White-box Attacks 18

B Experiment Setup: Most Stochastic Defenses Lack Sufficient Randomness **18**

B.1 Randomized Image Cropping [11] 18

B.2 Randomized Image Rescaling [40] 19

B.3 Randomized Activation Pruning [8] 19

B.4 Discontinuous Activation [39] 19

B.5 Statistical Detection [28] 19

C Theoretical Analysis: Trade-off between Robustness and Invariance **20**

C.1 Detailed Analysis of the Binary Classification Task 20

C.2 Theorem: Trade-off between Robustness and Invariance 24

C.3 Proofs 25

D Experiment Setup: Main Evaluation **28**

D.1 Datasets 28

D.2 Models 29

D.3 Defenses 29

D.4 Attacks 30

E More Experiment Results **31**

E.1 PGD Captures Randomness with Fine-grained Learning Rates 31

E.2 Inability to Remove Invariance that Does Not Hurt the Utility 31

E.3 Visualization of Adversarial Perturbation 32

E.4 Additional Experiments on CIFAR10 33

F More Discussions **34**

F.1 Discussions about DiffPure [24] 34

F.2 Insights for Designing Attacks and Defenses Regarding Randomness 35

F.3 Limitations and Potential Negative Societal Impacts 36

A More Preliminaries

In this section, we expand on the preliminaries provided in Section 3.

A.1 Aggregation Strategies for Stochastic Classifiers

As the stochastic classifier $f_{\boldsymbol{\theta}}$ returns varied outputs even for a fixed input, it needs to determine the final prediction by aggregating n independent inferences with a particular strategy.

Majority Vote. The most commonly used strategy is *majority vote*, which can be formulated as

$$F_{\boldsymbol{\theta}}^{\text{vote}}(\boldsymbol{x}) \coloneqq \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{n} \mathbb{1}\Big\{ F_{\boldsymbol{\theta}_i}(\boldsymbol{x}) = y \Big\}, \tag{9}$$

where $\{\boldsymbol{\theta}_i\}_{i=1}^{n} \overset{\text{iid}}{\sim} \Theta$ are sampled parameters. We adopt this strategy when the defended classifier computes its prediction. But when attacking a defended classifier, we use the gradient obtained from the original stochastic classifier $f_{\boldsymbol{\theta}}$.

Match All. A more restricted strategy is *match all*, which requires all predictions to be identical:

$$F_{\boldsymbol{\theta}}^{\text{all}}(\boldsymbol{x}) \coloneqq y, \quad \text{s.t.} \quad \sum_{i=1}^{n} \mathbb{1}\Big\{ F_{\boldsymbol{\theta}_i}(\boldsymbol{x}) = y \Big\} = n, \tag{10}$$

where $\{\boldsymbol{\theta}_i\}_{i=1}^{n} \overset{\text{iid}}{\sim} \Theta$ are sampled parameters. This strategy *rejects* the input if the condition is not satisfied, which can be used as a strict setting for targeted attacks. We do not choose this strategy in our work because it is overly strict and is hard to satisfy, even for benign inputs.

Averaged Logits. One may also determine the label from averaged logits over multiple inferences:

$$F_{\boldsymbol{\theta}}^{\text{logits}}(\boldsymbol{x}) \coloneqq \arg\max_{y \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} f_{\boldsymbol{\theta}_i, y}(\boldsymbol{x}), \tag{11}$$

where $\{\boldsymbol{\theta}_i\}_{i=1}^{n} \overset{\text{iid}}{\sim} \Theta$ are sampled parameters. Sitawarin et al. [33] leverage this strategy to design adaptive attacks against the BaRT [27] defense. Still, we do not use this strategy because our main objective is not to break defenses but to analyze their fundamental weaknesses. We only apply the prediction strategy when using a stochastic classifier to evaluate a given set of inputs.

The Choice of Prediction Times. The choice of n typically depends on the stochastic classifier's variance to its randomness. In our setting, the randomness comes from the applied pre-processing defense $t_{\boldsymbol{\theta}}$ with random variable $\boldsymbol{\theta}$ drawn from the randomization space Θ. When the randomization space is small, it suffices to set $n = 1$ for most such defenses [11, 40]. For defenses with a slightly larger randomization space, they can set $n = 30$, for example for randomized cropping [11]. Finally, defenses with even larger randomization spaces set $n = 500$ or more [6, 27]. We fix $n = 500$ in our main experiments in Section 6 for consistency.

A.2 Formulation of Projected Gradient Descent

In this paper, we mainly use PGD [23] to evaluate the robustness of a stochastically defended model. Given a benign example $\boldsymbol{x}^0$ and its ground-truth label y, each iteration of the *untargeted* PGD attack with ℓ_∞ norm budget ϵ can be formulated as

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i + \alpha \cdot \operatorname{sgn}\big\{ \nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y \big) \big\}, \tag{12}$$

where α is the step size, $\mathcal{L}$ is the loss function, and each iteration is projected to the ℓ_∞ ball around $\boldsymbol{x}^0$ of radius ϵ. As for *targeted* PGD attacks with a target label y', the above iteration becomes

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i - \alpha \cdot \operatorname{sgn}\big\{ \nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y' \big) \big\}, \tag{13}$$

where we switch the optimizing direction and the label for computing the loss.

Similarly, the untargeted attack with ℓ_2 norm budget ϵ is formulated as

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i + \alpha \cdot \big\| \nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y \big) \big\|_2, \tag{14}$$

where each iteration is projected to the ℓ_2 norm ball around $\boldsymbol{x}^0$ of radius ϵ, and the targeted attack

$$\boldsymbol{x}^{i+1} \leftarrow \boldsymbol{x}^i - \alpha \cdot \big\| \nabla \mathcal{L}\big(f_{\boldsymbol{\theta}}(\boldsymbol{x}^i), y' \big) \big\|_2. \tag{15}$$

A.3 Quantifying the Strength of White-box Attacks

In this work, we consider attacks with different combinations of PGD steps and EOT samples, denoted by PGD-k and EOT-m. For evaluations of deterministic defenses, quantifying the strength of PGD attacks by the number of steps k is valid. However, this quantification is not informative enough when the evaluated defense is stochastic and involves EOT. For example, it is hard to tell whether PGD-1 with EOT-100 or PGD-100 with EOT-1 has more strength in terms of the number of steps. For a fair comparison between such attacks, we quantify their strength by the total *number of gradient computations*, defined as

$$\texttt{strength}(\text{PGD-}k, \text{EOT-}m) := k \times m. \tag{16}$$

This concept is similar to the *query budget* in the black-box setting. Although we *do not* constrain white-box attacks like this, it allows for a fair comparison between attacks with different settings. For example, we can now argue that the two attacks above have the same strength due to $k \times m = 100$.

Moreover, the above quantification has realistic implications for the white-box attack's computational cost under finite computing resources (w.r.t. the number of evaluated samples). In such a case, the computation of EOT is not parallelizable by batching the EOT samples. For example, when attacking B samples with a maximally possible batch size of B, the attacker has to compute the gradients for $k \times m$ batches. Only when the maximally possible batch size becomes $m \times B$, the attacker can parallelize the EOT samples and only needs to compute gradients for k batches.

Potential Optimality Analysis. Although we evaluate various combinations of PGD-k and EOT-m, we are not interested in finding a heuristic for the best combination for two reasons. Firstly, this discussion is beyond the scope of the question that we want to answer in Section 6.2. Secondly, white-box attackers in the real world have sufficient incentive to adopt a sufficiently large value of k and m (to make sure their attack converges), regardless of the potential optimal choice.

However, it is still possible to correlate the choice of k and m with the convergence rate of stochastic gradient descent (SGD). For example, it is well-known that the convergence rate of SGD can be affected by the estimated gradient's variance [10], and this variance is again affected by the number of EOT samples m we choose due to the central limit theorem. As a result, one can analyze the attack's convergence behavior with different choices of PGD-k and EOT-m. Still, this discussion is beyond the scope of this work and is more beneficial in the context of black-box attacks.

B Experiment Setup: Most Stochastic Defenses Lack Sufficient Randomness

In Section 4, we replicate the evaluation of five previous stochastic defenses from Athalye et al. [2][5] and Tramèr et al. [37][6] without applying EOT. Here, we provide more details of these defenses and their evaluation settings.

Case Study: Random Rotation. In this case study, we evaluate this defense on 1,000 randomly chosen ImageNet images and a pre-trained ResNet-50 model. The settings are consistent with our main evaluation described later in Appendix D.

B.1 Randomized Image Cropping [11]

Defense Details. This defense randomly crops $m = 30$ patches of size 90×90 from each input image of size 299×299. These patches are sent to the classifier, and the final prediction is a majority vote over the predictions of these patches.

Original Evaluation. Athalye et al. [2] evaluate this defense with an ℓ_2-bounded adversary under the root-mean-square perturbation budget of 0.05. Their attack decreases the classification loss (averaged over m patches) using gradient descent with 1,000 iterations and a learning rate 0.1. They decrease the accuracy of an Inception-v3 [36] target model to 0% (i.e., 100% attack success rate) on 1,000 randomly sampled ImageNet [29] images.

Our Ablation Study. We replicate this evaluation by setting $m = 1$ when running the attack (the final defense still uses $m = 30$ patches). This means that we only attack a randomly cropped small

[5] https://github.com/anishathalye/obfuscated-gradients
[6] https://github.com/wielandbrendel/adaptive_attacks_paper

patch from the entire image at each iteration. We then change the learning rate from 0.1 to 0.001 and are able to achieve 99.0% attack success rate.

B.2 Randomized Image Rescaling [40]

Defense Details. This defense randomly rescales the input image of size 299×299 to $r \times r$, where $r \in [200, 331)$ is chosen uniformly at random, and then randomly pads the image with zeros to size 331×331. The resulting padded image is sent to the classifier for one-time prediction.

Original Evaluation. Athalye et al. [2] evaluate this defense with an ℓ_∞-bounded adversary under the perturbation budget of 8/255. They generate adversarial examples using PGD-1000 with a step size of 0.1, where each step applies EOT-30 to compute the gradients averaged over 30 samples processed from the evaluated input image. They decrease the accuracy of an Inception-v3 [36] target model to 0% (i.e., 100% attack success rate) on 1,000 randomly sampled ImageNet [29] images.

Our Ablation Study. We replicate this evaluation with PGD-200 and EOT-1, with all the other parameters unchanged. We are still able to achieve a 100% attack success rate in this case.

B.3 Randomized Activation Pruning [8]

Defense Details. This defense randomly drops (zeros out) some neurons of each layer with probability proportional to their absolute value. The defense considers several levels of probability, and we use the setting used by Athalye et al. [2].

Original Evaluation. Athalye et al. [2] evaluate this defense with an ℓ_∞-bounded adversary under the perturbation budget of 8/255. They decrease the margin between the correct label's logit and the wrong label's logit with gradient descent using the Adam [14] optimizer. The attack runs for 500 steps with a learning rate 0.1, where each iteration averages the gradient over 10 samples. The attack achieves 100% success rate on an Inception-v3 [36] target model and the CIFAR-10 [15] dataset.

Our Ablation Study. We replicate this evaluation by simply setting the number of EOT samples to 1 and are still able to obtain 100% success rate.

B.4 Discontinuous Activation [39]

Defense Details. This defense replaces the standard ReLU activation function inside the neural network with a discontinuous function, so that only the k largest elements are preserved. Although this defense is not stochastic by itself, we evaluate it because the existing evaluation relies heavily on the application of EOT.

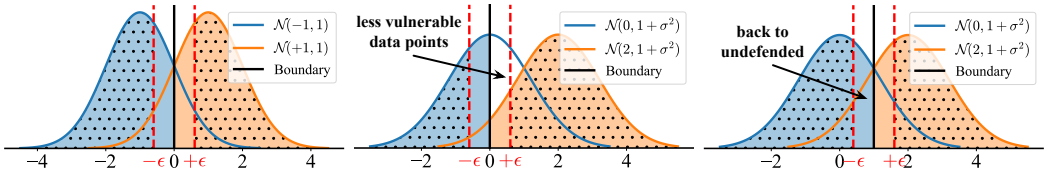
Original Evaluation. Tramèr et al. [37] evaluate this defense with several techniques that approximate the correct gradient. For each input, they estimate the average local gradient with $m = 20,000$ random perturbations drawn from a standard normal distribution with standard deviation $\epsilon = 8/255$. Given this estimated gradient, they consider an ℓ_∞-bounded adversary with perturbation budget 8/255 and run the PGD attack with 100 steps with step size 0.01. Their evaluation code uses a fine-grained choice of m, which is set to 100, 1K, and 20K at the 1st, 20th, and 40th iterations, respectively. We report 1K in the main paper.

As a result, their attack achieves 100% attack success rate on a ResNet-18 [39] model from the original defense and the CIFAR-10 [15] dataset.

Our Ablation Study. We replicate this evaluation by moving all gradient computations from the estimation side m to the attack's iteration side k. That is, instead of running PGD-100 with EOT-1K (100K gradient computations), we run PGD-40K and EOT-1 (40K gradient computations). This setting achieves 98.4% success rate on the same model and dataset. In Appendix E.1, we discuss an interesting observation when evaluating this defense; it shows that PGD may capture randomness as well as EOT with a carefully fine-tuned learning rate.

B.5 Statistical Detection [28]

Defense Details. This defense is a statistical test for detecting adversarial examples. It checks if a given input image is overly robust under Gaussian noise, which is a property of adversarial examples generated by PGD [23] and C&W [5]; benign images are sensitive to such noise.



(a) Undefended (b) Defended (Lack of Invariance) (c) Defended (Trained Invariance)

Figure 6: Illustration of the binary classification task we consider. The curves are the probability density function of two classes of data. Shadowed area denotes correct classification. Dotted area denotes robustly correct classification under the ℓ_∞-bounded adversary with perturbation budget ϵ.

Original Evaluation. Tramèr et al. [37] evaluate this defense with logit matching. Specifically, they generate adversarial examples with a logit that matches a given target image's logit in terms of (1) low mean squared error (MSE) distance and (2) similar robustness under the Gaussian noise. Their attack combines the above two objectives and runs for 100 steps with a learning rate 0.2/255, where the robustness under the Gaussian noise is measured under $m = 40$ samples. The resulting PGD-100 and EOT-40 attack achieves 100% success rate on a target ResNet [12] model and 1,000 randomly sampled ImageNet [29] images.

Our Ablation Study. We replicate the evaluation by moving all EOT samples to PGD steps. That is, we run PGD-4K and EOT-1 with a learning rate 0.1/255. As a result, our attack achieves 96.1% success rate, only 3.9% lower than the attack using EOT. We did not tune the step size further.

C Theoretical Analysis: Trade-off between Robustness and Invariance

We consider a class-balanced dataset $\mathcal{D}$ consisting of input-label pairs (x, y) with $y \in \{-1, +1\}$ and $x|y \sim \mathcal{N}(y, 1)$, where $\mathcal{N}(\mu, \sigma^2)$ is a normal distribution with mean μ and variance σ^2. An ℓ_∞-bounded adversary perturbs the input with a small δ to fool the classifier for $\|\delta\|_\infty \leq \epsilon$. We quantify the classifier's robustness by its robust accuracy, i.e., the ratio of correctly classified samples that remain correct after being perturbed by the adversary. We also consider a stochastic pre-processing defense $t_\theta(x) := x + \theta$, where $\theta \sim \mathcal{N}(1, \sigma^2)$ is the random variable parameterizing the defense.

We formalize our assumptions as follows. Assumptions 1 to 3 characterize the standard behavior of classifiers that employ the pre-processing defense, and Assumption 4 specifies a set of hyper-parameters to simplify the analysis without loss of generality.

Assumption 1 (pre-processing defense)**.** *The classifier only employs a pre-processing defense of the form $t_\theta(x) := x + \theta$. As such, the defended classifier is defined as $F_\theta(x) := \mathrm{sgn}(x + \theta - k)$, where k is the decision boundary it wants to optimize.*

Assumption 2 (trained invariance)**.** *The defended classifier controls its invariance to the defense's transformation through trained invariance, i.e., shifting the decision boundary k.*

Assumption 3 (majority vote)**.** *The defended classifier employs majority vote (for higher invariance) only after it improves the trained invariance. We only consider a sufficiently large number of votes.*

Assumption 4 (hyper-parameters)**.** *For simplicity, we assume that the defender applies $\theta \sim \mathcal{N}(1, 1)$ and the adversary is reasonably strong with a perturbation budget $\epsilon = 1$. Note that $\epsilon = 1$ allows the adversary to shift half of the data across the decision boundary in the undefended scenario.*

Disambiguation of Notations. We use $x + \delta$ to denote the perturbed input passed to a classifier, such as $F(x + \delta)$, but its actual value can be any value chosen from $[x - \epsilon, x + \epsilon]$. We use φ and Φ to denote the PDF and CDF of the standard normal distribution $\mathcal{N}(0, 1)$, respectively. We use φ' and Φ' to denote the PDF and CDF for non-standard normal distributions, whose parameters will be specified in the context.

C.1 Detailed Analysis of the Binary Classification Task

We outline the detailed computations of Section 5 below.

C.1.1 Undefended Classification

The Bayes optimal linear classifier $F(x) = \mathrm{sgn}(x)$ without any defense is illustrated in Figure 6a. This classifier has benign accuracy (i.e., shadowed area):

$$\begin{aligned}
\Pr[F(x) = y] &= \frac{1}{2}\Big(\Pr[F(x) = y \mid y = -1] + \Pr[F(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x < 0 \mid y = -1] + \Pr[x > 0 \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(-1, 1) < 0] + \Pr[\mathcal{N}(+1, 1) > 0]\Big) \\
&= \frac{1}{2}\Big(\Phi(1) + 1 - \Phi(-1)\Big) \\
&= \Phi(1).
\end{aligned} \tag{17}$$

We then compute its probability of making robustly correct predictions (i.e., dotted area)

$$\begin{aligned}
&\Pr[F(x+\delta) = y \wedge F(x) = y] \\
&= \frac{1}{2}\Big(\Pr[F(x+\delta) = y \wedge F(x) = y \mid y = -1] + \Pr[F(x+\delta) = y \wedge F(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x + \epsilon < 0 \wedge x < 0 \mid y = -1] + \Pr[x - \epsilon > 0 \wedge x > 0 \mid y = +1]\Big) \\
&\quad \text{(where we use } x - \epsilon \text{ when } y = +1 \text{ because the correctly classified sample must lie on the right)} \\
&= \frac{1}{2}\Big(\Pr[x < -\epsilon \mid y = -1] + \Pr[x > \epsilon \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(-1, 1) < -\epsilon] + \Pr[\mathcal{N}(+1, 1) > \epsilon]\Big) \\
&= \frac{1}{2}\Big(\Phi(1 - \epsilon) + 1 - \Phi(\epsilon - 1)\Big) \\
&= \Phi(1 - \epsilon),
\end{aligned} \tag{18}$$

which shows that this classifier has robust accuracy (i.e., dotted area over shadowed area)

$$\Pr[F(x+\delta) = y \mid F(x) = y] = \frac{\Pr[F(x+\delta) = y \wedge F(x) = y]}{F(x) = y} = \frac{\Phi(1-\epsilon)}{\Phi(1)}, \tag{19}$$

which verifies the computation in Equation (4).

C.1.2 Defended Classification

The defended classifier $F_\theta(x) = \mathrm{sgn}(x + \theta)$ is illustrated in Figure 6b, with benign accuracy

$$\begin{aligned}
\Pr[F_\theta(x) = y] &= \frac{1}{2}\Big(\Pr[F_\theta(x) = y \mid y = -1] + \Pr[F_\theta(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x + \theta < 0 \mid y = -1] + \Pr[x + \theta > 0 \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(0, 1 + \sigma^2) < 0] + \Pr[\mathcal{N}(2, 1 + \sigma^2) > 0]\Big) \\
&= \frac{1}{2}\Big(\Phi'(0) + \Phi'(2)\Big)
\end{aligned} \tag{20}$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1 + \sigma^2)$.

We then compute its probability of making robustly correct predictions (i.e., dotted area)

$$
\begin{aligned}
&\Pr[F_\theta(x+\delta) = y \wedge F_\theta(x) = y] \\
&= \frac{1}{2}\Big(\Pr[F_\theta(x+\delta) = y \wedge F_\theta(x) = y \mid y = -1] + \Pr[F_\theta(x+\delta) = y \wedge F_\theta(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x+\theta+\epsilon < 0 \wedge x+\theta < 0 \mid y = -1] + \Pr[x+\theta-\epsilon > 0 \wedge x+\theta > 0 \mid y = +1]\Big) \\
&\quad \text{(where we use } x+\theta-\epsilon \text{ when } y = +1 \text{ because the correctly classified sample must lie on the right)} \\
&= \frac{1}{2}\Big(\Pr[x+\theta < -\epsilon \mid y = -1] + \Pr[x+\theta > \epsilon \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(0, 1+\sigma^2) < -\epsilon] + \Pr[\mathcal{N}(2, 1+\sigma^2) > \epsilon]\Big) \\
&= \frac{1}{2}\Big(\Phi'(-\epsilon) + \Phi'(2-\epsilon)\Big),
\end{aligned} \tag{21}
$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1+\sigma^2)$.

It shows that this classifier has robust accuracy (i.e., dotted area over shadowed area)

$$
\Pr[F_\theta(x+\delta) = y \mid F_\theta(x) = y] = \frac{\Pr[F_\theta(x+\delta) = y \wedge F_\theta(x) = y]}{F_\theta(x) = y} = \frac{\Phi'(-\epsilon) + \Phi'(2-\epsilon)}{\Phi'(0) + \Phi'(2)}, \tag{22}
$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1+\sigma^2)$. This verifies the computation in Equation (5).

Here, we can make a quick observation under Assumption 4, where we assume $\sigma = 1$ and $\epsilon = 1$ for simplicity. It shows that the stochastic pre-processing defense in our setting explicitly reduces invariance and utility to gain robustness. The general case is proven in Theorem 1.

Observation 1. *The defended classifier $F_\theta(x) = \mathrm{sgn}(x+\theta)$ has higher robust accuracy (67.7% vs. 59.4%) yet lower benign accuracy (71.1% vs. 84.1%) than the undefended classifier $F(x) = \mathrm{sgn}(x)$.*

C.1.3 Defended Classification (Trained Invariance)

One critical step of stochastic pre-processing defenses is to preserve the defended model's utility by minimizing the risk over processed data $t_\theta(x)$, which leads to a new defended classifier $F_\theta^+(x) = \mathrm{sgn}(x+\theta-1)$ that is optimal on transformed data, as illustrated in Figure 6c. It has benign accuracy

$$
\begin{aligned}
\Pr[F_\theta^+(x) = y] &= \frac{1}{2}\Big(\Pr[F_\theta^+(x) = y \mid y = -1] + \Pr[F_\theta^+(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x+\theta-1 < 0 \mid y = -1] + \Pr[x+\theta-1 > 0 \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(0, 1+\sigma^2) < 1] + \Pr[\mathcal{N}(2, 1+\sigma^2) > 1]\Big) \\
&= \frac{1}{2}\Big(\Phi'(1) + 1 - \Phi'(-1)\Big) \\
&= \Phi'(1),
\end{aligned} \tag{23}
$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1+\sigma^2)$.

We then compute its probability of making robustly correct predictions (i.e., dotted area)

$$
\begin{aligned}
&\Pr[F_\theta^+(x+\delta) = y \wedge F_\theta^+(x) = y] \\
&= \frac{1}{2}\Big(\Pr[F_\theta^+(x+\delta) = y \wedge F_\theta^+(x) = y \mid y = -1] + \Pr[F_\theta^+(x+\delta) = y \wedge F_\theta^+(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x+\theta-1+\epsilon < 0 \wedge x+\theta-1 < 0 \mid y = -1] + \Pr[x+\theta-1-\epsilon > 0 \wedge x+\theta-1 > 0 \mid y = +1]\Big) \\
&\quad \text{(where we use } x+\theta-1-\epsilon \text{ when } y = +1 \text{ because the correctly classified sample must lie on the right)} \\
&= \frac{1}{2}\Big(\Pr[x+\theta < 1-\epsilon \mid y = -1] + \Pr[x+\theta > 1+\epsilon \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(0, 1+\sigma^2) < 1-\epsilon] + \Pr[\mathcal{N}(2, 1+\sigma^2) > 1+\epsilon]\Big) \\
&= \frac{1}{2}\Big(\Phi'(1-\epsilon) + 1 - \Phi'(\epsilon-1)\Big) \\
&= \Phi'(1-\epsilon),
\end{aligned} \tag{24}
$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1+\sigma^2)$.

It shows that this classifier has robust accuracy (i.e., dotted area over shadowed area)

$$
\Pr[F_\theta^+(x+\delta) = y \mid F_\theta^+(x) = y] = \frac{\Pr[F_\theta^+(x+\delta) = y \wedge F_\theta^+(x) = y]}{F_\theta^+(x) = y} = \frac{\Phi'(1-\epsilon)}{\Phi'(1)}, \tag{25}
$$

where $\Phi'(x) \coloneqq \Phi(x/\sqrt{1+\sigma^2})$ is the cumulative distribution function of $\mathcal{N}(0, 1+\sigma^2)$. This verifies the computation in Equation (6).

Here, we can make the following two observations under Assumption 4, where we assume $\sigma = 1$ and $\epsilon = 1$ for simplicity. They show that the defense has to increase the invariance, which was previously reduced to gain robustness, to recover utility. The general case is proven in Theorem 1.

Observation 2. *The defended classifier with trained invariance $F_\theta^+(x) = \operatorname{sgn}(x+\theta-1)$ is less robust (65.8% vs. 70.4%) than the defended classifier $F_\theta(x) = \operatorname{sgn}(x+\theta)$ without trained invariance.*

Observation 3. *The defended classifier with trained invariance $F_\theta^+(x) = \operatorname{sgn}(x+\theta-1)$ is more robust (65.8% vs. 59.4%) than the original undefended classifier $F(x) = \operatorname{sgn}(x)$ at the cost of utility (76.0% vs. 84.1%).*

C.1.4 Defended Classification (Perfect Invariance)

Furthermore, these defenses usually leverage majority vote to obtain stable predictions, which finally produces a perfectly invariant defended classifier:

$$
\begin{aligned}
F_\theta^*(x) &= \operatorname*{arg\,max}_{y \in \{-1,+1\}} \sum_{i=1}^{n} \mathbb{1}\{F_{\theta_i}^+(x) = y\} \\
&= \operatorname{sgn}\left(\frac{1}{n}\sum_{i=1}^{n} F_{\theta_i}^+(x)\right) \\
&= \operatorname{sgn}\left(\frac{1}{n}\sum_{i=1}^{n} \operatorname{sgn}(x+\theta_i-1)\right) \\
&\to \operatorname{sgn}\left(\mathop{\mathbb{E}}_{\theta \sim \mathcal{N}(1,\sigma^2)}\Big[\operatorname{sgn}(x+\theta-1) \,\Big|\, x\Big]\right) \\
&= \operatorname{sgn}\left(\mathop{\mathbb{E}}_{z \mid x \sim \mathcal{N}(x,\sigma^2)}\Big[\operatorname{sgn}(z) \,\Big|\, x\Big]\right) \\
&= \operatorname{sgn}\Big(\Pr\Big[\mathcal{N}(x,\sigma^2) > 0 \,\Big|\, x\Big] - \Pr\Big[\mathcal{N}(x,\sigma^2) < 0 \,\Big|\, x\Big]\Big) \\
&= \operatorname{sgn}(x),
\end{aligned} \tag{26}
$$

where the last equality holds because $\mathcal{N}(x, \sigma^2)$ has more probability on the positive side if and only if $x > 0$ and has more probability on the negative side if and only if $x < 0$. As we can observe,

the defended classifier with trained invariance and majority vote reduces to the original undefended classifier $F(x) = \text{sgn}(x)$, which verifies Equation (7).

C.2 Theorem: Trade-off between Robustness and Invariance

In this section, we extend the above coupling between robustness and invariance to a general trade-off, where we can control the invariance through shifting decision boundary and employing majority vote.

Recall that $x|y \sim \mathcal{N}(y, 1)$ and $\theta \sim \mathcal{N}(1, \sigma^2)$, we denote their density functions by

$$\varphi_x = \begin{cases} \varphi(x+1), & y = -1 \\ \varphi(x-1), & y = +1 \end{cases}, \Phi_x = \begin{cases} \Phi(x+1), & y = -1 \\ \Phi(x-1), & y = +1 \end{cases}, \varphi_\theta = \varphi\left(\frac{\theta - 1}{\sigma}\right), \Phi_\theta = \Phi\left(\frac{\theta - 1}{\sigma}\right), \tag{27}$$

where φ and Φ are the probability and cumulative density functions of $\mathcal{N}(0, 1)$, respectively.

Rate of Invariance. To facilitate our analysis, given the theoretical setting and assumptions specified in Appendix C.1, we define the *rate of invariance* for a defended classifier $F_\theta(x)$ as

$$R(k) \coloneqq \Pr[F_\theta(x) = F(x)], \tag{28}$$

where $F_\theta(x) = \text{sgn}(x + \theta - k)$, and $F(x) = \text{sgn}(x)$ is the undefended classifier.

We formalize the trade-off between robustness and invariance in the following theorem proven in Appendix C.3.4. It shows that stochastic pre-processing defenses provide robustness by intentionally reducing the model's invariance to added randomized transformations.

Theorem 1 (Trade-off between Robustness and Invariance)**.** *Given the above theoretical setting and assumptions, when the defended classifier $F_\theta(x)$ achieves higher invariance $R(k)$ under the defense's randomization space to preserve utility, the adversarial robustness provided by the defense strictly decreases.*

We prove this theorem by characterizing the (strictly opposite) monotonic behavior of invariance and robustness as the defended classifier shifts its decision boundary towards the optimal decision boundary $k = 1$ on transformed data (see Appendix C.1.3) and applies majority vote at the end. We formalize such characterizations in the following lemmas and corollaries.

First, we show in Lemma 1 that the defended classifier's rate of invariance strictly increases as the decision boundary shifts towards the optima; applying majority vote further yields perfect invariance, as we show in Corollary 1. We prove them in Appendix C.3.1.

Lemma 1 (Strictly Increasing Invariance)**.** *The defended classifier's invariance $R(k)$ strictly increases as the decision boundary approaches $k = 1$ without applying majority vote.*

Corollary 1 (Perfect Invariance by Majority Vote)**.** *When the defended classifier maximizes trained invariance at $k = 1$, employing majority vote further improves the rate of invariance $R(k)$ to one.*

Second, we show in Lemma 2 that the defended classifier's robust accuracy strictly decreases as the decision boundary shifts towards the optima. When the trained invariance is approximated, we show in Corollary 2 that applying majority vote strictly decreases the robust accuracy further. We prove them in Appendix C.3.2.

Lemma 2 (Strictly Decreasing Robustness)**.** *The defended classifier's robust accuracy strictly decreases as the decision boundary approaches $k = 1$ without applying majority vote.*

Corollary 2 (Strictly Decreasing Robustness by Majority Vote)**.** *When the defended classifier approximates the trained invariance by shifting its decision boundary to $k \in [0, 2]$, applying majority vote strictly decreases its robust accuracy.*

Finally, we show in Lemma 3 that the defended classifier indeed preserves its utility by shifting the decision boundary towards the optima, and applying majority vote recovers the full utility as we show in Corollary 3. We prove them in Appendix C.3.3.

Lemma 3 (Strictly Increasing Accuracy)**.** *The defended classifier's benign accuracy strictly increases as the decision boundary approaches $k = 1$ without applying majority vote.*

Corollary 3 (Strictly Increasing Accuracy by Majority Vote)**.** *When the defended classifier approximates the trained invariance by shifting its decision boundary to $k \in [0, 2]$, applying majority vote strictly increases its accuracy.*

C.3 Proofs

We provide complete proofs for the theorems, lemmas, and corollaries that we present above.

C.3.1 Strictly Increasing Invariance

Lemma 1 (Strictly Increasing Invariance)**.** *The defended classifier's invariance $R(k)$ strictly increases as the decision boundary approaches $k = 1$ without applying majority vote.*

Proof. We directly compute the rate of invariance $R(k)$ as

$$\begin{aligned}
R(k) &= \Pr[F_\theta(x) = F(x)] \\
&= \Pr[\operatorname{sgn}(x+\theta-k) = \operatorname{sgn}(x)] \\
&= \Pr[\operatorname{sgn}(x+\theta-k) = \operatorname{sgn}(x) \wedge x < 0] + \Pr[\operatorname{sgn}(x+\theta-k) = \operatorname{sgn}(x) \wedge x > 0] \\
&= \Pr[\theta < k - x \wedge x < 0] + \Pr[\theta > k - x \wedge x > 0] \\
&= \int_{-\infty}^{0} \int_{-\infty}^{k-x} \varphi_x(x) \cdot \varphi_\theta(\theta) \,\mathrm{d}\theta \,\mathrm{d}x + \int_{0}^{\infty} \int_{k-x}^{\infty} \varphi_x(x) \cdot \varphi_\theta(\theta) \,\mathrm{d}\theta \,\mathrm{d}x \\
&= \int_{-\infty}^{0} \varphi_x(x) \cdot \Phi_\theta(k-x) \,\mathrm{d}x - \int_{0}^{\infty} \varphi_x(x) \cdot \Phi_\theta(k-x) \,\mathrm{d}x + \int_{0}^{\infty} \varphi_x(x) \,\mathrm{d}x,
\end{aligned} \tag{29}$$

whose gradient with respect to k is

$$\begin{aligned}
\frac{\partial}{\partial k} R(k) &= \int_{-\infty}^{0} \varphi_x(x) \cdot \varphi_\theta(k-x) \,\mathrm{d}x - \int_{0}^{\infty} \varphi_x(x) \cdot \varphi_\theta(k-x) \,\mathrm{d}x \\
&= \frac{1}{2}\left(\int_{-\infty}^{0} \varphi(x+1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x - \int_{0}^{\infty} \varphi(x+1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x \right) \\
&\quad + \frac{1}{2}\left(\int_{-\infty}^{0} \varphi(x-1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x - \int_{0}^{\infty} \varphi(x-1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x \right).
\end{aligned} \tag{30}$$

From calculus and the error function erf we have

$$\begin{aligned}
p_1(x) &\coloneqq \int \varphi(x+1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x = \frac{1}{4\sqrt{\pi}} \exp\left(-\frac{k^2}{4}\right) \cdot \operatorname{erf}\left(1 - \frac{k}{2} + x\right) \\
p_2(x) &\coloneqq \int \varphi(x-1) \cdot \varphi_\theta(k-x) \,\mathrm{d}x = \frac{-1}{4\sqrt{\pi}} \exp\left(-\frac{(k-2)^2}{4}\right) \cdot \operatorname{erf}\left(\frac{k}{2} - x\right),
\end{aligned} \tag{31}$$

where we have assumed $\theta \sim \mathcal{N}(1, 1)$ to simplify the analysis by Assumption 4.

It shows that the gradient in Equation (30) is

$$\begin{aligned}
\frac{\partial}{\partial k} R(k) &= \frac{1}{2}\Big(p_1(0) - p_1(-\infty) - p_1(\infty) + p_1(0)\Big) + \frac{1}{2}\Big(p_2(0) - p_2(-\infty) - p_2(\infty) + p_2(0)\Big) \\
&= p_1(0) + p_2(0) \\
&\propto \exp\left(-\frac{k^2}{4}\right) \cdot \operatorname{erf}\left(1 - \frac{k}{2}\right) - \exp\left(-\frac{(k-2)^2}{4}\right) \cdot \operatorname{erf}\left(\frac{k}{2}\right).
\end{aligned} \tag{32}$$

Notice that $G(k) \coloneqq \frac{\partial}{\partial k} R(k)$ is a symmetric function with respect to the point $(1, 0)$:

$$G(1+z) + G(1-z) = 0, \quad \forall z \in \mathbb{R}, \tag{33}$$

which shows that $G(k)$ attains zero at $k = 1$.

Since both $\exp$ and erf are strictly increasing functions, for $k < 1$, we have

$$\begin{aligned}
-\frac{k^2}{4} > -\frac{(k-2)^2}{4} \quad &\implies \exp\left(-\frac{k^2}{4}\right) > \exp\left(-\frac{(k-2)^2}{4}\right), \\
1 - \frac{k}{2} > \frac{k}{2} \quad &\implies \operatorname{erf}\left(1 - \frac{k}{2}\right) > \operatorname{erf}\left(\frac{k}{2}\right),
\end{aligned} \tag{34}$$

which shows that $G(k) > 0$ when $k < 1$, and $G(k) < 0$ when $k > 1$ by symmetry.

Therefore, the rate of invariance $R(k)$ strictly increases for $k < 1$ and strictly decreases for $k > 1$. □

Corollary 1 (Perfect Invariance by Majority Vote)**.** *When the defended classifier maximizes trained invariance at $k = 1$, employing majority vote further improves the rate of invariance $R(k)$ to one.*

Proof. We showed in Appendix C.1.4 that the defended classifier $F_\theta(x) = \mathrm{sgn}(x + \theta - 1)$ converges to the optimal classifier $F_\theta(x) = \mathrm{sgn}(x)$ if given a sufficiently large number of votes.

In such a case, it is straightforward to show that the rate of invariance converges to one:

$$R(k = 1) = \Pr[F_\theta(x) = F(x)] \to \Pr[\mathrm{sgn}(x) = \mathrm{sgn}(x)] = 1. \tag{35}$$

□

C.3.2 Strictly Decreasing Robustness

Lemma 2 (Strictly Decreasing Robustness)**.** *The defended classifier's robust accuracy strictly decreases as the decision boundary approaches $k = 1$ without applying majority vote.*

Proof. We directly compute the robust accuracy of the defended classifier $F_\theta(x) = \mathrm{sgn}(x + \theta - k)$ and characterize its monotonic behavior. Recall that $x + \theta \sim \mathcal{N}(y + 1, 2)$.

We first compute the defended classifier's benign accuracy:

$$\begin{aligned}
&\Pr[F_\theta(x) = y] \\
&= \Pr[\mathrm{sgn}(x + \theta - k) = y] \\
&= \Pr[x + \theta < k \mid y = -1] \cdot \Pr[y = -1] + \Pr[x + \theta > k \mid y = +1] \cdot \Pr[y = +1] \\
&= \Pr[\mathcal{N}(0, 2) < k] \cdot \Pr[y = -1] + \Pr[\mathcal{N}(2, 2) > k] \cdot \Pr[y = +1] \\
&= \frac{1}{2}\Big(\Phi'(k) + \Phi'(2 - k)\Big),
\end{aligned} \tag{36}$$

where Φ' denotes the cumulative density function of $\mathcal{N}(0, 2)$.

We then compute the probability of robustly correct predictions, where we use $x + \delta$ to denote the adversarial example that actually can take any value from $[x - \epsilon, x + \epsilon]$ to change the prediction:

$$\begin{aligned}
&\Pr[F_\theta(x + \delta) = y \wedge F_\theta(x) = y] \\
&= \frac{1}{2}\Big(\Pr[F_\theta(x + \delta) = y \wedge F_\theta(x) = y \mid y = -1] + \Pr[F_\theta(x + \delta) = y \wedge F_\theta(x) = y \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x + \theta - k + \epsilon < 0 \wedge x + \theta - k < 0 \mid y = -1] + \Pr[x + \theta - k - \epsilon > 0 \wedge x + \theta - k > 0 \mid y = +1]\Big) \\
&\quad \text{(where we use } -\epsilon \text{ when } y = +1 \text{ because the correctly classified sample must lie on the right)} \\
&= \frac{1}{2}\Big(\Pr[x + \theta < k - \epsilon \wedge x + \theta < k \mid y = -1] + \Pr[x + \theta > k + \epsilon \wedge x + \theta > k \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[x + \theta < k - \epsilon \mid y = -1] + \Pr[x + \theta > k + \epsilon \mid y = +1]\Big) \\
&= \frac{1}{2}\Big(\Pr[\mathcal{N}(0, 2) < k - \epsilon] + \Pr[\mathcal{N}(2, 2) > k + \epsilon]\Big) \\
&= \frac{1}{2}\Big(\Phi'(k - \epsilon) + \Phi'(2 - k - \epsilon)\Big),
\end{aligned} \tag{37}$$

where Φ' denotes the cumulative density function of $\mathcal{N}(0, 2)$.

Now we can compute the robust accuracy at decision boundary k as

$$\Pr[F_\theta(x + \delta) = y \mid F_\theta(x) = y] = \frac{\Pr[F_\theta(x + \delta) = y \wedge F_\theta(x) = y]}{\Pr[F_\theta(x) = y]} = \frac{\Phi'(k - \epsilon) + \Phi'(2 - k - \epsilon)}{\Phi'(k) + \Phi'(2 - k)}. \tag{38}$$

While the argument holds for any fixed ϵ, we will show a simple example and assume a reasonably strong adversary with $\epsilon = 1$ (Assumption 4), which initializes the robust accuracy to:

$$\texttt{Rob}(k) \coloneqq \frac{\Phi'(k-1) + \Phi'(1-k)}{\Phi'(k) + \Phi'(2-k)}, \tag{39}$$

where Φ' denotes the cumulative density function of $\mathcal{N}(0, 2)$. Its gradient with respect to k is

$$G(k) \coloneqq \frac{\partial}{\partial k}\texttt{Rob}(k) = \frac{2\exp\left(-1 - \frac{k^2}{4}\right)(e^k - e)}{\sqrt{\pi}\left(2 - \operatorname{erf}\left(\frac{k-2}{2}\right) - \operatorname{erf}\left(-\frac{k}{2}\right)\right)^2} \propto e^k - e, \tag{40}$$

which shows that $G(k) < 0$ when $k < 1$, $G(k) > 0$ when $k > 1$, and $G(k) = 0$ when $k = 1$.

Therefore, the robust accuracy $\texttt{Rob}(k)$ strictly decreases as k approaches $k = 1$ from either side. □

Corollary 2 (Strictly Decreasing Robustness by Majority Vote)**.** *When the defended classifier approximates the trained invariance by shifting its decision boundary to $k \in [0, 2]$, applying majority vote strictly decreases its robust accuracy.*

Proof. For this proof, we assume the applied defense adopts $\theta \sim \mathcal{N}(1, \sigma^2)$, which reformulates the robust accuracy in Equation (39) as

$$\texttt{Rob}(k) \coloneqq \frac{\Phi'(k-1) + \Phi'(1-k)}{\Phi'(k) + \Phi'(2-k)} = 2\left(2 + \operatorname{erf}\left(\frac{2-k}{\sqrt{2}\sqrt{1+\sigma^2}}\right) + \operatorname{erf}\left(\frac{k}{\sqrt{2}\sqrt{1+\sigma^2}}\right)\right)^{-1}, \tag{41}$$

where Φ' is the cumulative density function of $\mathcal{N}(0, 1 + \sigma^2)$.

For $k \in [0, 2]$, where the argument for erf is non-negative, decreasing σ will also decrease the robust accuracy (the erf function is strictly increasing). Given that majority vote effectively reduces the noise's variance, having a larger number of votes will strictly decrease the robust accuracy. □

C.3.3 Strictly Increasing Accuracy

Lemma 3 (Strictly Increasing Accuracy)**.** *The defended classifier's benign accuracy strictly increases as the decision boundary approaches $k = 1$ without applying majority vote.*

Proof. In Equation (36), we showed that the defended classifier $F_\theta(x) = \operatorname{sgn}(x + \theta - k)$ has benign accuracy

$$\texttt{Acc}(k) \coloneqq \Pr[F_\theta(x) = y] = \frac{1}{2}\Big(\Phi'(k) + \Phi'(2-k)\Big), \tag{42}$$

where Φ' denotes the cumulative density function of $\mathcal{N}(0, 2)$.

Its gradient with respect to k is

$$G(k) \coloneqq \frac{\partial}{\partial k}\texttt{Acc}(k) = \frac{\exp\left(-1 - \frac{k^2}{4}\right)(e - e^k)}{2\sqrt{\pi}} \propto e - e^k, \tag{43}$$

which shows that $G(k) > 0$ when $k < 1$, $G(k) < 0$ when $k > 1$, and $G(k) = 0$ when $k = 1$.

Therefore, the benign accuracy $\texttt{Acc}(k)$ strictly increases as k approaches $k = 1$ from either side. □

Corollary 3 (Strictly Increasing Accuracy by Majority Vote)**.** *When the defended classifier approximates the trained invariance by shifting its decision boundary to $k \in [0, 2]$, applying majority vote strictly increases its accuracy.*

Proof. For this proof, we assume the applied defense adopts $\theta \sim \mathcal{N}(1, \sigma^2)$, which reformulates the benign accuracy in Equation (42) as

$$\texttt{Acc}(k) \coloneqq \frac{1}{2}\Big(\Phi'(k) + \Phi'(2-k)\Big) = 2 + \operatorname{erf}\left(\frac{2-k}{\sqrt{2}\sqrt{1+\sigma^2}}\right) + \operatorname{erf}\left(\frac{k}{\sqrt{2}\sqrt{1+\sigma^2}}\right), \tag{44}$$

where Φ' denotes the cumulative density function of $\mathcal{N}(0, 1 + \sigma^2)$.

For $k \in [0, 2]$, where the argument for erf is non-negative, decreasing σ will also decrease the robust accuracy (the erf function is strictly increasing). Given that majority vote effectively reduces the noise's variance, having a larger number of votes will strictly increase the robust accuracy.

As a special case, when $k = 1$ and $\sigma \to 0$, we have

$$\texttt{Acc}(k) = \frac{1}{2}\Big(\Phi'(k) + \Phi'(2-k)\Big) = \frac{1}{2}\left(\Phi\left(\frac{k}{\sqrt{1+\sigma^2}}\right) + \Phi\left(\frac{2-k}{\sqrt{1+\sigma^2}}\right)\right) \to \Phi(1), \tag{45}$$

which recovers the full utility of the undefended classifier in Equation (17). □

C.3.4 Trade-off between Robustness and Invariance

Theorem 1 (Trade-off between Robustness and Invariance)**.** *Given the above theoretical setting and assumptions, when the defended classifier $F_\theta(x)$ achieves higher invariance $R(k)$ under the defense's randomization space to preserve utility, the adversarial robustness provided by the defense strictly decreases.*

Proof. The proof follows by directly combining the lemmas and corollaries proven above.

By Lemma 3 and Corollary 3, when the defended classifier $F_\theta(x) = x + \theta - k$ shifts its decision boundary towards $k = 1$, its benign accuracy strictly increases and is maximized at $k = 1$ with the application of majority vote. This verifies that the defended classifier in our setting indeed preserves utility by shifting the decision boundary towards $k = 1$.

By Lemma 1 and Corollary 1, when the defended classifier shifts the decision boundary towards $k = 1$ to preserve utility, its rate of invariance strictly increases and is maximized at $k = 1$ with the application of majority vote. This verifies that the defended classifier in our setting strictly controls its invariance by shifting the decision boundary.

By Lemma 2 and Corollary 2, when the defended classifier shifts the decision boundary towards $k = 1$ to acquire more invariance, the adversarial robustness strictly decreases and is minimized at $k = 1$ with the application of majority vote.

The above arguments show that the defended classifier strictly improves its invariance by approaching $k = 1$, yet the adversarial robustness strictly decreases during this process. When perfect invariance is achieved, the utility and robustness go back to those of the undefended classifier, nullifying the initially applied stochastic pre-processing defense. □

D Experiment Setup: Main Evaluation

In this section, we provide more details of our main evaluation.

D.1 Datasets

We conduct all experiments on the public ImageNet [29] and ImageNette [9] datasets.

For ImageNet, our test data consists of 1,000 images randomly sampled from the validation set. These images are only sampled once and are fixed for all experiments. We did not train models on the ImageNet training data in our experiments.

ImageNette is a ten-class subset of ImageNet. Its original training set and validation set have 9,469 and 3,925 images, respectively. We randomly split its original training set into our 90% and 10% training and validation data, and adopt 1,000 images randomly sampled from its original validation set as our test data. The data split and test images are only sampled once and fixed for all experiments. We use the high-resolution version of ImageNette, where all images are larger than 320×320.

Because some of our experiments require fine-tuning models on processed training data, we switch to ImageNette to reduce the training cost. We evaluate on ImageNet only when (1) model fine-tuning is not needed, or (2) model fine-tuning is needed but a pre-trained model is publicly available.

D.2 Models

We adopt various ResNet [12] models mainly depending on the examined defense. All models make the prediction with majority vote over $n = 500$ samples if a stochastic defense is applied.

For defenses with low randomness, which require no model fine-tuning, we evaluate them on ImageNet with a ResNet-50 model pre-trained by TorchVision[7], which attains 76.13% Top-1 accuracy and 92.86% Top-5 accuracy on ImageNet.

For defenses with higher randomness, which require model fine-tuning, we evaluate them on ImageNette with our own ResNet-34 models, detailed as follows.

To first obtain a baseline model for ImageNette, we adopt a ResNet-34 model pre-trained by TorchVision, which attains 73.31% Top-1 accuracy and 91.42% Top-5 accuracy on ImageNet. We fine-tune this model on ImageNette's training set with gradient descent for 70 epochs using the AdamW [20] optimizer and the Cosine Annealing [19] learning rate scheduler, where we use batch size 256, initial learning rate 0.001, and weight decay 0.01. We choose the model that performs best on the validation set, which attains 96.9% Top-1 accuracy on the test set.

We then fine-tune the above baseline ResNet-34 model on training data pre-processed by the defense we examine in each experiment. We adopt the same training configs as those used to train the baseline model but reduce the number of epochs to 30.

As a special case, when we evaluate randomized smoothing in Section 6.2, which requires model fine-tuning with data perturbed by Gaussian noise, we adopt the ResNet-50 models pre-trained on such perturbed ImageNet from Cohen et al. [6]. These models attain 67% and 57% Top-1 accuracy when the input is perturbed with Gaussian noise of standard deviation 0.25 and 0.50, respectively.

D.3 Defenses

Our main evaluation focuses on two stochastic defenses, detailed as follows.

BaRT [27]. The original BaRT defense considers a randomization space of 25 diverse input transformations, and the parameters of each transformation are further randomized. At each inference, it randomly samples κ randomized transformations, composites them together in a random order, and applies the composited transformation to the input image.

Since our evaluation only aims to examine the limitations of BaRT but not to break it, it suffices to analyze a *subset* of transformations. Specifically, we consider a randomization space of $\kappa \leq 6$ input transformations and composite all κ transformations in a random order to pre-process the input image before feeding it to the classifier. We outline the chosen randomized transformations below and refer to Raff et al. [27] for more details. Our implementation is available in the code.

- *Noise Injection.* This transformation perturbs the input image with noise of distributions and parameters chosen uniformly at random. The set of candidate noise distributions includes Gaussian, Poisson, Salt, Pepper, Salt and Pepper, and Speckle.
- *Gaussian Blur.* This transformation blurs the input image using a Gaussian filter with the kernel size randomly chosen from $[2, 14]$ and the standard deviation randomly chosen from $[0.1, 3.1]$.
- *Median Blur.* This transformation blurs the input image using a median filter with the kernel size randomly chosen from $[2, 14]$.
- *Swirl Transformation.* This transformation applies the swirl transformation[8], a non-linear image deformation that creates a whirlpool effect. Its strength, radius, and location are chosen uniformly at random from $[0.1, 2.0]$, $[10, 200]$, and $[1, 200]$, respectively. We adopt BPDA [2] with the identity function to handle the non-differentiable problem.
- *Quantization.* This transformation quantizes the input image's pixel values within $[0, 1]$ to a limited number of bins, where the number of bins is chosen uniformly at random from $[8, 200]$. For example, if the number of bins is set to 4, all pixels will be quantized to $\{0.00, 0.25, 0.50, 0.75, 1.00\}$.

[7] https://pytorch.org/vision/stable/models.html
[8] https://scikit-image.org/docs/stable/auto_examples/transform/plot_swirl.html

- *FFT Perturbation.* This transformation perturbs the 2D FFT of each channel of the input image. For each channel in the frequency domain, it randomly zeros out a fraction of coefficients. The fraction is chosen uniformly at random from $[0.00, 0.95]$.

In our setting, we form the randomization space by compositing the first κ transformations in random order. While increasing the space of transformations typically leads to a more effective defense, randomly compositing transformations may not always lead to stronger defenses. For example, the quantization may decrease the effectiveness of other transformations. However, this drawback *does not* affect our evaluation, as our main objective is to compare the defense's performance before and after the defended model achieves higher invariance. Rigorous comparisons between the defense's performance before and after increasing the randomness are largely orthogonal to our work.

Randomized Smoothing [6]. Randomized smoothing adds Gaussian noise to the input image and makes predictions with majority vote over a large number of samples. This defense was initially proposed for certifiable adversarial robustness. In our evaluation, we adopt this defense to examine (1) how randomness affects the effectiveness of applying EOT and (2) how invariance affects the robustness provided by the defense. Specifically, we control the level of randomness by varying the added Gaussian noise's standard deviation σ. For evaluation on ImageNet, we choose $\sigma \in \{0.25, 0.50\}$ as models pre-trained on data perturbed by such noise are available from Cohen et al. [6]. For evaluation on ImageNette, we are able to scale the evaluation for σ from 0.10 to 0.50 with a step size of 0.05. We ignore the *abstain* output in the original defense, as we do not study the certification.

D.4 Attacks

We evaluate defenses with standard PGD combined with EOT and focus on the ℓ_∞-bounded adversary with a perturbation budget $\epsilon = 8/255$ in both untargeted and targeted settings. We do not introduce any techniques other than EOT to explicitly handle the randomness, such as random restarts [7] and momentum-based optimizers [33]. We also utilize AutoPGD [7] to avoid selecting the best step size when it is computationally expensive to repeat some experiments. More importantly, we only conduct adaptive evaluations, where the defense is always included in the attack loop with non-differentiable components approximated by the identity function [2]. For targeted attacks, we choose the last class of each dataset as the target label: ImageNet (999) and ImageNette (9).

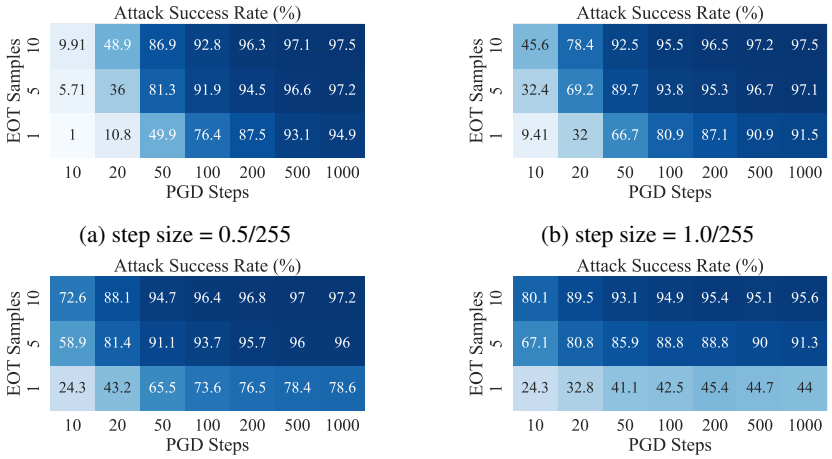
We adopt various attack settings depending on the experiment, as detailed below.

In Section 6.2, we aim to evaluate the benefits of applying EOT under different settings. For this end, we apply the standard PGD attack of $k \in \{10, 20, 50, 100, 200, 500, 1000\}$ steps and combine them with EOT of $m \in \{1, 5, 10, 20\}$ samples. For each combination, we further test several step sizes chosen from $\alpha \in \{0.5/255, 1/255, 2/255, 4/255\}$ and report their best performance.

In Section 6.3, we aim to evaluate the trade-off between the defense's robustness and the model's invariance to the added randomness. For this end, we evaluate the defense's performance when it applies to models of different levels of invariance. Since it is computationally expensive to repeat the attack for multiple step sizes, we utilize AutoPGD [7] to tune the step size automatically. In this experiment, we evaluate all defenses with AutoPGD of 200 steps and disable all techniques designed to capture the randomness, including EOT.

We make this choice due to three considerations. First, disabling EOT effectively reduces the computational cost. Second, we already showed in Section 6.2 that PGD attacks can already assess the robustness of stochastic defenses without applying EOT. Last but not least, our objective is to evaluate the defense's performance when it applies to different models *under the same attack*. Under this setting, we can observe that the same attack (regardless of its strength) that hardly works for the defense (before fine-tuning & low invariance) now becomes more effective (after fine-tuning & high invariance). We can surely run each attack for more iterations and samples, but the current setting suffices to show that the defense provides robustness by explicitly reducing invariance.

Computing Resources. All experiments are conducted on two Linux workstations, each with 48 Intel Xeon CPUs and 8 GeForce RTX 2080 Ti GPUs. We only train ResNet-34 models on ImageNette without the distributed setting. Standard training (70 epochs) takes 35 minutes. Training with data processed by Gaussian noise (30 epochs) takes 15 minutes. Training with data processed by BaRT (30 epochs) takes 18 minutes for $\kappa \in \{1, 2\}$, 70 minutes for $\kappa \in \{3, 4, 5\}$, and 3 hours for $\kappa \in \{6\}$.



(a) step size = 0.5/255

(b) step size = 1.0/255

(c) step size = 2.0/255

(d) step size = 4.0/255

Figure 7: Evaluation of randomized smoothing on ImageNet (targeted attacks, $\sigma = 0.25$).

The training time for $\kappa \geq 3$ is higher due to CPU-bounded transformations; implementing such transformations using native PyTorch operations on GPUs should decrease the training cost.

E More Experiment Results

In this section, we provide more experiment results that strengthen discussions in the main paper.

E.1 PGD Captures Randomness with Fine-grained Learning Rates

During our evaluation, we recognize that the effectiveness of PGD attacks, when combined with EOT, is sensitive to the choice of step size (i.e., learning rate). Here, we provide the full results of four different choices of step size when evaluating the randomized smoothing defense. The results with step size chosen from $\alpha \in \{0.5/255, 1.0/255, 2.0/255, 4.0/255\}$ are shown in Figure 7.

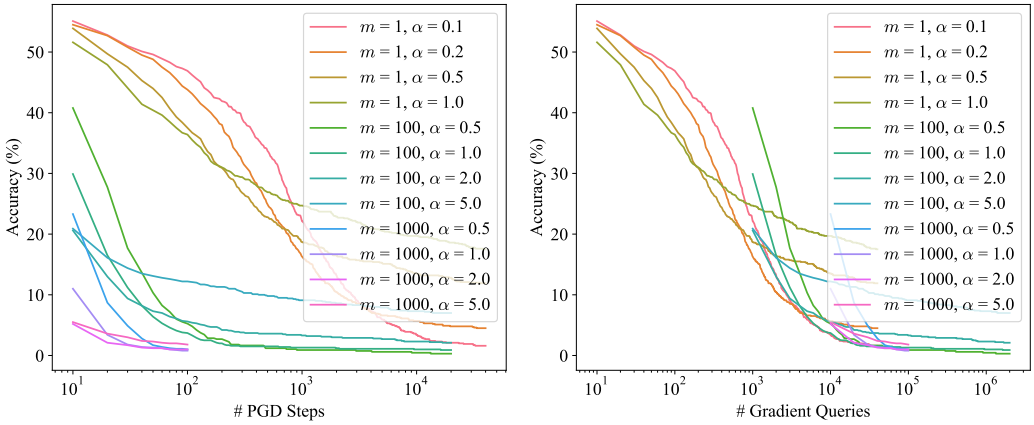
As we can observe, standard PGD without EOT achieves better performance when the step size is small, yet the application of EOT requires larger step sizes to perform better. We conjecture that this is because EOT reduces the variance of gradients so the attack algorithm can take a larger step, yet standalone PGD only gets noisy gradients and is only "confident" to take a small step.

However, this may not always prevent PGD from converging to a competitive solution. For example, we evaluate the discontinuous activation [39] defense with different attack settings, where the attack adds Gaussian noise around the input to estimate the correct gradient. The convergence curves of different settings are demonstrated in Figure 8.

When we examine the convergence in terms of PGD steps in Figure 8a, applying EOT obtains better gradients and quickly decreases the defended model's accuracy to zero. However, when we examine the convergence in terms of the total number of gradient queries in Figure 8b, we observe that (1) PGD without EOT given a smaller learning rate and (2) PGD with EOT given a larger learning rate have almost the same convergence behavior. This interesting observation suggests that standard PGD attacks may be sufficient in some cases if using a carefully fine-tuned learning rate. For example, Sitawarin et al. [33] showed that PGD attacks could be significantly improved by applying the AggMo [21] optimizer, which leverages multiple momentum terms.

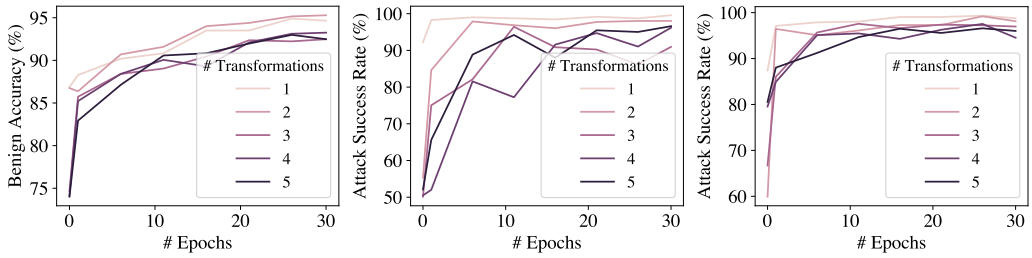
E.2 Inability to Remove Invariance that Does Not Hurt the Utility

Some recent works also suggest that one could gain robustness by removing invariance that does not hurt the utility [32]. However, this may not be the case for defenses with a larger randomization space. For example, we evaluate the performance of BaRT when it applies to models during the



(a) Accuracy under Attack (view by PGD Steps) (b) Accuracy under Attack (view by Gradient Queries)

Figure 8: Evaluation of the discontinuous activation [39] defense with EOT-m and step size α.



(a) No Attacks (b) Targeted Attacks (c) Untargeted Attacks

Figure 9: Performance of the BaRT defense on ImageNette and models during fine-tuning.

fine-tuning process (same experiment as in Section 6.3). As shown in Figure 9, the robustness has already dropped significantly before the model achieves invariance that preserves most of the utility.

E.3 Visualization of Adversarial Perturbation

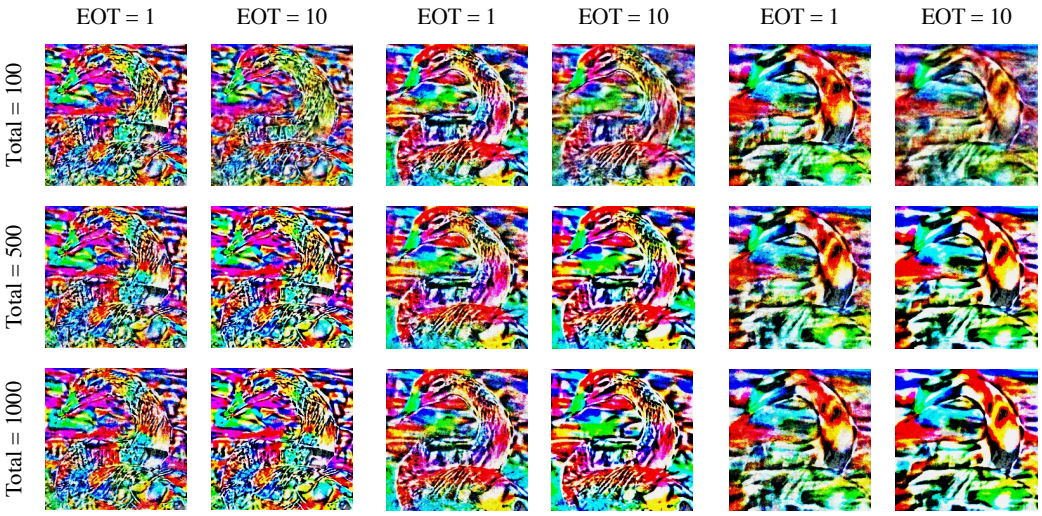
Figures 10 and 11 show the adversarial perturbation created by PGD attacks with and without EOT.

Settings. When given C gradient queries in total, we run PGD for (1) C steps without EOT and (2) $C/10$ steps with EOT of 10 samples. All attacks use ℓ_∞-norm budget $\epsilon = 8/255$ and step size $\alpha = 1/255$. The target model is a ResNet-50 defended by randomized smoothing with Gaussian noise $\sigma \in \{0.25, 0.50, 1.00\}$. We adopt pre-trained models from Cohen et al. [6].

Visualization. We randomly choose an image (id 5000) from the ImageNet validation set. For the benign image x and its adversarial example x', the perturbation is written as $\delta := x' - x \in [-\epsilon, \epsilon]$. We normalize it to $\delta' := \delta/(2\epsilon) + 0.5 \in [0, 1]$ and multiply it by 0.95 for better visualization.

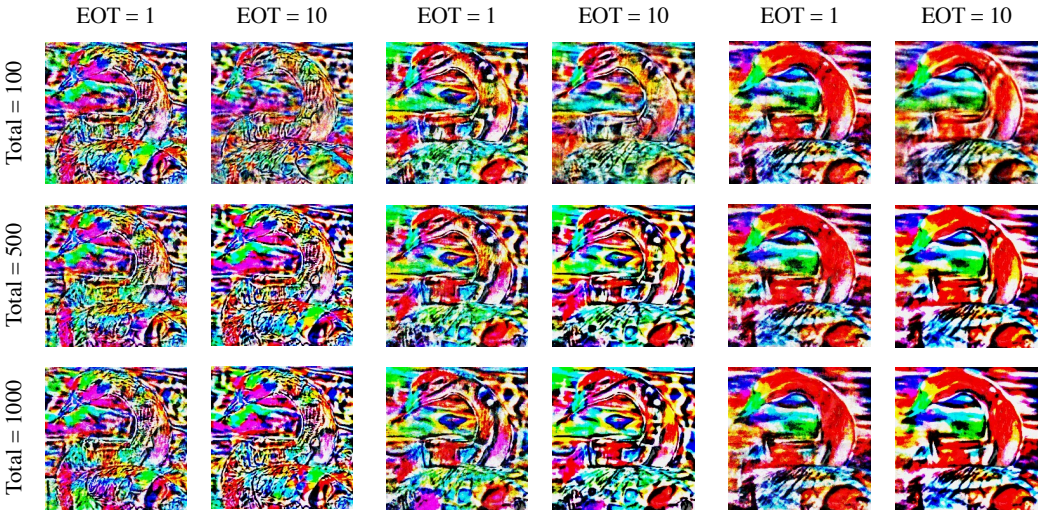
Compare PGD with and without EOT. For models of the same noise level, applying EOT leads to slightly smoother (or less noisy) adversarial perturbation. This observation shows that EOT computes more stable gradients. Besides, the above effect becomes more significant when (1) the model has a higher level of randomness (i.e., large σ), or (2) the attack runs in the targeted mode. These are the scenarios where applying EOT benefits more, which correspond to our findings in Section 6.2.

Compare PGD on models with different degrees of randomness. If we compare the visualization across different models, we can observe that models with a higher level of randomness produce smoother adversarial perturbation (even without applying EOT). While this observation seems counterintuitive, we note that these models are all fine-tuned on noisy data. As a result, making the model invariant to randomness also smoothes out the gradient, which removes the expected robustness



(a) $\sigma = 0.25$ (b) $\sigma = 0.50$ (c) $\sigma = 1.00$

Figure 10: Adversarial perturbation created by *untargeted* PGD attacks with and without EOT.



(a) $\sigma = 0.25$ (b) $\sigma = 0.50$ (c) $\sigma = 1.00$

Figure 11: Adversarial perturbation created by *targeted* PGD attacks with and without EOT.

provided by randomness. This observation corresponds to our theoretical model in Section 5 and empirical findings in Section 6.3.

E.4 Additional Experiments on CIFAR10

In Table 2, we evaluate a few defenses on CIFAR10, including randomized activation pruning [8] and discontinuous activation [39]. Such defenses cannot defend against standard PGD attacks without applying EOT. This shows that our findings hold on small and large input spaces.

In this section, we add an experiment to show that our findings in Section 6.3 also hold on CIFAR10. To this end, we evaluate the randomized smoothing defense [6] on CIFAR10 with the more challenging targeted attack. Specifically, we run the standard PGD attack for 100 steps, with budget $\epsilon = 8/255$, step size $\alpha = 1/255$, target label 9, and EOT of $m = 20$ samples. The target models are ResNet-110 pre-trained on noisy data by Cohen et al. [6]. We evaluate four different noise levels $\sigma \in$

$\{0.12, 0.25, 0.50, 1.00\}$. For each noise level, we run the *same* attack on models before and after fine-tuning on data perturbed by such noise. The results are shown in Tables 3 and 4.

For all noise levels, fine-tuning models to obtain invariance improves the benign accuracy as expected. During this procedure, however, we can observe that the defense becomes less effective when the model recovers more invariance. In particular, the attack is nearly ineffective for $\sigma \in \{0.25, 0.50, 1.00\}$ when the model has low invariance, yet starts to work as the model recovers invariance. This observation is consistent with our findings on ImageNet in Section 6.3.

Table 3: Benign accuracy of models with low and high invariance to the defense's randomness.

	$\sigma = 0.12$	$\sigma = 0.25$	$\sigma = 0.50$	$\sigma = 1.00$
Before Fine-tuning (Low Invariance)	23.4%	14.7%	12.3%	10.1%
After Fine-tuning (High Invariance)	**83.6%**	**77.9%**	**71.1%**	**56.7%**

Table 4: Attack success rate on models with low and high invariance to the defense's randomness.

	$\sigma = 0.12$	$\sigma = 0.25$	$\sigma = 0.50$	$\sigma = 1.00$
Before Fine-tuning (Low Invariance)	52.1%	1.1%	0.0%	0.0%
After Fine-tuning (High Invariance)	**63.1%**	**29.5%**	**18.1%**	**12.3%**

F More Discussions

F.1 Discussions about DiffPure [24]

In parallel to our work, DiffPure [24] adopts a complicated stochastic diffusion process to purify the input images. This defense belongs to an existing line of research that leverages generative models to pre-process input images and hence removing the potential adversarial perturbation [18, 31, 34]. In this section, we elaborate on the implications of our work for DiffPure.

Firstly, DiffPure is the defense that our work expects to avoid. As we indicated in Section 1, a thorough evaluation of stochastic pre-processing defenses typically requires significant modeling and computational efforts. DiffPure is a new example of such defenses — it has a complicated solver of stochastic differential equations (SDE) and requires "high-end NVIDIA GPUs with 32 GB of memory[9]." Our initial experiment shows that it takes several hours to attack even one batch of 8 CIFAR10 images on an Nvidia RTX 2080 Ti GPU with 11 GB of memory, and we received an out-of-memory error when attempting ImageNet with batch size 1. Because of these complications and computational costs, fully understanding its robustness requires substantially more effort than a previous stochastic pre-processing defense BaRT [27].

Given this challenging arms race between attacks and defenses, our work provides empirical and theoretical evidence to show that stochastic pre-processing defenses are fundamentally flawed. They cannot provide inherent robustness (like that from adversarial training) to prevent the existence of adversarial examples. Hence, future attacks may break it. As a result of these findings, future research should look for new ways of using randomness, such as those discussed in Section 7.

Secondly, DiffPure matches our theoretical model. DiffPure has two consecutive steps:

1. Forward SDE adds noise to the image to decrease invariance like Equation (5). The model becomes more robust because the input distribution is shifted.
2. Reverse SDE removes noise from the image to recover invariance like Equation (6). The model becomes less robust because the shifted input distribution is recovered.

These two steps are consistent with our characterization of stochastic pre-processing defenses in Section 5. While our work mainly focuses on trained invariance (through model fine-tuning), an auxiliary denoiser (like Reverse SDE) can achieve a similar notion of invariance. Hence, we expect our arguments about the robustness-invariance trade-off to hold here as well.

[9] https://github.com/NVlabs/DiffPure

Finally, Our findings raise concerns with the way DiffPure claims to obtain robustness. The above discussion finds no evident difference between DiffPure and our model in Section 5. When the Reverse SDE is perfect, we should achieve full invariance in Equation (7) and expect no improved robustness — attacking the whole procedure is equivalent to attacking the original model (if non-differentiable and randomized components are handled correctly). Hence, our findings raise concerns with the way DiffPure claims to obtain robustness.

Driven by the above concerns, we carefully review DiffPure's evaluation and identify red flags:

1. They only used 100 PGD steps and 20–30 EOT samples in AutoAttack [7]. This setting is potentially inadequate based on our empirical results in Table 2. Even breaking a less complicated defense requires far more steps and samples.
2. Previous purification defenses cannot prevent adversarial examples on the manifold of their underlying generative model or denoiser [2]. However, DiffPure did not discuss this attack, i.e., whether it is possible to find an adversarial example of the diffusion model such that it remains adversarial (to the classifier) after the diffusion process. This strategy is different from its current evaluation, which attacks the whole pipeline with BPDA and EOT.

These red flags suggest that there is still room for improving DiffPure's evaluation.

Summary. DiffPure matches our theoretical characterization of previous stochastic pre-processing defense. Thus, we expect our findings to hold here as well. Unfortunately, we cannot finish the evaluation of the above discussions due to their high computational costs. However, this challenge is exactly what our work aims to mitigate — we can identify concerns with the way robustness is achieved without needing to design adaptive attacks, and our findings have motivated us to identify red flags in their evaluation. We hope our work can increase the confidence of future research towards understanding the robustness of defenses sharing a similar assumption.

F.2 Insights for Designing Attacks and Defenses Regarding Randomness

While systematic guidance for designing defenses (and their attacks) remains an open question, we attempt to summarize some critical insights for this direction as follows.

Guidance for Attacks.

1. Attackers aiming to evaluate defenses (i.e., not merely breaking them) should start with standard attacks before resorting to more involved attack strategies like EOT. This helps form a better understanding of the defense's fundamental weakness.
2. Stochastic pre-processors cannot provide inherent robustness, so an effective attack should exist. Although there has not been a systematic way to design or find such attacks, our work provides general guidelines to help with this task.
3. Stochastic pre-processors provide robustness by invariance, so attackers can examine the model invariance to check the room for improvements.

Guidance for Defenses.

1. The current use of randomness is not promising. Defenses should decouple robustness and invariance; otherwise, future attacks may break them.
2. Defenses should look for new ways of using randomness, such as those below or beyond the input space. Below-input randomness divides the input into orthogonal components, like modalities [41] and independent patches [17]. Beyond-input randomness routes the input to separate components, like non-transferable models [43].
3. Randomness should force the attack to target all possible (independent) subproblems, where the model performs well on each (independent and) non-transferable subproblem. In this case, defenses can decouple robustness and invariance, hence avoiding the pitfall of previous randomized defenses.
4. Randomness alone does not provide robustness. Defenses must combine randomness with other inherently robust concepts to improve robustness.

F.3 Limitations and Potential Negative Societal Impacts

Finally, we discuss the limitations and potential negative societal impacts of this work.

Limitations. This paper mainly focuses on stochastic *pre-processing* defenses, thus we cannot comment on the effectiveness of stochastic defenses that are not based on input transformations. However, we do evaluate a few such defenses in Table 2 and observe similar results for our own interests, such as randomized activation pruning [8] and discontinued activation [39]. Given this observation, we believe our findings on stochastic pre-processing defenses are potentially generalizable to all stochastic defenses. We leave this exploration to future work.

Due to the limitation of computing resources, we are unable to evaluate the full BaRT [27] defense on ImageNet. To mitigate this problem, we evaluate a subset of BaRT on the smaller ImageNette dataset. Since the primary objective of this work is to study the limitations of such defenses but not to break them, we believe the limitations that we observe on a subset of BaRT are reasonably generalizable to the full set of BaRT. Other work studying this defense made a similar choice [33].

We are unable to evaluate the parallel defense DiffPure [24] due to its significantly high computational requirements. Given this limitation, we provide a thorough discussion in Appendix F.1 and explain that DiffPure is consistent with our model, hence we expect our findings to hold here as well.

Potential Negative Societal Impacts. This paper investigates the limitations of stochastic pre-processing defenses against adversarial examples. While the publication of this research may be used by attackers to create stronger attacks, we argue such considerations are out-weighted by the benefits of enabling defenders to understand the weaknesses of existing defenses. Moreover, our evaluation mainly involves existing attacks and previously broken defenses, thus we do not observe novel negative societal impacts. Our main objective is to uncover the fundamental weaknesses of such defenses, both empirically and theoretically, thereby raising the awareness of how to design proper stochastic defenses that avoid inadvertently weak evaluations and overestimated security.