IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 16, 2021, accepted January 22, 2022, date of publication January 25, 2022, date of current version February 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3146198

Mitigating Black-Box Adversarial Attacks via Output Noise Perturbation

MANJUSHREE B. AITHAL AND XIAOHUA LI, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY 13902, USA
Corresponding author: Xiaohua Li (xli@binghamton.edu)

This work was supported in part by the U.S. Air Force Office of Scientific Research (AFOSR) under Grant FA9550-21-1-0229.

ABSTRACT In black-box adversarial attacks, attackers query the deep neural network (DNN) and use the query results to optimize the adversarial samples iteratively. In this paper, we study the method of adding white noise to the DNN output to mitigate such attacks. One of our unique contributions is a theoretical analysis of gradient signal-to-noise ratio (SNR), which shows the trade-off between the defense noise level and the attack query cost. The attacker's query count (QC) is derived mathematically as a function of noise standard deviation. This will guide the defender to find the appropriate noise level for mitigating attacks to the desired security level specified by QC and DNN performance loss. Our analysis shows that the added noise is drastically magnified by the small variation of DNN outputs, which makes the reconstructed gradient have an extremely low SNR. Adding slight white noise with a very small standard deviation, e.g., less than 0.01, is enough to increase QC by many orders of magnitude yet without introducing any noticeable classification accuracy reduction. Our experiments demonstrate that this method can effectively mitigate both soft-label and hard-label black-box attacks under realistic QC constraints. We also prove that this method outperforms many other defense methods and is robust to the attacker's countermeasures.

INDEX TERMS Deep learning, adversarial machine learning, black-box attack, noise perturbation, performance analysis.

I. INTRODUCTION

Along with the rapid development of deep neural networks (DNNs), there are a lot of online services, such as Clarifai API, Google Photos, advertisement detection and fake news filtering, etc., that highly rely on DNNs. Nevertheless, an intriguing issue is that DNNs are highly susceptible to small variations in input data [1]. Online DNN servers suffer from adversarial attacks where the attackers can slightly change the input data to make DNNs give false results or misclassification [2].

Depending on the knowledge about the DNNs that the attackers have, adversarial attacks can be classified into white-box attacks [1], [3]–[5] and black-box attacks [6]–[13]. The former assumes that the attackers have complete knowledge of the deep network, while the latter assumes that the attackers have limited knowledge, typically some output information of the DNNs. Compared with white-box attacks, black-box attacks are more realistic threats to real-world practical applications.

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

In general, black-box attacks need to estimate certain gradients via the output information of the deep networks obtained through querying to optimize iteratively their adversarial samples. This is true even for attacks that are claimed "gradient-free". The query cost is thus a critical constraint to attackers. Over the recent years, more and more efficient black-box attack methods have been developed and they can now generate adversarial samples with only a few hundred of queries [7], [14]. Considering this fast-increasing threat, it is the right time to develop effective defense methods [15], [16] since most existing defense techniques are shown to provide a false sense of defense [17].

In this paper, we study the performance of the simple output noise perturbation technique as a defense against black-box attacks, where the defender (or the DNN) adds white noise to the DNN outputs. Since it is impossible to find a technique that can completely stop attackers of unlimited resources, we focus on mathematical analysis of the attack-defense trade-off in terms of defense noise level and attack query count (QC). Such a theoretical analysis is critical for defense study because it is both computationally intractable and theoretically incredible to guarantee defense just with

experiments. Specifically, we express QC as a function of noise standard deviation σ, with which the defender can easily apply appropriate noise to prevent attacks up to certain performance loss and security levels. For example, our results demonstrate that small noise with $\sigma \leq 0.01$ can prevent black-box attacks with 10^6 query count budget over the MNIST, CIFAR10 and IMAGENET datasets without any noticeable classification accuracy loss.

The major contributions of this paper are outlined as follows.

- We develop a novel analysis framework to study the trade-off between defense noise level and attack QC mathematically instead of heuristically only via experiments. The signal-to-noise ratio (SNR) of the noisy gradients is derived, and it exhibits that small noise is magnified by the small DNN outputs. The attacker's QC is shown to be increased by many orders of magnitude even with an extremely small noise perturbation.
- We analyze the properties of the proposed noise perturbation method and show that the method is robust to various countermeasures of the attackers. We also observe that quantization and output-correlated noise do not perform well. The latter explains that output noise perturbation is better than other randomization or gradient obfuscation methods.
- We experiment with a list of representative black-box attack algorithms, including both soft-label and hard-label attacks. The results fit well with the analysis and demonstrate the effectiveness of the output noise perturbation method against black-box attacks.

This paper is organized as follows. Related works are presented in Section II. The noise perturbation method is studied in Section III. Experiments are conducted in Section IV. Conclusions are given in Section V.

II. RELATED WORK

Black-box attacks can be subdivided into three major classes: transfer-learning-based attacks, soft-label attacks, and hard-label attacks [8]. Transfer-learning-based attacks exploit the fact that an adversarial input to one deep network may also be adversarial to another deep network [2].

Soft-label attacks assume that the logit information is available to the attacker, either fully or partially. Narodytska and Kasiviswanathan [18] used random perturbation and local search to look for adversarial samples. Hayes and Danezis [19] trained a generator neural network to generate adversarial samples. Chen *et al.* developed the zeroth-order optimization (ZOO)-based attacks [6], where they reconstructed gradients from output logits using zeroth-order gradient estimators. Ilyas *et al.* [8] applied the natural evolution strategies (NES) to estimate the gradients. Tu *et al.* [7] improved the ZOO-based attacks with the AutoZOOM algorithm, which used autoencoders to generate gradient search directions. Cheng *et al.* [20] combined the transfer-learning and ZOO-based attack techniques.

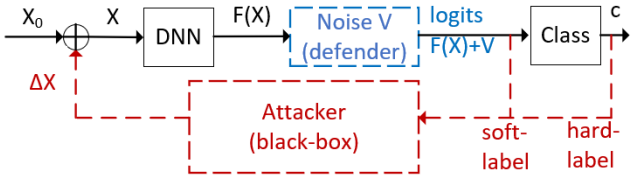

FIGURE 1. The model of black-box attack (both soft-label and hard-label attack) and output noise perturbation defense. Blue-dashed line shows defender (or DNN)'s activity, while red-dashed line shows attacker's activity.

Hard-label attacks assume that only hard decisions of DNN outputs are available. Within this class, Brendel *et al.* [13] exploited large perturbation to generate adversarial samples and used fine-tuning to reduce adversarial image distortion. Ilyas *et al.* [8] picked a target image and fine-tuned it toward the original image. Cheng *et al.* [9], [10] applied randomized gradient-free ZOO techniques.

On the defense side, a majority of existing studies are focused on white-box attacks. Most existing black-box defense techniques are in fact borrowed from their white-box version. A large number of defense techniques were proposed based on the idea of gradient masking or gradient obfuscation, e.g., defensive distillation [16], non-differentiable classifiers [21], input randomization [22], network structure randomization [23], etc. Unfortunately, almost all of them were defeated shortly after their publications via the so-called expectation-over-transformation (EOT) technique [17], [24], [25]. Today, the most effective way is perhaps adversarial training where adversarial samples are used to train the network [26]–[28], but the performance is not reliable for unknown attacks.

To the best of our knowledge, the simple output noise perturbation method has not been studied in-depth. Dong *et al.* [14] experimented with a long list of white-box/black-box attack/defense algorithms, but without this one. All the other reported noise perturbation techniques injected noise into the input or the network, not the output [28]–[31]. The reason is perhaps they were obtained from white-box attacks where adding noise to network outputs was of no use. Lee *et al.* [32] used output noise perturbation but for model stealing attack only and also without mathematical analysis.

III. ANALYSIS OF OUTPUT NOISE PERTURBATION

A. BLACK-BOX ATTACK AND DEFENSE MODEL

Consider a DNN that classifies an image x_0 into class c. The DNN outputs (softmax) logits $F(x_0)$, where F is the DNN's nonlinear mapping function. The classification result is $c = \arg\max_i F_i(x_0)$, where F_i denotes the ith element function of F.

The objective of the adversarial attacker is to generate an image $x = x_0 + \Delta x$ such that the DNN classifies it as $t = \arg\max_i F_i(x) \neq c$. The difference Δx should be as small as possible. For soft-label black-box attacks, the attackers query the DNN to obtain the input-output pair $(x, F(x))$, as shown in Fig. 1, with which they can minimize the following loss

function to search for the adversarial sample x [7],

$$f(x) = \mathcal{D}(x, x_0) + \lambda \mathcal{L}(F(x), t), \tag{1}$$

where $\mathcal{D}(\cdot,\cdot)$ is a distance function and $\mathcal{L}(\cdot, t)$ is the loss function. Typical distance functions are norms $\|x - x_0\|_p$. Typical loss functions include the cross-entropy [8] and the C&W loss [24].

If the logit $F(x)$ is not available, the attackers can adopt the hard-label attack strategy with the queried input-output pair (x, c). A common approach is to first find an image x_t in the target class, i.e., $\arg\max_i F_i(x_t) = t$. Then, starting from x_t, the attackers iteratively estimate new x in the target class so as to minimize (1) under the constraint $\lambda = 0$ and $\arg\max_i F_i(x) = t$.

In this paper, we consider that the DNN defends itself by adding noise v to the logit and providing either the perturbed output $F(x) + v$ or the perturbed class decision $\arg\max_i[F(x) + v]_i$. The objective is to prevent the attacker from optimizing (1). We assume that v is an independent and identically distributed (i.i.d.) Gaussian random vector with zero mean and standard deviation σ, i.e., $v \sim \mathcal{N}(0, \sigma^2 I)$, where I is an identity matrix. We consider low magnitude or small noise throughout our analysis.

Definition 1: Small noise is defined as the noise v whose standard deviation σ is small so that $\log(1 + v) \approx v$ is valid almost surely.

In other words, the standard deviation σ is several orders-of-magnitude smaller than 1, i.e., $\sigma \ll 1$. We also assume that the DNN satisfies $\|F(x) - F(y)\| \leq L\|x - y\|$ with a local Lipschitz constant L.

The performance of defense can be measured by three metrics: **attack success rate (ASR)**, **query count (QC)**, and **input distortion**. Robust defense makes the black-box attacks to have low ASR, high QC, or high distortion. While reducing ASR to near 0 is the major defense goal, defenses leading to high query count are also effective. High query count means the attack is infeasible due to query cost limit.

Since ASR is hard to analyze theoretically, in this paper, we analyze QC as a function of noise standard deviation σ. We will show that small σ can lead to prohibitively high QC to the attacker while introducing minimal classification performance loss to the DNN. Such analysis results will be demonstrated by extensive experiments which show that ASR can be reduced to a very small level under small σ and pre-set QC limit.

B. OUTPUT NOISE PERTURBATION TO MITIGATE NES TARGETED ATTACK

In this section, we analyze the defense performance. To save space, detailed analysis is presented only for the soft-label targeted attack with the NES method [8]. We will extend the analysis to other attacks in Section III-C to show that our analysis framework is general.

In [8], the soft-label NES targeted attack towards class t is conducted by minimizing the softmax cross-entropy loss function $f(x) = -\log F_t(x)$, where $F_t(x)$ is the softmax value of the target class. We skip the distance term $\mathcal{D}(x, x_0)$ from (1) in order to consider the most challenging defense situation. For the attacker, it is easier to find an adversarial sample without the distortion constraint. The NES algorithm uses gradient descent to minimize the loss function. In each iteration, the attacker conducts J queries with the randomly perturbed inputs $x + \beta u_j$, where u_j is a random tensor and β is the search variance. With the so-called antithetic sampling, both $x + \beta u_j$ and $x - \beta u_j$ are used as query inputs. According to the NES principle [8], the attacker can estimate the gradient from the derivative of the average loss $f(x)$ as

$$\bar{g} = -\frac{1}{J}\sum_{j=1}^{J/2}\left[\frac{u_j}{\beta}\log F_t(x + \beta u_j) - \frac{u_j}{\beta}\log F_t(x - \beta u_j)\right], \tag{2}$$

which can be written as

$$\bar{g} = \frac{1}{J}\sum_{j=1}^{J/2} g_j, \quad g_j = u_j \frac{1}{\beta}\log\frac{F_t(x - \beta u_j)}{F_t(x + \beta u_j)} = a u_j. \tag{3}$$

See (10)(11) for more explanation. Note that g_j can be expressed as the attacker-generated search-direction tensor u_j multiplying a deterministic scalar multiplication factor a.

Theorem 1: Under white Gaussian noise perturbation with $v \sim \mathcal{N}(0, \sigma^2 I)$, the gradient $g_j = a u_j$ becomes $g_j = A u_j$, where $A = a + \frac{1}{\beta}\log Z$ with random variable $Z \sim \mathcal{N}(1, \sigma_Z^2)$,

$$\sigma_Z^2 = \frac{\sigma^2}{F_t^2(x - \beta u_j)} + \frac{\sigma^2}{F_t^2(x + \beta u_j)}. \tag{4}$$

With small noise Definition 1, we have $A \sim \mathcal{N}(a, \sigma_Z^2/\beta^2)$.

The proof is presented in Appendix A. Theorem 1 tells us that the noise randomizes the gradient estimation. To understand the degree that v randomizes the estimated gradient, we can evaluate the signal-to-noise ratio (SNR) of A defined as $SNR = \frac{a^2}{E[|\beta^{-1}\log Z|^2]}$, where $E[.]$ denotes mathematical expectation. We call it the SNR of the noisy gradient.

Lemma 1: Under small β and small noise, the SNR of A is

$$\begin{aligned} SNR &= \frac{\left[F_t(x - \beta u_j) - F_t(x + \beta u_j)\right]^2 F_t^2(x - \beta u_j)}{\sigma^2\left[F_t^2(x - \beta u_j) + F_t^2(x + \beta u_j)\right]} \\ &\leq \frac{2L^2\beta^2}{\sigma^2}. \end{aligned} \tag{5}$$

See Appendix B for the proof. Lemma 1 shows that the SNR is very low because the output variation $\Delta F_t = |F_t(x - \beta u_j) - F_t(x + \beta u_j)|$ and β are very small in practice. A very small SNR makes A to have signs opposite to a with high probability, which changes the gradient descent toward the wrong direction and thus prevents the attacker's optimization from converging.

The ill-convergence can be studied through the attack QC. To derive QC as a function of noise level σ, we consider the following approach since it is difficult to find QC expressions for deep networks. Consider the iterative gradient-descent minimization of $f(x) = 1/2[F(wx) - F(wx^*)]^2$, where w is

the weight of the input DNN layer and x is the DNN input. We assume that the function $F(wx)$ is monotone between the starting point wx_0 and the optimal point wx^* because otherwise there is no guarantee of convergence. The minimization is conducted as $x_{n+1} = x_n - a\partial f(x_n)/\partial x_n$, $n = 0, 1, \cdots$. Our objective is to find the ratio R, i.e., the ratio of the iteration number needed when using a constant learning rate a to that when using the random learning rate $A = a + \sqrt{SNR}v$ with noise v.

Theorem 2: If the learning rate a is small such that $(1 - a\lambda)^n \approx 1 - na\lambda$, then

$$R = \frac{1}{4}\left(\sqrt{K^2+4} - K\right)^2, \quad K = \frac{\Phi^{-1}(\epsilon)\sqrt{a\lambda}}{\sqrt{SNR(1-\eta/v_0)}}, \tag{6}$$

where η and ϵ are small probabilities, λ and v_0 are constants related to w, x_0 and x^*, and $\Phi^{-1}(\epsilon)$ is the inverse of the standard normal cumulative distribution function.

The proof is in Appendix C. From the proof, we also see that R can be used as an estimation of $QC(noise)/QC(noiseless)$, i.e., the ratio of QCs between the case with noise perturbation and the case without noise perturbation.

The relationship between the defense noise level σ and the attack QC can be readily analyzed based on (5) and (6). In particular, if σ is small, then $R \sim 1/SNR$, i.e., increases with 1/SNR. As a rule of thumb, R, $1/SNR$, and $\sigma/\Delta F_t$ change linearly with each other.

Now we can summarize the reasons for the noise perturbation method being effective. First, from Lemma 1, the SNR of the estimated gradient becomes very low since the noise power σ^2 is amplified by the small β and small ΔF_t. Numerical results in Section IV-A show that SNR can be -100 dB or near 0. Second, from Theorem 1, the gradient becomes so random that it changes the search direction to the opposite with high probability, which prevents gradient search from converge. Finally, according to Theorem 2, low SNR makes the attack QC prohibitively high. Results in Section IV-A show R of 10^{10} and QC of 10^{15} for attacking IMAGENET images.

C. PROPERTIES OF OUTPUT NOISE PERTURBATION

In this subsection, we first show that our analysis framework and the noise perturbation method are general enough for many other black-box attack methods. Then, we show an important property, i.e., quantization noise and output-correlated noise are not effective. This will explain why the output noise perturbation method is better than other randomization or gradient obfuscation methods.

First, within the black-box attack community, the NES and ZOO are two major gradient estimation approaches. For ZOO, we consider the **AutoZOOM** algorithm [7] that minimizes the C&W loss function $f(t) = \log\left(F_{\max}(x)/F_t(x)\right)$, where $F_{\max}(x) = \{F_i(x) : i = \arg\max_j F_j(x), \forall j \neq t\}$, with the gradient estimator $g_j = \beta^{-1}(f(x + \beta u_j) - f(x))$ $u_j = au_j$.

Theorem 3: Under white Gaussian noise perturbation, the multiplication factor a becomes the noisy factor $A = a + \frac{1}{\beta}\log(Z_1 Z_2)$, where $Z_1 \sim \mathcal{N}(1, \sigma^2/F^2_{\max}(x) + \sigma^2/F^2_{\max}(x + \beta u_j))$ and $Z_2 \sim \mathcal{N}(1, \sigma^2/F^2_t(x) + \sigma^2/F^2_t(x + \beta u_j))$. In addition, when σ is small, we have

$$A \sim \mathcal{N}\left(a, \frac{\sigma^2}{\beta^2}\left(\frac{1}{F^2_{\max}(x)} + \frac{1}{F^2_{\max}(x+\beta u_j)} + \frac{1}{F^2_t(x)} + \frac{1}{F^2_t(x+\beta u_j)}\right)\right) \tag{7}$$

and the SNR of A satisfies $SNR \leq \frac{L^2\beta^2}{2\sigma^2}$.

See Appendix D for its proof. This theorem tells us that noise perturbation randomizes the AutoZOOM's gradient estimation similarly as it does for NES.

Second, our analysis method represented by Theorem 1 and Theorem 2 can be applied to analyze other black-box attacks as well. For example, the $\mathcal{N}$**attack** algorithm [33] uses the NES-estimated gradients to learn adversarial distributions. Assume the adversarial samples have a certain distribution with mean μ, then the $\mathcal{N}$attack algorithm finds μ via optimization $\mu_{t+1} = \mu_t - \eta\bar{g}$. Obviously, the noise perturbed $\bar{g}$ can hardly make the updating converge. As another example, for the **partial-information NES attack** [8], the authors propose to start from a target image and then apply the NES algorithm to estimate the gradient so as to modify the target image to become similar to the original image. Noise perturbation is still effective to randomize the estimated gradients.

Third, untargeted attacks can be analyzed similarly with just a change of loss functions. i.e., change the loss function to $f(x) = \log F_t(x)$ for the cross-entropy loss or $f(x) = \log\frac{F_t(x)}{F_{\max}(x)}$ for the C&W loss, where $F_t(x)$ is the true logit value of the input x, and $F_{\max}(x)$ denotes the maximum logit value excluding the true logit. Our analysis and conclusions are still valid. Specifically, we still have noisy multiplication factor $A = a + \beta^{-1}\log Z$ with an extremely low SNR which leads to high QC and low ASR.

Next, one especially interesting case is the attacks that claim "gradient-free", i.e., they do not aim to estimate the DNN's true gradient. Nevertheless, they still need to estimate a gradient-like searching direction for iterative optimization. An example is the **SimBA** attack [12] which searches over an orthogonal set of basis u such as Fourier basis. During each iteration it gets a new sample $x_{n+1} = x_n - \eta u$ that minimizes the loss $P(Y|X)$. With noise perturbation $P(Y|X)+v$, it can be shown that the probability of choosing a basis in each iteration is randomized by a random variable $Z \sim \mathcal{N}(1, \sigma_Z^2)$, where $\sigma_Z^2 = \sigma^2/P^2(Y|X)$. The SNR is extremely small since noise σ^2 is amplified by the small probability $P(Y|X)$.

Another example is the **GenAttack** [11] which uses the genetic algorithm to search for adversarial samples. In each iteration, it selects the sample x_n that maximizes the target label $F_t(x_n)$. Since noise perturbation changes $F_t(x_n)$, it makes such a sample selection procedure unreliable, and suffers from a noisy random variable $Z \sim \mathcal{N}(1, \sigma_Z^2)$ where $\sigma_Z^2 = \sigma^2/F_t^2(x_n)$. The SNR is also very small.

Furthermore, another especially interesting case is the hard-label attack. The **label-only NES attack** [8] starts from

a target image and uses its random variations to query the DNN. The binary query results are used to construct a measure similar to $F_t(x)$. Obviously, noise perturbation can change the hard-label which makes this measure very noisy and reduces the SNR of the estimated gradients. Our analysis framework can still be applied. In addition, the **BoundaryAttack** [13] uses the query results of random updating $x_n + \eta u$, i.e., whether $x_n + \eta u$ remains as the adversarial sample, to determine whether using this random u to update x_n or not. Obviously, noise perturbation leads to unreliable query results so that x_n is updated in the wrong direction with high probability. The unreliable query decision can be expressed as the SNR of query results following our analysis framework.

Finally, as to the important properties of noise perturbation, an interesting question is whether output quantization noise can be used. Another interesting question is whether the noise must be white.

Lemma 2: Noises created by output quantization (to 2 or more bits) or noises highly correlated with DNN outputs make the random variable Z have very small σ_Z^2, and thus are not effective to mitigate black-box attacks.

The proof is shown in Appendix E. The lemma gives a good explanation for the limited or failed defending performance of existing network randomization defenses. For example, Liu *et al.* [29] suggest adding noise to each convolutional layer but not the final output layer, whose net effect is to create output perturbations that are highly correlated with the true output logits. Its noise perturbation effect is in fact reduced by the network. Another drawback is that adding noise to the DNN input or mid layers makes it harder for the DNN to maintain classification accuracy since the perturbation effect to DNN output is out of control.

D. ROBUSTNESS TO ATTACKER's COUNTERMEASURES AND ADAPTIVE ATTACKS

The output noise perturbation method is robust to various counter-defense techniques that the attacker may adopt. Countermeasures are also called adaptive attack in [34]. First, the attacker may try to increase ΔF_t and β to reduce their noise amplification effects. However, ΔF_t is usually out of the attacker's control. Large β leads to worsening gradient estimation accuracy, which in fact reduces SNR of A.

Second, the attacker may adopt the EOT or gradient averaging strategy that has been shown effective to invalidate gradient obfuscation defenses in white-box attack scenarios [17]. Nevertheless, EOT is not as effective in our case as one would expect. In principle, EOT finds the average gradient $\bar{g} = 1/J \sum_j g_j$, similar to (3). Transformed images that the attacker uses to query the DNN can be written as $x + \Delta x_j$, where Δx_j denotes the difference caused by the random transform. The attacker still gets a noise perturbed output $F(x + \Delta x_j) + v_j$ to construct g_j. The estimated gradient is still random which is worse because of the randomized Δx_j. In this scenario, the accuracy of $\bar{g}$ can not be guaranteed, even if J is large. In the worst case, independent g_j may make $\bar{g} \to 0$.

TABLE 1. Statistical DNN output parameters obtained from validation datasets (without noise perturbation). ACC: classification accuracy. Mean $F_t(x)$: average softmax output values (excluding the top-1 pick). Mean/median/std ΔF_t: mean, median, and standard deviation of output variation ΔF_t.

Model	ACC	Mean $F_t(\mathbf{x})$	β	Mean ΔF_t	Median ΔF_t	std ΔF_t
MNIST model	0.99	4e-4	1e-3	5e-6	1e-21	1e-4
CIFAR10 model	0.83	1e-3	1e-3	2e-4	3e-8	1e-3
IMAGENET (InceptionV3)	0.78	3e-4	1e-5	5e-8	4e-9	1e-6

Finally, the best countermeasure is perhaps to estimate a by querying the DNN repeatedly with the same x. This is the optimal strategy to estimate a constant from noisy samples. Note that this is different from finding average $\bar{g}$ with different x. Theoretically the attacker can average out noise and get a reliable estimation with a large number of repeated queries. We are interested to study the QC in this case.

Theorem 4: If the attacker conducts N repeated queries with the same data x, it gets N samples $y_n = a + 1/\beta \log z_n$, $n = 1, \cdots, N$. Assume $a > 0$. The minimum number of samples N required to estimate a as $\hat{a}$ with $P[\hat{a} < 0] < \epsilon$ for some ϵ is

$$N = \frac{2\sigma^2}{F_t^2(x)} \left(\frac{\Phi^{-1}(\epsilon)}{e^{\frac{2\sigma^2}{F_t^2(x)} - a\beta} - 1} \right)^2, \tag{8}$$

where $\Phi^{-1}(\epsilon)$ is the inverse of the standard normal cumulative distribution function.

The proof is shown in Appendix F. We can see that small $F_t(x)$ leads to large N.

IV. EXPERIMENTS

A. NUMERICAL EVALUATION OF SNR AND QC

To evaluate numerically the tradeoff between performance loss (specified by σ or SNR) and defense security (specified by QC), we need to know the averages of output logits F_t and its variation ΔF_t. For this we trained a 4-layer CNN model for the MNIST dataset (conv(32), conv(64), dense(1024), dropout(0.2), dense(10)), a 7-layer CNN model for the CIFAR10 dataset (conv(64), conv(128), conv(128), conv(256), conv(256), conv(512), conv(10), averagepool), and used the Inception V3 model for the IMAGENET dataset. First, using their validation datasets (10,000 images for MNIST and CIFAR10, and 50,000 images for IMAGENET), we calculated the statistical parameters of DNN outputs, which are shown in Table 1. We applied random u_j to calculate output variation ΔF_t. Second, we added noise with various σ to the outputs and calculated output SNR and ACC degradation. The results in Fig. 2(a) clearly show that there is almost no ACC degradation when noise $\sigma \leq 0.02$. From Fig. 2(b) we also find that the output SNR is high.

Furthermore, using the mean ΔF_t data in Table 1, we calculated the SNRs of A of the NES attack (Lemma 1) and showed them in Fig. 2(b). The SNRs were drastically reduced to very

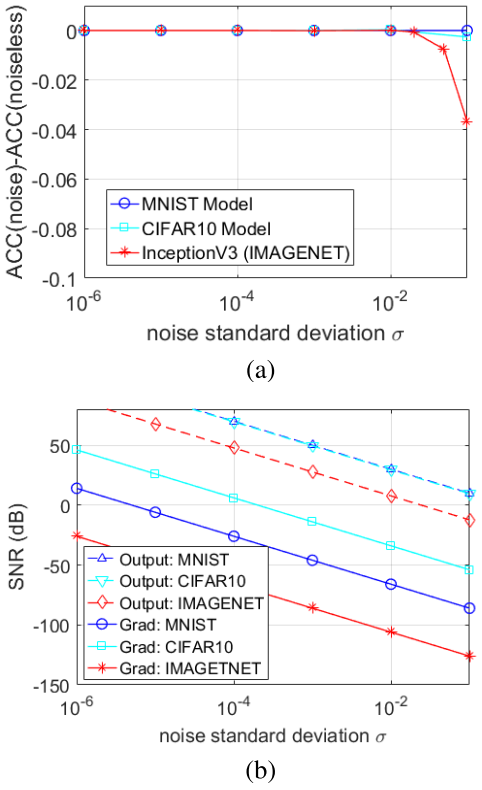

FIGURE 2. (a) Classification accuracy degradation due to noise perturbation. (b) SNR of noisy model outputs $F(x) + v$ (Output) and SNR of gradient multiplication factor A (Grad).

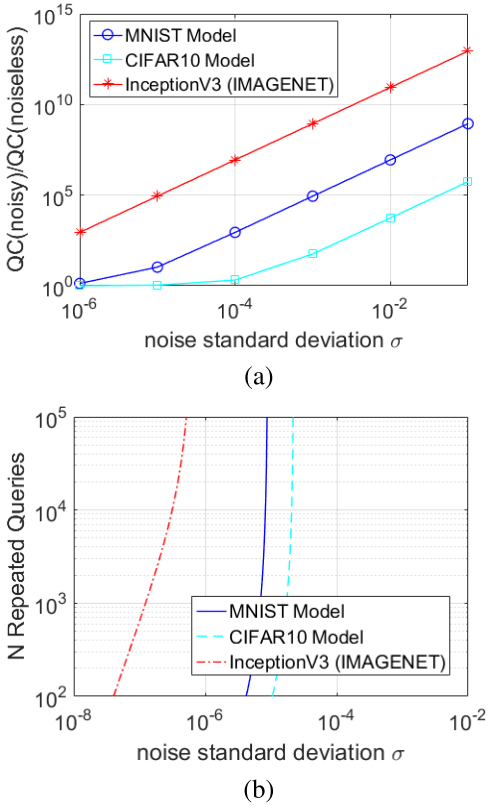

FIGURE 3. (a) Ratio R of query counts of noisy case to noiseless case. (b) Number of repeated queries N required to estimate a so that $P[\hat{a} < 0] < 0.3$ when $a > 0$.

small numbers. At $\sigma = 0.01$, the SNRs were -66dB, -34dB and -106dB for the MNIST, CIFAR10 and IMAGENET models, respectively. Such low SNRs made A completely different from a.

Next, to evaluate the QC ratio R with (6), we assume $\eta = \epsilon = 0.01$, $a = 0.1$, $\lambda = 2$, $v_0 = 1$. The increase of R as a function of σ is shown in Fig. 3(a). At $\sigma = 0.01$, we have $R = 9 \times 10^6, 5 \times 10^3, 9 \times 10^{10}$ for the three models, respectively. Considering that today's state-of-the-art attack methods need around 10^3 queries to attack MNIST/CIFAR10 images and 10^5 queries to attack IMAGENET images, small noise perturbation with $\sigma = 0.01$ would increase the number of queries to 10^6 to 10^{15}, prohibitively high to attackers. Note that the much smaller median ΔF_t values shown in Table 1 will lead to even higher QCs.

From Fig. 3(a), for attackers with 1 million query budget, the defender can simply add very small noise with $\sigma = 0.001, 0.01$, and 0.0001 to mitigate them over the MNIST, CIFAR10 and IMAGENET datasets, respectively. Even smaller noise, such as $\sigma \leq 10^{-4}$, is effective for well-trained models (such as MNIST) or models with a large number of classes (such as IMAGENET) that have very small ΔF_t. The defender can conveniently apply appropriate small noise according to its output parameters and required security level.

Finally, to evaluate the QC of EOT-based countermeasure, we would like to calculate N of (8). Adopting mean $F_t(x)$ and β data in Table 1, $\epsilon = 0.3$ and $a = 1$, the N values as function of σ are shown in Fig. 3(b). A huge number of repeated queries was needed to estimate each gradient g_j, which made this countermeasure technique impractical. Especially, when $\sigma \geq 10^{-4}$, no realistic N could be found to estimate the gradient g_j to the correct direction with 70% probability.

B. EXPERIMENTS OVER TARGETED ATTACKS

From Section III-B, the QC needed for generating an adversarial image under our noise perturbation defense can be 10^{15} or more. This means that it is computationally prohibitive to conduct experiments to search for the limit of QC. Considering this, as an alternative, instead of looking for QC limit, we followed the common practice to look for ASR under a pre-set realistic QC in our subsequent experiments.

We conducted experiments on the three widely used benchmark datasets in adversarial machine learning: MNIST, CIFAR10 and IMAGENET. We randomly sampled 1,000 images from the validation dataset for both MNIST and CIFAR10 to evaluate the performance of attack and defense algorithms. For IMAGENET, we randomly sampled 200 images from the validation dataset. We stuck to the default MNIST and CIFAR10 models used in the original

TABLE 2. Targeted soft-label attacks: Attack success rate (ASR%) versus defense noise standard deviation σ. ZOO [6], ZOO+AE and AZ+AE and AZ+Bi (AutoZOOM) [7], GenAttack [11], SimBA-pixel and SimBA-DCT [12], NES [8] and NES/PI (NES Partial Information attack where only the top-1 pick's confidence score is available [8]). QC limit is the maximum number of iterations the attack algorithms run.

Dataset	Attack Method	QC Limit	No Noise	Noise Standard Deviation σ					
				10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
MNIST	ZOO	1e5	99.44	77.78	58.00	28.67	10.67	10.11	10.33
	ZOO+AE	1e5	99.64	47.44	31.56	19.78	16.44	13.44	12.11
	AZ+AE	1e5	100.00	73.33	58.89	37.33	20.33	15.22	12.89
	AZ+Bi	1e5	99.89	91.67	24.78	17.00	14.00	12.44	13.44
	GenAttack	1e5	95.38	38.19	30.85	23.61	12.75	9.85	5.13
CIFAR10	ZOO	1e5	97.00	20.67	14.22	11.67	13.56	10.00	8.56
	ZOO+AE	1e5	99.00	52.33	42.00	32.67	23.44	17.00	17.22
	AZ	1e5	100.00	70.44	56.78	42.11	31.22	19.56	16.67
	AZ+Bi	1e5	99.33	38.00	17.56	14.22	12.67	13.00	12.56
	GenAttack	1e5	98.76	34.75	29.65	21.96	16.13	10.42	6.46
	Simba-pixel	2e4	96.31	96.29	90.93	43.5	21.23	12.16	6.9
	Simba-DCT	2e4	97.14	97.14	89.77	49.85	27.12	16.39	10.21
IMAGE -NET	ZOO	2e5	76.00	0.00	0.00	0.00	0.00	0.00	0.00
	ZOO+AE	2e5	92.00	10.00	0.00	0.00	0.00	0.00	0.00
	AZ+AE	1e5	100.00	0.00	0.00	0.00	0.00	0.00	0.00
	AZ+Bi	1e5	100.00	0.00	0.00	0.00	0.00	0.00	0.00
	NES	1e6	100.00	80.00	58.00	20.00	12.00	2.00	0.00
	GenAttack	1e6	100.00	70.00	20.00	0.00	0.00	0.00	0.00
	SimBA-pixel	6e4	100.00	92.00	62.00	2.16	0.00	0.00	0.00
	SimBA-DCT	6e4	96.5	55.00	50.00	2.13	0.05	0.00	0.00
	NES/PI	1e6	93.6	89.4	21.1	0.00	0.00	0.00	0.00

attack source code. For IMAGENET, all the attack algorithms used the pretrained Inception V3 model with a clean ACC of 0.78.

We experimented with a list of state-of-the-art black-box attack algorithms. For soft-label attacks, we experimented with ZOO [6], ZOO+AE and AZ+AE and AZ+Bi (AutoZOOM) [7], GenAttack [11], SimBA-pixel and SimBA-DCT [12], P-RGF [20], NES [8] and NES/PI (NES Partial Information attack where only the top-1 pick's confidence score is available [8]). For hard-label attacks, we experimented with OPT attack [9], Sign-OPT attack [10], NES/Label-Only attack [8], Boundary attack [13], and Hop-SkipJump attack [35]. We considered only the ℓ_2 attack version of these algorithms. Noise with standard deviation σ from 1e-6 to 0.1 was added to the softmax logit in the range of [0, 1]. We compared the performance of our defense algorithm with the JPEG Compression [15] and Input Randomization [22] defense algorithms.

For fair comparison, we used the original source code of the attack algorithms with their default hyper-parameter settings (represented as **no-noise** results). Especially, default QC limit was kept, which was in fact set as the maximum number of attack iterations. We inserted our noise addition defense subroutine to the source code. In practice, we could not add truly i.i.d. noise since the DNN should have softmax outputs in [0, 1]. We replaced negative elements with their absolute values and clipped the values over 1. The ACC of the noise perturbed DNN is not shown because the model accuracy degrades very little as shown in Fig. 2(a).

1) ASR OF TARGETED ATTACKS

Table 2 shows the ASR of the soft-label attack methods under our proposed noise perturbation defense method. Compared with near 100% ASR of the original noiseless case, the ASR of all the attack methods reduced significantly in presence of the proposed defense. On MNIST, small noise with standard deviation as small as $\sigma = 0.001$ was enough to degrade ASR from 100% to below 20%. On CIFAR10, small noise with σ as small as 0.01 was enough to degrade ASR to below 20%. Note that ASR can not be smaller than 10% theoretically for these two datasets because a random guess among 10 classes will result to 10% correctness. On IMAGENET, even smaller noise with σ as small as 10^{-4} was enough to reduce ASR to below 20%. Note that adding such small noise to DNN's output would lead to negligible (near 0%) ACC degradation according to Fig. 2(a). All these results fit well with our analysis in Section III-B and numerical results in Section IV-A. By all means, QC over 10^{15} is needed to break our noise perturbation defense over IMAGENET, which is much higher than the preset QC limits in these attack algorithms.

For hard-label attacks, experiment results in Table 3 showed that a low noise standard deviation $\sigma = 0.001$ was effective and $\sigma = 0.01$ successfully reduced ASR to below 25%. Note that the observation in Fig. 2(b), i.e., the ACC did not degrade for such small σ, was for normal images with high enough classification confidence only. The added noise could change the top-1 labels in case the classification confidence was not high enough, which happened frequently in the mid of the attacker's optimization procedure. This prevented the attack algorithm from converging. We should also note that in hard-label attacks, the attacker needs a large number of queries, over 2.5 million queries, to generate an adversarial image even when there is no noise perturbation. This is because the gradient is already very noisy with low SNR.

2) ASR COMPARISON WITH EXISTING DEFENSE METHODS

In Table 4 we compare the output noise perturbation defense method with two existing defense algorithms: JPEG Compression [15] and Input Randomization [22]. As seen from

TABLE 3. Targeted hard-label attacks: Attack success rate (ASR%) of the hard-label OPT attack [9], Sign-OPT attack [10], NES/Label-Only attack [8], Boundary attack [13], and HopSkipJump attack [35].

Dataset	Attack Method	No Noise	Noise Standard Deviation σ 10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
MNIST	OPT	100.0	76.0	41.0	17.0	7.0	**5.0**	**13.0**
	Sign-OPT	90.0	71.0	42.0	14.0	5.0	**0.0**	**0.0**
CIFAR10	OPT	81.0	67.0	58.0	35.0	33.0	**25.0**	**16.0**
	Sign-OPT	81.0	75.0	49.0	39.0	7.0	**7.0**	**4.0**
IMAGENET	NES/Label-Only	90.0	90.0	90.0	85.0	45.0	**21.0**	**0.0**
	OPT	100.0	40.0	20.0	20.0	10.0	**20.0**	**10.0**
	Sign-OPT	100.0	60.0	60.0	30.0	0.0	**0.0**	**0.0**
	Boundary	100.0	90.0	75.0	32.0	0.0	**0.0**	**0.0**
	HopSkipJump	100.0	100.0	90.0	75.0	45.0	**0.0**	**0.0**

TABLE 4. ASR (%) comparison of three defense methods: output noise perturbation method, and two input randomization methods. Soft-label targeted attack.

Dataset	Attack Method	no Def.	JPEG [15]	Rand Input [22]	Our Method
CIFAR10	ZOO	97.0	11.2	-	**8.56**
	GenAttack	98.76	88.0	70.0	**10.42**
IMAGENET	NES	100.0	66.5	45.9	**0.0**
	GenAttack	100.0	89.0	-	**0.0**

the table, the proposed output noise perturbation method had the best defense result with the lowest ASR. Specifically, the two existing methods could not mitigate NES and GenAttack attacks on IMAGENET data satisfactorily, while our output noise perturbation method could reduce the ASR to near 0%.

3) QUANTIZATION AND OUTPUT-CORRELATED NOISE

For quantization noise, we quantized the outputs from 32-bit to 2-, 4-, and 8-bit. For output-correlated noise, the noise was generated as $v = \alpha F(x) + \epsilon$, where α was the correlation coefficient and $\epsilon \sim \mathcal{N}(0, 10^{-16}I)$ was the residual noise with a very small standard deviation 10^{-8}. Results in Table 5 clearly show that quantization did not mitigate the attacks. There was no change in ASR between the original (32-bit float) and the quantized cases. Similarly, correlated noise could not mitigate the attacks as well. The slight reduction in ASR at $\alpha = 0.001$ and $\alpha = 0.1$ was solely caused by the small residual noise ϵ.

TABLE 5. ASR (%) of the ZOO and AutoZOOM black-box attack algorithms under quantization noise and output-correlated noise. Soft-label targeted attack.

Dataset	Attack Method	Quantization 8-bit	4-bit	2-bit	Noise Corr α 10^{-3}	10^{-1}
MNIST	ZOO	100	100	100	-	-
	AZ+AE	94.5	94.5	94.5	-	-
CIFAR10	ZOO	100.00	100.00	100.00	-	-
	AZ+AE	99.89	99.89	99.89	86.78	84.89

Robust to Attacker's Countermeasures: We evaluated the performance of the output noise perturbation method under attacker's countermeasures or adaptive attacks. We considered that the attackers changed the parameter β of (4) or took EOT-like countermeasures, where the attackers used N repeated queries to average each g_j (8), or used more non-repeated queries (large J) to look for better average gradients (3).

First, we experimented with the NES targeted attack where the attackers chose various β to optimize attacks. Results shown in Table 6 clearly demonstrate that if β was deviated from the default (optimal) value of 1e-3, the ASR degraded. This means that our defense was robust to this type of adaptive attack.

TABLE 6. Attack success rate (ASR%) of the NES targeted attack [8] under noise perturbation defense along with the countermeasure where the attacker used different β. IMAGENET dataset. β =1e-3 is the default value.

β	No Noise	Noise Standard Deviation σ 10^{-6}	10^{-5}	10^{-4}	10^{-3}
1e-5	100.0	0.0	0.0	0.0	0.0
1e-4	100.0	10.0	0.0	0.0	0.0
1e-3	100.0	80.0	58.0	20.0	12.0
1e-2	100.0	75.0	44.0	0.0	0.0
1e-1	80.0	60.0	45.0	0.0	0.0

Second, for the countermeasure with N repeated queries, from the results in Table 7, we can say that our method was robust against this type of EOT-like countermeasures. There was no drastic change in ASR even when queries were increased to $N = 1000$, which means 3 orders of magnitude more QCs. Finally, for the countermeasure with

TABLE 7. Attack success rate (ASR) of the AutoZOOM targeted attack [7] under noise perturbation defense along with the countermeasure where the attacker used *N* repeated queries to average gradients.

Dataset	Repeated Queries (N)	Noise Standard Deviation σ 10^{-4}	10^{-2}
MNIST	1	37.33%	15.22%
	10	39.92%	16.73%
	100	45.49%	18.33%
	1000	48.28 %	22.71%
CIFAR10	1	42.11%	19.56%
	10	45.59%	23.95%
	100	58.73%	27.50%
	1000	54.29%	29.70 %

TABLE 8. Attack success rate (ASR) of the NES [8] and P-RGF [20] under noise perturbation defense along with the countermeasure where the attacker used higher *J* non-repeated queries to average gradients. Noise standard deviation $\sigma = 10^{-4}$. IMAGENET.

Attack	$J = 50$	$J = 100$
NES (targeted)	20.00%	15.78%
P-RGF (untargeted)	60.00%	57.89%

TABLE 9. Untargeted soft-label attacks: Attack success rate (ASR%) versus defense noise standard deviation σ. ZOO [6], AZ+AE [7], SimBA-pixel, SimBA-DCT [12], and P-RGF attack [20].

Dataset	Attack Method	No Noise	Noise Standard Deviation σ					
			10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
MNIST	ZOO	100.00	63.99	60.99	53.99	17.99	**0.99**	**0.0**
	AZ+AE	100.00	22.99	21.00	13.99	20.03	**0.00**	**0.00**
CIFAR10	ZOO	100.00	47.00	42.00	33.99	27.00	**26.00**	**19.99**
	AZ+AE	100.00	60.00	53.00	52.00	39.99	**14.00**	**10.00**
	SimBA-pixel	92.59	91.84	59.84	22.48	7.75	**2.69**	**1.21**
	SimBA-DCT	93.3	92.33	64.53	27.53	8.52	**3.47**	**1.56**
IMAGENET	ZOO	86.00	20.00	10.00	0.00	0.00	**0.00**	**0.00**
	AZ+AE	100.00	100.00	100.00	50.00	20.00	**20.00**	**10.00**
	SimBA-pixel	93.69	93.72	93.53	76.8	0.00	**0.00**	**0.00**
	SimBA-DCT	90.01	90.01	90.01	86.57	0.00	**0.00**	**0.00**
	P-RGF	98.0	96.00	90.0	60.0	46.0	**36.0**	**30.0**

TABLE 10. Untargeted hard-label attacks: Attack success rate (ASR%) of the hard-label OPT attack [9], Sign-OPT attack [10], Boundary attack [13], and HopSkipJump attack [35].

Dataset	Attack Method	No Noise	Noise Standard Deviation σ					
			10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
MNIST	OPT	98.0	95.0	86.0	35.0	22.0	**16.0**	**9.0**
	Sign-OPT	98.0	78.0	37.0	5.0	3.0	**4.0**	**0.0**
	Boundary	100.0	100.0	100.0	100.0	96.0	**64.0**	**2.0**
	HopSkipJump	100.0	100.0	100.0	100.0	90.0	**70.0**	**0.0**
CIFAR10	OPT	100.0	81.0	63.0	18.0	18.0	**17.0**	**13.0**
	Sign-OPT	100.0	69.9	59.1	31.3	2.4	**0.0**	**0.0**
	Boundary	100.0	100.0	100.0	100.0	98.0	**88.0**	**9.0**
	HopSkipJump	100.0	100.0	70.0	70.0	67.0	**60.0**	**10.0**
IMAGENET	OPT	100.0	60.0	60.0	0.0	0.0	**0.0**	**0.0**
	Sign-OPT	100.0	50.0	50.0	30.0	0.0	**0.0**	**0.0**
	HopSkipJump	100.0	100.0	95.0	90.0	80.0	**65.0**	**0.0**

larger J values, while the original NES and P-RGF attack algorithms both used $J = 50$, we experimented with $J = 100$ and the results are summarized in Table 8. We can see that using a higher J did not necessarily lead to better ASR. The results demonstrated that our noise perturbation method was also robust to this type of countermeasures. Note that the ASR of the untargeted attack (P-RGF) critically depends on the distortion threshold. We used the original relatively high distortion threshold for P-RGF which resulted in relatively high ASR.

C. EXPERIMENTS OVER UNTARGETED ATTACKS

For untargeted attacks, we also used their original source code with their default hyperparameters and just inserted our noise addition subroutine to the source code. The ASR results are shown in Table 9 for soft-label attacks and Table 10 for hard-label attacks. As can be seen, the additive noise with $\sigma = 0.01$ successfully mitigated all these attack methods.

A special note is that although the noise perturbation method could not be applied to mitigate transfer-learning attacks, this experiment demonstrated that the method was effective against the transfer-learning strengthened attack P-RGF [9]. The P-RGF attack applied a transfer-learning model to assist gradient estimation. Experimental data in Table 9 showed that the P-RGF had ASR reduced from 98% to 36% under $\sigma = 0.01$. The relatively high ASR of 36% was due to the strong transfer model and the larger L_2 distortion threshold used in the original source code. It is well known that untargeted attacks can be always successful as long as large distortion can be allowed. The ASR would drop to very low levels if the transfer model did not fit well with the black-box DNN or a small L_2 distortion level was used. As a matter of fact, the ASR of pure transfer-learning attacks, especially targeted attacks, is very low in practical applications [36].

TABLE 11. ASR (%) comparison of three defense methods in untargeted attack setting.

Dataset	Attack Method	Without Defense	JPEG [15]	Rand Input [22]	Our Method
CIFAR10	ZOO	100.0	28.7	-	**20.0**
IMAGENET	P-RGF	98.0	81.1	82.3	**30.0**

In Table 11 we compare the noise perturbation defense method with two existing defense algorithms: JPEG Compression [15] and Input Randomization [22]. As seen from the table, the proposed noise perturbation method had the best defense result.

D. DISTORTION AND SAMPLE IMAGES OF TARGETED/UNTARGETED ATTACKS

There were some successful adversarial images depending on the level of noise perturbations. We are interested to check whether these images were truly successful attacks. A successful attack requires low enough distortion. Therefore, we checked the L_2 distortion of these adversarial images. The comparison of L_2 distortion between the noiseless attacks and noise defenses is shown in Table 12. We can see that

TABLE 12. Per-pixel L_2 distortion ($\times 10^{-4}$) versus noise perturbation standard deviation σ.

	Dataset	Attack Method	No Noise	Noise Standard Deviation σ					
				10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
Targeted Attacks	MNIST	ZOO	29.27	79.34	81.12	95.56	111.7	112.32	115.28
		AZ+AE	77.17	82.79	95.17	109.6	143.77	163.73	167.87
		OPT	1.85	1.78	1.79	1.93	2.39	3.86	4.05
		Sign-OPT	1.54	1.25	0.88	0.29	0.25	3.79	0.27
	CIFAR10	ZOO	3.54	13.74	23.54	34.07	41.59	48.19	57.27
		AZ+AE	19.63	22.22	24.8	28.87	35.13	62.03	68.71
		SimBA-pixel	1.48	1.57	8.59	7.56	5.8	5.3	5.18
		SimBA-DCT	1.23	1.22	9.61	7.53	5.75	5.26	5.12
		OPT	0.37	0.42	0.39	0.5	1.17	2.83	3.3
		Sign-OPT	0.25	0.23	0.19	0.15	0.19	0.3	0.31
	IMAGENET	ZOO	-	-	-	-	-	-	-
		AZ+AE	-	-	-	-	-	-	-
		NES	0.41	2.99	4.3	4.22	11.18	18.46	21.97
		SimBA-pixel	14.22	7.22	6.3	-	-	-	-
		SimBA-DCT	12.22	6.02	5.35	5.25	5.09	-	-
		OPT	53.37	58.19	69.52	68.58	109.69	126.28	129.11
		Sign-OPT	5.34	5.12	5.48	7.62	-	-	-
Untargeted Attacks	MNIST	ZOO	20.36	33.08	34.03	36.54	52.3	66.81	74.9
		AZ+AE	35.8	49.2	52.89	56.46	61.35	71.74	73.31
		OPT	1.07	1.07	1.07	1.1	1.94	5.12	5.52
		Sign-OPT	1.03	1.04	1.04	1.04	1.26	1.25	5.37
		Boundary	61.93	61.49	61.15	61.41	61.14	61.41	74.35
	CIFAR10	ZOO	2.4	34.5	32.4	54.4	61.8	73.2	84.4
		AZ+AE	13.6	21.5	14.16	20.52	23.26	21.92	35.92
		SimBA-pixel	3.81	4.94	14.11	7.43	5.58	5.28	5.14
		SimBA-DCT	3.579	5.36	13.68	7.61	5.54	5.28	5.16
		OPT	0.15	0.15	0.15	0.18	0.6	2.03	2.5
		Sign-OPT	0.12	0.12	0.12	0.13	0.3	1.36	2.51
		Boundary	28.68	28.57	28.62	28.63	29.12	29.05	29.69
	IMAGENET	ZOO	0.0023	21.8	23.3	25.4	25.4	25.35	27.00
		AZ+AE	0.27	0.27	0.26	1.02	1.65	2.07	2.37
		P-RGF	8.34	6.92	8.89	23.63	29.3	34.01	37.55
		SimBA-pixel	0.19	0.19	0.19	1.02	-	-	-
		SimBA-DCT	0.19	0.2	0.21	0.32	-	-	-
		OPT	21.54	21.53	22.65	-	-	-	-
		Sign-OPT	1.70	8.52	9.68	24.47	-	-	-

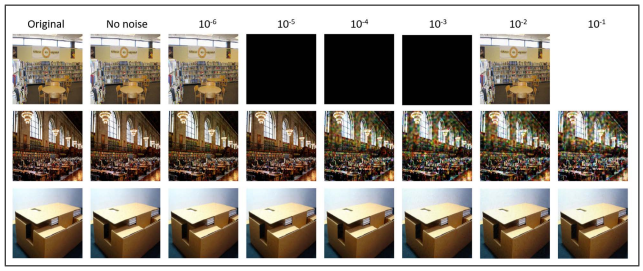

FIGURE 4. IMAGENET successful adversarial samples. From top to bottom: Samples obtained by NES (targeted), AutoZOOM (untargeted), P-RGF (untargeted) attack algorithms, respectively. From left to right: Original images and adversarial images obtained at $\sigma = 0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$, respectively.

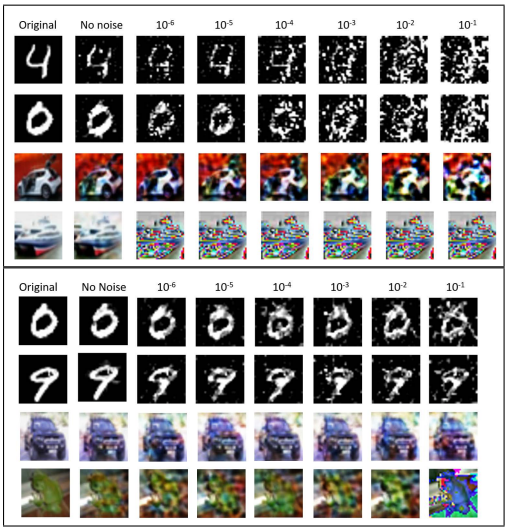

FIGURE 5. MNIST and CIFAR10 successful adversarial samples obtained by AutoZOOM. Top: targeted attack. Bottom: untargeted attack. From left to right: Original images, Adversarial outputs at $\sigma = 0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$.

the L_2 distortion increased with noise level σ. The distortion under $\sigma = 0.01$ was several times larger than those of noiseless attack. This means that even if the attacks were considered successful, the adversarial samples had high distortion. On the IMAGENET dataset, the ZOO and AutoZOOM algorithms had no data in targeted attacks because their ASR was 0. P-RGF, NES, AutoZOOM, and ZOO had L_2 distortion thresholds ranked from large to small. Their ASR also ranked from high to low under noise perturbation.

Fig. 4 shows some sample IMAGENET images generated by the adversarial algorithms. Heavier distortions can be seen when $\sigma \geq 10^{-4}$. Especially, some images obtained

by the NES targeted attacks were black-out but were still classified as successful attacks. Fig. 5 shows the visual effects of adversarial MNIST and CIFAR10 images. Similarly, the adversarial examples at higher σ could no longer deceive human perception, especially in targeted attacks.

V. CONCLUSION

In this paper, we studied the addition of white noise to DNN's output as a defense against black-box adversarial attacks. Noisy gradient is theoretically analyzed, which shows that the added noise is drastically amplified by the small logit variation. Small noise is thus effective to mitigate attacks while without degrading the DNN performance. The trade-off between the defender's noise level and the attacker's query count is analyzed mathematically. Extensive experiments verified the theoretical analysis and demonstrated that white noise perturbation can effectively mitigate black-box attacks under realistic query cost constraints.

APPENDIX A
PROOF OF THEOREM 1

As outlined in Section III-B, the NES targeted attack algorithm minimizes the cross-entropy loss

$$f(x) = -\log F_t(x) \tag{9}$$

assuming the label is hot-one coded, where $F_t(x)$ is the DNN output corresponding to the target class t. The NES algorithm minimizes the loss iteratively via gradient descent and in each iteration the gradient is estimated as

$$\bar{g} = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{\beta}u_j f(x+\beta u_j), \tag{10}$$

where J queries with random direction tensors u_j are conducted to obtain DNN output $F_t(x+\beta u_j)$ as well as loss $f(x+\beta u_j)$. Antithetic sampling is adopted in [8] which changes (10) to (3). Antithetic sampling means that both $x+\beta u_j$ and $x-\beta u_j$ are used to query the DNN.

To study the noise perturbation effect on the estimated gradient, it is sufficient to focus on just

$$g_j = u_j\frac{1}{\beta}\log\frac{F_t(x-\beta u_j)}{F_t(x+\beta u_j)}. \tag{11}$$

To simplify notation, we can write g_j as the attacker-generated tensor u_j multiplying a scalar multiplication factor a, i.e.,

$$g_j = au_j, \quad a = \frac{1}{\beta}\log h(x),\ h(x) = \frac{F_t(x-\beta u_j)}{F_t(x+\beta u_j)}. \tag{12}$$

With white Gaussian noise v added to the DNN output $F(x)$, the equation (12) becomes

$$g_j = Au_j, \quad A = \frac{1}{\beta}\log\tilde{h}(x) \tag{13}$$

where

$$\tilde{h}(x) = \frac{F_t(x-\beta u_j)+v_t(j+J/2)}{F_t(x+\beta u_j)+v_t(j)}. \tag{14}$$

The variables $v_t(j)$ and $v_t(j+J/2)$ are the noises added to the target class logits $F_t(x+\beta u_j)$ and $F_t(x-\beta u_j)$, respectively. Note that in antithetic sampling, we denote the noise added to the query $F_t(x-\beta u_j)$ as $v_t(j+J/2)$, where $j+J/2$ denotes the $(j+J/2)$th query.

The connection between the noiseless $h(x)$ and the noisy $\tilde{h}(x)$ is

$$\tilde{h}(x) = h(x)\frac{1+\frac{v_t(j+J/2)}{F_t(x-\beta u_j)}}{1+\frac{v_t(j)}{F_t(x+\beta u_j)}} \triangleq h(x)Z, \tag{15}$$

where we use the random variable Z to include all the noise terms. As a result, we have

$$A = a + \frac{1}{\beta}\log Z. \tag{16}$$

Since Z is the ratio of two independent Gaussian random variables, from [37] we can readily see that it can be approximated as a single Gaussian random variable $Z \sim \mathcal{N}(1,\sigma_Z^2)$ with unit mean and variance σ_Z^2 described by (4).

Furthermore, let $Z = 1+S$, where $S \sim \mathcal{N}(0,\sigma_Z^2)$. From the small noise Definition 1, we have that σ^2 is small enough so that $\log Z = \log(1+S) \approx S$. Therefore, from (16) we can get $A \sim \mathcal{N}(a, \sigma_Z^2/\beta^2)$. Theorem 1 is proved.

Remark 1: To understand why the noisy g_j can prevent the NES attack, it is helpful to have some idea about the value distribution of $h(x)$, $\log h(x)$ and Z. Since β is very small, we expect that $h(x)$ is near 1 due to the bounded local Lipschitz constant L. Then, $\log h(x)$ is around 0 and can be positive or negative. The value of a can also be positive and negative, and $|a|$ is usually small. This means that the factor a controls the gradient descent direction. The noise Z and thus A make the estimated gradient $g_j = Au_j$ randomized, with the gradient descent direction randomized in particular. For example, even if a is positive, A may become negative (see the numerical example in Remark 2). The random multiplication factor A has an accurate probability density function

$$p_A(x) = \beta e^{\beta(x-a)}\frac{1}{\sqrt{2\pi}\sigma_Z}e^{-\frac{1}{2\sigma_Z^2}e^{2\beta(x-a)}} \tag{17}$$

according to (16). However, (17) is too complex to conduct our subsequent SNR and QC analysis. Therefore, we have applied a further simplification to approximate A as a Gaussian random variable.

Remark 2: As an example, let $\beta = 10^{-3}$ as [8]. For a well designed DNN, $F_t(x)$ is usually around $1/C$ for a total of C classes. We consider $F_t(x) = 0.1, 0.01$, respectively. For positive a, we evaluate the probability that A becomes negative, which means that the gradient search direction becomes opposite to the true direction. With the cumulative distribution function (CDF) $P_A[A<x] = P_Z[Z<e^{-\beta(x-a)}]$, we can calculate $P_A[A<0]$. From the results shown in Fig. 6, we can see that a very small noise standard deviation $\sigma = 10^{-3}$ is enough to make $P[A<0] \approx 0.5$.

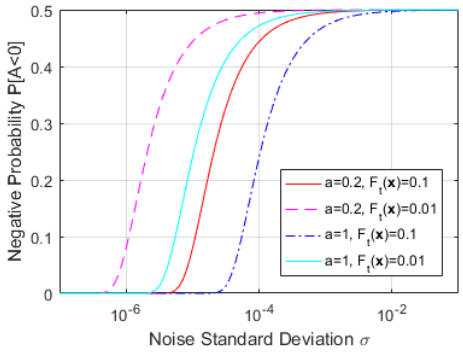

FIGURE 6. Probability of noisy multiplication factor A becoming negative when the true value a is positive. $\beta = 10^{-3}$.

APPENDIX B
PROOF OF LEMMA 1

From (3), i.e., the definition

$$a = \frac{1}{\beta}\log\frac{F_t(x-\beta u_j)}{F_t(x+\beta u_j)}, \tag{18}$$

we have

$$\begin{aligned} a^2 &= \frac{1}{\beta^2}\log^2\left(1+\frac{F_t(x-\beta u_j)-F_t(x+\beta u_j)}{F_t(x+\beta u_j)}\right) \\ &\approx \frac{\left[F_t(x-\beta u_j)-F_t(x+\beta u_j)\right]^2}{\beta^2 F_t^2(x+\beta u_j)}. \end{aligned} \tag{19}$$

We have applied the approximation $\log(1+x) \approx x$ for small x when deriving the approximation in (19). Because $\|F_t(x-\beta u_j)-F_t(x+\beta u_j)\| \le 2\beta\|u_j\|L$, under the assumption of small β, we can guarantee $\|F_t(x-\beta u_j)-F_t(x+\beta u_j)\| \ll F_t(x+\beta u_j)$ and thus the validity of (19). From (19) and utilizing the approximation $\log Z = Z-1$, the SNR is then

$$\begin{aligned} SNR &= \frac{a^2}{\frac{1}{\beta^2}E[(Z-1)^2]} \\ &= \frac{\left[F_t(x-\beta u_j)-F_t(x+\beta u_j)\right]^2}{F_t^2(x+\beta u_j)\sigma_Z^2}. \end{aligned} \tag{20}$$

Replacing σ_Z^2 with (3), after some straightforward deductions we can get (5).

Next, to derive the simplified upper bound in (5), consider the Lipschitz constraint assumption. From the left hand side of (5), we get

$$SNR \le \frac{L^2 4\beta^2\|u_j\|^2 F_t^2(x-\beta u_j)}{\sigma^2[F_t^2(x-\beta u_j)+F_t^2(x+\beta u_j)]}. \tag{21}$$

Without loss of generality, assuming $\|u_j\| = 1$ and using $F_t(x-\beta u_j) \approx F_t(x+\beta u_j)$, we get the SNR upper bound in the right hand side of (5). The lemma is proved.

Remark 3: Note that the SNR can be calculated numerically without applying the approximation in (19). The reason we apply the approximation here is to get a simplified SNR expression that outlines the major contribution factor $\Delta F_t = |F_t(x-\beta u_j) - F_t(x+\beta u_j)|$. Note also that the assumption of small β is not a severe constraint at all in practice. In most black-box attacks, such as [7], β is selected (and proved) to be less than or equal to the inverse of DNN input dimension d. Obviously, d is much larger than the DNN output dimension C (class number). Since $F_t(x+\beta u_j)$ on average is around $1/C$, β is thus much less than $F_t(x+\beta u_j)$ in most cases. This may be violated occasionally, but such occasional violations do not affect the SNR because the SNR is the average over all possible DNN outputs $F_t(x)$.

APPENDIX C
PROOF OF THEOREM 2

Consider the problem of minimizing

$$f(x) = \frac{1}{2}\|F(wx) - F(wx^*)\|^2 \tag{22}$$

with iterative gradient descent

$$x_{n+1} = x_n - a\frac{\partial f(x_n)}{\partial x_n}, \tag{23}$$

where a is a constant and small learning rate. In $F(wx)$, F denotes the mapping of DNN, w denotes the weight of the input layer, and x denotes the input. For notation simplicity, w and x are treated as matrix and vector. x^* denotes the optimal solution. In order for the gradient descent to converge to x^* from a starting point x_0 so we can count the total number of iterations, we have to assume that $F(wx)$ is a monotone function between the starting point wx_0 and the optimal point wx^*.

To further simplify our notation, without loss of generality, we assume $F(wx)$ is a monotonously decreasing function from wx_0 to wx^*. We also assume $wx_n \le wx^*$ for $n = 0, 1, \cdots$, which can be guaranteed with a small enough learning rate a and a starting point $wx_0 \le wx^*$. Our subsequent deduction can be easily extended to include other cases such as $F(wx)$ monotonously increasing, or some elements of $F(wx)$ monotonously increasing and others decreasing, or some elements of wx_n becomes greater than wx^*. In these cases, we just need to treat each element in each case individually.

The gradient is

$$\frac{\partial f(x_n)}{\partial x_n} = w^T F'[F(wx_n) - F(wx^*)] \tag{24}$$

where F' denotes the derivative of F with respect to its argument wx_n and w^T denotes the transposition of w. Then, the gradient updating is

$$x_{n+1} = x_n - aw^T F'[F(wx_n) - F(wx^*)]. \tag{25}$$

Next, we consider

$$wx_{n+1} = wx_n - aww^T F'[F(wx_n) - F(wx^*)] \tag{26}$$

instead to exploit the assumption of $wx_n \le wx^*$. From the Lipschitz assumption and monotonicity, we have $F(wx_n) - F(wx^*) \le L(wx^* - wx_n)$ for some constant L. Therefore,

$$wx_{n+1} \ge wx_n - aww^T F'L(wx^* - wx_n). \tag{27}$$

Using wx^* to subtract both sides, we get

$$\begin{aligned} wx^* - wx_{n+1} &\le wx^* - wx_n + aLww^T F'(wx^* - wx_n) \\ &= (I + aLww^T F')(wx^* - wx_n) \\ &= (I + aLww^T F')^{n+1}(wx^* - wx_0). \end{aligned} \tag{28}$$

Denote the largest eigenvalue of the matrix $Lww^T F'$ as $-\lambda$ where λ is a positive value. Note that the eigenvalues must be negative because otherwise (28) does not converge, which contradicts with the convergence assumption. In this special case, F' is negative because F is assumed monotonously decreasing. Let $v_n = \|wx^* - wx_n\|$. From (28), we have

$$v_n \le |1 - a\lambda|^n v_0, \tag{29}$$

where $v_0 = \|wx^* - wx_0\|$ or the initial distance from wx^*. If a is small so that $(1 - a\lambda)^n \approx 1 - na\lambda$, then, in order to guarantee $v_n \le \eta$ where η is a small constant, the number of iterations n must satisfy

$$N_a \ge \frac{1 - \eta/v_0}{a\lambda}. \tag{30}$$

Next, consider the case when the learning rate a is replaced by the noisy learning rate $A = a + \sqrt{SNR}v$ with noise $v \sim \mathcal{N}(0, 1)$. Equation (28) becomes

$$wx^* - wx_{n+1} \le \left(\prod_{i=0}^{n}(I + A_i Lww^T F')\right)(wx^* - wx_0) \tag{31}$$

where A_i denotes the learning rate in the ith iteration. Similarly, (29) becomes

$$v_n \le v_0 \prod_{i=0}^{n-1} |1 - A_i\lambda|. \tag{32}$$

In order to guarantee $v_n \le \eta$, a sufficient condition is

$$\prod_{i=0}^{n-1} |1 - A_i\lambda| \le \frac{\eta}{v_0}. \tag{33}$$

If both a and SNR is small, then (33) can be simplified to

$$1 - \lambda \sum_{i=0}^{n-1} A_i \le \frac{\eta}{v_0}, \tag{34}$$

which leads to

$$\sum_{i=0}^{n-1} A_i \ge \frac{1 - \eta/v_0}{\lambda}. \tag{35}$$

Since $A_i \sim \mathcal{N}(a, SNR)$ are independent Gaussian random variables, in order to make

$$P\left[\sum_{i=0}^{n-1} A_i < \frac{1 - \eta/v_0}{\lambda}\right] \le \epsilon \tag{36}$$

for some small probability ϵ, we need

$$\frac{(1 - \eta/v_0)/\lambda - na}{\sqrt{nSNR}} \le \Phi^{-1}(\epsilon). \tag{37}$$

Solving (37) for n, we get that the number of iterations needed when the learning rate becomes random A must satisfy

$$N_A \ge \frac{1}{4}\left[-\frac{1}{\sqrt{SNR}}\Phi^{-1}(\epsilon) + \sqrt{\frac{1}{SNR}\Phi^{-2}(\epsilon) + 4\frac{1 - \eta/v_0}{a\lambda}}\right]^2. \tag{38}$$

Using the lower bound of (30) and (38), we can get the ratio of required iterations between the case of a and the case of A as

$$\begin{aligned} R &= \frac{N_A}{N_a} \\ &= \frac{\left[-\frac{1}{\sqrt{SNR}}\Phi^{-1}(\epsilon) + \sqrt{\frac{1}{SNR}\Phi^{-2}(\epsilon) + 4\frac{1-\eta/v_0}{a\lambda}}\right]^2}{4\frac{1-\eta/v_0}{a\lambda}} \end{aligned} \tag{39}$$

which is just (6). The theorem is proved.

Remark 4: First, the proof is easier to understand if we consider w as a row vector and F as a scalar nonlinear monotone function. We present the general case with the matrix w in the proof. One can actually treat each row of w separately to get the same result. Second, although R is defined as the ratio of iterations, it equals to the ratio of query counts because there are a fixed number of queries conducted to estimate the gradients in each iteration.

Third, we argue that R can be used as an approximate estimation of $QC(noise)/QC(noiseless)$, i.e., the ratio of QCs between the case with noise perturbation and the case without noise perturbation in our black-box attack and defense models. It is well known that the QC expression is hard or impossible to derive for black-box attack to general DNNs because $F(x)$ is highly nonlinear/nonconvex and the black-box estimated gradient is not the true gradient. The key concept of our approach is that we consider a fixed optimization trajectory of the attacker from a starting input x_0 to the final adversarial input x^*. This trajectory is obtained by the attack's gradient descent minimization without noise perturbation. Along this trajectory, the mapping $F(x)$ can approximately be assumed as monotonously decreasing or piece-wise monotonously decreasing from x_0 to x^*. The attacker's estimated gradients can also be looked as true gradients with a on this trajectory. The effect of noise perturbation is changing the value a in each iteration to a random value A with certain SNR. As a result, the model and assumptions we made for deriving R in this theorem are valid for analyzing the DNN attack-defense models. Therefore, it is reasonable to claim that if the attacker uses N_a iterations to get the adversarial input x^*, it would needs R times more iterations in case the output noise perturbation changes a to A.

Finally, based on the $QC(noiseless)$ needed by the attackers when there is no noise perturbation (which can be obtained by experiments), we can estimate the $QC(noise)$ needed when there is noise perturbation by multiplying $QC(noiseless)$ with R. By this way, we can avoid the difficulty of finding the $QC(noise)$ directly with experiments. As shown by

our analysis, noise perturbation may increase $QC(noise)$ to some computationally prohibitive level, such as 10^{15} or more. When calculating R numerically, we can use a very small η/v_0 (because η is the desired small distance of final results wx_n to the targeting result wx^* and v_0 is the initial distance), and a very small ϵ (because $1-\epsilon$ is the confidence probability). We can use the average of a defined in (3) as $a\lambda$ in R. As a matter of fact, because SNR is usually small, the value R is not very sensitive to these parameters. From (6), it can be easily seen that $R \approx C_0/SNR$ where C_0 is a small constant.

APPENDIX D
PROOF OF THEOREM 3

Consider the targeted attack toward class t that is conducted by minimizing the loss function

$$f(x) = \log \frac{F_{\max}(x)}{F_t(x)}, \tag{40}$$

where $F_{\max}(x) = \{F_i(x) : i = \arg\max_j F_j(x), \forall j \neq t\}$ is the logit (softmax) value of the largest non-target element, and $F_t(x)$ is the logit value of the target element. Note that this is just the C&W loss function $\max\{0, \max_{j\neq t} \log F_j(x) - \log F_t(x)\}$. The first max is skipped because we consider the adversarial search stage when the target has not been reached yet.

If the DNN adds noise $v \sim \mathcal{N}(0, \sigma^2 I)$ to its output, then the attacker's loss becomes

$$f(x) = \log \frac{[F(x)+v]_{\max}}{[F(x)+v]_t}, \tag{41}$$

where $[F(x)+v]_{\max}$ and $[F(x)+v]_t$ denotes the maximum-valued element (exclude the tth element) and the tth element, respectively. According to the AutoZOOM attack algorithm [7], the gradient estimator used by the attacker is

$$\begin{aligned} g_j &= \frac{1}{\beta} u_j \big(f(x+\beta u_j) - f(x)\big) \\ &= \frac{1}{\beta} u_j \log \frac{\frac{[F(x+\beta u_j)+v_j]_{\max}}{[F(x+\beta u_j)+v_j]_t}}{\frac{[F(x)+v]_{\max}}{[F(x)+v]_t}}. \end{aligned} \tag{42}$$

where u_j is the vector of gradient direction which is pre-set and fixed, β is the smoothing parameter, v_j is the noise added by the DNN when the attacker queries with $x+\beta u_j$.

The estimated gradient equals to the vector u_j multiplying a scalar multiplication factor A, i.e.,

$$g_j = Au_j, \quad A = \frac{1}{\beta} \log \tilde{h}(x), \tag{43}$$

where

$$\tilde{h}(x) = \frac{[F(x+\beta u_j)+v_j]_{\max}/[F(x+\beta u_j)+v_j]_t}{[F(x)+v]_{\max}/[F(x)+v]_t}. \tag{44}$$

Define the noiseless term

$$h(x) = \frac{F_{\max}(x+\beta u_j)/F_t(x+\beta u_j)}{F_{\max}(x)/F_t(x)}. \tag{45}$$

Since the noise is small, the location of the maximum element does not change almost surely. We have

$$\begin{aligned} \tilde{h}(x) = h(x) &\times \frac{1 + v_{j,\max}/F_{\max}(x+\beta u_j)}{1 + v_{\max}/F_{\max}(x)} \\ &\times \frac{1 + v_t/F_t(x)}{1 + v_{j,t}/F_t(x+\beta u_j)}, \end{aligned} \tag{46}$$

where v_t and $v_{j,t}$ are the tth entry of the noise vectors v and v_j, respectively. The random variables $v_{\max}$ and $v_{j,\max}$ are the noises added to the maximum-valued elements of $F(x)$ and $F(x+\beta u_j)$, respectively.

Define

$$Z_1 = \frac{1 + v_{j,\max}/F_{\max}(x+\beta u_j)}{1 + v_{\max}/F_{\max}(x)}, \tag{47}$$

$$Z_2 = \frac{1 + v_t/F_t(x)}{1 + v_{j,t}/F_t(x+\beta u_j)}. \tag{48}$$

Each of Z_1 and Z_2 is the ratio of two independent Gaussian random variables and can be approximated as a single Gaussian random variable [37]. Specifically,

$$Z_1 \sim \mathcal{N}\left(1, \frac{\sigma^2}{F^2_{\max}(x)} + \frac{\sigma^2}{F^2_{\max}(x+\beta u_j)}\right) \tag{49}$$

$$Z_2 \sim \mathcal{N}\left(1, \frac{\sigma^2}{F^2_t(x)} + \frac{\sigma^2}{F^2_t(x+\beta u_j)}\right) \tag{50}$$

The probability density function (PDF) $p_Z(z)$ of the product $Z = Z_1 Z_2$ can be found based on [38].

Define $a = \frac{1}{\beta} \log h(x)$. Then from (43) and (46) we have

$$A = a + \frac{1}{\beta} \log Z_1 Z_2. \tag{51}$$

Therefore, we can see that noise perturbation randomizes the AutoZOOM's gradient estimation similarly as it does for NES-based attack method.

To derive A's distribution and SNR bound, when noise variance σ^2 is small enough, we have $\log Z_1 \approx Z_1 - 1$ and $\log Z_2 \approx Z_2 - 1$. Therefore, $A \approx a + \frac{1}{\beta}(Z_1 - 1) + \frac{1}{\beta}(Z_2 - 1)$, from which we can verify (7). In addition, the SNR bound can be proved following strictly the proof of (5). The theorem is proved.

Remark 5: When deriving (46), we have assumed $[F(x)+v]_{\max} = F_{\max}(x) + v_{\max}$ and also $[F(x+\beta u_j)+v_j]_{\max} = F_{\max}(x+\beta u_j) + v_{j,\max}$, which means small noise does not change the index of the maximum-valued elements. This is true almost surely under small noise perturbation. On the other hand, the noise may accidentally change the index of the maximum-valued element. In this case, the two elements, old $F_{max}(x)$ and new $F_{newmax}(x)$, have similar (almost identical) values since even tiny noise can switch their order. Therefore, (46) is still valid.

APPENDIX E
PROOF OF LEMMA 2

First, for output quantization, instead of outputting the full precision 32-bit logit values, the DNN can output $Q \geq 2$ bit

quantized logit values. Note that 1-bit quantization is actually hard-label outputs, and only the special hard-label attack methods can work. It is well known that quantization method introduces quantization noise. Under coarse quantization, attacks with the cross-entropy loss do not work because $a = 1/\beta \log(F_t(x-\beta u)/F_t(x+\beta u))$ quite often results in $a = 0$. However, the attacks with the C&W loss still work well. In other words, the quantization noise can not mitigate such attacks. To explain it, let us look at the proof of Theorem 4 and consider the noise term Z_2. The noises are the quantization residues of $F_t(x)$ and $F_t(x+\beta u_j)$, whose quantized values are the same, say Q, almost surely. This means $v_t = F_t(x) - Q$ and $v_{j,t} = F_t(x+\beta u_j) - Q$. We then have

$$Z_2 = \frac{2 - \frac{Q}{F_t(x)}}{2 - \frac{Q}{F_t(x+\beta u_j)}}. \tag{52}$$

Obviously, $\log Z_2$ is no longer randomly positive and negative. In other words, the variance of Z_2 is zero. Therefore, quantization noise can not mitigate the attacks.

Second, for output-correlated noise, let us look at the proof of Theorem 1. If the noise v is correlated to the output $F_t(x)$, then we have $v_t(j) = \sum_i \alpha_i F_t^i(x+\beta u_j) + \epsilon$ for a very small $\epsilon \to 0$, where α_i are correlation coefficients. From (15), it is easy to see that Z is now randomized by ϵ only, which means a very small σ_Z^2 with much-reduced noise perturbation effect. The attack mitigation effect is also reduced.

APPENDIX F
PROOF OF THEOREM 4

Let us re-iterate the problem setting first. In order to improve the accuracy of the estimation of g_j, or specifically, the estimation of

$$a = \frac{1}{\beta} \log \frac{F_t(x-\beta u_j)}{F_t(x+\beta u_j)}, \tag{53}$$

the attacker can repeatedly query the DNN with inputs $x-\beta u_j$ and $x+\beta u_j$. The noisy outputs are $F_t(x-\beta u_j) + v_{t1,n}$ and $F_t(x+\beta u_j) + v_{t2,n}$ in the nth query, where $v_{t1,n}$ and $v_{t2,n}$ are independent Gaussian random variables $\mathcal{N}(0,\sigma^2)$, $n = 1, \cdots, N$. The attacker uses the query results to calculate y_n as

$$y_n = \frac{1}{\beta} \log \frac{F_t(x-\beta u_j) + v_{t1,n}}{F_t(x+\beta u_j) + v_{t2,n}}, \tag{54}$$

for each n. From Theorem 1, we have

$$y_n = a + \frac{1}{\beta} \log z_n, \tag{55}$$

where $z_n \sim \mathcal{N}(1, \sigma_Z^2)$. To simplify notation, we let

$$\sigma_Z^2 = \sigma^2 \left(\frac{1}{F_t^2(x-\beta u_j)} + \frac{1}{F_t^2(x+\beta u_j)} \right) \approx \frac{2\sigma^2}{F_t^2(x)} \tag{56}$$

because $F_t(x-\beta u_j) \approx F_t(x+\beta u_j)$. The problem is to estimate a from y_n, $n = 1, \cdots, N$. We would like to find the N that is needed for estimating a reliably.

Lemma 3: Under small noise perturbation, the optimal estimator for the attacker to estimate a from y_n is the maximum likelihood estimator

$$\hat{a} = \frac{1}{N} \sum_{n=1}^{N} y_n - \frac{1}{\beta} \sigma_Z^2. \tag{57}$$

Proof: The distribution of y_n is

$$p_Y(y_n) = \beta e^{\beta(y_n - a)} p_Z\left(e^{\beta(y_n-a)}\right). \tag{58}$$

From the joint distribution $p(y_1, \cdots, y_N) = \Pi_{n=1}^N p_Y(y_n)$, we can obtain the maximum likelihood estimator by making the derivative $\partial \log p(y_1, \cdots, y_N)/\partial a = 0$. With some straightforward deductions, we have

$$\sum_{n=1}^{N} \left(e^{2\beta(y_n-a)} - e^{\beta(y_n-a)} - \sigma_Z^2 \right) = 0. \tag{59}$$

For small σ, $\log z_n$ is close to 0. Therefore, $\beta(y_n - a)$ is also very close to 0. We can apply the approximation $e^x \approx 1 + x$ to simplify (59) to

$$\sum_{n=1}^{N} \left((1+2\beta(y_n-a)) - (1+\beta(y_n-a)) - \sigma_Z^2 \right) = 0. \tag{60}$$

Then (57) is readily available. □

Now we are ready to prove Theorem 3. From (57), after some deductions, we can get

$$\begin{aligned} P[\hat{a} < 0] &= P\left[\frac{1}{N} \sum_{n=1}^{N} \left(a + \frac{1}{\beta} \log z_n \right) < \frac{\sigma_Z^2}{\beta} \right] \\ &= P\left[\frac{1}{N} \sum_{n=1}^{N} \log z_n < \sigma_Z^2 - a\beta \right]. \end{aligned} \tag{61}$$

According to Jensen's inequality, $1/N \sum_n \log z_n \leq \log 1/N \sum_n z_n$. Therefore,

$$\begin{aligned} \epsilon &> P[\hat{a} < 0] \\ &\geq P\left[\log \frac{1}{N} \sum_{n=1}^{N} z_n < \sigma_Z^2 - a\beta \right] \\ &= P\left[\frac{1}{N} \sum_{n=1}^{N} z_n < e^{\sigma_Z^2 - a\beta} \right]. \end{aligned} \tag{62}$$

Because $1/N \sum_{n=1}^N z_n \sim \mathcal{N}\left(1, \sigma_Z^2/N\right)$, (8) can be easily found from (62). Theorem 3 is thus proved.

Remark 6: There are two ways for the attacker to estimate $\hat{a}$. Besides (57), the second way is that the attacker calculates first

$$\hat{F}_1 = \frac{1}{N} \sum_{n=1}^{N} \left(F_t(x-\beta u_j) + v_{t1,n} \right), \tag{63}$$

$$\hat{F}_2 = \frac{1}{N} \sum_{n=1}^{N} \left(F_t(x+\beta u_j) + v_{t2,n} \right), \tag{64}$$

and then estimates

$$\hat{\hat{a}} = \frac{1}{\beta} \log \frac{\hat{F}_1}{\hat{F}_2}. \tag{65}$$

In this case, we have the following result:

Lemma 4: Under small noise perturbation, we have $\hat{\hat{a}} \approx \hat{a} \approx \frac{1}{N} \sum_{n=1}^{N} y_n$ if N is large.

Proof: First, from (57), if noise standard deviation $\sigma^2 \ll \beta F_t^2(x)$, then $\hat{a} \approx \frac{1}{N} \sum_{n=1}^{N} y_n$. Next, starting from (65), with the definition of y_n in (54), we have

$$\hat{\hat{a}} = a + \frac{1}{\beta} \log \frac{\frac{1}{N} \sum_{n=1}^{N} \left(1 + v_{t1,n}/F_t(x - \beta u_j)\right)}{\frac{1}{N} \sum_{n=1}^{N} \left(1 + v_{t2,n}/F_t(x + \beta u_j)\right)}. \tag{66}$$

If the noise is small, we can apply the first-order Taylor series approximation $E[X/Y] \approx E[X]/E[Y]$ to get

$$\hat{\hat{a}} \approx a + \frac{1}{\beta} \log \left(\frac{1}{N} \sum_{n=1}^{N} z_n \right). \tag{67}$$

Next, further applying first-order Taylor series approximation $\log E[X] \approx E[X] - 1$, we have

$$\begin{aligned} \hat{\hat{a}} &\approx a + \frac{1}{\beta} \left(\frac{1}{N} \sum_{n=1}^{N} z_n - 1 \right) \\ &= a + \frac{1}{N\beta} \sum_{n=1}^{N} (z_n - 1) \\ &\approx a + \frac{1}{N\beta} \sum_{n=1}^{N} \log z_n \\ &= \frac{1}{N} \sum_{n=1}^{N} \left(a + \frac{1}{\beta} \log z_n \right) \\ &= \frac{1}{N} \sum_{n=1}^{N} y_n, \end{aligned} \tag{68}$$

which proves the lemma. □

Therefore, if σ^2 is small enough and N is large enough, the estimator $\hat{a}$ in (57) and the estimator $\hat{\hat{a}}$ in (65) give the same result, both equal to $1/N \sum_{n=1}^{N} y_n$. As a result, this second way needs the similar N repeated queries as (8) in order to estimate the gradient up to certain accuracy.

REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014, pp. 1–10.

[2] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Apr. 2017, pp. 506–519.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Stat*, vol. 1050, p. 20, Dec. 2015.

[4] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2017, pp. 39–57.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–28.

[6] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[7] C.-C. Tu, P. Ting, P.-Y. Chen, S. Liu, H. Zhang, J. Yi, C.-J. Hsieh, and S.-M. Cheng, "AutoZOOM: Autoencoder-based zeroth order optimization method for attacking black-box neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 742–749.

[8] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.

[9] M. Cheng, T. Le, P.-Y. Chen, H. Zhang, J. Yi, and C.-J. Hsieh, "Query-efficient hard-label black-box attack: An optimization-based approach," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–12.

[10] M. Cheng, S. Singh, P. H. Chen, P.-Y. Chen, S. Liu, and C.-J. Hsieh, "Sign-opt: A query-efficient hard-label adversarial attack," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–16.

[11] M. Alzantot, Y. Sharma, S. Chakraborty, H. Zhang, C.-J. Hsieh, and M. B. Srivastava, "GenAttack: Practical black-box attacks with gradient-free optimization," in *Proc. Genetic Evol. Comput. Conf.*, 2019, pp. 1111–1119.

[12] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2484–2493.

[13] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[14] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness on image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 321–331.

[15] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Represent.*, Sep. 2018, pp. 1–12.

[16] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2016, pp. 582–597.

[17] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[18] N. Narodytska and S. P. Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," 2016, *arXiv:1612.06299*.

[19] J. Hayes and G. Danezis, "Learning universal adversarial perturbations with generative models," 2017, *arXiv:1708.05207*.

[20] S. Cheng, Y. Dong, T. Pang, H. Su, and J. Zhu, "Improving black-box adversarial attacks with a transfer-based prior," in *Advances in Neural Information Processing Systems*, 2019, pp. 10934–10944.

[21] J. Lu, T. Issaranon, and D. Forsyth, "SafetyNet: Detecting and rejecting adversarial examples robustly," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 446–454.

[22] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[23] X. Wang, S. Wang, P.-Y. Chen, Y. Wang, B. Kulis, X. Lin, and S. Chin, "Protecting neural networks with hierarchical random switching: Towards better robustness-accuracy trade-off for stochastic defenses," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1–16.

[24] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 3–14.

[25] A. Athalye and N. Carlini, "On the robustness of the CVPR 2018 white-box adversarial example defenses," 2018, *arXiv:1804.03286*.

[26] F. Tramèr, D. Boneh, A. Kurakin, I. Goodfellow, N. Papernot, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. 6th Int. Conf. Learn. Represent.*, 2018, pp. 1–22.

[27] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–17.

[28] Z. He, A. S. Rakin, and D. Fan, "Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 588–597.

[29] X. Liu, M. Cheng, H. Zhang, and C.-J. Hsieh, "Towards robust neural networks via random self-ensemble," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 369–385.

[30] W. Fan, G. Sun, Y. Su, Z. Liu, and X. Lu, "Integration of statistical detector and Gaussian noise injection detector for adversarial example detection in deep neural networks," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20409–20429, Jul. 2019.
[31] B. Li, C. Chen, W. Wang, and L. Carin, "Certified adversarial robustness with additive noise," in *Advances in Neural Information Processing Systems*, 2019.
[32] T. Lee, B. Edwards, I. Molloy, and D. Su, "Defending against machine learning model stealing attacks using deceptive perturbations," 2018, *arXiv:1806.00054*.
[33] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "NATTACK: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3866–3876.
[34] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[35] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *Proc. IEEE Symp. Secur. Privacy*, May 2020, pp. 1277–1294.
[36] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–24.
[37] E. Díaz-Francés and F. J. Rubio, "On the existence of a normal approximation to the distribution of the ratio of two independent normal random variables," *Stat. Papers*, vol. 54, no. 2, pp. 309–323, May 2013.
[38] G. Cui, X. Yu, S. Iommelli, and L. Kong, "Exact distribution for the product of two correlated Gaussian random variables," *IEEE Signal Process. Lett.*, vol. 23, no. 11, pp. 1662–1666, Nov. 2016.

MANJUSHREE B. AITHAL received the B.E. degree from Shivaji University, Maharashtra, India, in 2012, and the M.E. degree from Savitribai Phule Pune University, Maharashtra, in 2015. She is currently pursuing the Ph.D. degree in electrical and computer engineering with Binghamton University, Binghamton, NY. Her research interests include signal processing, machine learning, deep learning, and adversarial networks.

XIAOHUA LI (Senior Member, IEEE) received the B.S. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, in 2000. From 2000 to 2006, he was an Assistant Professor with the Department of Electrical and Computer Engineering, Binghamton University, Binghamton, NY, where he has been an Associate Professor, since 2006. His research interests include signal processing, machine learning, deep learning, wireless communications, and wireless information assurance.