IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2023.0322000

Knowing is Half the Battle: Enhancing Clean Data Accuracy of Adversarial Robust Deep Neural Networks via Dual-model Bounded Divergence Gating

HOSSEIN ABOUTALEBI[1, 4], MOHAMMAD JAVAD SHAFIEE[2, 3], CHI-EN AMY TAI[2], and ALEXANDER WONG [2, 3, 4]

[1]Department of Computer Science, University of Waterloo, Ontario, Canada
[2]Department of Systems Design, University of Waterloo, Ontario, Canada
[3]DarwinAI, Waterloo, Ontario, Canada
[4]Waterloo Artificial Intelligence Institute, Waterloo, Ontario, Canada

Corresponding author: Hossein Aboutalebi (e-mail: haboutal@uwaterloo.ca)

ABSTRACT Significant advances have been made in recent years in improving the robustness of deep neural networks, particularly under adversarial machine learning scenarios where the data has been contaminated to fool networks into making undesirable predictions. However, such improvements in adversarial robustness has often come at a significant cost in model accuracy when dealing with uncontaminated data (i.e., clean data), making such defense mechanisms challenging to adapt for real-world practical scenarios where data is primarily clean and accuracy needs to be high. Motivated to find a better balance between adversarial robustness and clean data accuracy, we propose a new model-agnostic adversarial defense mechanism named **D**ual-model **B**ounded **D**ivergence (DBD), driven by a theoretical and empirical analysis of the bias-variance trade-off within an adversarial machine learning context. More specifically, the proposed DBD mechanism is premised on the observation that the variance in deep neural networks tends to increase in the presence of adversarial perturbations in the input data. As such, DBD employs a gating mechanism to decide on the final model prediction output based on a novel dual-model variance measure (coined DBD Variance), which is a bounded version of KL-Divergence between models. Not only is the proposed DBD mechanism itself training-free, but it can be combined with existing adversarial defense mechanisms to boost the balance between clean data accuracy and adversarial robustness. Comprehensive experimental results across over 10 different state-of-the-art adversarial defense mechanisms using both CIFAR-10 and ImageNet benchmark datasets across different adversarial attacks (e.g., APGD, AutoAttack) demonstrates that the integration of DBD can lead to as much as a 6% improvement on clean data accuracy without compromising much on adversarial robustness.

INDEX TERMS adversarial attack, computer vision, deep learning, neural networks

I. INTRODUCTION

Deep learning has achieved remarkable success in various domains including computer vision [1]–[4], machine translation [5], [6], and healthcare [7], [8]. However, concerns arise about their robustness and reliability in real-world scenarios due to their susceptibility to adversarial machine learning attacks.

An adversarial perturbation, denoted as ϵ, when added in a specific direction to an input, can mislead the model into making erroneous predictions. This phenomenon is observed in both classification [9], [10] and regression tasks [11], [12]. Notably, these perturbations are often so subtle that they are undetectable to the human eye. The magnitude of ϵ is typically restricted to ensure its imperceptibility. The vulnerability of deep neural networks to such perturbations was first highlighted by Szegedy *et al.* [10]. They discovered that state-of-the-art models, when subjected to nearly imperceptible non-random perturbations, tend to make confidently incorrect predictions. They speculated that these inconsistencies arise from unnoticed gaps in the training of deep neural networks. Subsequently, Goodfellow *et al.* [13] posited that the suscep-

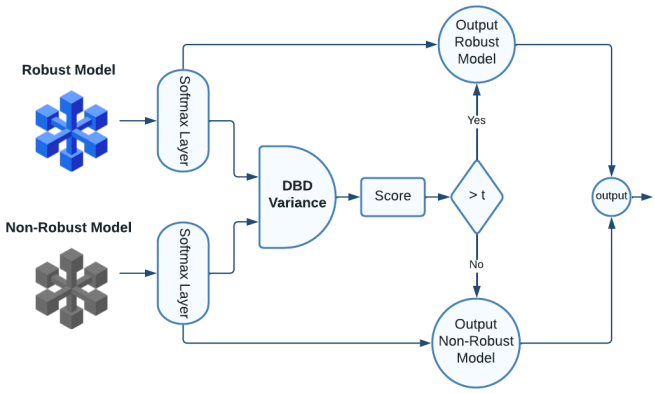

FIGURE 1: Overview of the proposed DBD defense mechanism. The DBD Variance is computed between the softmax outputs of an adversarially trained model (robust model) and a conventionally trained model (non-robust model) and a score is produced. Based on the score, a gating mechanism is used to decide whether the final model prediction is based on the output of the robust model or that of the non-robust model.

tibility of these networks to adversarial examples stems from their inherent linear nature in high-dimensional spaces. Since then, several studies have introduced different approaches to generate adversarial perturbation and fool the deep neural networks [9], [13]–[15].

Despite significant improvements in building more robust deep neural networks against adversarial attacks, these models still suffer from a substantial loss of accuracy on clean data after undergoing adversarial training. For instance, various studies [16]–[18] have reported a drop in accuracy of around $5\% - 15\%$ for clean data after making a model robust against well-known adversarial attacks. This issue poses a challenge for real-world applications of deploying trained robust models since, in most cases, it is safe to assume that the input data is mainly drawn from a clean distribution (uncontaminated data distribution).

In this study, we propose an unsupervised algorithm, called Dual-model Bounded Divergence (DBD), to address the issue of reduced accuracy on clean data while maintaining robust accuracy against adversarial datasets. DBD is a novel yet straightforward approach, illustrated in Figure 1, that can be seamlessly integrated with various robust models to enhance clean accuracy while imposing minimal computational complexity compared to that imposed by adversarial attacks. Our results demonstrate that the proposed DBD framework can increase clean accuracy by up to 6% while maintaining model robustness. Moreover, DBD offers a hyperparameter threshold that enables a trade-off between clean and robust accuracy, making it flexible in determining which performance to prioritize.

Our intuition behind the DBD framework's design stems from our in-depth theoretical analysis of bias-variance decomposition when the model is attacked by adversarial input. We observed that any adversarial attack could cause a significant increase in the model variance, which is further supported by our empirical experiments. Observing this flaw in the design of adversarial attacks, we exploited variance measure in the proposed DBD framework to mitigate the adversarial attack impact.

In previous studies, although the bias-variance trade-off has been used to analyze different aspects of deep neural networks, to the best of the authors' knowledge, this is the first time the theoretical analysis of bias-variance within the context of adversarial attacks has been studied. Moreover, the theoretical insight further supports the core design of the proposed DBD framework, which relies on the variance measure.

To fully leverage the variance measure in the DBD framework and maintain the algorithm's unsupervised and model-agnostic nature, we could not utilize some of the measures proposed in prior research that rely on unbounded measures like KL-Divergence [19]. Consequently, we introduce a new variance measure called DBD Variance in this study. This measure is bounded and does not result in data leakage. The inspiration for DBD Variance comes from a recently developed bounded version of KL Divergence (bounded KL Divergence) [20].

Although the proposed DBD approach shares similarities with algorithms that utilize data refinement and batch normalization to determine whether a sample is perturbed, it stands out by not necessitating retraining and solely focusing on enhancing clean accuracy performance within an unsupervised manner. For instance, Xie *et al.* [21] proposed using auxiliary batch normalization to enhance adversarial training performance, whereas Liu *et al.* [22] suggested a similar approach using multiple batch normalization gates. Unlike the approaches proposed in [21], [22], the DBD framework does not require architectural changes, making it an easier choice for real-world scenarios. Furthermore, the proposed DBD framework differs significantly from the algorithm introduced by Maini *et al.* [23] where an ensemble of robust classifiers is utilized to enhance robust accuracy. Here, the primary emphasis is solely on improving clean dataset accuracy, which is critical for real-world applications. The main contributions of the proposed work are as follows:

- A novel unsupervised Dual-model Bounded Divergence (DBD) framework is proposed as a post-processing step to increase the performance of a robust model on clean data while maintaining robust accuracy with a negligible difference.
- A new dual-model measure for computing variance in practice based on a bounded version of KL-Divergence is proposed, which does not compromise model accuracy and does not cause data leakage problems during test time.
- The bias-variance decomposition of a deep neural network facing an adversarial attack is derived and decomposed for both MSE and cross-entropy loss functions.

- Theoretically and empirically, the variance increase under adversarial attack is illustrated and exploited as a weak point of adversarial attacks in the DBD framework.
- Extensive experimental results show the efficacy of the new Dual-model Bounded Divergence (DBD) algorithm. The results show one of the main weaknesses of adversarial attacks is the variance increase which is observable/measurable by the proposed variance measure.

A. RELATED WORKS

1) Adversarial Attacks

Madry *et al.* [18] proposed a multi-step attack called the projected gradient descent (PGD) algorithm, which generalizes the prior first-order adversarial attack algorithms [10] and can produce adversarial examples that are harder to learn and defeat. Furthermore, Croce *et al.* [24] introduces two new attacks, APGD and AutoAttack. APGD is an improvement over PGD attack through optimizing its steps. AutoAttack is an ensemble of four parameter-free attacks including APGD, Square Attack [25], and FAB [26], which is known to be the most effective attack, and it reported state-of-the-art performance in fooling different deep neural network models.

2) Adversarial Defense

In tandem with the development of adversarial machine learning techniques, substantial research has delved into devising countermeasures to thwart these adversarial threats. One seminal work in this area is adversarial training, proposed by Goodfellow *et al.* [13]. This technique strengthens neural networks against adversarial onslaughts by supplementing the training dataset with adversarial instances. Building on this foundational idea, Kurakin *et al.* [27] showcased the efficacy of adversarial training for the Inception V3 model [28] using the ImageNet dataset [29]. Their work also highlighted the superior transferability of one-step attacks, such as FGSM, making them more potent in scenarios resembling black-box attacks, where an adversary lacks knowledge of the model's internals. Madry *et al.* [18] demonstrated that the higher the capacity of the deep neural network model, the more resilient it is against adversarial attacks. The adversarial training method is further extended by Liu *et al.* [30] (so-called Adv-BNN), where Bayesian techniques benefit the adversarial training method to improve the robustness of the model.

Beyond the realm of adversarial training, alternative defensive strategies have also been explored. For instance, the defensive distillation technique [31] crafts a fresh training set by substituting the Softmax outputs of a neural network with more nuanced, smoothed values. Subsequently, a similarly structured neural network is trained on this refined dataset. While this methodology elevates the model's resilience against certain rudimentary adversarial techniques, it falters when confronted with more sophisticated attacks, such as those leveraging the CW adversarial machine learning approach [32].

Generative models, such as GANs and diffusion models [33], [34], have played a pivotal role in bolstering the robustness of deep learning architectures [35]–[38]. A prime example of this is Defense-GAN [36], which harnesses the power of GANs to model the distribution of unaltered images, thereby enhancing the resistance of deep learning models to adversarial attacks. More recently, DiffPure [37] has utilized diffusion models to purify adversarial images by denoising them in the backward pass of the diffusion model to recover a clean image. Despite the effectiveness of generative model approaches in improving model robustness, they often require solving optimization problems or multiple forward passes during inference, leading to a significant increase in computation, typically by a factor of dozens.

In this regard, designing a standard benchmark to evaluate the performance of new attack algorithms and their defence counterparts is another active field of research in this area. Several works have tried to standardize this process by providing benchmark libraries [17], [18], [24], [39]. RobustBench [39] is one of the well-known and well-designed libraries for this purpose which we will use in our experiments.

3) Bias and Variance

Finally, to further elaborate on the topic of bias and variance is considered one of the long-standing and well-known procedures to analyze the generalization and reliability of machine learning models. The seminal work by Geman *et al.* [40] showed that while a model's variance increases, the model's bias decreases monotonically with the increase in the model complexity. They derived a well-formed decomposition of the bias and variance of the loss function for the regression learning task.

II. METHODOLOGY

One of the main problems in adversarial training is a drastic drop of accuracy over clean data, usually in the range of $10\% - 15\%$ drop to maintain higher robustness [16]–[18]. In this context, a clean data sample is a sample without any perturbation.

The proposed framework, called Dual-model Bounded Divergence (DBD), achieves a balance between clean accuracy and robust accuracy by applying it to any arbitrary robust deep neural network without prior training or parameter optimization. Therefore, DBD can be executed in an unsupervised manner without requiring any prior training or access to the data distribution. DBD leverages the variance increase anomaly in the robust model relative to the non-robust model when the input sample is perturbed. This observation is supported by our theoretical analysis, indicating that the increase in variance can be even more significant for non-robust models against adversarial input. In our terminology, the robust model refers to any model trained for adversarial attacks, while the non-robust model denotes any model that has not undergone any form of adversarial training.

Figure 1 illustrates the flow diagram of the proposed algorithm. Firstly, DBD receives the Softmax layers of both the robust and non-robust models as input. It is worth mentioning that the robust and non-robust models can have different

architectures and are not required to be the same architecture. The DBD framework incorporates a gating mechanism that can determine which model, either the robust or non-robust, should be employed for prediction based on the DBD Variance calculated by the gating mechanism. The DBD Variance is a cross-model variance obtained from the Softmax layer outputs extracted from both models. If the DBD Variance value surpasses a pre-determined gating threshold, it suggests that the input may have been perturbed by an adversarial attack, and the robust model is activated for prediction. Otherwise, the non-robust model is utilized in the prediction process. The pseudo-code of the proposed algorithm is presented in Algorithm 2.

The gating threshold can be identified by cross-validation or based on the user's preference for balancing the trade-off between the model's accuracy on non-perturbed (clean) samples versus perturbed ones.

A. DBD VARIANCE

The core element of the proposed framework is DBD Variance, used as a gating mechanism to perform the decision process. Here we motivate this approach and provide detailed formulation on how to apply this technique.

1) Motivation

Although KL Divergence is a well-known technique for measuring variance [19], it has some limitations. Firstly, it is a non-negative measure, meaning that the measure can result in NaN (Not a Number) during computations, which can cause the variance measured via KL Divergence to be undefined or out of scale for any comparison.

Secondly, the use of data splitting suggested by [19] to compute variance can potentially reduce the model's accuracy during testing as the model has access to only a subset of the training data. Additionally, there is a risk of data leakage as the average probability should be based solely on the training data, not the test data. Therefore, if we only have access to the test set, the only viable method to compute the mean for obtaining variance is by utilizing the test set, which can lead to data leakage. Consequently, weakly-supervised solutions using previously proposed methods for variance computation are challenging to envision.

To address the issues mentioned above, we propose a new variance measure, referred to as DBD Variance, for robust models that can accurately measure variance against adversarial attacks. The definition of DBD Variance is inspired by the work proposed by Chen *et al.* [20], where they introduced a bounded version of KL Divergence.

2) DBD Variance as Gating Mechanism

Definition (DBD Variance): *Given two models M, M', each with n outputs which are independently trained on the training set D, the variance of the model M from the model M' at data input (x, y) is DBD Variance of the model for data input (x, y).*

As such the DBD Variance is measured as follows:

$$Var(M|M')_x = \sum_{i=1}^{n} p_i(\bar{M}) \log_2(|p_i(\bar{M}) - p_i(M)| + 1) \quad (1)$$

where, p_i refers to i^{th} output of the model for data input x. $\bar{M}$ is the average probability of the model computed for data input x from the models M and M' which is computed as below:

$$p_i(\bar{M}) \propto \exp(\log(p_i(M')) + \log(p_i(M))) \quad i \in \{1, ..., n\}$$

Properties: The main properties of DBD Variance are as follows:

- Similar to KL-Divergence $Var(M|N) \neq Var(N|M)$.
- The range of $Var(M|N) \in [0, 1]$.

The proposed equation is not limited to two models and can be easily extended to more than two models. By utilizing the DBD Variance computation outlined in (1), data leakage issues during testing and undefined values during variance calculation can be avoided. Since the DBD Variance computation is based on the discrepancy between the outputs of the two models rather than the training data, calculating the average probability does not result in any data leakage problems on the test set. In the DBD framework, we have utilized the Robust and Non-Robust models instead of M and M', respectively.

B. DBD VARIANCE BEHAVIOUR

The DBD framework's effectiveness in improving model performance on clean data samples is attributed to the increase in the model's variance facing an adversarial attack, which is quantifiable using the proposed DBD Variance. In this study, we examine the behavior of adversarial attacks and their impact on the proposed DBD Variance. Table 1 illustrates the results of the DBD Variance for various models and architectures (specified in parentheses in the table) on the ImageNet dataset under adversarial attacks. We investigate several robust models [16], [17], [41] proposed for ResNet-50 [1] and WideResNet-50 [42] architectures. The variance of the robust models is compared against the same non-robust model (i.e., ResNet-50) trained on the ImageNet dataset for all evaluated robust models. The adversarial data were generated using AutoAttack and APGD algorithms.

As shown in Table 1, the DBD Variances of both robust and non-robust models increase significantly when faced with adversarial inputs compared to their variance on clean data samples. Additionally, it is observed that the variance increase in non-robust models is higher than that of robust models, as studied in our subsequent theoretical analysis.

The observed phenomenon can be expressed as a gating mechanism, where the maximum of the DBD Variance of the two models is utilized as a score. This score can then be employed to determine whether the robust model (with superior performance on adversarial input) or the non-robust model (with superior performance on clean input) should be utilized for the final prediction.

TABLE 1: DBD Variance measure. RN-50 stands for ResNet-50, WRN-50 for WideResNet-50, and AA for AutoAttack. C-Data Var. stands for clean dataset variance. Adv-Data Var. stands for adversarial dataset.

Model	Attack	C-Data Var.	Adv-Data Var.
Salman *et al.* [16] (RN-50)	AA	0.14	**0.18**
Non-robust (RN-50)		0.08	**0.39**
Engstrom *et al.* [17] (RN-50)	AA	0.14	**0.18**
Non-robust (RN-50)		0.08	**0.38**
Wong *et al.* [41] (RN-50)	AA	0.13	**0.18**
Non-robust (RN-50)		0.10	**0.37**
Salman *et al.* [16] (WRN-50)	AA	0.12	**0.18**
Non-robust (RN-50)		0.07	**0.42**
Salman *et al.* [16] (RN-50)	APGD	0.14	**0.15**
Non-robust (RN-50)		0.08	**0.22**
Engstrom *et al.* [17] (RN-50)	APGD	0.14	**0.15**
Non-robust (RN-50)		0.08	**0.21**
Wong *et al.* [41] (RN-50)	APGD	0.13	**0.14**
Non-robust (RN-50)		0.10	**0.19**
Salman *et al.* [16] (WRN-50)	APGD	0.12	**0.16**
Non-robust (RN-50)		0.07	**0.24**

In this regard, for faster computation of DBD, only single model execution is required in the inference time after it is identified that queries are being initiated from a malicious source when the gating mechanism is triggered. It is worth noting that adversarial attacks usually query the model for several times to be able to generate an effective perturbation.

C. GATING THRESHOLD AND DBD INTEGRATION

The only hyper-parameter of the proposed DBD framework is the gating threshold. As discussed earlier, this parameter lets the user balance the trade-off between clean and robust accuracy. As a result, the choice of this parameter is dependent on the problem context and how much the user wants to give weight to robust accuracy against clean accuracy. The threshold value is between $[0-1]$ as the DBD Variance cannot be higher than 1 or lower than 0.

III. THEORETICAL ANALYSIS

In this section, we will analyze the bias-variance decomposition of the loss function under an adversarial attack and demonstrate its effect on the model's behavior, resulting in an increase in variance. Specifically, we will investigate the impact of adversarial attacks on the cross-entropy loss function as the most well-known classification loss. The analysis illustrates that a non-robust models are more susceptible to higher level of variance increase compared to its robust version.

A. NOTATION

$\|x\|$ denotes a generic norm function. Notations $\|x\|_2$, $\|x\|_\infty$ refers to the l_2 and l_∞ norms respectively. A set is denoted with capital letters such as $\mathcal{X}, \mathcal{Y}$ while vectors are denoted by small letters such as x and y. The training set is denoted with

Algorithm 1: DBD Framework

Input: S=$\{x | x \in D\}$, Threshold t
Input: Robust model M and Non-Robust Model M' with c distinct classes, $L_M(x)$: Softmax layer
Result: $R = \{\hat{y}(x) \mid x \in D\}$
$R = [\,]$
for x *in* S **do**
Forward Pass $M(x) \rightarrow L_M(x)$
Forward Pass $M'(x) \rightarrow L_{M'}(x)$
$Score = Max\Big(Var(M'|M), Var(M|M')\Big)$ (1)
if $Score > t$ **then**
$R.add(M(x))$
else
$R.add(M'(x))$

$\mathcal{D}$ and the target function by $f : \mathcal{X} \rightarrow \mathcal{Y}$. In case of regression learning tasks, the set $\mathcal{Y}$ is a continuous one dimensional space while in the classification task it contains discrete values. In our setting, a prediction model is denoted by $\hat{f}$ which is an estimation of the ground truth function f over the training set $\mathcal{D}$. In order to consider adversarial perturbation, we denote the perturbation added to each data sample x by an adversary with the vector $\beta(x)$[1]. $\beta(x)$ is not generated naturally and is designed specifically for each data sample and usually has the property $l_\infty\big(\beta(x)\big) < \delta$. Throughout our analysis, whenever we use the notation $\nabla f(x)$, it is the gradient of the function f with respect to x.

B. CASE I: REGRESSION WITH MSE LOSS

Assume the goal is to estimate the target function $f : \mathcal{X} \rightarrow \mathcal{Y}$. Each element $x \in \mathcal{X}$ has dimension $|x| = d$. Given the training data, $\mathcal{D} = \{(x_1, y_1), ..., (x_m, y_m)\}$, a learner produces a prediction model $\hat{f}(x)$. As such, the configuration of the parameters in $\hat{f}(x)$ is dependent on the training data $\mathcal{D}$. Let us also assume the training data $\mathcal{D}$ is accompanied with a natural noise γ such that:

$$y_i = f(x_i) + \gamma \tag{2}$$

where $1 \leq i \leq m$ with m total number of data samples in $\mathcal{D}$, and γ is a random variable where $\mathbb{E}[\gamma] = 0$, and $\mathbb{E}[\gamma^2] = \sigma_\gamma^2$. It is worth to note that, we keep this assumption mainly for the regression task and we will drop it for the classification problems with cross-entropy loss function for simplicity. Geman *et al.* [40] decomposed a MSE loss function in terms of its bias and variance of a prediction model by Theorem 1.

Theorem 1: For a prediction model $\hat{f}(x)$ trained on the training data $\mathcal{D}$ to estimate the target function $f(x)$ with MSE loss

[1]Without loss of generality in our derivation, we can also rewrite the β to be the function of parameters of the model θ as $\beta(x, \theta)$. However, in order to also consider black-box attack and have a more general derivations, in our notation we consider β as a function of x.

function, the bias variance trade-off is [40]:

$$\mathbb{E}_{x,\mathcal{D},\gamma}\left[(y-\hat{f}(x))^2\right] = \mathbb{E}_{x,\mathcal{D}}\left[\left(\mathbb{E}_{\mathcal{D}}\left[\hat{f}(x)\right]-f(x)\right)^2\right]+ \mathbb{E}_{x,\mathcal{D}}\left[\left(\hat{f}(x)-\mathbb{E}_{\mathcal{D}}\left[\hat{f}(x)\right]\right)^2\right]+\sigma_\gamma^2 = Bias[\hat{f}]+Var[\hat{f}]+Var[\gamma]. \quad (3)$$

The $Var[\gamma]$ is the intrinsic noise of the system. Given (3), it is possible to break down and decouple the effect of different factors on model performance based on the bias, variance and intrinsic noise in the model. However, (3) does not take the effect of adversarial perturbation into account. The perturbation $\beta(x)$ added to each data sample x during the test time aims to increase the loss value of the model. It is assumed that $f(x) = f(x+\beta(x))$, this assumption is to make sure the added perturbation magnitude is reasonable and follows the imperceptibility of the adversarial perturbation.

This perturbation can have a great impact on the final loss which is significantly different from (3). Following, we propose a new theorem to account for the adversarial perturbation in deriving bias and variance of a model.

Theorem 2: Assume $\bar{f}(x) = \mathbb{E}_{\mathcal{D}}[\hat{f}(x)]$ and the target function is $f(x)$. The bias-variance trade-off for MSE loss function with a prediction model $\hat{f}(x)$ trained on dataset $\mathcal{D}$ with noise γ in the presence of adversarial perturbation $\beta(x)$ via the adversarial algorithm is:

$$\mathbb{E}_{x,\mathcal{D},\gamma}\left[(y-\hat{f}(x+\beta(x)))^2\right] \approx \mathbb{E}_{x,\mathcal{D}}[(f(x)-\bar{f}(x)-c_x)^2] + Var[\gamma] + Var[\hat{f}] + \mathbb{E}_{x,\mathcal{D}}[c'_x] \quad (4)$$

$$\text{where,} \quad c_x = \nabla\bar{f}(x)^T\beta(x)$$

$$\text{and,} \quad c'_x = 2\Big(\hat{f}(x)-\bar{f}(x)\Big)\Big(\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x)\Big) \quad (5)$$

Proof: Given the adversarial perturbation $\beta(x)$ with the condition $l_\infty\big(\beta(x)\big) < \delta$ and the assumption that $f(x+\beta(x)) = f(x)$, the MSE loss can be decomposed as follows:

$$\begin{aligned}
&\mathbb{E}_{x,\mathcal{D},\gamma}\left[(y-\hat{f}(x+\beta(x)))^2\right] = \\
&\mathbb{E}_{x,\mathcal{D},\gamma}\left[(f(x)-\hat{f}(x+\beta(x))+\gamma)^2\right] \\
&= \mathbb{E}_{x,\mathcal{D},\gamma}\left[(f(x)-\bar{f}(x+\beta(x))+\bar{f}(x+\beta(x))-\hat{f}(x+\beta(x))+\gamma)^2\right] \\
&= \mathbb{E}_{x,\mathcal{D}}\left[(f(x)-\bar{f}(x+\beta(x)))^2\right] \\
&+ \mathbb{E}_{x,\mathcal{D}}\left[(\hat{f}(x+\beta(x))-\bar{f}(x+\beta(x)))^2\right]+\sigma_\gamma^2 \\
&+2\,\mathbb{E}_\gamma[\gamma]\,\mathbb{E}_{x,\mathcal{D}}\left[(f(x)-\bar{f}(x+\beta(x))+(\bar{f}(x+\beta(x))-\hat{f}(x+\beta(x)))\right] \\
&+2\,\mathbb{E}_x\left[(f(x)-\bar{f}(x+\beta(x))\times\mathbb{E}_{\mathcal{D}}\left[(\bar{f}(x+\beta(x))-\hat{f}(x+\beta(x)))\right]\right] \\
&= \mathbb{E}_{x,\mathcal{D}}\left[(f(x)-\bar{f}(x+\beta(x)))^2\right] \\
&+ \mathbb{E}_{x,\mathcal{D}}\left[(\hat{f}(x+\beta(x))-\bar{f}(x+\beta(x)))^2\right]+\sigma_\gamma^2 \\
&\approx \mathbb{E}_{x,\mathcal{D}}\left[\left(f(x)-\bar{f}(x)-\nabla\bar{f}(x)^T\beta(x)\right)^2\right] \\
&+ \mathbb{E}_{x,\mathcal{D}}\left[(\hat{f}(x+\beta(x))-\bar{f}(x+\beta(x)))^2\right]+\sigma_\gamma^2 \quad (6)
\end{aligned}$$

In (6), we are using the fact that $\mathbb{E}_\gamma[\gamma] = 0$ and

$$\mathbb{E}_{\mathcal{D}}\left[(\bar{f}(x+\beta(x))-\hat{f}(x+\beta(x)))\right] = 0.$$

Also for the term $\mathbb{E}_{x,\mathcal{D}}\left[(\hat{f}(x+\beta(x))-\bar{f}(x+\beta(x)))^2\right]$ in (6), by using Taylor polynomial of order one [43], we have:

$$\begin{aligned}
&\mathbb{E}_{x,D}\left[(\hat{f}(x+\beta(x))-\bar{f}(x+\beta(x)))^2\right] \\
&\approx \mathbb{E}_{x,D}\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2+2(\hat{f}(x)-\bar{f}(x))(\nabla\hat{f}(x)-\nabla\bar{f}(x))^T\beta(x)\right] \\
&= Var[\hat{f}]+\mathbb{E}_{x,D}\left[2(\hat{f}(x)-\bar{f}(x))(\nabla\hat{f}(x)-\nabla\bar{f}(x))^T\beta(x)\right] \quad (7)
\end{aligned}$$

Putting together (7) and (6), we have:

$$\mathbb{E}_{x,\mathcal{D},\gamma}\left[(y-\hat{f}(x+\beta(x)))^2\right] \approx \mathbb{E}_{x,\mathcal{D}}[(f(x)-\bar{f}(x)-c_x)^2] + Var[\gamma] + Var[\hat{f}] + \mathbb{E}_{x,\mathcal{D}}[c'_x] \quad (8)$$

$$\text{where,} \quad c_x = \nabla\bar{f}(x)^T\beta(x) \quad (9)$$

$$\text{and,} \quad c'_x = 2\Big(\hat{f}(x)-\bar{f}(x)\Big)\Big(\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x)\Big) \quad (10)$$

■

As illustrated, one important conclusion from the Theorem 2 is that the adversarial machine learning increases the bias term (i.e., c_x in (10)) and it can also increase the variance of the model by c'_x in (10). Experimental results verify these findings as well. The following corollaries are the direct results of theorem 2.

Corollary I: The maximum expected increase in the bias of a model with MSE loss function for a regression task is when the adversarial perturbation is added in the direction of $-\nabla\bar{f}(x)$.

Corollary II: The maximum expected increase in the variance of a model trained for a regression task with MSE loss function is when the adversarial perturbation is added in the direction of $(\hat{f}(x)-\bar{f}(x))(\nabla\hat{f}(x)-\nabla\bar{f}(x))$.

C. CASE II: CLASSIFICATION WITH CROSS-ENTROPY LOSS

The notion of bias and variance can be analyzed for the classification models trained with cross-entropy loss as well. To this end, followed by the work done in [19], [44] let c be the number of classes for classification and $\hat{\pi}_{\mathcal{D}}(x) \in [0,1]^c$ be the output of a neural network trained on the training set $\mathcal{D}$. This function measures the confidence values over classes. Let $\pi(x) \in [0,1]^c$ be a one-hot vector encoding ground truth label that we wish to estimate via $\hat{\pi}$. Then cross-entropy loss can be formulated as:

$$L(\pi,\hat{\pi}) = -\,\mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\pi_i(x)\log\hat{\pi}_i(x)\right)\right] \quad (11)$$

where $\pi_i(x)$ refers to ith component of the output vector $\pi(x)$. As explained in [19], the loss function in (11) can be decomposed:

$$L(\pi,\hat{\pi}) = D_{KL}\big(\pi(x)||\bar{\pi}(x)\big) + \mathbb{E}_{x,\mathcal{D}}\left[D_{KL}\left(\bar{\pi}(x)||\hat{\pi}(x)\right)\right] \quad (12)$$

where $\bar{\pi}(x) \propto \exp\Big(\mathbb{E}_{\mathcal{D}}\left[\log(\hat{\pi}(x))\right]\Big)$. $\bar{\pi}(x)$, as described in [19], is the average of log probability after normalization. In this regard, we can consider it as the mean of the prediction model $\hat{\pi}$ which is defined in terms of KL-Divergence. This

perspective which is further elaborated in Domingos's seminal work [45] is somehow different from the mean defined in the previous section for the regression task because of the cross-entropy loss function form. As a result, following [19], [40] it is possible to consider $D_{KL}\big(\pi(x)||\bar{\pi}(x)\big)$ as the factor which drives the bias and $D_{KL}\big(\bar{\pi}(x)||\hat{\pi}(x)\big)$ as the one deriving the variance in the model. However to account for the adversarial perturbation instead of input x, the function $\pi^*(\cdot)$ needs to be calculated for $x+\beta(x)$. This leads to Theorem 3 which illustrates the behavior of a model trained based on cross-entropy loss in the presence of adversarial perturbation $\beta(x)$. It is assumed that the target function $\pi(x)$ is constant on a small blob around x, and $\beta(x)$ magnitude does not exceed the limits of that blob; in other words, $\pi(x)=\pi\big(x+\beta(x)\big)$.

Theorem 3: Assume for input x, the ground truth class is t_x. For a cross-entropy loss function, the bias-variance tradeoff of a prediction model $\hat{\pi}(x)$ with training data $\mathcal{D}$ for a target function $\pi(x)$ in the presence of adversarial algorithm injecting perturbation $\beta(x)$ to the system is:

$$L(\pi,\hat{\pi}) = \mathbb{E}_{x,\mathcal{D}}\left[D_{KL}(\pi(x)||\bar{\pi}(x)) + D_{KL}(\bar{\pi}(x)||\hat{\pi}(x))\right] + \mathbb{E}_x\left[c_x\right] + \mathbb{E}_{x,\mathcal{D}}\left[c'_x\right] \tag{13}$$

where,

$$c_x = -\,\mathbb{E}_x\left[\left(\nabla_x \log \bar{\pi}_{t_x}(x)\right)^T \beta(x)\right]$$
$$c'_x = -\,\mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\nabla_x \bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right)^T \beta(x)\right] \tag{14}$$

Proof: Assuming that for input x, the ground truth class is t_x. By using Taylor polynomial of order one [43], the loss can be decomposed as follow:

$$\begin{aligned}
L(\pi,\hat{\pi}) &= \mathbb{E}_{x,\mathcal{D}}\left[D_{KL}(\pi(x)||\bar{\pi}(x+\beta(x)))\right] \\
&+ D_{KL}\left[(\bar{\pi}(x+\beta(x))||\hat{\pi}(x+\beta(x)))\right] \\
&= -\,\mathbb{E}_x\left[\log \bar{\pi}_{t_x}(x+\beta(x))\right] \\
&- \mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\bar{\pi}_i(x+\beta(x))\log\frac{\hat{\pi}_i(x+\beta(x))}{\bar{\pi}_i(x+\beta(x))}\right] \\
&= -\,\mathbb{E}_x\left[\log \bar{\pi}_{t_x}(x)\right] - \mathbb{E}_{x,D}\left[\sum_{i=1}^{c}\bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right] \\
&- \mathbb{E}_x\left[\left(\nabla_x \log \bar{\pi}_{t_x}(x)\right)^T \beta(x)\right] \\
&- \mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\nabla_x \bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right)^T \beta(x)\right] \\
&= \mathbb{E}_{x,\mathcal{D}}\left[D_{KL}(\pi(x)||\bar{\pi}(x)) + D_{KL}(\bar{\pi}(x)||\hat{\pi}(x))\right] \\
&- \mathbb{E}_x\left[\left(\nabla_x \log \bar{\pi}_{t_x}(x)\right)^T \beta(x)\right] \\
&- \mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\nabla_x \bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right)^T \beta(x)\right]
\end{aligned} \tag{15}$$

■

This derivation is aligned with finding in [19], [44], where the bias variance decomposition for cross-entropy loss function is in the form of KL-Divergence.

Corollary III: From Theorem 2 and Theorem 3, the variance of a deep neural network trained with cross-entropy loss or MSE loss can increase via perturbation caused by adversarial attack by the added term of c'_x in (14) and (10) in its bias-variance decomposition. Note that c'_x is non-negative.

Proof: We can rewrite the c'_x in the following format:

For Regression with MSE Loss:

$$\begin{aligned}
c'_x &= 2\Big(\hat{f}(x)-\bar{f}(x)\Big)\Big(\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x)\Big) \\
&= \Big(\nabla(\hat{f}(x)-\bar{f}(x))^2\Big)^T\beta(x)
\end{aligned} \tag{16}$$

Assuming that $\beta(x)$ is a non-negative perturbation (similar in using the sign of gradient as a perturbation in FGSM and PGD) and $(\hat{f}(x)-\bar{f}(x))^2$ is a non-negative function, their dot product should be non-negative.

For cross-entropy Loss we can provide a similar argument:

$$\begin{aligned}
c'_x &= -\,\mathbb{E}_{x,\mathcal{D}}\left[\sum_{i=1}^{c}\left(\nabla_x\bar{\pi}_i(x)\log\frac{\hat{\pi}_i(x)}{\bar{\pi}_i(x)}\right)^T\beta(x)\right] \\
&= \mathbb{E}_{x,\mathcal{D}}\left[\nabla_x KL(\bar{\pi}(x)||\hat{\pi})\right)^T\beta(x)\Big]
\end{aligned} \tag{17}$$

Again in equation 17, KL divergence is always non-negative and assuming that $\beta(x)$ consists of non-negative perturbation, c'_x would be non-negative.

Corollary IV: From corollary III, given that $\beta(x)$ is positive and the robust model M denoises the $\beta(x)$ to $\beta'(x)$ where the $|\beta'(x)|_2 \le |\beta(x)|_2$, the upper bound of variance increase of robust model M trained with cross-entropy loss or MSE loss is lower against perturbation compared to non-robust model M'.

Proof: We show here the proof for MSE loss. The proof for Cross Entropy loss is similar. Given that $\beta'(x)$ has lower l_2 norm than $\beta(x)$ we have for $c'_x = 2\Big(\hat{f}(x)-\bar{f}(x)\Big)\Big(\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x)\Big)$, considering the constant $\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x)$ in c'_x, we show that this constant is upper bounded by:

$$\Big(\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big)^T\beta(x) \le \Big|\nabla\hat{f}(x)-\nabla\bar{f}(x)\Big|_2\Big|\beta(x)\Big|_2 \tag{18}$$

As a result, the upperbound of c'_x for the robust model will be lower than the non-robust model as it has lower l_2 norm of $\beta'(x)$ due to denoising process.

DBD Framework: The theorem derived in this study confirms that adversarial attacks result in an increase in model variance. Quantifying the increase of variance can be leveraged as a defence mechanism against adversarial attacks. As illustrated by the theorem, non-robust models exhibit a greater increase in variance compared to robust models. Thus, the difference in variance between robust and non-robust models, as measured by the DBD Variance, can be an effective means of detecting if input samples are affected by adversarial attacks.

Empirical investigation of this variance increase due to adversarial attacks, as presented in Table 1, supports the

theoretical findings. Additional insights into these results will be discussed in the experiment section.

IV. EXPERIMENTS

The effectiveness of the proposed DBD framework is evaluated on various datasets and compared with state-of-the-art methods. One of the main advantages of the DBD framework is its plug-and-play integration with other defense models and the ease of use. The proposed framework strikes a balance between clean and robust accuracy, maintaining both accuracies at the highest level. In addition, the DBD's unsupervised nature as a post-processing step can be used in scenarios where there is limited access to training data or when the process cannot be trained and necessitates unsupervised learning. In this regard, although cross-validation is recommended for determining the optimal threshold value in DBD, using a threshold value of 0.5 provides acceptable performance. Here, we present the optimal threshold value determined through cross-validation.

To further demonstrate the effectiveness of our approach, we utilize the DBD framework on various attacks and defense models with diverse network architectures, while employing the same non-robust model trained exclusively on clean data. Furthermore, for all experiments conducted in this section, we use the version of DBD described in Algorithm 2.

For the initial experiment, we studied the distribution of DBD Variance on the adversarial and clean datasets. We then performed comprehensive experiments to verify the effectiveness of the DBD framework against adversarial attacks. In particular, we tested over ten different defence mechanisms [16], [17], [41], [46]–[53] integrated with DBD against both white-box and black-box attacks, including AutoAttack [24], APGD [24], FAB [26], and Square [25] attacks.

To ensure a fair evaluation and avoid obfuscated gradients [54], we employ the BPDA technique [54] to prevent gradient obfuscation. Accordingly, in our evaluations, we replace the gradient with the backward pass generated by the model selected using the Variance Score. Moreover, we utilize targeted loss in the APGD, FAB and PGD attacks included in our evaluation to increase their perturbation impact and provide a fair evaluation.

Both ImageNet [55], and CIFAR10 [56] datasets were used for the comprehensive evaluation. We also tested our DBD framework on several different architectures, including ResNet-50 (RN-50), WideResNet-50 (WRN-50), WideResNet-28 (WRN-28), WideResNet-34 (WRN-34), ResNet-18 (RN-18), WideResNet-70 (WRN-70), and PreActResNet-18 (PA-RN-18) [1], [42], [57]. In this regard, the experimental implementation is developed based on the toolkit provided by RobustBench library [39] to ensure that the results are reproducible.

It is worth noting that the choice of threshold t in the proposed DBD framework depends on the problem's context and a trade-off between clean accuracy and robust accuracy for the model.

The source code for all experiments can be found here.

A. DBD VARIANCE DISTRIBUTION

One of the main novelties of the proposed DBD framework is DBD Variance. The question looming is how practical and discriminative this measure is in distinguishing between adversarial data samples from clean data samples. While we showed in Table 1 that adversarial data samples have higher DBD Variance on average, compared to clean data samples, here we provide more evidence on this observation which is supported by our theoretical analysis. Figure 2 demonstrates the distribution of DBD Variance of adversarial data samples compared with clean data samples on three different robust models designed on WideResNet-34, PreActResNet-18, and WideResNet-70, respectively. The non-robust model used to measure DBD Variance in this experiment for all three robust models was a non-robust WideResNet-28 trained on the clean dataset. The attack used here is AutoAttack.

Figure 2 demonstrates that the density mass of adversarial data is consistently higher near the maximum DBD Variance value of 1 compared to clean data samples, which have a higher density mass near the minimum DBD Variance of zero. This indicates that the DBD Variance measure can effectively partition clean data from adversarial data in an unsupervised manner, making it a useful tool for identifying manipulated data and determining whether a robust model is needed for prediction.

B. IMAGENET DATASET

Table 2 presents the results of integrating the proposed DBD framework with defence models used in [16], [17], [41] against APGD, targeted FAB, and AutoAttack (AA) on the ImageNet dataset with epsilon set at $\frac{4}{255}$ for the infinite norm. Each row in the table indicates the performance of the robust model with and without integration with the DBD framework. The last column in the table, denote the average accuracy of the model given both clean and adversarial input. To evaluate the impact of model selection on the proposed algorithm's performance, we use a ResNet-50 architecture as the non-robust model passed to the DBD framework.

As seen, the proposed DBD framework can improve the clean accuracy (C-Acc) by $1.2\% - 6\%$ across all models with a minor drop in robust accuracy (R-Acc).

Table 2 shows that AutoAttack (AA) has the highest success rate in fooling the model, resulting in the lowest robust accuracy. The proposed DBD framework improves clean accuracy without compromising much on robust accuracy and achieves the highest average of both accuracies in all cases for ResNet-50 (RN-50) and WideResNet-50 (WRN-50) architectures.

C. CIFAR10 DATASET

Table 3 demonstrates the results of defense models [17], [46]–[53] integrated with the DBD framework on CIFAR-10 dataset. The epsilon was set at $\frac{8}{255}$ for the infinite norm. We use a WideResNet-28 trained on the clean dataset for all models as a non-robust model passed to the DBD framework. The attack used in this experiment was AutoAttack (AA). As

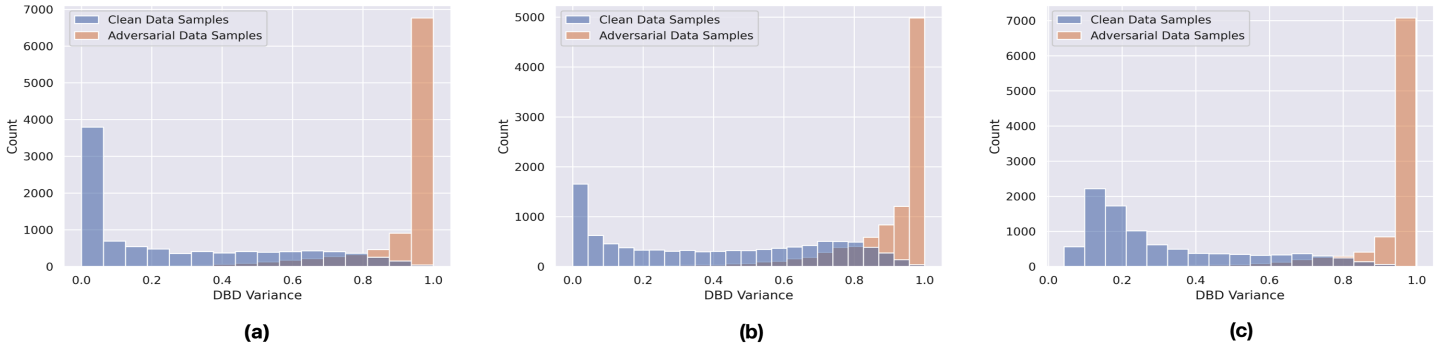

FIGURE 2: Distribution of DBD Variance of adversarial data against clean data on CIFAR10 dataset. The architecture used in (a) is a robust WideResNet-34 while (b) uses a robust PreActResNet-18 and (c) a robust WideResNet-70. The adversarial attack is AutoAttack.

TABLE 2: DBD performance on **ImageNet**. The abbreviations stand for C-Acc: Clean Accuracy, R-Acc: Robust Accuracy, Avg-Acc: Average Accuracy, RN-50: ResNet-50, WRN-50: WideResNet-50 and AA: AutoAttack.

Model	Attack	C-Acc	R-Acc	Avg-Acc
Salman *et al.* (RN-50) [16]	AA	64.02%	**34.96**	49.49%
Salman *et al.* + DBD		**71.32%**	34.24%	**52.78%**
Engstrom *et al.* (RN-50) [17]	AA	62.56%	29.2%	45.88%
Engstrom *et al.* + DBD		**64.88%**	29.2%	**47.04%**
Wong *et al.* (RN-50) [41]	AA	53.44%	25.06%	39.25%
Wong *et al.* + DBD		**59.2%**	25.04%	**42.12%**
Salman *et al.* (WRN-50) [16]	AA	68.46%	**38.14%**	53.3%
Salman *et al.* + DBD		**72.3%**	38.04%	**55.17%**
Salman *et al.* (RN-50) [16]	APGD	64.02%	**34.96%**	49.49%
Salman *et al.* + DBD		**66.10%**	34.42%	**50.26%**
Engstrom *et al.* (RN-50) [17]	APGD	62.56%	**29.32%**	45.96%
Engstrom *et al.* + DBD		**66.74%**	28.98%	**47.86%**
Wong *et al.* (RN-50) [41]	APGD	53.44%	25.06%	39.25%
Wong *et al.* + DBD		**54.52%**	**25.10%**	**39.81%**
Salman *et al.* (WRN-50) [16]	APGD	68.46%	**38.22%**	53.34%
Salman *et al.* + DBD		**71.12%**	37.70%	**54.51%**
Salman *et al.* (RN-50) [16]	FAB	64.06%	**36.82%**	50.44%
Salman *et al.* + DBD		**66.10%**	36.02%	**51.06%**
Engstrom *et al.* (RN-50) [17]	FAB	62.52%	31.44%	46.98%
Engstrom *et al.* + DBD		**64.88%**	31.12%	**48.00%**
Wong *et al.* (RN-50) [41]	FAB	53.44%	30.80%	42.12%
Wong *et al.* + DBD		**54.52%**	30.60%	**42.56%**
Salman *et al.* (WRN-50) [16]	FAB	68.46%	**40.68%**	54.57%
Salman *et al.* + DBD		**71.12%**	39.74%	**55.43%**

TABLE 3: DBD performance on **CIFAR10**. The abbreviations stands for C-Acc: Clean Accuracy, R-Acc: Robust Accuracy, Avg-Acc: Average Accuracy, RN-50: ResNet-50, WRN-50: WideResNet-50, PA-RN: PreActResNet and AA: AutoAttack.

Model	Attack	C-Acc	R-Acc	Avg-Acc
Gowal *et al.* (WRN-28) [46]	AA	87.50%	**63.45%**	75.47%
Gowal *et al.* + DBD		**89.39%**	63.07%	**76.23%**
Wu *et al.* (WRN-28) [47]	AA	88.25%	**60.03%**	74.14%
Wu *et al.* + DBD		**90.11%**	59.56%	**74.83%**
Sridhar *et al.* (WRN-28) [48]	AA	89.46%	**59.66%**	74.56%
Sridhar *et al.* + DBD		**92.72%**	58.61%	**75.66%**
Zhang *et al.* (WRN-28) [49]	AA	89.36%	**59.21%**	74.28%
Zhang *et al.* + DBD		**91.80%**	58.42%	**75.11%**
Sridhar *et al.* (WRN-34) [48]	AA	86.53%	**60.41%**	73.47%
Sridhar *et al.* + DBD		**87.88%**	59.90%	**73.89%**
Sehwag *et al.* (WRN-34) [50]	AA	86.68%	**60.27%**	73.47%
Sehwag *et al.* + DBD		**92.05%**	59.87%	**75.96%**
Rebuffi *et al.* (WRN-70) [52]	AA	92.23%	66.56%	79.39%
Rebuffi *et al.* + DBD		**94.78%**	66.56%	**80.67%**
Engstrom *et al.* (RN-50) [17]	AA	87.03%	49.25%	68.14%
Engstrom *et al.* + DBD		**92.00%**	49.16%	**70.58%**
Sehwag *et al.* (RN-18) [50]	AA	84.59%	**55.54%**	70.06%
Sehwag *et al.* + DBD		**90.74%**	55.17%	**72.82%**
Andriushchenko *et al.* (PA-RN-18) [51]	AA	79.84%	**43.93%**	61.88%
Andriushchenko *et al.* + DBD		**86.52%**	43.60%	**65.06%**
TRADES (WRN-28) [53]	AA	84.92%	**52.51%**	68.71%
TRADES (WRN-28) [53] + DBD		**88.02%**	52.12%	**70.07%**

seen, the of the accuracy in clean data is up to 6%. Also, the drop of the robust accuracy in some models is almost negligible across different network architectures and defence mechanisms. The results illustrate that the proposed DBD framework is not dependent on either the network architectures or the defence mechanism used to train the robust model or even having similar network architectures for both robust and non-robust models. It is observed that the proposed DBD framework is an effective defence mechanism that can boost the clean accuracy performance and achieve the highest average over clean and robust accuracy of various established and recently proposed defence mechanisms to a higher level on different architectures.

D. GATING THRESHOLD TRADE-OFF

To analyze the impact of the choice of gating threshold on the performance of the proposed DBD framework, Figures 3, 4 demonstrate the robust and clean accuracy of DBD framework on different defence algorithms based on various gating thresholds against AutoAttack and APGD attacks, respectively. The experiment was conducted on the ImageNet dataset. The non-robust model used here is ResNet-50 for all eight plots, while the robust model used for plot (a) is Salman *et al.* [16] with architecture ResNet-50; (b) is Engstrom *et al.* [17] with architecture ResNet-50; (c) is Wong *et al.* [41] with architecture ResNet-50, and (d) is Salman *et al.* [16] with architecture WideResNet-50.

As evident by the plots, as the threshold increases from 0 to 1, the clean accuracy rises while the robust accuracy decreases. It is evident that we have the highest clean accuracy at threshold 1 while the robust accuracy hits zero. This analysis highlights the impact of threshold in helping users balance the trade-off between robust and clean accuracy based on the circumstance and the expectation of the model's behaviour.

As seen by all plots, the proposed DBD framework is reliable in selecting various gating thresholds, which makes it self-sustain and reduces the need for rigorous hyper-parameter tuning. The proposed framework is a plug-and-play approach for improving the model's performance with any detailed hyper-parameter tuning.

E. MULTI MODEL DBD

The extension of the DBD framework beyond two models is straightforward. We still assume at least one non-robust and one robust model are in the set of models. For DBD variance, here, we take a simple decision making where the maximum DBD variance among all models is calculated, and if it is higher than the threshold, we use the model with the highest robust accuracy for the final output. In this regard, if the DBD variance is less than the threshold, we use the model with the highest clean accuracy for the final output. However, it is possible to take advantage of different ensemble techniques in the literature to predict the final output based on the measured DBD variance.

An example of three models is evaluated in Table 4. It can be seen that the robust accuracy does not change with the DBD framework, yet we have an increase of around 1.5% on clean accuracy. In this example, we used two robust models from Salman *et al.* [16] and Wong *et al.* [41]. The non-robust model is ResNet-50, trained on clean data. The attack used here is AutoAttack on the ImageNet dataset.

TABLE 4: Multi-model DBD framework performance on **ImageNet Dataset**. The abbreviations stand for C-Acc: Clean Accuracy, R-Acc: Robust Accuracy, RN-50: ResNet-50, WRN-50: WideResNet-50s.

Model	Attack	C-Acc	R-Acc
Salman *et al.* (RN-50) [16]	AutoAttack	64.06%	34.64%
Wong *et al.* (RN-50) [41]		53.44%	25.06%
Salman *et al.* + Wong *et al.* + DBD		**65.64%**	34.64%

V. DBD VARIANCE DISTINGUISHES ADVERSARIAL ATTACKS EVEN IF THEY ARE NOT OPTIMAL

While in general the original version of DBD needs to execute two network architectures to identify whether the input data sample is perturbed or it is a clean data sample for the gating mechanism; it is possible to reduce the computational complexity of the proposed framework significantly in real-world applications.

The well-known adversarial attacks generate the final perturbation to fool the target machine learning model by querying the model iteratively. One of the main benefits of the proposed DBD variance is to identify whether a sample is malicious or not. Therefore, if it is possible to identify whether the query originated from a source is perturbed in early stages of adversarial attack generation, then only the robust model needs to be used without requiring further DBD variance calculations for the consecutive queries originated from the malicious source. Therefore, the DBD variance only needs to be calculated once for all samples originated from that source. To this end, we illustrate that the proposed DBD variance can distinguish the clean data sample from perturbed sample even if the perturbation is not optimal and in early stage of adversarial generation/optimization.

In Figure 5, the DBD variance scores are presented for both unperturbed samples and those subjected to perturbations at different steps of the adversarial attack generation process. It is worth to note that an adversarial attack takes multiple steps (e.g., $\sim$100 steps) to generate an optimal adversarial sample which can fool the target network architecture.

As mentioned in main manuscript, the DBD Algorithm formulates the DBD variance as below:

$$Score = \max\Big(Var(M'|M), Var(M|M')\Big) \qquad (19)$$

where, M and M' refer to the robust and non-robust models, respectively. Here, the score is computed for different steps of the targeted APGD attack on the test set of ImageNet. Both the robust and non-robust models used are ResNet-50 architectures, with the robust model trained based on the approach proposed by Salman *et al.* [16]. As shown, the DBD variance is higher for adversarial data compared to normal data, particularly for multi-step attacks even in the first iteration of the APGD algorithm.

Based on the illustrated results in Figure 5 the proposed DBD framework can identify whether a sample is perturbed by a malicious source (i.e., possibly an adversarial attack) even in the first step of the attack algorithm. As such, it is possible to only execute the robust model from the first iteration of an attack and there is no need to calculate DBD variance for samples originated from that source. This can potentially reduce the computational expense of the algorithm

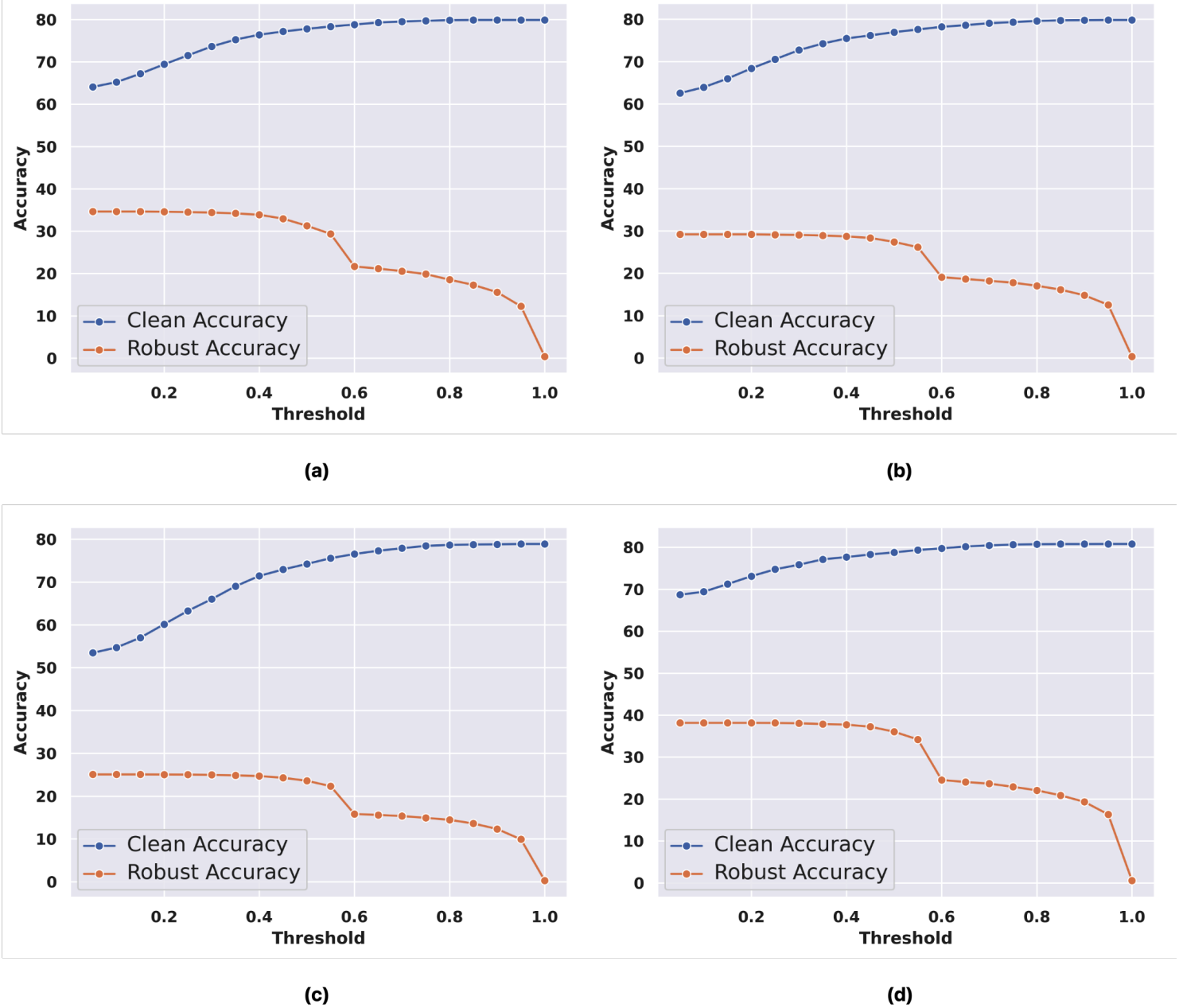

FIGURE 3: The impact of different threshold settings on DBD framework. The plots show the robust versus clean accuracy against AutoAttack for four different defense algorithms used with the proposed DBD framework. The experiment was conducted on ImageNet dataset. The non-robust model in this experiment was ResNet-50. The robust model for (a) is from Salman *et al.* [16] with architecture ResNet-50; for (b) is from Engstrom *et al.* [17] with architecture ResNet-50; (c) is from Wong *et al.* [41] with architecture ResNet-50 and (d) is from Salman *et al.* [16] with architecture WideResNet-50.

significantly, as the DBD variance does not need to be computed for every step of the attack which means there is no need to execute more than one model to predict the result. This approach makes the amortized computational complexity of the proposed DBD framework negligible and makes it closer to execute only one model to predict the output on average computation. Here we name this version of the proposed framework as Fast DBD. The pseudocode for Fast DBD is provided in Algorithm 2. In the next section, we analyze what is the amortized computational complexity of the proposed Fast DBD framework if we assume 50% samples are generated from clean sources and 50% are generated from malicious sources.

A. AMORTIZED COMPUTATIONAL COMPLEXITY

Let us assume the model services $2N$ users from independent sources which 50% are originated from benevolent sources and the second 50% originated from malicious sources. The malicious sources query the model on average 100 times to try to fool the system. Therefore, the model is executed $101N$ times in total to address all requests within a regular framework (i.e., having a robust model in place to service the

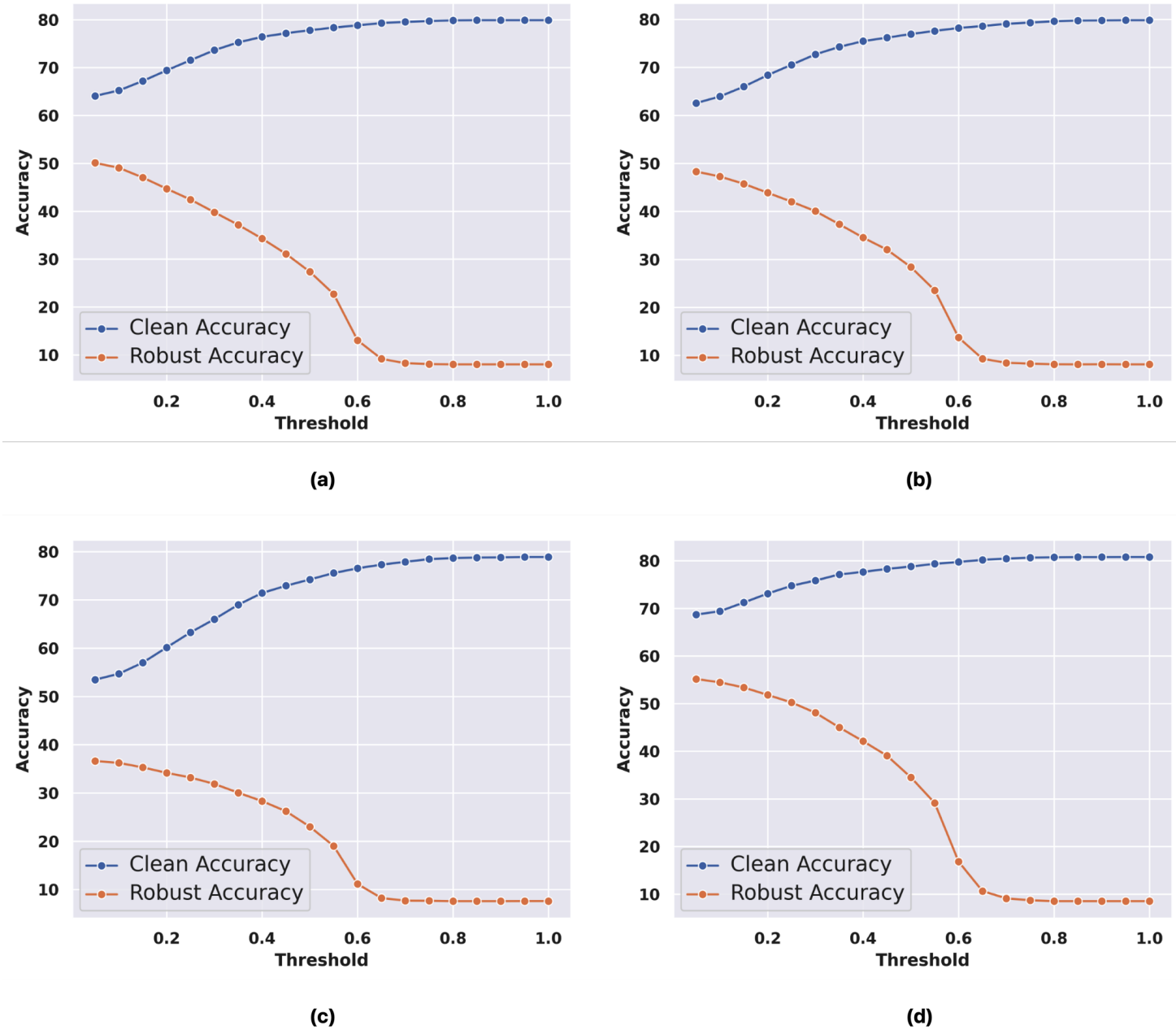

FIGURE 4: The impact of different threshold settings on DBD framework. The plots show the robust versus clean accuracy against APGD attack for four different defense algorithms used with the proposed DBD framework. The experiment was conducted on ImageNet dataset. The non-robust model in this experiment was ResNet-50. The robust model for (a) is from Salman *et al.* [16] with architecture ResNet-50; for (b) is from Engstrom *et al.* [17] with architecture ResNet-50; (c) is from Wong *et al.* [41] with architecture ResNet-50 and (d) is from Salman *et al.* [16] with architecture WideResNet-50.

queries). This is calculated based on the fact that the malicious sources query the robust model 100 times and we have N sources while the N benevolent sources only query the system once.

Now let us calculate how many times the Fast DBD framework executes one of the models to address all the requests. As illustrated in the previous section, the DBD framework can identify whether a sample is perturbed by a malicious source in the very first iteration queried by the source. As such, for those samples, the DBD framework executes both models (i.e., robust and non-robust models) in the first iteration of the query and executes only the robust model from the second iteration moving forward as it flags the source. This means that $101N$ times (i.e., 100 times execution of the robust model and 1 time executing the non-robust model in the first iteration of the attack) a model needs to be executed. For the clean samples, the DBD framework execute both robust and non-robust model and, therefore, $2N$ times model executions is the cost of processing all requests coming from benevolent sources (i.e., clean data samples). As such, the total cost for using DBD framework is $103N$ times model execution for all queries. Therefore, the proposed DBD framework only

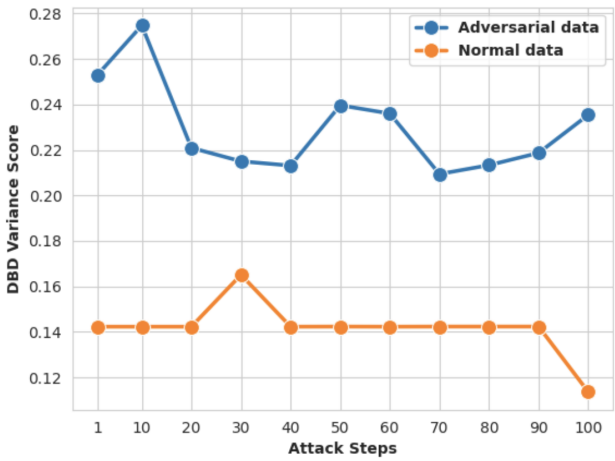

FIGURE 5: DBD variance score for targeted APGD at different steps. For DBD variance score, similar to DBD algorithm, we have reported the maximum of the variance distance between robust and non-robust model used in Algorithm 1 of main paper.

TABLE 5: Performance comparison of Fast DBD, DBD, and Salman et al.'s (2020) robust model on the WideResNet-50 architecture, using a batch size of 128. The graph displays average times calculated over multiple samples.

Model	Attack	C-Acc	R-Acc	Time
Salman *et al.* (WRN-50) [16]	APGD	68.46%	38.14%	**0.02501**
Salman *et al.* + Fast DBD		71.12%	38.14%	0.02630
Salman *et al.* + DBD		**72.3%**	38.04%	0.05210

Algorithm 2: Fast DBD Framework

Input: S=$\left\{\text{x}|\ \text{x} \in D\right\}$, Threshold t

Input: Robust model M and Non-Robust Model $M^{'}$ with c distinct classes, $L_M(x)$: Softmax layer

Result: $R = \left\{\hat{y}(x) \mid x \in D\right\}$

$R = [\,]$

for x *in* S **do**

  x from a new source

  Forward Pass $M(x) \rightarrow L_M(x)$

  if *Benevolent-Flag* **then**

    Forward Pass $M'(x) \rightarrow L_{M'}(x)$

    $Score = Max\left(Var(M'|M), Var(M|M')\right)$

    if $Score > t$ **then**

      $R.add(M(x))$

      Benevolent-Flag=False

    else

      $R.add(M'(x))$

  else

    $R.add(M(x))$

incurs and additional $\sim$ 2% extra computational cost in amortized computation to services all sources and satisfies their requests.

To validate the amortized computational analysis in practice here we demonstrate the potential of Fast DBD in improving performance using a real-world scenario. Specifically, we consider a scenario in which two clients make queries to the model: one client is normal, while the other client uses APGD targeted attack with 100 steps and an epsilon value of $\frac{4}{255}$. We assess the model's performance using the ImageNet test set and evaluate the time efficiency based on the average running time of total queries to the network. The average time is calculated over the entire test dataset, encompassing both normal and adversarial queries, with the understanding that adversarial inputs necessitate a greater number of queries (on the order of 100) to the model. The results, shown in Table 5, indicate that while DBD provides the highest clean accuracy, its running time is twice as long as Fast DBD. Moreover, we observe that Fast DBD incurs a much smaller increase (in the magnitude of 5% on average which is consistent with the the amortized analysis above) in the running time of the robust model proposed by Salman *et al.* [16], while still increasing the clean accuracy.

VI. DISCUSSION

In this study, we introduced the DBD framework, a novel and effective defense mechanism for deep learning models against adversarial attacks. The DBD framework not only elevates performance on clean data but also upholds robust accuracy. Central to our approach is the use of DBD Variance as a gating mechanism, allowing the system to discern whether a given sample has been manipulated by an adversarial attack, and subsequently, if a robust model is essential for accurate prediction. Our inspiration for this method stems from a comprehensive exploration of the bias-variance trade-offs' influence on adversarial attacks, both from empirical and theoretical perspectives. Our findings indicate that adversarial attacks amplify variance, a key insight that guided the formulation of our DBD defense mechanism. Notably, the DBD framework operates unsupervised and seamlessly integrates with any defense model, adeptly reconciling the trade-off between clean data accuracy and robust accuracy. Our empirical evaluations underscore the potency of DBD, showcasing an enhancement in clean data accuracy by up to 6%, while incurring a minuscule decrement in robust accuracy. At the heart of DBD lies the innovative DBD Variance, which adeptly gauges the model's variance, sidestepping data leakage issues and performance pitfalls that plagued previous variance-centric methodologies.

A. LIMITATION AND FUTURE WORK

While this work proposes a novel approach to mitigate the sharp drop in clean accuracy caused by adversarial training, it requires the use of two models simultaneously, which is computationally expensive. We provided heuristic solutions to mitigate this additional cost, but future work should in-

vestigate less computationally expensive approaches that can approximate the variance.

As a forward-looking avenue, utilizing DBD Variance as an alternative loss function presents an intriguing prospect, potentially outperforming traditional loss functions like cross-entropy in enhancing model performance across both clean and adversarial datasets.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[3] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[4] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[6] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[7] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7:46450, 2017.

[8] Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, et al. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9):1342–1350, 2018.

[9] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

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

[11] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[12] Liang Tong, Sixie Yu, Scott Alfeld, and Yevgeniy Vorobeychik. Adversarial regression with multiple learners. *arXiv preprint arXiv:1806.02256*, 2018.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[14] Ruitong Huang, Bing Xu, Dale Schuurmans, and Csaba Szepesvári. Learning with a strong adversary. *arXiv preprint arXiv:1511.03034*, 2015.

[15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[16] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[17] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[19] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. *arXiv preprint arXiv:2002.11328*, 2020.

[20] Min Chen and Mateu Sbert. On the upper bound of the kullback-leibler divergence and cross entropy. *arXiv preprint arXiv:1911.08334*, 2019.

[21] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 819–828, 2020.

[22] Aishan Liu, Shiyu Tang, Xianglong Liu, Xinyun Chen, Lei Huang, Zhuozhuo Tu, Dawn Song, and Dacheng Tao. Towards defending multiple adversarial perturbations via gated batch normalization. *arXiv preprint arXiv:2012.01654*, 2020.

[23] Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Perturbation type categorization for multiple adversarial perturbation robustness. In *Uncertainty in Artificial Intelligence*, pages 1317–1327. PMLR, 2022.

[24] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[25] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

[26] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[30] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-bnn: Improved adversarial defense through robust bayesian neural network. *arXiv preprint arXiv:1810.01279*, 2018.

[31] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016.

[32] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[33] IJ Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, A Courville, and Y Bengio. Generative adversarial networks. 2014 jun. doi: 10.48550. *arXiv preprint arXiv.1406.2661*.

[34] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Yingxia Shao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*, 2022.

[35] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17:151–178, 2020.

[36] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. *arXiv preprint arXiv:1805.06605*, 2018.

[37] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

[38] Xuanqing Liu and Cho-Jui Hsieh. Rob-gan: Generator, discriminator, and adversarial attacker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11234–11243, 2019.

[39] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[40] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[41] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[42] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
[43] George Brinton Thomas, Maurice D Weir, Joel Hass, Frank R Giordano, and Recep Korkmaz. *Thomas' calculus*. Pearson Boston, 2010.
[44] D. Pfau. A generalized bias-variance decomposition for bregman divergences. 2013.
[45] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pages 231–238, 2000.
[46] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021.
[47] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.
[48] Kaustubh Sridhar, Oleg Sokolsky, Insup Lee, and James Weimer. Improving neural network robustness via persistency of excitation. In *2022 American Control Conference (ACC)*, pages 1521–1526. IEEE, 2022.
[49] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020.
[50] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? *arXiv preprint arXiv:2104.09425*, 2021.
[51] Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.
[52] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021.
[53] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.
[54] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.
[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
[56] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

HOSSEIN ABOUTALEBI is a PhD candidate at the University of Waterloo, specializing in computer vision and large language models, with a particular emphasis on generative AI. Throughout his doctoral studies, Hossein has contributed numerous publications in the realm of computer vision applications. He has also undertaken internships with prominent organizations such as Amazon AWS, Cerebras, and EAIGLE. Prior to his PhD journey, he served as an applied research scientist at Deeplite.

MOHAMMAD JAVAD SHAFIEE is currently the Cofounder and VP Research at DarwinAI and an Adjunct Professor at the Department of Systems Design Engineering at University of Waterloo. He received the B.Sc. and M.Sc. degrees in Computer Science and Artificial Intelligence, in 2008 and 2011 respectively; and the Ph.D. degree in systems design engineering with the focus on Machine Learning and Deep Learning from the University of Waterloo, Canada in 2017. His main research focus is on statistical learning and graphical models with random fields and deep learning approaches. His research interests include Computer Vision, Machine Learning and Biomedical Image Processing.

CHI-EN AMY TAI is a graduate student in the Vision and Image Processing Lab at the University of Waterloo. Her research areas include computer vision, machine learning, and explainable AI. She focuses on cancer diagnosis and prognosis using medical imaging, personalized nutrition intervention for the aging population, and improved rendering of 3D models. Previously, she completed her Bachelor of Applied Science in Management Engineering at the University of Waterloo.

ALEXANDER WONG Alexander Wong, P.Eng., is currently the Canada Research Chair in Artificial Intelligence and Medical Imaging, co-director of the Vision and Image Processing Research Group, and a professor in the Department of Systems Design Engineering at the University of Waterloo. Dr. Wong is a Fellow of the Royal Society for Public Health, the Institution of Engineering and Technology, Institute of Physics, the International Society for Design and Development of Education, and Member of the College of the Royal Society of Canada. He has published over 600 refereed journal and conference papers, as well as patents, in various fields such as computational imaging and artificial intelligence, and has received numerous research, industry, and best paper awards.