Dual-model Bounded Divergence Gating for Improved Clean Accuracy in Adversarial Robust Deep Neural Networks

Hossein Aboutalebi* Mohammad Javad Shafiee†‡ Amy Tai† Alexander Wong†‡

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 ImageNet benchmark datasets across different adversarial attacks (e.g., APGD, AutoAttack, and FAB) demonstrate that the integration of DBD can lead to as much as a* 6% *improvement on clean data accuracy without compromising much on adversarial robustness.*

1. Introduction

A perturbation ϵ in a specific direction added to the input sample fools the model, which results in an incorrect prediction; this process can be applied in both classification [9, 11] or regression problems [1, 12]. In this regard, perturbation ϵ is imperceptible by a human eye and is enforced by bounding the norm of ϵ when generated. Szegedy *et al.* [11] introduced this drawback for deep neural networks in their seminal paper.

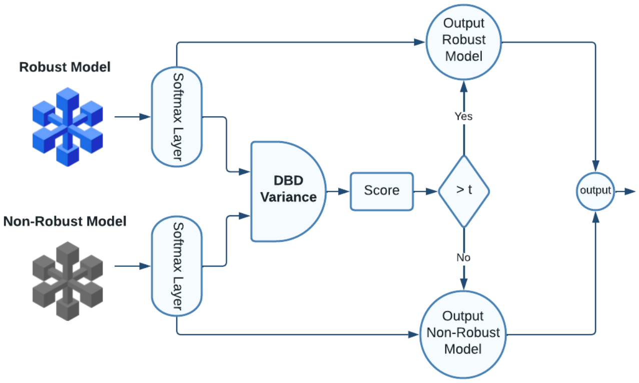

Figure 1: Overview of the proposed DBD defense mechanism. The DBD Variance is computed between the softmax outputs of an adversarially trained model (robust model) and a conventionally trained model (non-robust model) and a score is produced. Based on the score, a gating mechanism is used to decide whether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

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.

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

[*] haboutal@uwaterloo.ca, Cheriton School of Computer Science, University of Waterloo
[†] Department of Systems Design Engineering, University of Waterloo
[‡] DarwinAI Corp., Canada

proposed in prior research that rely on unbounded measures like KL-Divergence [14]. 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) [2].

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.

2. 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 [10, 5, 8]. 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.

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 1.

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.

2.1. 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.

2.1.1 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:

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

Algorithm 1: DBD Framework

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

3. Experiments

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.

Algorithm 2: Fast DBD Framework

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

ImageNet [4] dataset was 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) [6, 15, 7]. In this regard, the experimental implementation is developed based on the toolkit provided by RobustBench library [3] to ensure that the results are reproducible.

The source code for all experiments can be found here.

3.1. ImageNet Dataset

Table 1 presents the results of integrating the proposed DBD framework with defence models used in [10, 5, 13] 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. 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 1 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.

Table 1: 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) [10] Salman *et al.* + DBD	AA	64.02% **71.32%**	**34.96** 34.24%	49.49% **52.78%**
Engstrom *et al.* (RN-50) [5] Engstrom *et al.* + DBD	AA	62.56% **64.88%**	29.2% 29.2%	45.88% **47.04%**
Wong *et al.* (RN-50) [13] Wong *et al.* + DBD	AA	53.44% **59.2%**	25.06% 25.04%	39.25% **42.12%**
Salman *et al.* (WRN-50)[10] Salman *et al.* + DBD	AA	68.46% **72.3%**	**38.14%** 38.04%	53.3% **55.17%**
Salman *et al.* (RN-50) [10] Salman *et al.* + DBD	APGD	64.02% **66.10%**	**34.96%** 34.42%	49.49% **50.26%**
Engstrom *et al.* (RN-50) [5] Engstrom *et al.* + DBD	APGD	62.56% **66.74%**	**29.32%** 28.98%	45.96% **47.86%**
Wong *et al.* (RN-50) [13] Wong *et al.* + DBD	APGD	53.44% **54.52%**	25.06% **25.10%**	39.25% **39.81%**
Salman *et al.* (WRN-50)[10] Salman *et al.* + DBD	APGD	68.46% **71.12%**	**38.22%** 37.70%	53.34% **54.51%**
Salman *et al.* (RN-50) [10] Salman *et al.* + DBD	FAB	64.06% **66.10%**	**36.82%** 36.02%	50.44% **51.06%**
Engstrom *et al.* (RN-50) [5] Engstrom *et al.* + DBD	FAB	62.52% **64.88%**	31.44% 31.12%	46.98% **48.00%**
Wong *et al.* (RN-50) [13] Wong *et al.* + DBD	FAB	53.44% **54.52%**	30.80% 30.60%	42.12% **42.56%**
Salman *et al.* (WRN-50)[10] Salman *et al.* + DBD	FAB	68.46% **71.12%**	**40.68%** 39.74%	54.57% **55.43%**

3.2. Fast DBD

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 pseudocode for Fast DBD is provided in Algorithm 2. Basically in Fast DBD, we switch to the robust model the DBD variance score for the model exceeds the threshold. This is motivated by the fact that well-known adversarial attacks generate the final perturbation to fool the target machine learning model by querying the model iteratively (multi-step attacks). One of the main benefits of the proposed DBD variance is to identify whether a sample is malicious or not. Therefore, 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

Table 2: 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) [10]	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

from that adversarial source reducing the computational time substantially.

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 2, 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.* [10], while still increasing the clean accuracy.

4. Conclusion

The DBD framework improves the performance of a deep learning model on clean data samples while maintaining its robust accuracy. The proposed DBD framework benefits from DBD Variance as a gating mechanism to determine if a sample is perturbed by an adversarial attack and if a robust model is required for the prediction. The experimental results show the efficacy of the proposed defence model, which could improve the clean data accuracy of the model by up to 6% with a negligible drop in robust accuracy. The core element of the proposed framework is based on a novel DBD Variance which can be used to determine the model's variance without any data leakage problem and decrease of performance which existed in prior variance measure works.

References

[1] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1

[2] Min Chen and Mateu Sbert. On the upper bound of the kullback-leibler divergence and cross entropy. *arXiv preprint arXiv:1911.08334*, 2019. 2

[3] 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. 3

[4] 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. 3

[5] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 2, 3

[6] 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. 3

[7] 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. 3

[8] 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. 2

[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. 1

[10] 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. 2, 3, 4

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

[12] Liang Tong, Sixie Yu, Scott Alfeld, and Yevgeniy Vorobeychik. Adversarial regression with multiple learners. *arXiv preprint arXiv:1806.02256*, 2018. 1

[13] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020. 3

[14] 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. 2

[15] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 3