DVERGE: Diversifying Vulnerabilities for Enhanced Robust Generation of Ensembles

Huanrui Yang[1*], Jingyang Zhang[1*], Hongliang Dong[1*], Nathan Inkawhich[1], Andrew Gardner[2], Andrew Touchet[2], Wesley Wilkes[2], Heath Berry[2], Hai Li[1]

[1]Duke University, [2]Radiance Technologies

NeurIPS 2020 (Oral)

Duke RADIANCE TECHNOLOGIES

NEURAL INFORMATION PROCESSING SYSTEMS

Outline

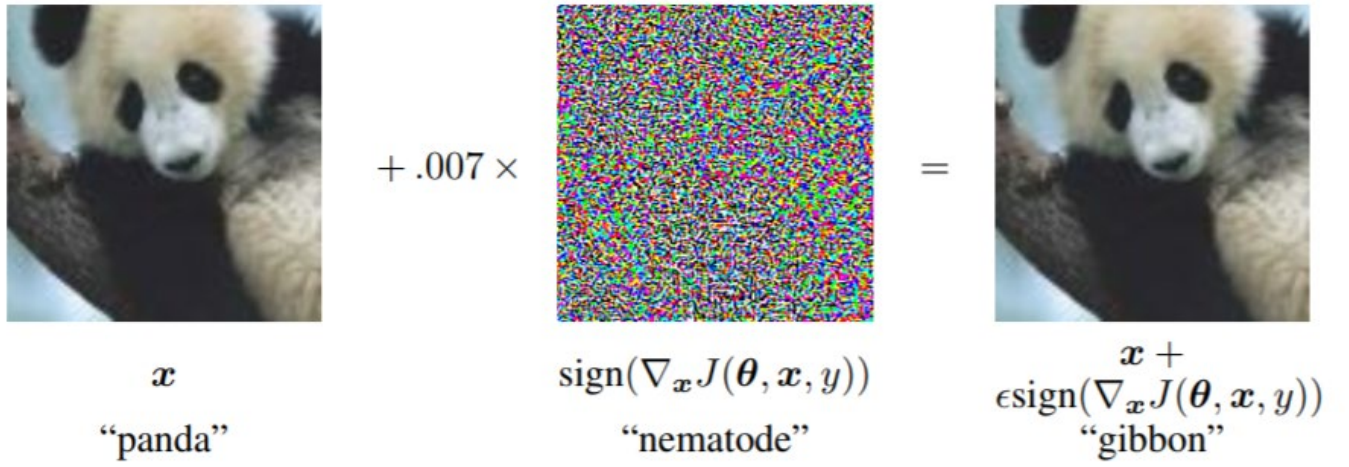
- **Issue with robustness: Adversarial attack and adversarial training**
- **Vulnerability vs. Learnability: Robust and non-robust features**
- **Vulnerability diversification: Combining diverse non-robust sub-models into a robust ensemble**
- **Experiment results of DVERGE**

Lack of Robustness in Deep Learning Models

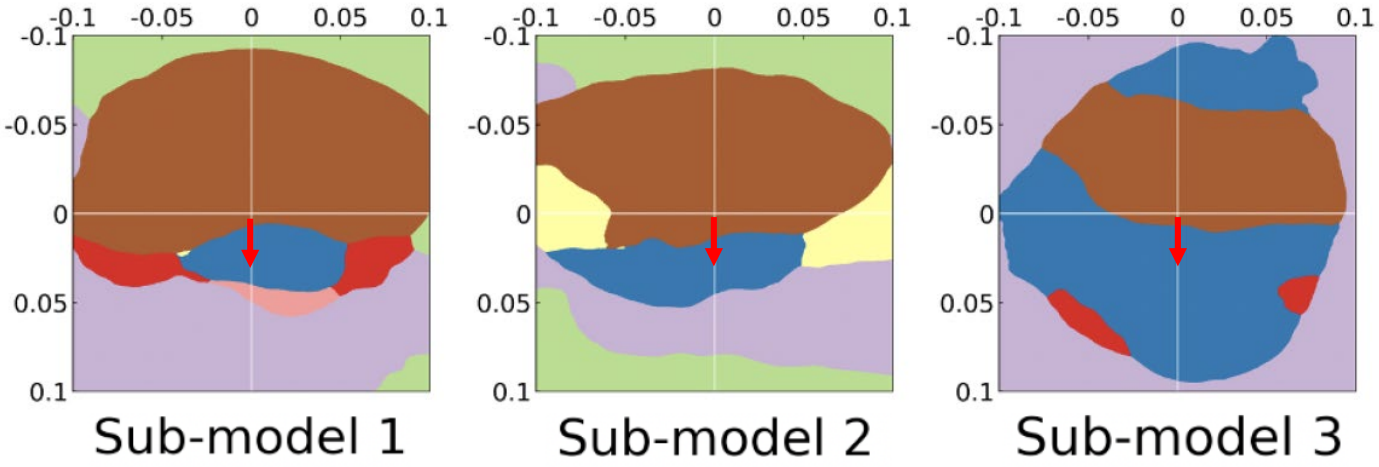
- **Small adversarial noises can mislead the model**

$$\max \mathcal{L}\big(x+\delta, f(x)\big)\ s.t.\ \delta \in S$$

- **Models suffer from black-box (transfer) attacks generated on a surrogate model -> threat real-world applications**



Ian J Goodfellow, et al. Explaining and harnessing adversarial examples.



We must mitigate attack transferability while maintaining high clean accuracy.

↓: Adversarial direction of a surrogate standard model

General Robustness Improvement: Adversarial Training

- **Mini-max objective**

$$\min_\theta \mathbb{E}_{(x,y)\sim D}[\max_{\delta\in S} \mathcal{L}_\theta(x+\delta, y)]$$

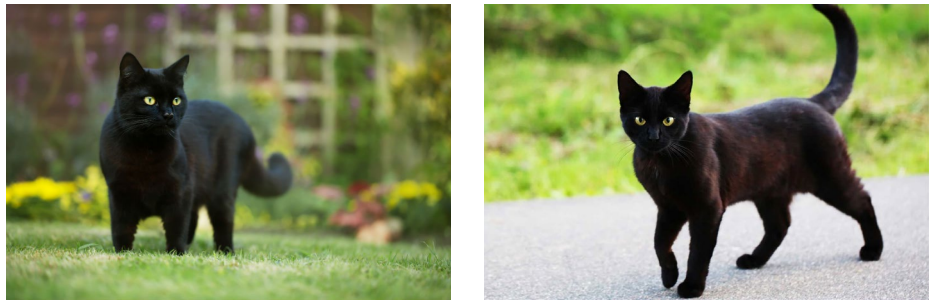
  - Training with online generated adversarial examples, forcing the model to correctly classify adversarial examples

- **Effectively improves robustness, but hinders clean accuracy**
  - Results evaluated with ResNet-20 ensembles under transfer PGD attack ($\epsilon = 0.01$) on CIFAR-10 dataset

Ensemble size	Baseline Acc	Baseline Rob	AdvT Acc	AdvT Rob
3	93.9	9.6	77.2	76.3
5	93.9	9.4	78.6	77.8
8	94.4	9.7	79.4	78.2

Why is adversarial training hard?

- **Why deep learning models are naturally non-robust?**
  - Models tends to learn “simplest” feature correlated to the class label, not robust
- **Example: Distinguish class “A” against class “B”**

A  B 

  - Human: Dog -> A, Cat -> B, using “robust” feature
  - CNN: White -> A, Black -> B, highly-correlated but “non-robust” feature
- **Non-robust features are common in image dataset, easy to learn**
- **Adversarial training enforces the utilization of robust features**
  - Hard to represent in CNN model, leading to low clean accuracy

Towards Robustness Improvement with Ensemble

- **Can we improve robustness without enforcing robust features?**
  - Impossible with single model: Not using the robust feature inevitably leads to some vulnerability in the model's classification

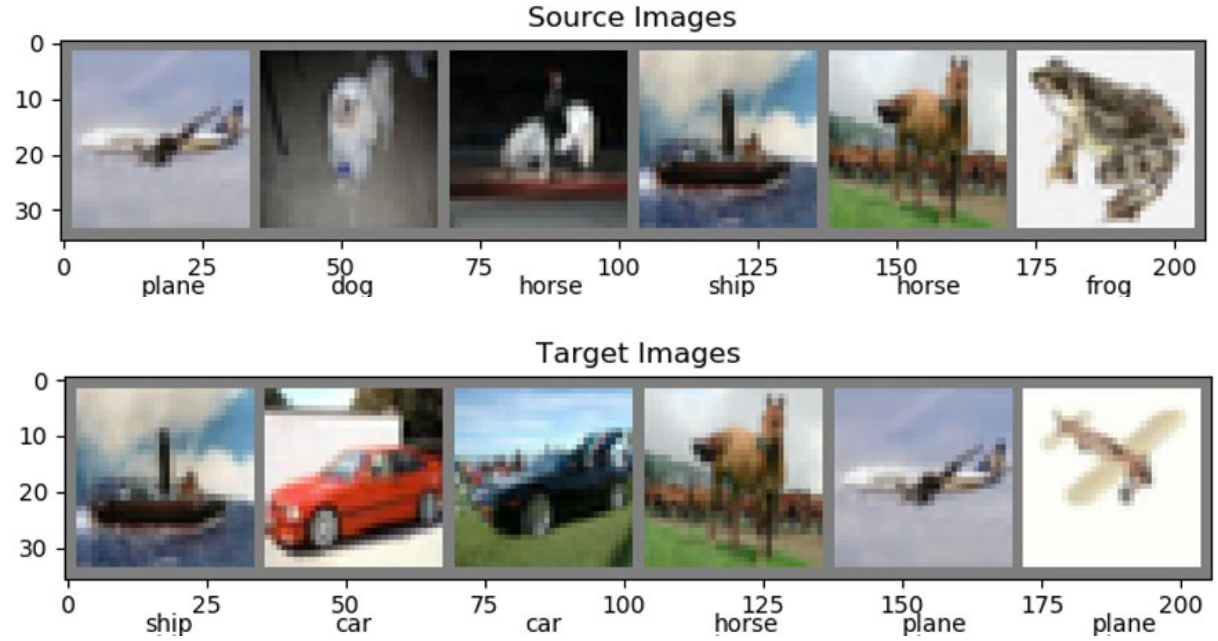
- **Robust improvement with diverse ensemble**
  - Weak learners with diverse output can combine into a high-performance ensemble
  - Hypothesis: Non-robust sub-models with diverse outputs against transfer attack can combine into a robust ensemble -> minimize transferability between sub-models

- **Need a diversity metric and a training method to induce such diversity**
  - Previous output/gradient-based diversity metrics not effective enough against transfer attack

Adversarial Vulnerability Isolation

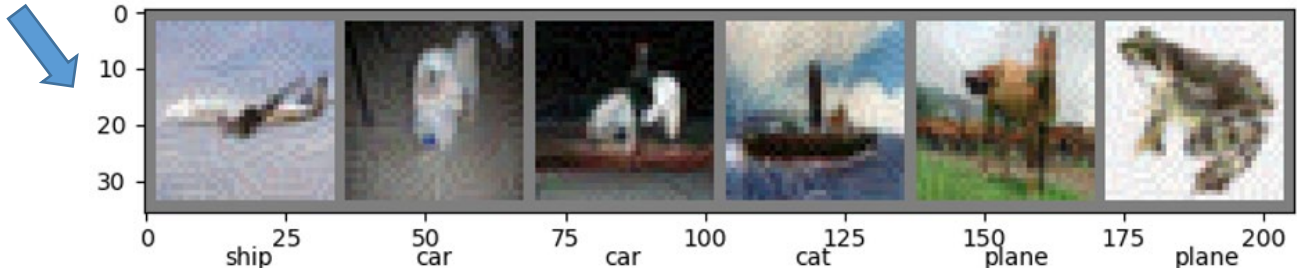
- **Why adversarial attacks transfer?**
  - Ilyas et al.: models trained on the same dataset captures a similar set of “non-robust features”, highly correlated to the labels yet sensitive to noise -> **vulnerability**
- **Vulnerability isolation**
  - Non-robust feature distillation: find image x' that’s **visually close** to source image x_s, but has the **same hidden feature** as another target image x in a layer

Ilyas, Andrew, et al. "Adversarial examples are not bugs, they are features." *Advances in Neural Information Processing Systems*. 2019.



$$x'_{f_i^l}(x, x_s) = \underset{z}{\operatorname{argmin}} \left\| f_i^l(z) - f_i^l(x) \right\|_2^2, \; s.t. \; ||z - x_s||_\infty \leq \epsilon,$$

Distilled non-robust images



Looks like source, but classified as target class label by the model, contain non-robust feature relating to the target class, characterize vulnerability

Vulnerability Diversification

- **Vulnerability diversity metric**
  - Models with similar vulnerability will both classify distilled non-robust image as the target class rather than the source class

$$d(f_i, f_j) := \frac{1}{2}\mathbb{E}_{(x,y),(x_s,y_s),l}\left[\mathcal{L}_{f_i}\left(x'_{f_j^l}(x, x_s), y\right) + \mathcal{L}_{f_j}\left(x'_{f_i^l}(x, x_s), y\right)\right]$$

Image from f_j classified by f_i Image from f_i classified by f_j

- **Diversification objective**
  - In practice, encourage sub-model to classify distilled image from other sub-models "correctly" as the visually-similar source class label, can also effectively contribute to minimize clean loss

$$\min_{f_i} \mathbb{E}_{(x,y),(x_s,y_s),l} \sum_{j \neq i} \mathcal{L}_{f_i}(x'_{f_j^l}(x, x_s), y_s)$$

DVERGE Training Algorithm

- **Round-robin adversarial diversity training**
  - Ask sub-model f_i to classify extracted feature from all other sub-models f_j as the source label Y_s, dynamically increases the diversity

    -> **Higher transfer robustness**
  - Not forcing the usage of robust feature: objective can be minimized if sub-model using different set of features, including non-robust features

    -> **Higher clean accuracy**

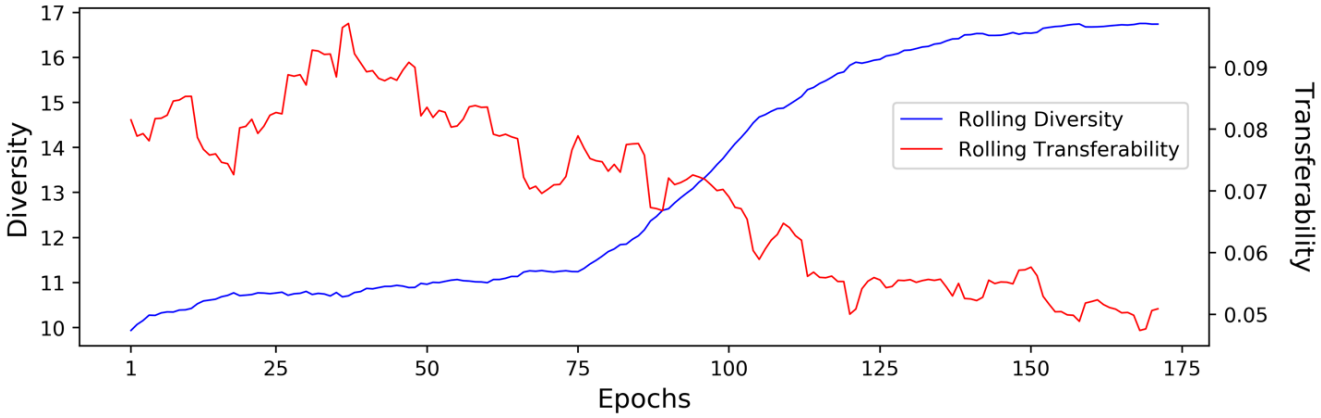
Objective: $\min_{f_i} \mathbb{E}_{(x,y),(x_s,y_s),l} \sum_{j \neq i} \mathcal{L}_{f_i}(x'_{f_j^l}(x, x_s), y_s)$

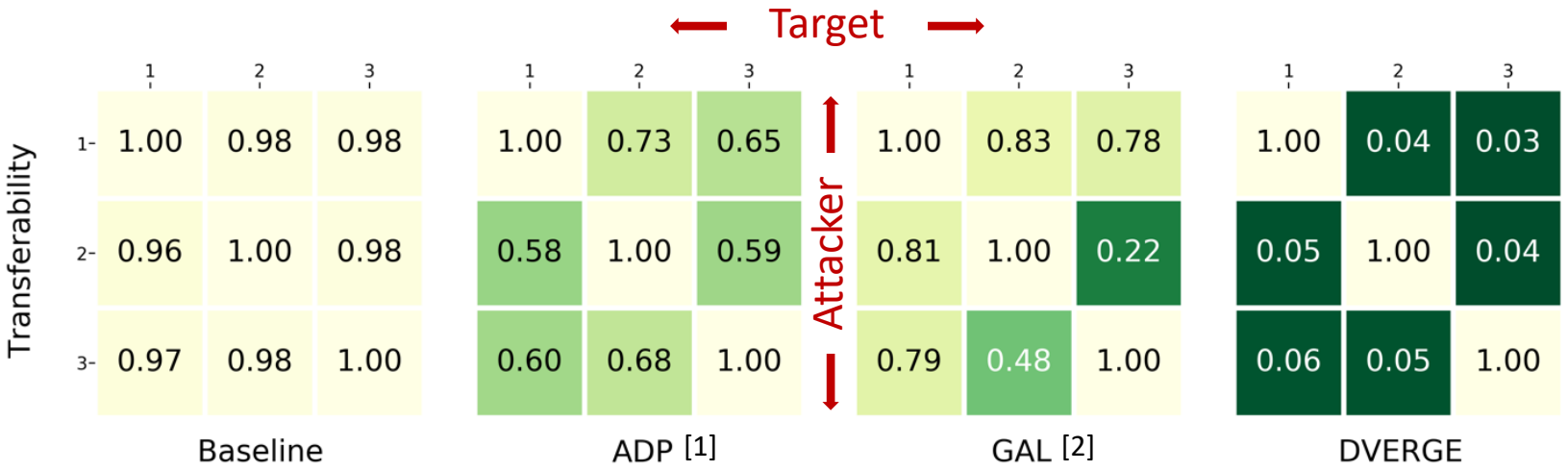
Algorithm 1 DVERGE training routine for a N-sub-model ensemble.

```
# initialization and pretraining
for i = 1, ..., N do
    Randomly initialize sub-model f_i
    Pretrain f_i with clean dataset
# round-robin feature diversification
for e = 1, ..., E do
    Uniformly randomly choose layer l for feature distillation
    for b = 1, ..., B do
        (X, Y) ← get batched input-label pairs
        (X_s, Y_s) ← uniformly sample batched source input-label pairs
        # get distilled batch for each model
        for i = 1, ..., N do
            X'_i := x'_{f_i^l}(X, X_s) ← non-robust feature distillation with Equation (1)
        # calculate loss and perform SGD update for all sub-models
        for i = 1, ..., N do
            ∇_{f_i} ← ∇[Σ_{j≠i} L_{f_i}(f_i(X'_j), Y_s)]
            f_i ← f_i − lr · ∇_{f_i}
```

Results: Diversity and Transferability

- DVERGE effectively maximizes the proposed vulnerability diversity and reduces the adversarial transfer within the ensemble





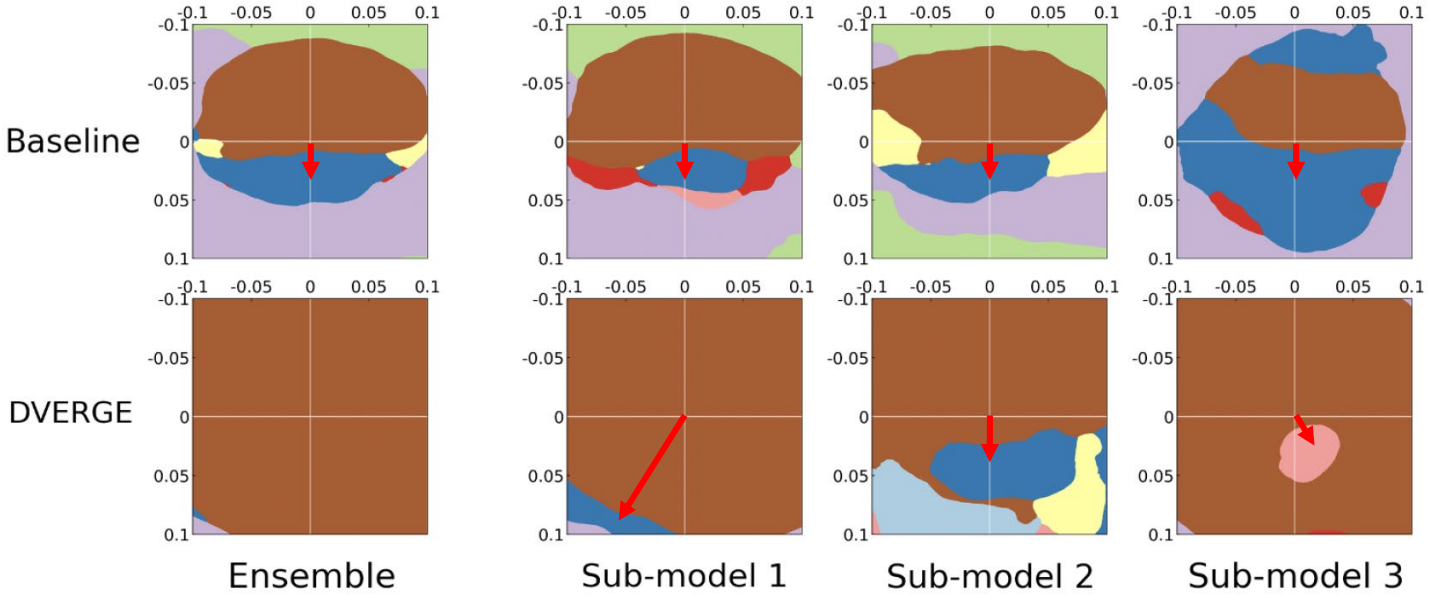
PGD attack ($\epsilon = 0.03$) success rate: **Lower == Better**

- DVERGE achieves significantly lower attack transferability between sub-models

[1] Pang, T., Xu, K., Du, C., Chen, N. and Zhu, J., "Improving Adversarial Robustness via Promoting Ensemble Diversity," *arXiv:1901.08846 [cs, stat]*, Jan 2019 [Online]. Available: https://arxiv.org/abs/1901.08846.

[2] Kariyappa, Sanjay, and Moinuddin K. Qureshi., "Improving adversarial robustness of ensembles with diversity training," *arXiv:1901.09981 [cs, stat]*, Jan 2019 [Online]. Available: https://arxiv.org/abs/1901.09981.

Decision Region Visualization



The vertical axis along the adversarial direction of a surrogate standard ensemble, and the horizontal axis along a random vector

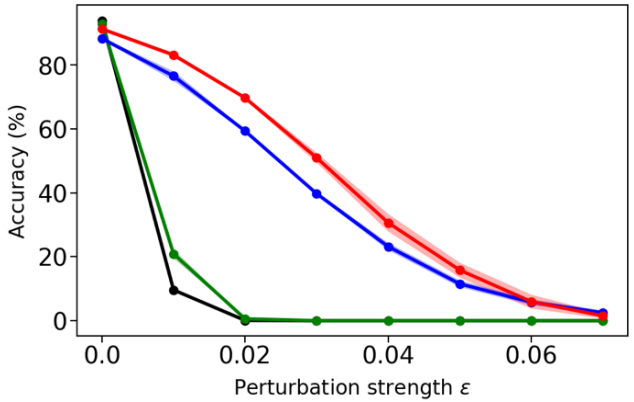
The color indicates the prediction label. Adversarial vulnerability can be inferred from the closest decision boundary and class

- Baseline sub-models demonstrates similar vulnerability, no adversarial robustness achieved in the ensemble
- Vulnerabilities allowed to persist in each sub-model of DVERGE; but are diversified
- Diverse vulnerabilities combine to yield an ensemble with greater robustness

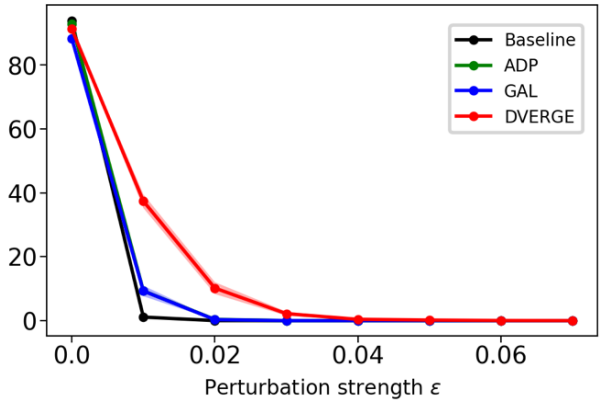
Ensemble Robustness - Results

- Greater robustness across various attack strengths, minimal accuracy loss

Black-box



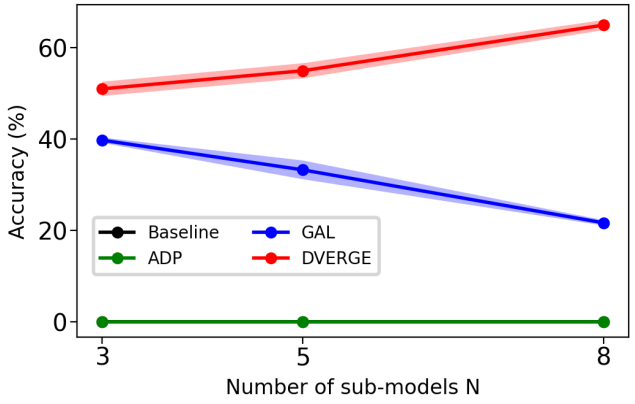
White-box



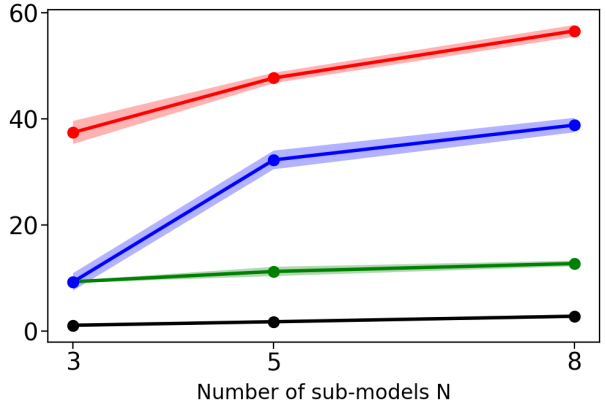
The robustness of ensembles with 3 sub-models.

- Consistently improves robustness as ensemble size increases

Black-box



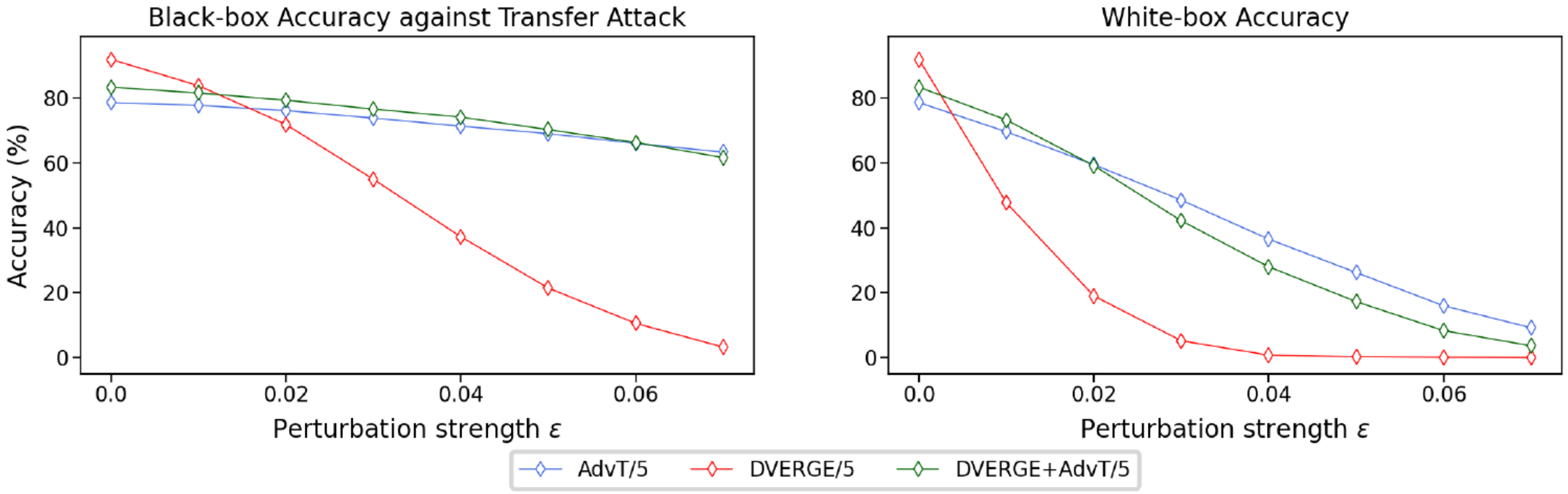
White-box



The robustness of ensembles with varying sizes under $\epsilon = 0.03$ (black-box) and $\epsilon = 0.01$ (white-box).

DVERGE + Adversarial Training

- Learning both robust and diverse non-robust features
- Higher clean accuracy and higher transfer robustness than AdvT
- Explore tradeoff between clean accuracy and white-box robustness



Ensembles with 5 sub-models.

Conclusions

- DVERGE: Diversifies the adversarial vulnerability in each sub-model to improve the overall transfer robustness without significant accuracy loss
  - **Vulnerability** characterized by **distilled non-robust features**
  - Diversifies vulnerability between sub-models with **round-robin training**, thereby **blocking attack transferability** within the ensemble
  - **Higher robustness** compared to previous ensemble training methods **with minimal clean accuracy loss**, **better with more sub-models**
  - **Higher clean accuracy** and **higher transfer attack robustness** when augmented to adversarial training

- Additional experiments and ablation studies results can be found in our paper

Thanks

More details can be found at:

Paper: https://arxiv.org/abs/2009.14720

Code: https://github.com/zjysteven/DVERGE

Duke RADIANCE TECHNOLOGIES

This work is supported by the DARPA HR00111990079 (QED for RML) program.

NEURAL INFORMATION PROCESSING SYSTEMS