Adversarial Vulnerability of Randomized Ensembles

Hassan Dbouk & Naresh Shanbhag

Department of Electrical and Computer Engineering

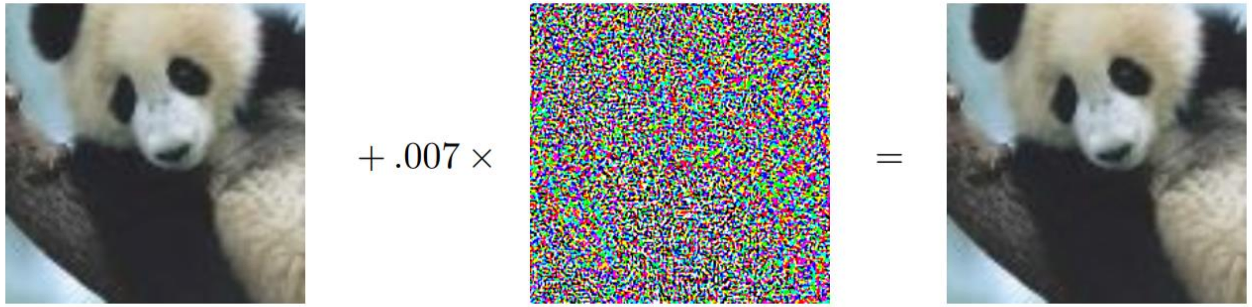
University of Illinois at Urbana-Champaign

Robust and Efficient Inference

deep nets are <u>vulnerable</u>

original sample

adversarial sample



decision: ‘panda’

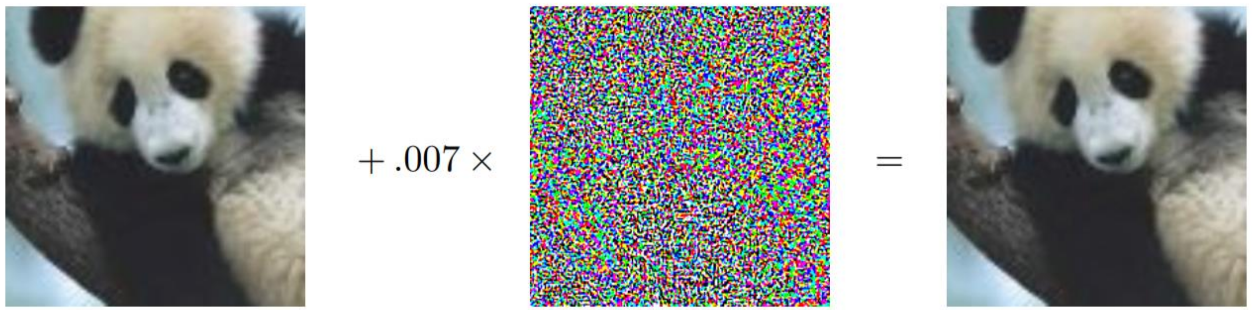
decision: ‘gibbon’

Robust and Efficient Inference

deep nets are vulnerable

original sample

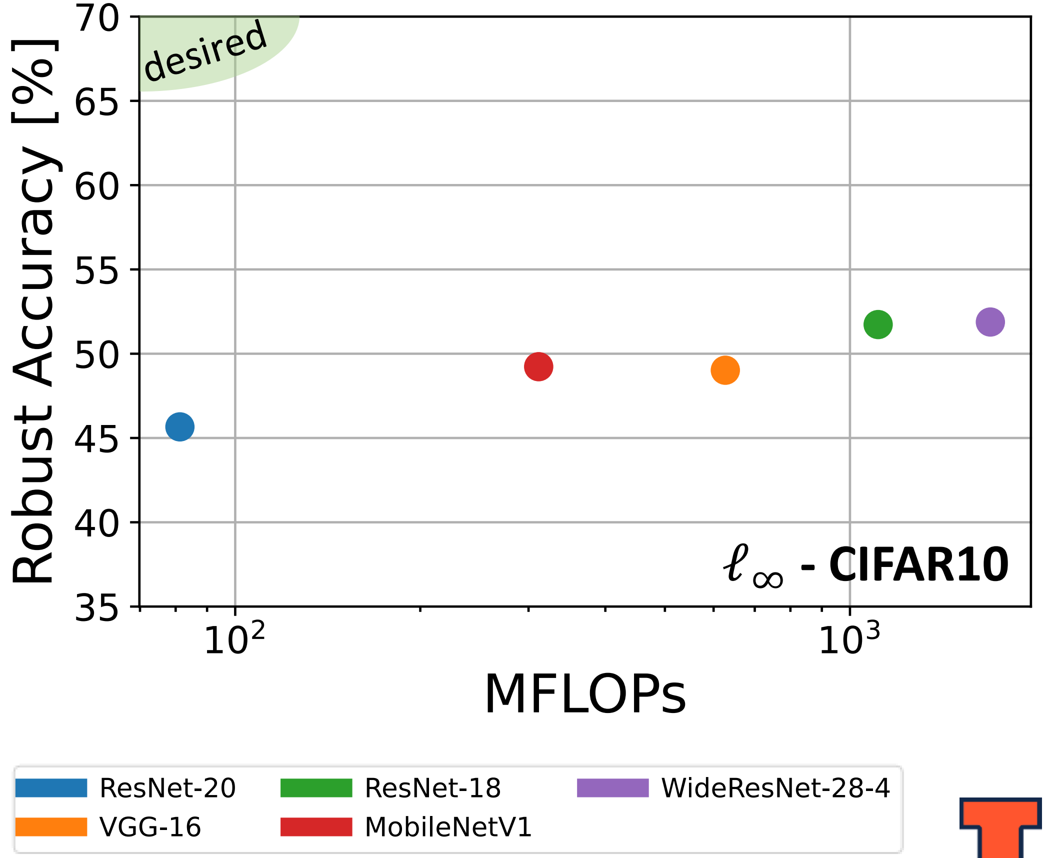
adversarial sample



decision: ‘panda’

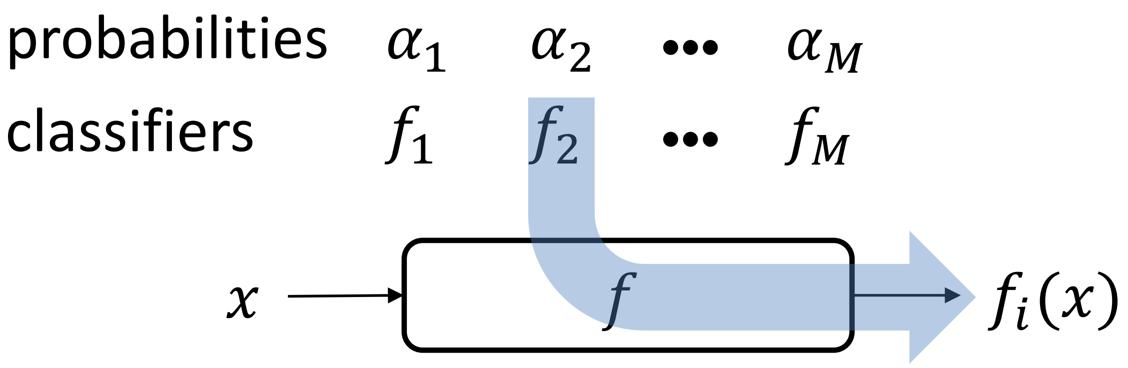
decision: ‘gibbon’

robustness is expensive



Robustness via Randomized Ensembles

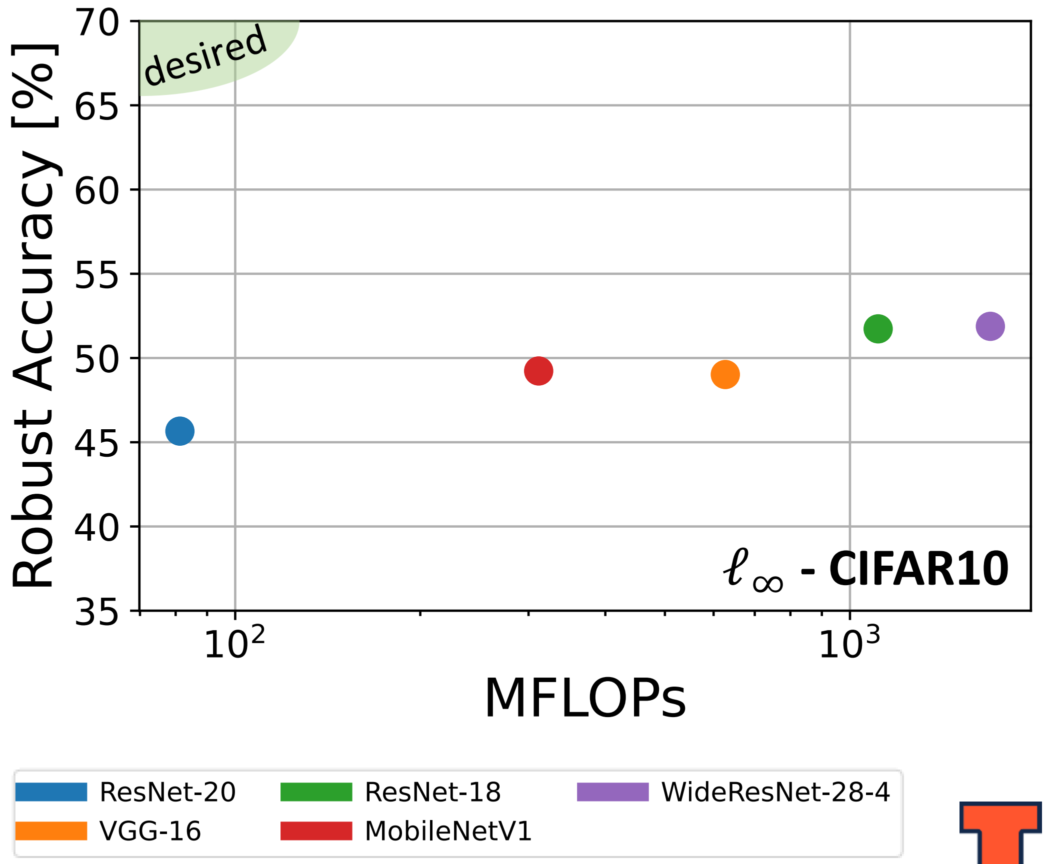
multiple classifiers $f_1, \dots, f_M$



inference: pick **one** at random

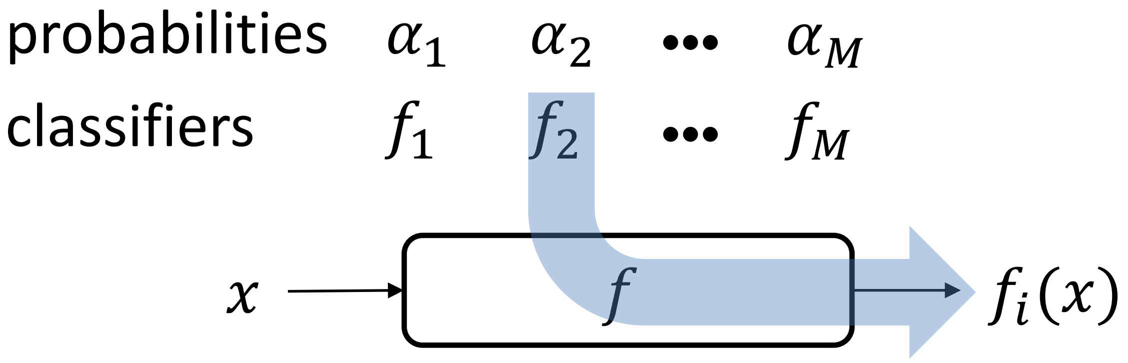
no increase in # of FLOPS

robustness is <u>expensive</u>



Robustness via Randomized Ensembles

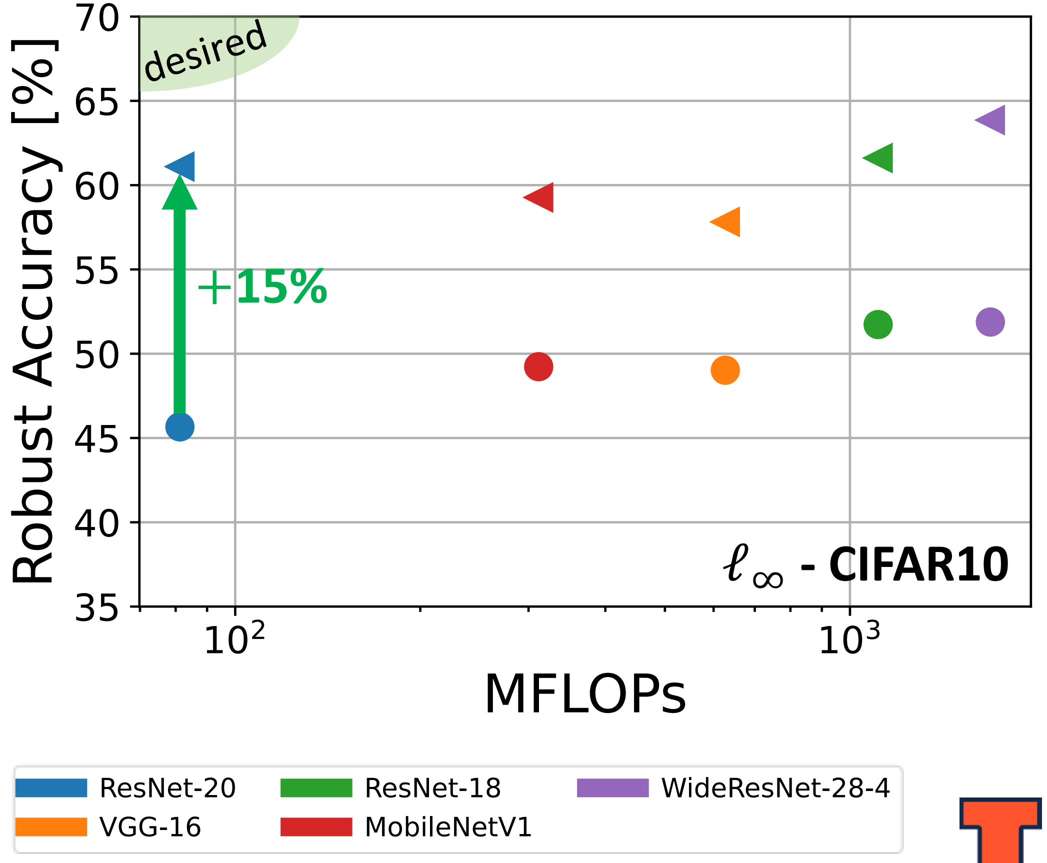
multiple classifiers $f_1, \ldots, f_M$



inference: pick **one** at random

no increase in # of FLOPS

using two classifiers trained via BAT [Pinot et al, 2020]



Robustness via Randomized Ensembles

multiple classifiers $f_1, \dots, f_M$

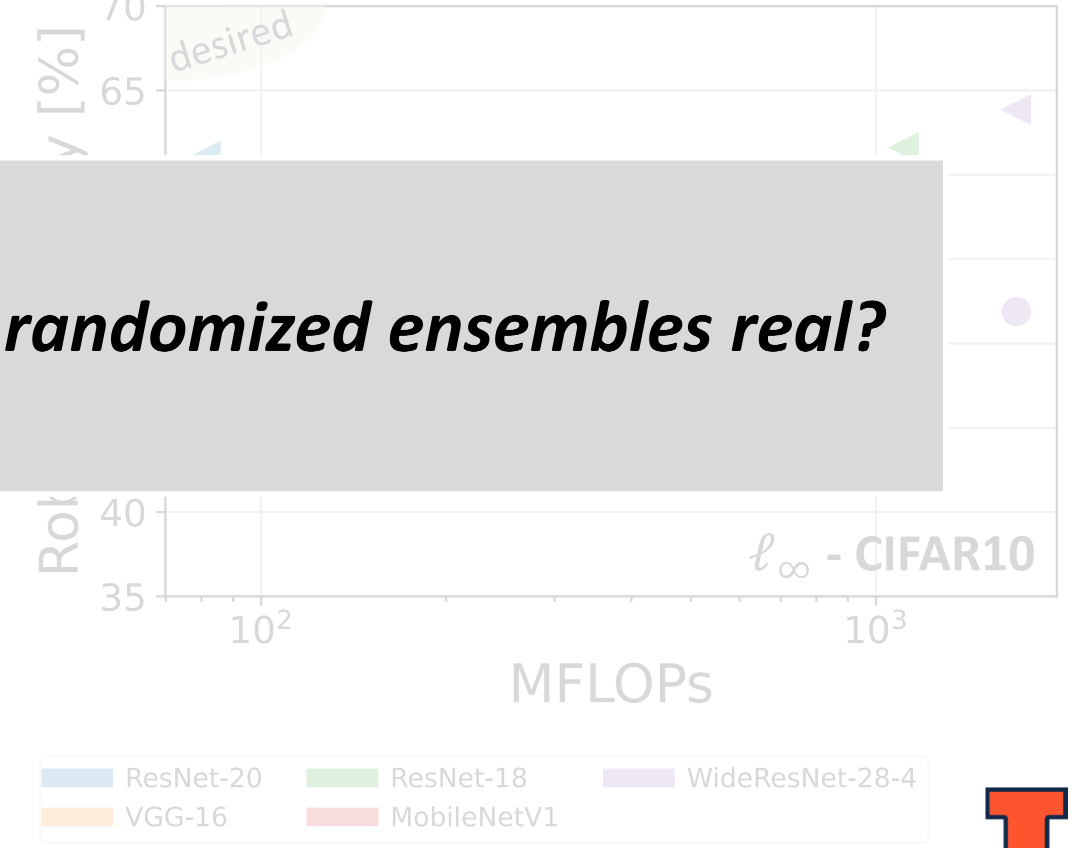
probabilities α_1 α_2 ••• α_M

clas

inference: pick **one** at random

no increase in # of FLOPS

using two classifiers trained via BAT [Pinot et al, 2020]



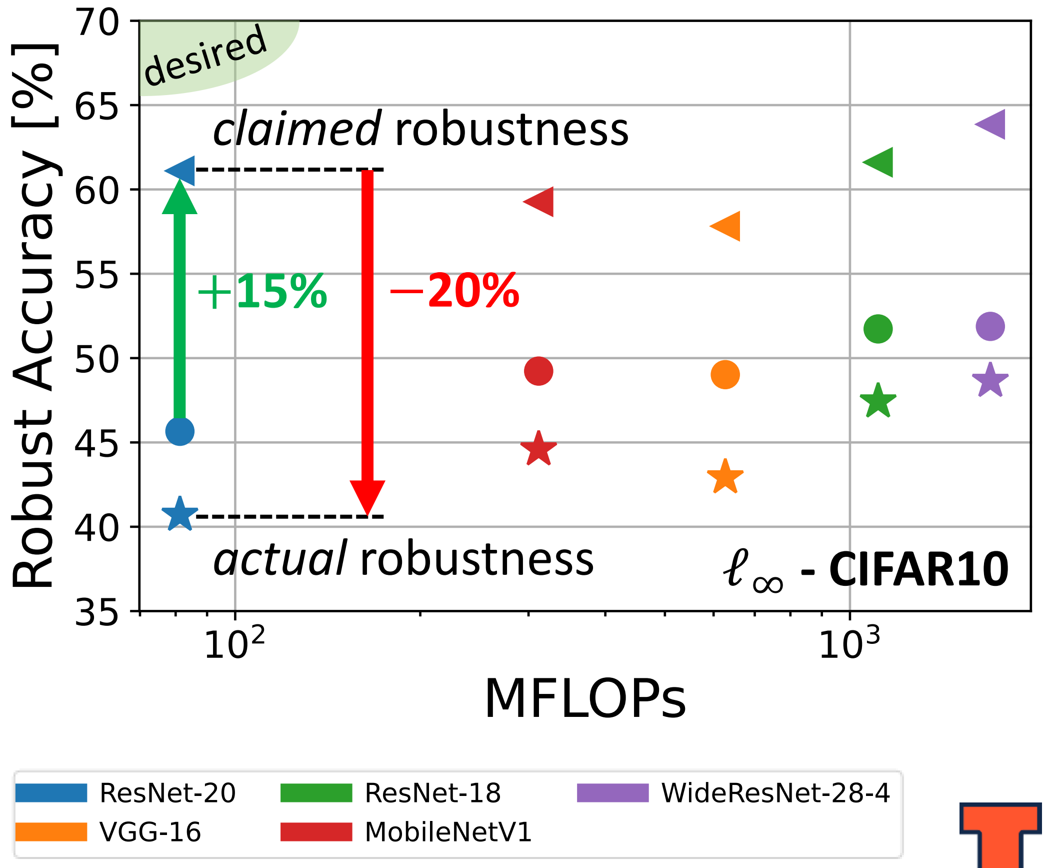
Are the robustness gains provided by randomized ensembles real?

This Work: Revealing the Vulnerability

main contributions

- show that adaptive PGD is ill-suited for evaluating robustness of RECs
  - no guarantees even for linear classifiers
- propose a provably consistent and efficient adversarial <u>attack</u> algorithm – **ARC: A**ttacking **R**andomized ensembles of **C**lassifiers
- demonstrate that existing randomized ensembles defenses are in fact more vulnerable than standard AT

BAT defense <u>compromised</u>



The ARC Algorithm – Binary Linear Classifiers

Algorithm 1 The ARC Algorithm for BLCs

Input: REC $(\mathcal{F}, \boldsymbol{\alpha})$, labeled data-point $(\mathbf{x}, y)$, norm p, and radius ϵ.
Output: Adversarial perturbation $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_p \leq \epsilon$.
Initialize $\boldsymbol{\delta} \leftarrow \mathbf{0}$, $v \leftarrow L(\mathbf{x}, y, \boldsymbol{\alpha})$, $q \leftarrow \frac{p}{p-1}$
Define $\mathcal{I}$ such that $\alpha_i \geq \alpha_j \ \forall i, j \in \mathcal{I}$ and $i \leq j$.
for $i \in \mathcal{I}$ **do**
  /* optimal unit ℓ_p norm adversarial direction for f_i
  $\mathbf{g} \leftarrow -y \frac{|\mathbf{w}_i|^{q-1} \odot \operatorname{sgn}(\mathbf{w}_i)}{\|\mathbf{w}_i\|_q^{q-1}}$
  /* shortest ℓ_p distance between $\mathbf{x}$ and f_i
  $\zeta \leftarrow \frac{|f_i(\mathbf{x})|}{\|\mathbf{w}_i\|_q}$
  if $\zeta \geq \epsilon \vee i = 1$ **then**
    $\beta \leftarrow \epsilon$
  else
    $\beta \leftarrow \frac{\epsilon}{\epsilon - \zeta} \left| \frac{y \mathbf{w}_i{}^{\top} \boldsymbol{\delta}}{\|\mathbf{w}_i\|_q} + \zeta \right| + \rho$
  end if
  $\hat{\boldsymbol{\delta}} \leftarrow \epsilon \frac{\boldsymbol{\delta} + \beta \mathbf{g}}{\|\boldsymbol{\delta} + \beta \mathbf{g}\|_p}$ ▷ candidate $\hat{\boldsymbol{\delta}}$ such that $\|\hat{\boldsymbol{\delta}}\|_p = \epsilon$
  $\hat{v} \leftarrow L(\mathbf{x} + \hat{\boldsymbol{\delta}}, y, \boldsymbol{\alpha})$
  /* if robustness does not increase, update $\boldsymbol{\delta}$
  if $\hat{v} \leq v$ **then**
    $\boldsymbol{\delta} \leftarrow \hat{\boldsymbol{\delta}}$, $v \leftarrow \hat{v}$
  end if
end for

- greedily iterate over all classifiers <u>once</u>

The ARC Algorithm – Binary Linear Classifiers

Algorithm 1 The ARC Algorithm for BLCs

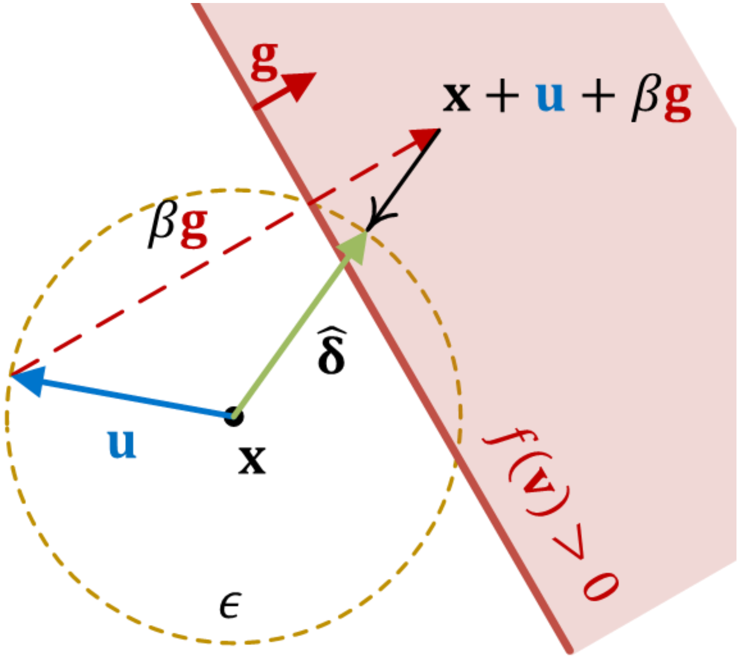
Input: REC $(\mathcal{F}, \boldsymbol{\alpha})$, labeled data-point $(\mathbf{x}, y)$, norm p, and radius ϵ.
Output: Adversarial perturbation $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_p \leq \epsilon$.
Initialize $\boldsymbol{\delta} \leftarrow \mathbf{0}$, $v \leftarrow L(\mathbf{x}, y, \boldsymbol{\alpha})$, $q \leftarrow \frac{p}{p-1}$
Define $\mathcal{I}$ such that $\alpha_i \geq \alpha_j \ \forall i, j \in \mathcal{I}$ and $i \leq j$.
for $i \in \mathcal{I}$ **do**
  /$*$ optimal unit ℓ_p norm adversarial direction for f_i
  $\mathbf{g} \leftarrow -y \frac{|\mathbf{w}_i|^{q-1} \odot \mathrm{sgn}(\mathbf{w}_i)}{\|\mathbf{w}_i\|_q^{q-1}}$
  /$*$ shortest ℓ_p distance between $\mathbf{x}$ and f_i
  $\zeta \leftarrow \frac{|f_i(\mathbf{x})|}{\|\mathbf{w}_i\|_q}$
  if $\zeta \geq \epsilon \vee i = 1$ **then**
    $\beta \leftarrow \epsilon$
  else
    $\beta \leftarrow \frac{\epsilon}{\epsilon - \zeta} \left| \frac{y \mathbf{w}_i{}^{\mathsf{T}} \boldsymbol{\delta}}{\|\mathbf{w}_i\|_q} + \zeta \right| + \rho$
  end if
  $\hat{\boldsymbol{\delta}} \leftarrow \epsilon \frac{\boldsymbol{\delta} + \beta \mathbf{g}}{\|\boldsymbol{\delta} + \beta \mathbf{g}\|_p}$ ▷ candidate $\hat{\boldsymbol{\delta}}$ such that $\|\hat{\boldsymbol{\delta}}\|_p = \epsilon$
  $\hat{v} \leftarrow L(\mathbf{x} + \hat{\boldsymbol{\delta}}, y, \boldsymbol{\alpha})$
  /$*$ if robustness does not increase, update $\boldsymbol{\delta}$
  if $\hat{v} \leq v$ **then**
    $\boldsymbol{\delta} \leftarrow \hat{\boldsymbol{\delta}}$, $v \leftarrow \hat{v}$
  end if
end for

- greedily iterate over all classifiers <u>once</u>
- novel <u>adaptive step size</u> computation:

smallest $\beta > 0$ such that

$$\hat{\boldsymbol{\delta}} = \gamma(\mathbf{u} + \beta \mathbf{g})$$

can fool f



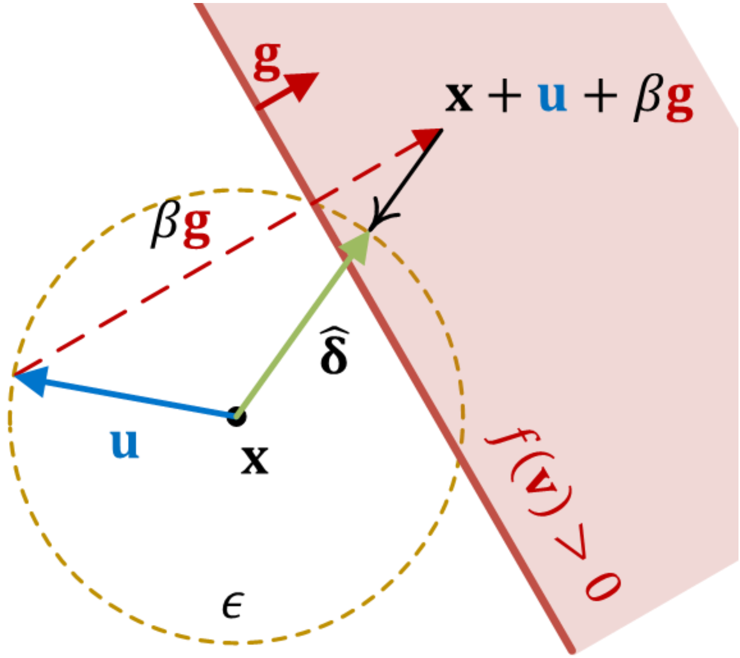
The ARC Algorithm – Binary Linear Classifiers

Algorithm 1 The ARC Algorithm for BLCs

Input: REC $(\mathcal{F}, \boldsymbol{\alpha})$, labeled data-point $(\mathbf{x}, y)$, norm p, and radius ϵ.
Output: Adversarial perturbation $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_p \leq \epsilon$.
Initialize $\boldsymbol{\delta} \leftarrow \mathbf{0}$, $v \leftarrow L(\mathbf{x}, y, \boldsymbol{\alpha})$, $q \leftarrow \frac{p}{p-1}$
Define $\mathcal{I}$ such that $\alpha_i \geq \alpha_j \; \forall i, j \in \mathcal{I}$ and $i \leq j$.
for $i \in \mathcal{I}$ **do**
  /* optimal unit ℓ_p norm adversarial direction for f_i
  $\mathbf{g} \leftarrow -y \frac{|\mathbf{w}_i|^{q-1} \odot \text{sgn}(\mathbf{w}_i)}{\|\mathbf{w}_i\|_q^{q-1}}$
  /* shortest ℓ_p distance between $\mathbf{x}$ and f_i
  $\zeta \leftarrow \frac{|f_i(\mathbf{x})|}{\|\mathbf{w}_i\|_q}$
  if $\zeta \geq \epsilon \vee i = 1$ **then**
    $\beta \leftarrow \epsilon$
  else
    $\beta \leftarrow \frac{\epsilon}{\epsilon - \zeta} \left| \frac{y \mathbf{w}_i^\top \boldsymbol{\delta}}{\|\mathbf{w}_i\|_q} + \zeta \right| + \rho$
  end if
  $\hat{\boldsymbol{\delta}} \leftarrow \epsilon \frac{\boldsymbol{\delta} + \beta \mathbf{g}}{\|\boldsymbol{\delta} + \beta \mathbf{g}\|_p}$ ▷ candidate $\hat{\boldsymbol{\delta}}$ such that $\|\hat{\boldsymbol{\delta}}\|_p = \epsilon$
  $\hat{v} \leftarrow L(\mathbf{x} + \hat{\boldsymbol{\delta}}, y, \boldsymbol{\alpha})$
  /* if robustness does not increase, update $\boldsymbol{\delta}$
  if $\hat{v} \leq v$ **then**
    $\boldsymbol{\delta} \leftarrow \hat{\boldsymbol{\delta}}$, $v \leftarrow \hat{v}$
  end if
end for

- greedily iterate over all classifiers once
- novel adaptive step size computation:

smallest $\beta > 0$ such that
$\hat{\boldsymbol{\delta}} = \gamma(\mathbf{u} + \beta \mathbf{g})$
can fool f



- extend to multiclass differentiable classifiers

The ARC Algorithm – Binary Linear Classifiers

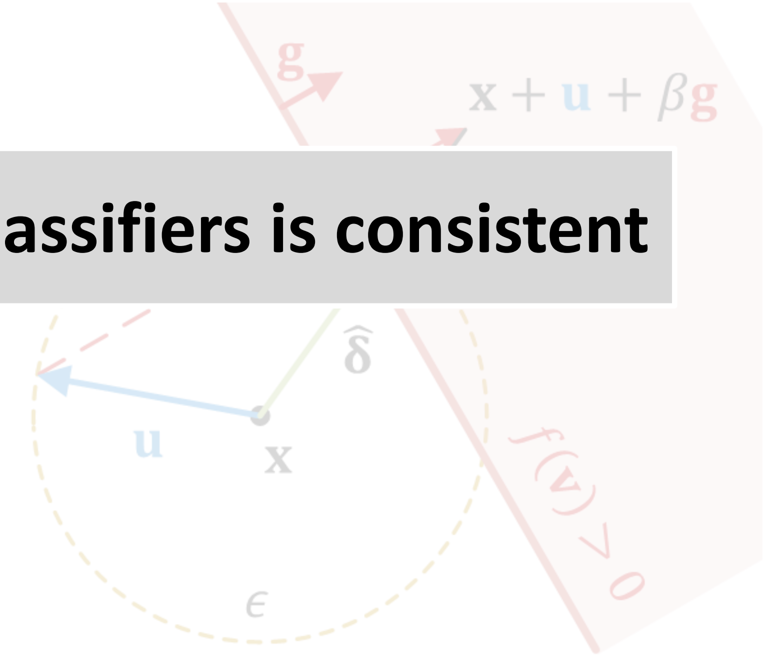
Algorithm 1 The ARC Algorithm for BLCs

Input: REC $(\mathcal{F}, \boldsymbol{\alpha})$, labeled data-point $(\mathbf{x}, y)$, norm p, and radius ϵ.
Output: Adversarial perturbation $\boldsymbol{\delta}$ such that $\|\boldsymbol{\delta}\|_p \leq \epsilon$.
Initialize $\boldsymbol{\delta} \leftarrow \mathbf{0}$, $v \leftarrow L(\mathbf{x}, y, \boldsymbol{\alpha})$, $q \leftarrow \frac{p}{p-1}$
Define $\mathcal{I}$ such that $\alpha_i \geq \alpha_j \ \forall i, j \in \mathcal{I}$ and $i \leq j$.
for $i \in \mathcal{I}$ **do**
    $\zeta \leftarrow \frac{|f_i(\mathbf{x})|}{\|\mathbf{w}_i\|_q}$
    if $\zeta \geq \epsilon \vee i = 1$ **then**
        $\beta \leftarrow \epsilon$
    else
        $\beta \leftarrow \frac{\epsilon}{\epsilon - \zeta} \left| \frac{y \mathbf{w}_i^{\top} \boldsymbol{\delta}}{\|\mathbf{w}_i\|_q} + \zeta \right| + \rho$
    end if
    $\hat{\boldsymbol{\delta}} \leftarrow \epsilon \frac{\boldsymbol{\delta} + \beta \mathbf{g}}{\|\boldsymbol{\delta} + \beta \mathbf{g}\|_p}$ ▷ candidate $\hat{\boldsymbol{\delta}}$ such that $\|\hat{\boldsymbol{\delta}}\|_p = \epsilon$
    $\hat{v} \leftarrow L(\mathbf{x} + \hat{\boldsymbol{\delta}}, y, \boldsymbol{\alpha})$
    /* if robustness does not increase, update $\boldsymbol{\delta}$
    if $\hat{v} \leq v$ **then**
        $\boldsymbol{\delta} \leftarrow \hat{\boldsymbol{\delta}}$, $v \leftarrow \hat{v}$
    end if
end for

- greedily iterate over all classifiers once
- novel adaptive step size computation:

Theorem: the ARC algorithm for binary linear classifiers is consistent

$\hat{\boldsymbol{\delta}} = \gamma(\mathbf{u} + \beta \mathbf{g})$
can fool f



- extend to multiclass differentiable classifiers

Results Summary

Table 1. Comparison between ARC and adaptive PGD when attacking randomized ensembles trained via BAT (Pinot et al., 2020) across various network architectures and norms on the CIFAR-10 dataset. We use the standard radii $\epsilon_2 = 128/255$ and $\epsilon_\infty = 8/255$ for ℓ_2 and ℓ_∞-bounded perturbations, respectively.

NETWORK	NORM	ROBUST ACCURACY [%] AT ($M = 1$)	REC ($M = 2$)		
		PGD	APGD	ARC	DIFF
RESNET-20	ℓ_2	62.43	69.21	55.44	**−13.77**
	ℓ_∞	45.66	61.10	40.71	**−20.39**
MOBILENETV1	ℓ_2	66.39	67.92	59.43	**−8.49**
	ℓ_∞	49.23	59.27	44.59	**−14.68**
VGG-16	ℓ_2	66.08	66.96	59.20	**−7.76**
	ℓ_∞	49.02	57.82	42.93	**−14.89**
RESNET-18	ℓ_2	69.16	70.16	65.88	**−4.28**
	ℓ_∞	51.73	61.61	47.43	**−14.18**
WIDERESNET-28-4	ℓ_2	69.91	71.48	62.95	**−8.53**
	ℓ_∞	51.88	63.86	48.65	**−15.21**

Table 2. Comparison between ARC and adaptive PGD when attacking randomized ensembles trained via BAT (Pinot et al., 2020) across various datasets and norms. We use ResNet-18 for ImageNet and and ResNet-20 for SVHN, CIFAR-10, and CIFAR-100 datasets.

DATASET	NORM	RADIUS (ϵ)	ROBUST ACCURACY [%] AT ($M = 1$)	REC ($M = 2$)		
			PGD	APGD	ARC	DIFF
SVHN	ℓ_2	128/255	68.35	74.66	60.15	**−14.51**
	ℓ_∞	8/255	53.55	65.99	52.01	**−13.98**
CIFAR-10	ℓ_2	128/255	62.43	69.21	55.44	**−13.77**
	ℓ_∞	8/255	45.66	61.10	40.71	**−20.39**
CIFAR-100	ℓ_2	128/255	34.60	41.91	28.92	**−12.99**
	ℓ_∞	8/255	22.29	33.37	17.45	**−15.92**
IMAGENET	ℓ_2	128/255	47.61	49.62	42.09	**−7.53**
	ℓ_∞	4/255	24.33	35.92	19.54	**−16.38**

BAT defense compromised

- ARC outperforms APGD across various datasets, norms, and network topologies

Next Steps

- develop a <u>complete</u> theoretical framework for better understanding randomized ensembles of classifiers

- how can we design <u>truly</u> robust randomized ensembles in practice?

Thank You!

code available at https://github.com/hsndbk4/ARC

Acknowledgement:
This work was supported by the Center for Brain-Inspired Computing (C-BRIC) and Artificial Intelligence Hardware (AIHW) funded by the Semiconductor Research Corporation (SRC) and the Defense Advanced Research Projects Agency (DARPA).