Guest Lecture:

Mixing Neural Network Classifiers to Balance Accuracy and Adversarial Robustness

Presenter: Yatong Bai yatong_bai@berkeley.edu

May 19, 2024

Berkeley
UNIVERSITY OF CALIFORNIA

About Myself

- Rising 5th-year Ph.D. candidate at UC Berkeley advised by Professor Somayeh Sojoudi.
- Research focus:
  - Reconciling adversarial robustness and accuracy of classification models.
  - Efficient audio generation through consistency models.
- Teaching:
  - Convex optimization and approximation.

Berkeley UNIVERSITY OF CALIFORNIA

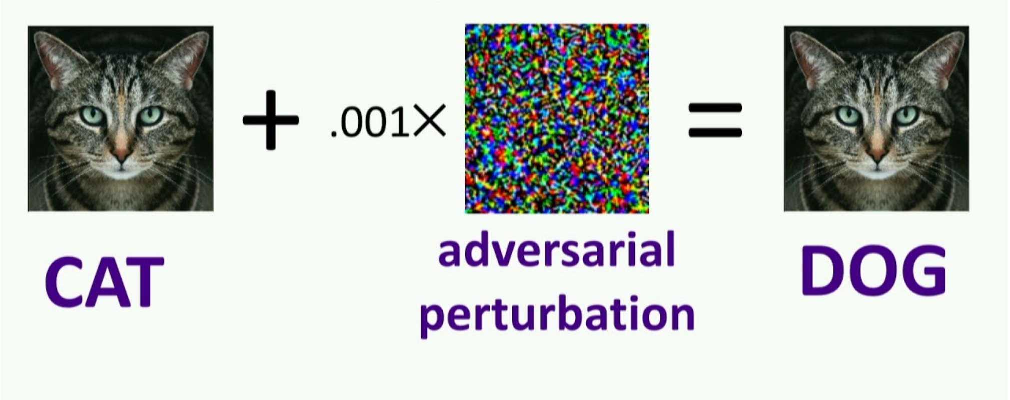
Overview of This Presentation

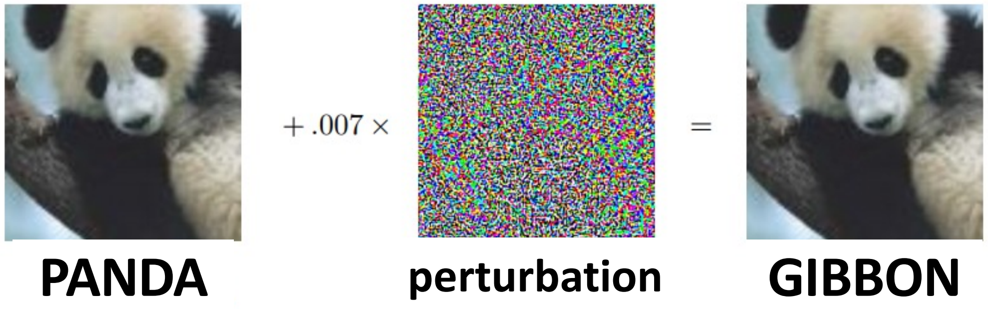
- Brief intro to adversarial robustness.
- Improving the accuracy-robustness trade-off.
  - Mixing classifiers to balance robustness and accuracy.
  - Adaptive Smoothing: adaptive mixing ratio.
    https://arxiv.org/abs/2301.12554
  - MixedNUTS: mix in a nonlinear fashion.
    https://arxiv.org/abs/2402.02263

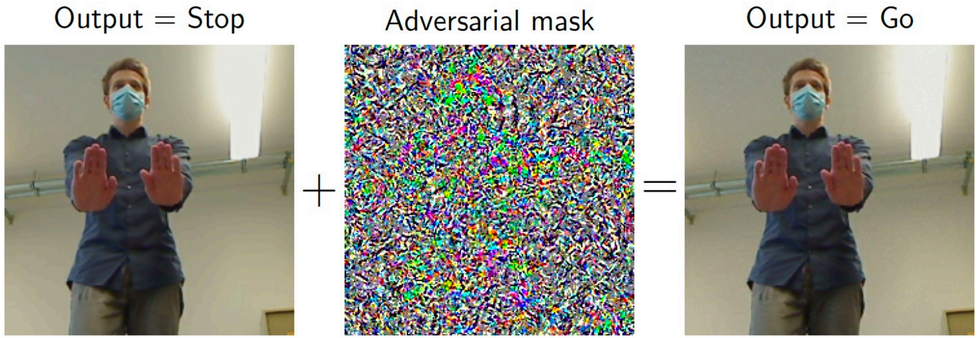
Berkeley
UNIVERSITY OF CALIFORNIA

Adversarial Robustness

- Neural networks are vulnerable
  - Small input perturbations elicit unexpected outputs.
- For classifiers: misclassifications.
- For control systems: dangerous actions

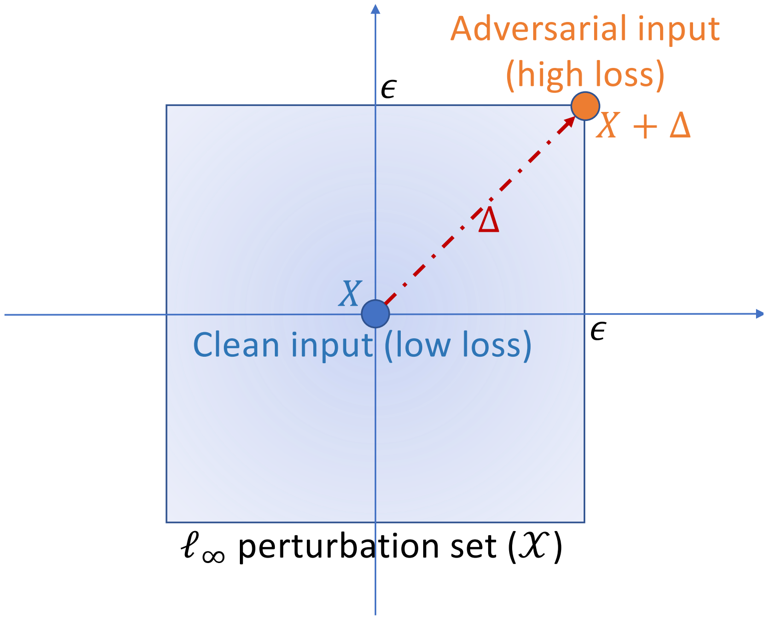






Berkeley
UNIVERSITY OF CALIFORNIA

Adversarial example generation **(An optimization formulation)**



- We need a budget for the attack, since the adversarial perturbations should be inperceptible by human.

  -- A common uncertainty set is an ℓ_∞-norm-bounded additive set with radius ϵ:

  -- I. e., a cube around each clean input.

- The adversarial examples are usually generated via the following optimization problem:

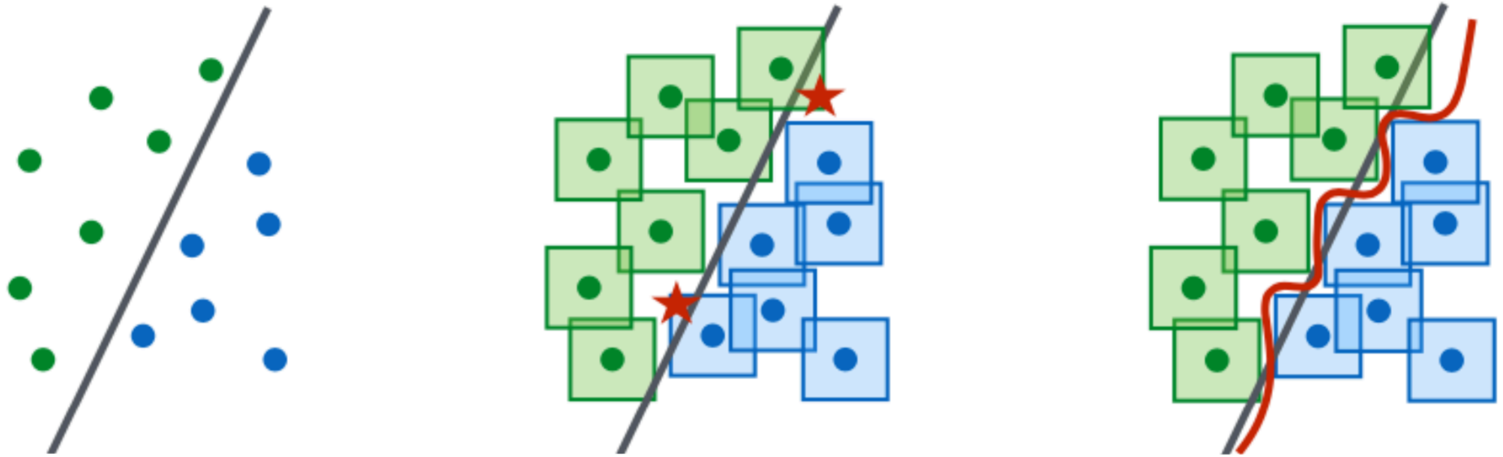
$$\max_{\delta: x+\delta \in \mathcal{X}} \underset{\substack{\uparrow \\ \text{Loss fn}}}{\ell} \Big(\underbrace{g(x+\delta)}_{\text{NN output for attacked input}}, \underbrace{Y}_{\text{Target output}} \Big),$$ where g represents the NN as a function.

Berkeley
UNIVERSITY OF CALIFORNIA

Defending attacks -- Adversarial training **(Robust Optimization)**

- One defense method: Adversarial training (train with adversarial data) [Madry et al., 2018, Goodfellow et al., 2015] .
  -- Train robust models via robust optimization. For an uncertainty set $\mathcal{X}$, solve the optimization problem

$$\underbrace{\min_{\theta}}_{\text{Optimize NN weights}} \left(\underbrace{\max_{\delta: x+\delta \in \mathcal{X}}}_{\text{Generate attack}} \ell(g_\theta(x+\delta), Y) \right) + \underset{\substack{\uparrow \\ \text{Regularization}}}{r_\theta} \tag{1}$$



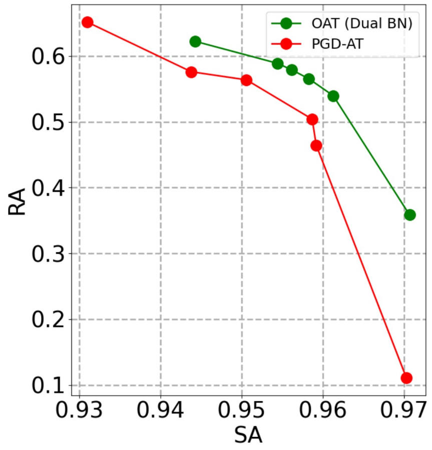
Decision boundaries of standard training and adversarial training [Madry et al., 2018].

Alternative methods:

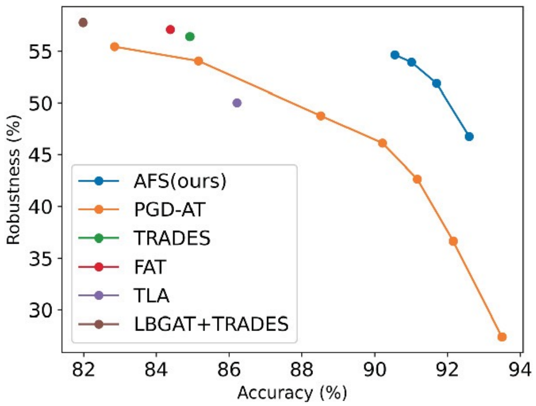
- TRADES, Randomized Smoothing.

Accuracy-Robustness Trade-Off

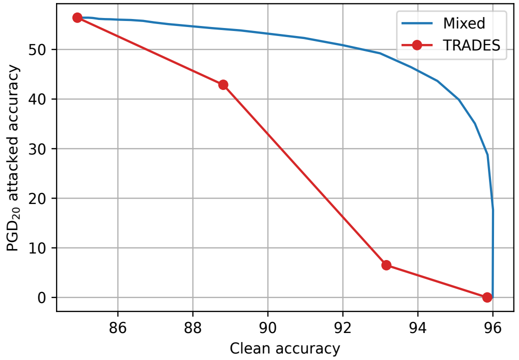
- Robust models often sacrifice clean accuracy.
- Theoretically, robust generalization needs much more training data.
- Existing methods for alleviating the trade-off:
  - Additional real/synthetic training data;
  - Attack purification;
  - Alternative training loss functions.



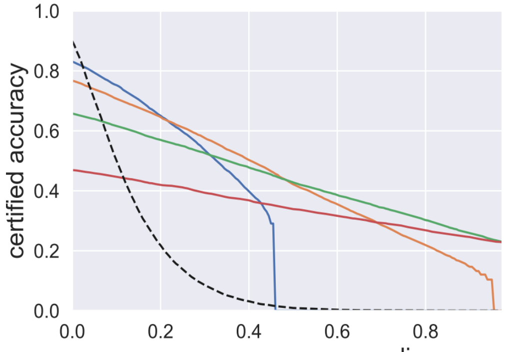
Once-for-All Adversarial Training: In-Situ Tradeoff between Robustness and Accuracy for Free



Towards Both Accurate and Robust Neural Networks Without Extra Data



Improving the Accuracy-Robustness Trade-Off of Classifiers via Adaptive Smoothing



Certified Adversarial Robustness via Randomized Smoothing

Berkeley
UNIVERSITY OF CALIFORNIA

Mixing Classifiers for Better Trade-Off

- What if we combine the wisdom of an **accurate model** and a **robust model**?
- Specifically, we "mix" their outputs, resulting in a **mixed classifier**.

$$f_i(x) := (1 - \alpha) \cdot g_i(x) + \alpha \cdot h_i(x)$$

Mixed Classifier

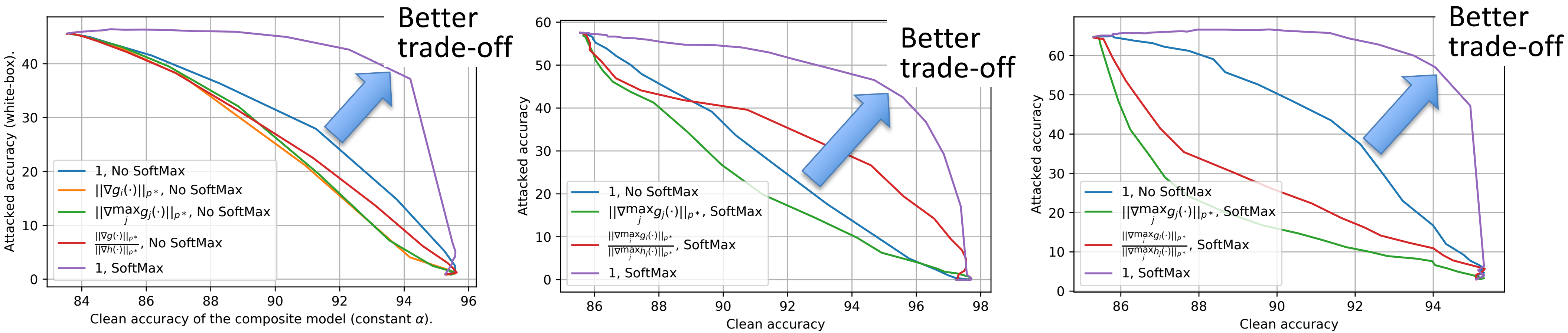
Accurate Base Classifier (ABC)

Robust Base Classifier (RBC)

- Should we mix the logits or probabilities?
  - Classifiers often use a "Softmax" operation to convert "logits" $(-\infty, +\infty)$ to prediction probabilities $(0, 1)$.

Empirically comparing the design choices

- We compare the cases with various values of α via the clean accuracy versus attacked accuracy plot:



ResNet18+AT, ℓ_∞ ConvNeXT+TRADE, ℓ_∞ ResNet18+AT, ℓ_2

Figure 1: Adaptive PGD_{10} accuracy versus clean accuracy for the three different choices of $R(x)$ on CIFAR-10.

- Blue: smoothing with logits. Purple: smoothing with probabilities.
- Conclusion: smoothing should be done on probabilities.

Mixing Probabilities is Better

- Conclusion: we should mix the base classifiers' **prediction probabilities.**
- The resulting class-wise mixing formulation is:

Convert back to logits

Softmax

$$f_i(x) := \log\left((1-\alpha)\cdot\sigma\circ g(x)_i + \alpha\cdot\sigma\circ h(x)_i\right)$$

Mixed Classifier

Accurate Base Classifier (ABC)

Robust Base Classifier (RBC)

Intuition for mixing the probabilities

- The robust classifier $h(\cdot)$ is typically smooth or Lipschitz, and we want $g^{\alpha}_{\text{CNN}}(\cdot)$ to inherent these properties.

- The accurate classifier $g(\cdot)$ is in general non-smooth and non-robust.

- If $g(\cdot) \in [0,1]$ (probabilities), then the ''level of incorrectness'' can be bounded. It is then possible for the smoothness of $h(\cdot)$ to overshadow the turbulence of $g(\cdot)$, ultimately making $g^{\alpha}_{\text{CNN}}(\cdot)$ robust.
  -- Will present a Lemma to formalize this.

- If $g(\cdot) \in \mathbb{R}$ (logits), then it can be arbitrarily unsmooth. $h(\cdot)$ may not be possible to correct $g(\cdot)$.

Certifiably robust with a margin **(Theoretically guaranteed robustness)**

To facilitate the proof for certified robust radii, we first introduce the notion ''robust with a margin''.

Definition

Consider an arbitrary input $x \in \mathbb{R}^d$ and let $y = \arg\max_i h_i(x)$, $\mu \in [0, 1]$, and $r \geq 0$.
Then, $h(\cdot)$ is said to be certifiably robust at x with margin μ and radius r
if $h_y(x + \delta) \geq h_i(x + \delta) + \mu$ for all $i \neq y$ and all $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$.

Lemma

Let $x \in \mathbb{R}^d$ and $r \geq 0$.
If it holds that $\alpha \in [\frac{1}{2}, 1]$ and $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius r,
then the smoothed classifier $g^{\alpha}_{\text{CNN}}(\cdot)$ is robust in the sense that $\arg\max_i g^{\alpha}_{\text{CNN},i}(x + \delta) = \arg\max_i h_i(x)$ for all $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$.

- Intuition: if $h(\cdot)$ is robust and confident, then it can override whatever $g(\cdot)$ predicts.

Certifiably robust with a margin -- Proof

Lemma

(Restated.) If it holds that $\alpha \in [\frac{1}{2}, 1]$ and $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius r, then $\arg\max_i g^{\alpha}_{\mathrm{CNN},i}(x+\delta) = \arg\max_i h_i(x)$ for all $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$.

Proof

Since $\alpha \in [\frac{1}{2}, 1]$, it holds that $\frac{1-\alpha}{\alpha} \in [0, 1]$.
Suppose that $h(\cdot)$ is certifiably robust at x with margin $\frac{1-\alpha}{\alpha}$ and radius r.
Let $y = \arg\max_i h_i(x)$. Consider an arbitrary $i \in [c] \setminus \{y\}$ and $\delta \in \mathbb{R}^d$ such that $\|\delta\|_p \leq r$. It holds that

$$\begin{aligned}
\exp\left(g^{\alpha}_{\mathrm{CNN},y}(x+\delta)\right) - \exp\left(g^{\alpha}_{\mathrm{CNN},i}(x+\delta)\right) &= (1-\alpha)(g_y(x+\delta) - g_i(x+\delta)) + \alpha(h_y(x+\delta) - h_i(x+\delta)) \\
\text{(Because } g_i(x+\delta) \in [0,1]\text{)} \qquad &\geq (1-\alpha)(0-1) + \alpha(h_y(x+\delta) - h_i(x+\delta)) \\
&\geq (\alpha - 1) + \alpha\left(\frac{1-\alpha}{\alpha}\right) = 0.
\end{aligned}$$

Thus, it holds that $g^{\alpha}_{\mathrm{CNN},y}(x+\delta) \geq g^{\alpha}_{\mathrm{CNN},i}(x+\delta)$ for all $i \neq y$, and thus
$\arg\max_i g^{\alpha}_{\mathrm{CNN},i}(x+\delta) = y = \arg\max_i h_i(x)$. $\square$

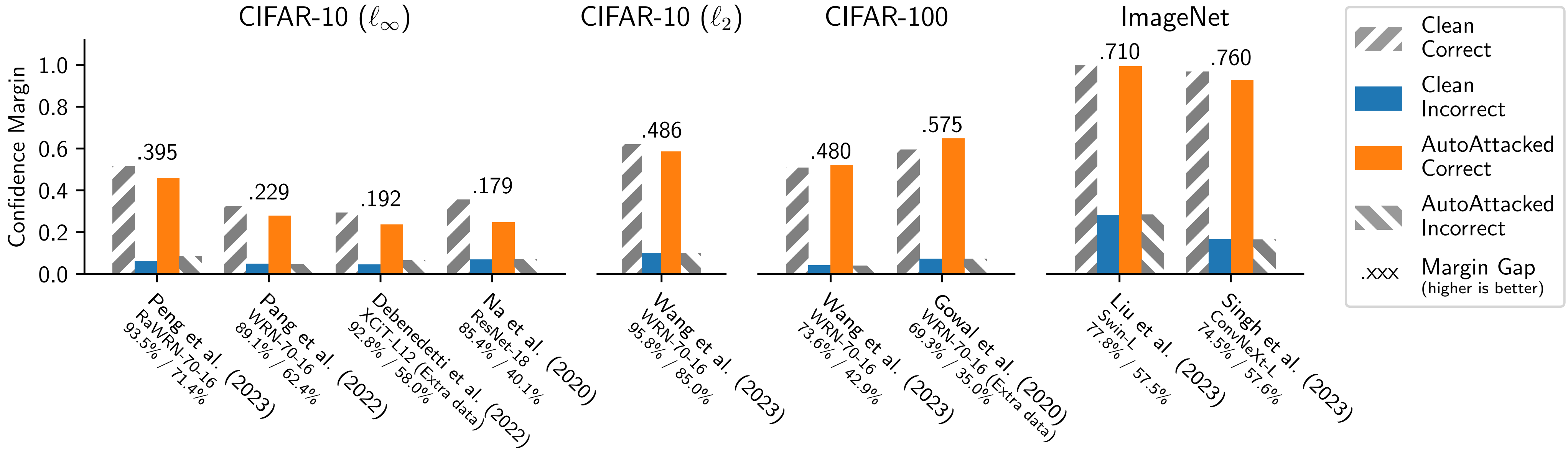
Berkeley
UNIVERSITY OF CALIFORNIA

Mechanism for Improved Accuracy Trade-Off

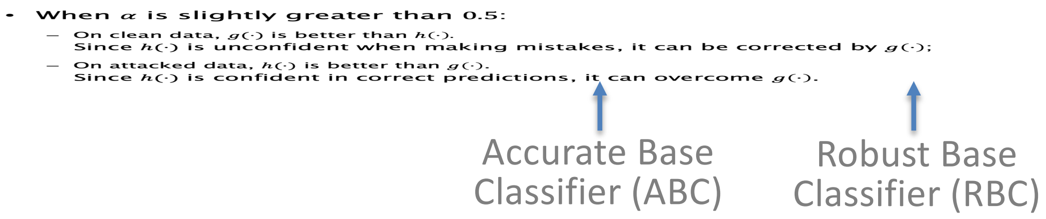
- Empirically robust models are more confident when correct than when incorrect, even on attacked data.

Definition 1. Consider a model $h : \mathbb{R}^d \to \mathbb{R}^c$, an arbitrary input $x \in \mathbb{R}^d$, and its associated predicted label $\widehat{y} \in [c]$. The *confidence margin* is defined as $m_h(x) := \sigma \circ h_{\widehat{y}}(x) - \max_{i \neq \widehat{y}} \sigma \circ h_i(x)$.

- Some examples (SOTA models on various datasets):



Mechanism for Improved Accuracy Trade-Off



- When α is slightly greater than 0.5:
  - On clean data, $g(\cdot)$ is better than $h(\cdot)$.
    Since $h(\cdot)$ is unconfident when making mistakes, it can be corrected by $g(\cdot)$;
  - On attacked data, $h(\cdot)$ is better than $g(\cdot)$.
    Since $h(\cdot)$ is confident in correct predictions, it can overcome $g(\cdot)$.

Adaptive Smoothing: Flexible Mixing Ratio

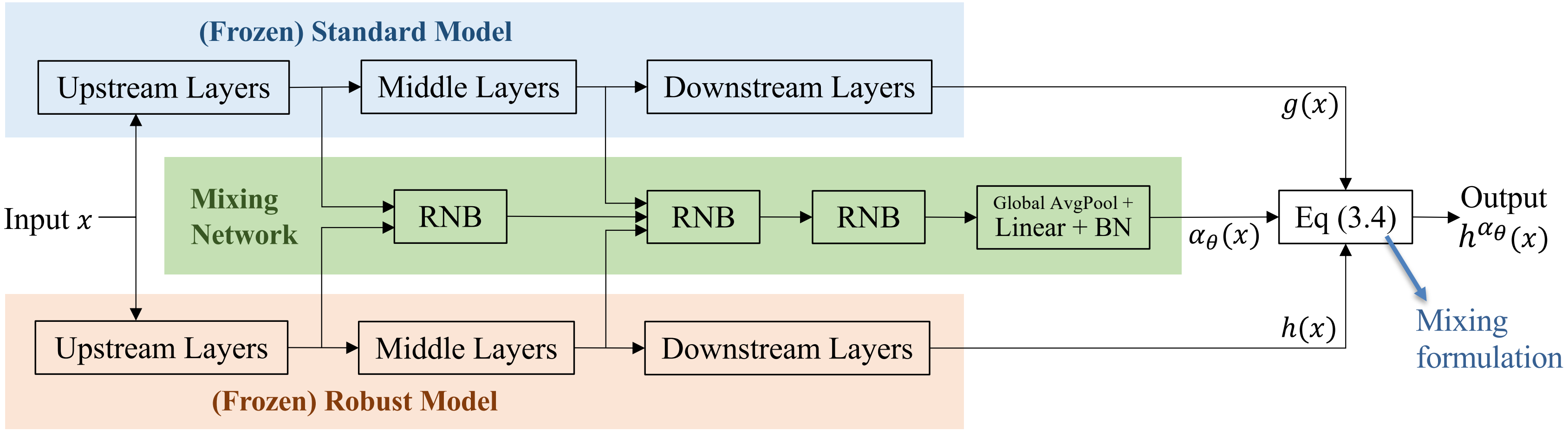
- Recall the mixed classifier formulation:

$$\underbrace{f_i(x)}_{\text{Mixed Classifier}} := \overbrace{\log}^{\text{Convert back to logits}} \left((1-\alpha) \cdot \overbrace{\sigma}^{\text{Softmax}} \circ \underbrace{g(x)_i}_{\text{Accurate Base Classifier (ABC)}} + \alpha \cdot \overbrace{\sigma}^{\text{Softmax}} \circ \underbrace{h(x)_i}_{\text{Robust Base Classifier (RBC)}} \right)$$

- It makes sense to **make the mixing ratio α a function of x.**

Adaptive Smoothing: Flexible Mixing Ratio

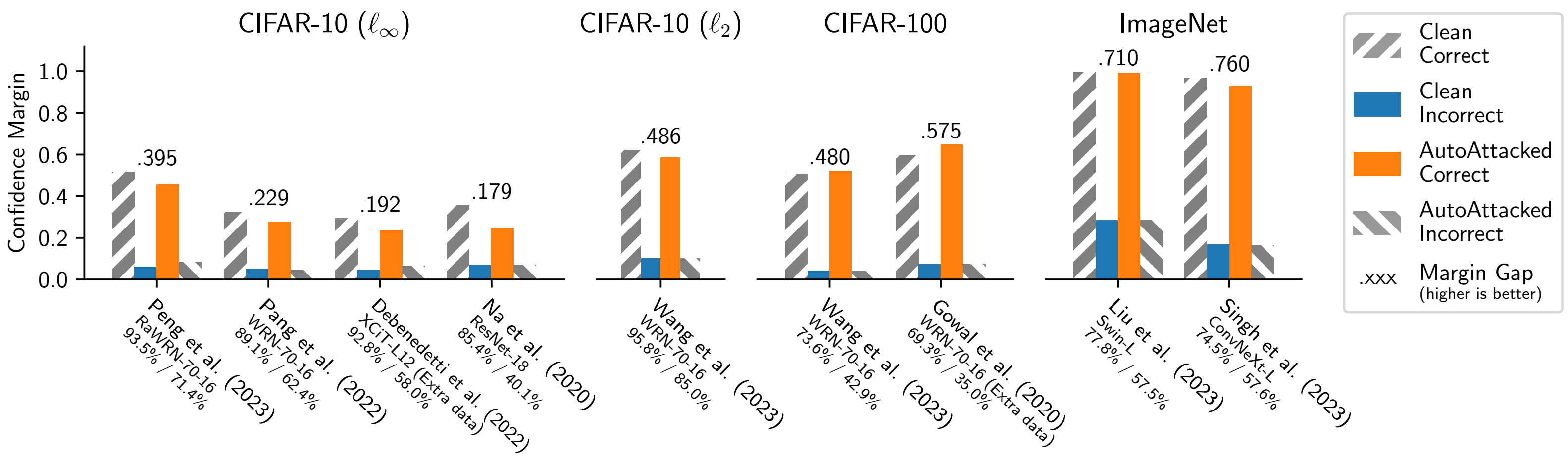
- It makes sense to make the mixing ratio α a function of x.
  - Make $\alpha(x)$ **small** and prefer the **ABC** $\boldsymbol{g}(\boldsymbol{x})$ when x is **natural** (no attack).
  - Make $\alpha(x)$ **large** and prefer the **RBC** $\boldsymbol{h}(\boldsymbol{x})$ when x is **adversarial**.
- Parameterizing $\alpha(x)$: an additional neural network module.



Berkeley
UNIVERSITY OF CALIFORNIA

MixedNUTS: Nonlinear Mixed Classifier

- **Recall:** Mixed classifiers rely on the RBC $h(\cdot)$'s benign confidence properties.
  - More confident in correct examples than incorrect ones.



- Confidence can be adjusted without changing predictions.
  - (e.g., temperature scaling).
- **Can we augment the benign properties to improve the mixed classifier?**

Berkeley
UNIVERSITY OF CALIFORNIA

MixedNUTS: Nonlinear Mixed Classifier

- **How to augment the benign properties?**
- Apply a non-linear transformation $M(\cdot)$ to RBC $h(\cdot)$'s logits before Softmax and mixing.
  - Notation: $h^M(x) = M\big(h(x)\big)$.
  - Temperature scaling is a special case where $M(\cdot)$ is linear.
- Apply temperature scaling to ABC $g(\cdot)$'s logits before Softmax and mixing.
  - Ablation study shows that zero temperature (one-hot probabilities) works the best.

Berkeley
UNIVERSITY OF CALIFORNIA

MixedNUTS: Nonlinear Mixed Classifier

- Goal: optimize $M(\cdot)$'s clean accuracy for a given robust accuracy r_f.

$$\max_{M \in \mathcal{M},\ \alpha \in [1/2, 1]} \mathbb{P}_{(X,Y)\sim\mathcal{D}} \left[\arg\max_i f_i^M(X) = Y \right] \quad (2)$$
$$\text{s.t. } \mathbb{P}_{(X,Y)\sim\mathcal{D}} \left[\arg\max_i f_i^M(X + \delta^\star_{f^M}(X)) = Y \right] \geq r_{f^M},$$

Maximize mixed classifier clean accuracy while maintaining robust accuracy

- Consider the approximate problem

$$\begin{aligned} \min_{M \in \mathcal{M},\ \alpha \in [1/2, 1]} \quad & \mathbb{P}_{X \sim \mathcal{X}_{ic}} \left[m_{h^M}(X) \geq \tfrac{1-\alpha}{\alpha} \right] \\ \text{s.t.} \quad & \mathbb{P}_{Z \sim \mathcal{X}_{ca}} \left[\underline{m}^\star_{h^M}(Z) \geq \tfrac{1-\alpha}{\alpha} \right] \geq \beta, \end{aligned} \quad (3)$$

Minimize $h^M(\cdot)$'s confidence margin at mispredicted clean data
while maintaining $h^M(\cdot)$'s margin at correctly predicted worst-case adversarial data

where $\mathcal{X}_{ic}$ is the distribution formed by clean examples incorrectly classified by $h^M(\cdot)$, $\mathcal{X}_{ca}$ is the distribution formed by attacked examples correctly classified by $h^M(\cdot)$, X, Z are the random variables drawn from these distributions, and $\beta \in [0, 1]$ controls the desired level of robust accuracy with respect to the robust accuracy of $h(\cdot)$.

  – The approximate problem decouples the optimization from $g(\cdot)$.

Berkeley
UNIVERSITY OF CALIFORNIA

Quality of Approximation

- Original goal:

$$\max_{M\in\mathcal{M},\ \alpha\in[1/2,1]} \mathbb{P}_{(X,Y)\sim\mathcal{D}}\big[\arg\max_i f_i^M(X) = Y\big] \qquad (2)$$
$$\text{s.t. } \mathbb{P}_{(X,Y)\sim\mathcal{D}}\big[\arg\max_i f_i^M(X+\delta^\star_{f^M}(X)) = Y\big] \geq r_{f^M},$$

- Approximate problem:

$$\min_{M\in\mathcal{M},\ \alpha\in[1/2,1]} \mathbb{P}_{X\sim\mathcal{X}_{ic}}\left[m_{h^M}(X) \geq \tfrac{1-\alpha}{\alpha}\right] \qquad (3)$$
$$\text{s.t. } \mathbb{P}_{Z\sim\mathcal{X}_{ca}}\left[\underline{m}^\star_{h^M}(Z) \geq \tfrac{1-\alpha}{\alpha}\right] \geq \beta,$$

- The objectives are equivalent, (3)'s constraint is more conservative

Assumption 4.1. On unattacked clean data, if $h^M(\cdot)$ makes a correct prediction, then $g(\cdot)$ is also correct.

Assumption 4.2. The transformation $M(\cdot)$ does not change the predicted class due to, *e.g.*, monotonicity. Namely, it holds that $\arg\max_i M(h(x))_i = \arg\max_i h_i(x)$ for all x.

Theorem 4.3. *Suppose that Assumption* 4.2 *holds. Let r_h denote the robust accuracy of $h(\cdot)$. If $\beta \geq r_{f^M}/r_h$, then a solution to* (3) *is feasible for* (2).

Theorem 4.4. *Suppose that Assumption* 4.1 *holds. Furthermore, consider an input random variable X and suppose that the margin of $h^M(X)$ is independent of whether $g(X)$ is correct. Then, minimizing the objective of* (3) *is equivalent to maximizing the objective of* (2).

Berkeley
UNIVERSITY OF CALIFORNIA

Nonlinear Transformation Parameterization

- **Step 1: Layer Norm (LN)**
  - Nonlinear transformations' effect depends on the logits range.
  - LN unifies the range.
  - For each image x, we standardize the logits $h(x)$ to have zero mean and variance one.
- **Step 2: Clamp**
  - We use a ReLU-like function to clamp the logits smaller than a positive threshold toward zero.
    - Introduce the threshold parameter c.
  - Since correct predictions have greater margins, clamping enlarges the margin difference between correct and incorrect examples.
  - We select GELU based on ablation studies.

So far, $h^M(x) = \mathrm{GELU}(\mathrm{LN}(h(x)) + c)$

Berkeley
UNIVERSITY OF CALIFORNIA

Nonlinear Transformation Parameterization

- **Step 3: Exponentiation**
  - Amplify large logits (common in correct predictions) to further enlarge the margin difference.
  - Use absolute value to preserve logit sign.
  - Introduce the exponent parameter p.
- **Step 4: Temperature Scaling**
  - Softmax “saturates” with large logits.
  - Temperature scaling allows for adjusting the level of saturation.
  - Introduce the scale parameter s.

Final formulation:

$$h^{\mathrm{Clamp},c}(x) = \mathrm{Clamp}\big(\mathrm{LN}(h(x)) + c\big)$$
$$h^{M\substack{s\\p\\c}}(x) = s \cdot \left|h^{\mathrm{Clamp},c}(x)\right|^{p} \cdot \mathrm{sgn}\left(h^{\mathrm{Clamp},c}(x)\right)$$

Optimizing s, p, c, α

- The resulting problem is then

$$\min_{s,p,c,\alpha\in\mathbb{R}} \mathbb{P}_{X\sim\mathcal{X}_{ic}}\left[m_{h^{\mathrm{map},s,p,c}}(X) \geq \tfrac{1-\alpha}{\alpha}\right]$$
$$\text{s.t.} \quad \mathbb{P}_{Z\sim\mathcal{X}_{ca}}\left[\underline{m}^{\star}_{h^{\mathrm{map},s,p,c}}(Z) \geq \tfrac{1-\alpha}{\alpha}\right] \geq \beta$$
$$s \geq 0, \quad p \geq 0, \quad 1/2 \leq \alpha \leq 1.$$

  - $\beta = 0.985$ works well in practice.

- Only three degrees of freedom.
  - Because the robust accuracy constraint is always active.
- Algorithm: grid search over s, p, c and calculate α via the constraint.
- Approximation for efficiency:
  - Use $h(\cdot)$ as a surrogate for $h^M(\cdot)$ in margin calculations, so that grid search doesn't need to include attack.

Berkeley
UNIVERSITY OF CALIFORNIA

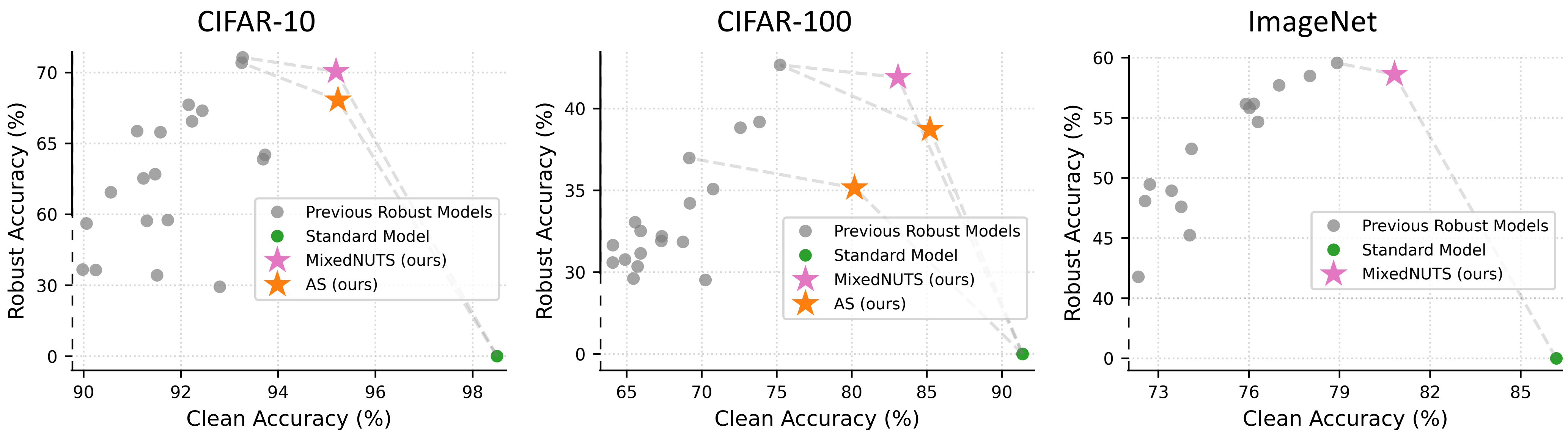
Optimizing s, p, c, α

Algorithm 1 Algorithm for optimizing s, p, c, and α.

Given an image set, save the predicted logits associated with mispredicted clean images $\{h^{\mathrm{LN}}(x) : x \in \tilde{\mathcal{X}}_{ic}\}$.
Run MMAA on $h^{\mathrm{LN}}(\cdot)$ and save the logits of correctly classified perturbed inputs $\{h^{\mathrm{LN}}(x) : x \in \tilde{\mathcal{A}}_{ca}\}$.
Initialize candidate values $s_1, \ldots, s_l, p_1, \ldots, p_m, c_1, \ldots, c_n$.
for s_i for $i = 1, \ldots, l$ **do**
    for p_j for $j = 1, \ldots, m$ **do**
        for c_k for $k = 1, \ldots, n$ **do**
            Obtain mapped logits $\{h^{M^{s_i}_{p_j, c_k}}(x) : x \in \tilde{\mathcal{A}}_{ca}\}$.
            Calculate the margins from the mapped logits $\{m_{h^{M^{s_i}_{p_j, c_k}}}(x) : x \in \tilde{\mathcal{A}}_{ca}\}$.
            Store the bottom $1 - \beta$-quantile of the margins as $q^{ijk}_{1-\beta}$ (corresponds to $\frac{1-\alpha}{\alpha}$ in (6)).
            Record the current objective $o^{ijk} \leftarrow \mathbb{P}_{X \in \tilde{\mathcal{X}}_{ic}}\left[m_{h^{M^{s_i}_{p_j, c_k}}}(X) \geq q^{ijk}_{1-\beta}\right]$.
        end for
    end for
end for
Find optimal indices $(i^\star, j^\star, k^\star) = \arg\min_{i,j,k} o^{ijk}$.
Recover optimal mixing weight $\alpha^\star := {}^1\!/\!\left(1+q^{i^\star j^\star k^\star}_{1-\beta}\right)$.
return $s^\star := s_{i^\star}, p^\star := p_{j^\star}, c^\star := c_{k^\star}, \alpha^\star$.

Main Experiment Result

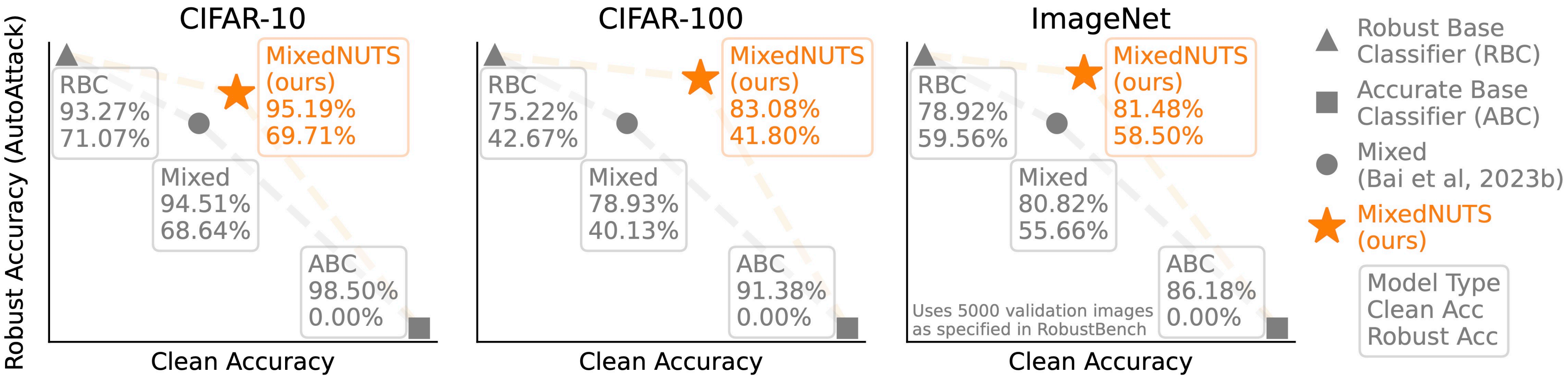
- Mixed classifiers achieve state-of-the-art accuracy-robustness trade-off.



Berkeley UNIVERSITY OF CALIFORNIA

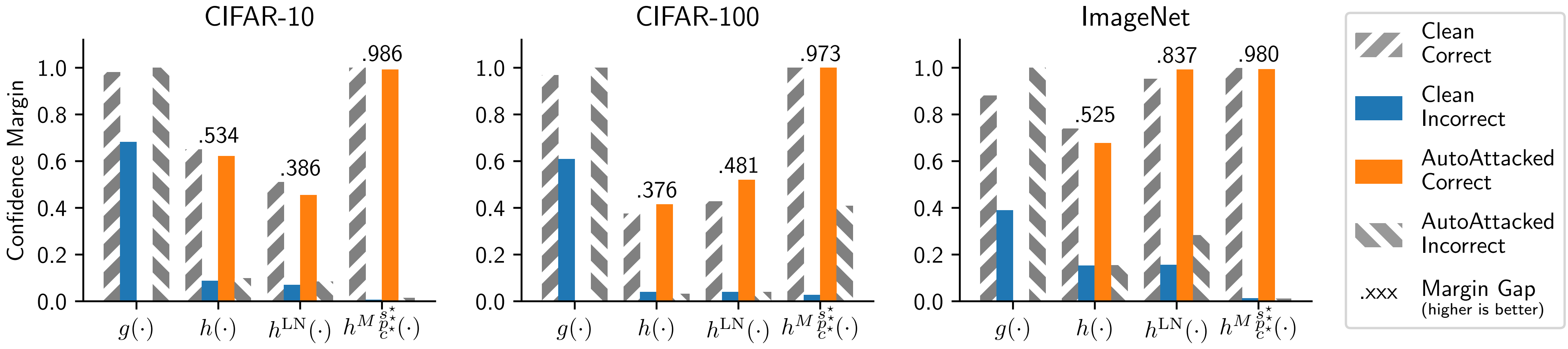
Main Experiment Result

- MixedNUTS' nonlinear logit transformations improve the accuracy–robustness trade-off.



Augmented Benign Margin Property

- MixedNUTS' nonlinear logit transformation augments the RBC's benign confidence margin properties.



Berkeley
UNIVERSITY OF CALIFORNIA

Future – Beyond Adversarial Robustness

- Beyond adversarial robustness:
  - Generalized case: Model A specializes in Distribution A; Model B specializes in Distribution B; Distributions A, B share the same classes.
- Beyond classification:
  - Language models: output the probabilities of candidate next word tokens.
    - Existing models use mixtures of experts (MoE) to save computation (not all weights are activated).

Berkeley
UNIVERSITY OF CALIFORNIA

Thank you!

Adaptive Smoothing: https://arxiv.org/abs/2301.12554
MixedNUTS: https://arxiv.org/abs/2402.02263

Presenter: Yatong Bai yatong_bai@berkeley.edu

May 19, 2024

Berkeley
UNIVERSITY OF CALIFORNIA