Confidence-Calibrated Adversarial Training
Lessons for Evaluating Defenses
David Stutz
with Bernt Schiele, Matthias Hein
Part 0
Background
Adversarial Examples

Images $x$ → Perturbations $δ$ → Images $\tilde{x} = x + δ$

Classifier

$\text{argmax}_δ \mathcal{L}(f(x + δ; w), y)$

s.t. $\|δ\|_p ≤ \epsilon$

Cross-Entropy loss

Perceptual Similarity

Common: $p = \infty$
Part 1
Where to Find Adversarial Examples
Adversarial examples *leave* the data manifold.

(Thanay and Griffin, 2016)
On- and Off-Manifold Adversarial Examples

Class Manifold “5”

Classifier’s Decision Boundary

True Decision Boundary

Class Manifold “6”

Confidence-Calibrated Adversarial Training – David Stutz
On- and Off-Manifold Adversarial Examples

regular adversarial example

Class Manifold “5”

Classifier’s Decision Boundary

Class Manifold “6”

True Decision Boundary
On- and Off-Manifold Adversarial Examples

regular adversarial example

on-manifold adversarial example

Classifier’s Decision Boundary

Class Manifold “5”

True Decision Boundary

Class Manifold “6”
Implications for Robustness

Vulnerability due to unpredictable behavior off-manifold:

Data Manifold

Classification Boundary
Part 2
Confidence Calibration of Adversarial Training
Revisiting Adversarial Training

Min-max robust optimization:

$$\min \mathbb{E} \left[ \max_{\|\delta\|_{\infty} \leq \epsilon} \mathcal{L}(f(x + \delta; w), y) \right].$$
Adversarial Training (AT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do

2: for \(i = 1, \ldots, B/2\) do

3: \{maximize cross-entropy loss:\}

4: \(\tilde{x}_i := x_i + \arg\max_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)\)

5: \{enforce “label constancy” in \(\epsilon\)-ball:\}

6: \(\tilde{y}_i = y_i\)

7: update parameters using

\[
\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) \{50\% \text{ adversarial examples}\} \\
+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b) \{50\% \text{ clean examples}\}
\]
Adversarial Training (AT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do
2:   for \(i = 1, \ldots, B/2\) do
3:     \{maximize cross-entropy loss:\}
4:       \(\tilde{x}_i := x_i + \arg\max_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)\)
5:     \{enforce “label constancy” in \(\epsilon\)-ball:\}
6:       \(\tilde{y}_i = y_i\)
7:   update parameters using
8:     \(\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)\) \{50\% adversarial examples\}
9:     + \(\sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)\) \{50\% clean examples\}
Adversarial Training: Pseudo Code

Adversarial Training (AT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do
2:   for \(i = 1, \ldots, B/2\) do
3:       \{maximize cross-entropy loss:\}
4:       \(\tilde{x}_i := x_i + \arg\max_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)\)
5:       \{enforce “label constancy” in \(\epsilon\)-ball:\}
6:       \(\tilde{y}_i = y_i\)
7:   update parameters using
8:       \(\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)\) \{50\% adversarial examples\}
9:       \(+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)\) \{50\% clean examples\}
Adversarial Training: Pseudo Code

50%/50% Adversarial Training (AT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do
2:   for \(i = 1, \ldots, B/2\) do
3:     \{maximize cross-entropy loss:\}
4:     \(\tilde{x}_i := x_i + \text{argmax } \|\delta\|_\leq \epsilon \mathcal{L}(f(x_i + \delta ; w), y_i)\)
5:     \{enforce “label constancy” in \(\epsilon\)-ball:\}
6:     \(\tilde{y}_i = y_i\)
7:   update parameters using
   \[
   \sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) \{50\% \text{ adversarial examples}\}
   + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b) \{50\% \text{ clean examples}\}
   \]
Robustness does not Generalize

$L_\infty$ Perturbation in Adversarial Direction

Confidence $\leq \epsilon$ seen

SVHN:
- Correct
- Adversarial

training $\epsilon = 0.03$
Robustness does *not* Generalize

SVHN:
- Correct
- Adversarial

$L_\infty$ Perturbation
in Adversarial Direction

\begin{align*}
\text{training } \epsilon = 0.03
\end{align*}

\begin{align*}
\leq \epsilon \text{ seen } & \quad > \epsilon \text{ unseen }
\end{align*}
Robustness does not Generalize

SVHN:
- Correct
- Adversarial
Robustness does not Generalize

$\mathbf{L}_2$ Perturbation in Adversarial Direction

High confidence not meaningful beyond $\varepsilon$-ball.

SVHN:
- **Correct**
- **Adversarial**

Confidence-Calibrated Adversarial Training – David Stutz
Reduced Accuracy
Reduced Accuracy
Reduced Accuracy

Overlapping $\epsilon$-balls cause conflicts.
Reduced Accuracy: Illustration

\[ x = 2 \quad 7 = x' \]

Confidence-Calibrated Adversarial Training – David Stutz
Encourage uniform distribution on adversarial examples within the $\epsilon$-ball:

- Idea: low-confidence extrapolated beyond $\epsilon$-ball.
Confidence-Calibrated Adversarial Training

Reject (adversarial) examples with low-confidence by confidence-thresholding:

- Idea: adversarial examples receive low-confidence.
Confidence-Calibrated Adversarial Training

1. Encourage **low confidence on adversarial examples** within the $\epsilon$-ball.

2. Reject (adversarial) examples with low-confidence by **confidence-thresholding**: 

![Confidence on Adversarial Examples](image)

- **CCAT** ← reject
Confidence-Calibrated Adversarial Training (CCAT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do

2: \hspace{1em} for \(i = 1, \ldots, \frac{B}{2}\) do

3: \hspace{2em} \{maximizes adversarial confidence:\}

4: \hspace{3em} \tilde{x}_i := x_i + \arg\max_{\|\delta\|_{\infty} \leq \epsilon} \max_{k \neq y_i} f_k(x_i + \delta; w)

5: \hspace{2em} \{target distribution tends towards uniform:\}

6: \hspace{3em} \tilde{y}_i = \lambda \text{one_hot}(y_i) + \frac{(1-\lambda)}{K} 1 \text{ with } \lambda \propto \frac{1}{\|\delta\|_{\infty}}

7: update parameters using

\[
\sum_{b=1}^{\frac{B}{2}} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=\frac{B}{2}}^{B} \mathcal{L}(f(x_b), y_b)
\]
Training: Pseudo-Code
Confidence-Calibrated Adversarial Training (CCAT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do
2:     for \(i = 1, \ldots, B/2\) do
3:         \{maximizes adversarial confidence:\}
4:         \(\tilde{x}_i := x_i + \arg\max_{\|\delta\|_{\infty} \leq \epsilon} \max_{k \neq y_i} f_k(x_i + \delta; w)\)
5:         \{target distribution tends towards uniform:\}
6:         \(\tilde{y}_i = \lambda \text{one\_hot}(y_i) + \frac{(1-\lambda)}{K} 1\) with \(\lambda \propto \frac{1}{\|\delta\|_{\infty}}\)
7:     update parameters using
8:     \[\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)\]
**1 Training: Pseudo-Code**

Confidence-Calibrated Adversarial Training (CCAT):

1. **for** batches \((x_1, y_1), \ldots, (x_B, y_B)\) **do**
2. **for** \(i = 1, \ldots, B/2\) **do**
3. {maximizes adversarial confidence:}
   \[
   \tilde{x}_i := x_i + \text{argmax} \|\delta\|_\infty \leq \epsilon \max_{k \neq y_i} \ f_k(x_i + \delta; w)
   \]
4. {target distribution tends towards uniform:}
   \[
   \tilde{y}_i = \lambda \text{one_hot}(y_i) + \frac{(1-\lambda)}{K} 1 \text{ with } \lambda \propto \frac{1}{\|\delta\|_\infty}
   \]
5. update parameters using
   \[
   \sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)
   \]
Training: Pseudo-Code

Confidence-Calibrated Adversarial Training (CCAT):

1: for batches \((x_1, y_1), \ldots, (x_B, y_B)\) do

2: for \(i = 1, \ldots, B/2\) do

3: \{maximizes adversarial confidence:\}

4: \(\tilde{x}_i := x_i + \arg\max_{\|\delta\|_{\infty} \leq \epsilon} \max_{k \neq y_i} f_k(x_i + \delta; w)\)

5: \{target distribution tends towards uniform:\}

6: \(\tilde{y}_i = \lambda \text{one\_hot}(y_i) + \frac{(1-\lambda)}{K} 1\) with \(\lambda \propto 1/\|\delta\|_{\infty}\)

7: update parameters using

\[
\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)
\]
Confidence Thresholding: Robustness

\[ L_\infty \text{ Perturbation in Adversarial Direction} \]

\[ \text{training } \epsilon = 0.03 \]

SVHN:
- Correct
- Adversarial

Confidence Thresholding: Robustness

Confidence-Calibrated Adversarial Training – David Stutz
Confidence Thresholding: Robustness

$\mathcal{L}_\infty$ Perturbation in Adversarial Direction

$\leq \epsilon$ seen $\geq \epsilon$ unseen

Training $\epsilon = 0.03$

SVHN:
- Correct
- Adversarial

Confidence Thresholding: Robustness

Confidence-Calibrated Adversarial Training – David Stutz
Confidence Thresholding: Robustness

Confidence Threshold $\tau$

SVHN:
- Correct
- Adversarial

$L_\infty$ Perturbation in Adversarial Direction
Confidence Thresholding: Robustness

SVHN:
- Correct
- Adversarial
2 Confidence Thresholding: Robustness

SVHN:
- Correct
- Adversarial

Uniform confidence extrapolates beyond $\epsilon$-ball.
Improved Accuracy

\[ x = 2 \rightarrow 7 = x' \]

Confidence

Interpolation Factor \( \kappa \) for

\[ \kappa x + (1 - \kappa) x' \]
Improved Accuracy

\[ x = \begin{array}{c} 2 \end{array} \quad \begin{array}{c} 7 \end{array} = x' \]

Interpolation Factor \( \kappa \) for \( \kappa x + (1 - \kappa)x' \)

Overlapping \( \epsilon \)-balls no problem.
Part 3

Lessons for Evaluation
Lessons for Evaluation

1 Define fair evaluation metrics:
   ▶ Reviewers *do not* like unnecessary new metrics.

Confidence-Calibrated Adversarial Training – David Stutz
Define fair evaluation metrics:

- Reviewers *do not* like unnecessary new metrics.

Define proper adversaries:

- Avoid “cracking” your defense 2 days before the NeurIPS deadline!
Lessons for Evaluation

1. Define fair evaluation metrics:
   - Reviewers *do not* like unnecessary new metrics.

2. Define proper adversaries:
   - Avoid “cracking” your defense 2 days before the NeurIPS deadline!

3. Worst-case evaluation:
   - Results might look better than they are.
1 “Standard” Robust Test Error RErr

= error on test examples that are “attacked”.

Adversarial Training (AT):
57.3% RErr

Ours (CCAT):
97.8% RErr
“Standard” Robust Test Error RErr

= error on test examples that are “attacked”.

Adversarial Training (AT):
57.3% RErr

Total: 539/1000

Confidence on Adversarial Examples

Ours (CCAT):
97.8% RErr

Total: 949/1000

Confidence on Adversarial Examples
1 “Standard” Robust Test Error RErr

= error on test examples that are “attacked”.

Adversarial Training (AT):

57.3% RErr

Ours (CCAT):

97.8% RErr
Confidence-Thresholded RErr

= error on test examples that are “attacked” and pass confidence thresholding.

Adversarial Training (AT):
56% (−1.3%)

Ours (CCAT):
39.1% (−58.7%)
Confidence Threshold

- Fix confidence threshold $\tau$ at 99% TPR.

Confidence on Test Examples

Confidence on Adversarial Examples
1 Confidence Threshold

- independent of adversarial examples;

Confidence on Test Examples

Confidence on Adversarial Examples
1 Confidence Threshold

- independent of adversarial examples;
- and avoid incorrectly rejecting (clean) test examples.

Confidence on Test Examples

Confidence on Adversarial Examples
Confidence Threshold

- independent of adversarial examples;
- and avoid incorrectly rejecting (clean) test examples.

Fix confidence threshold $\tau$ at 99% TPR.
2 Adversaries: Basics

“Adapted” objective:

\[
\arg\max_{\delta \in \mathbb{R}^d} \max_{k \neq y} f_k(x + \delta)
\]

\[\|\delta\|_\infty \leq \epsilon\]

Confidence in Class \(k\):

Applicable to many white- and black-box attacks:

<table>
<thead>
<tr>
<th>Attack</th>
<th>Iterations</th>
<th>Restarts</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGD</td>
<td>200-1000</td>
<td>10-50</td>
</tr>
<tr>
<td>Query-Limited†</td>
<td>1000</td>
<td>11</td>
</tr>
<tr>
<td>Simple†</td>
<td>1000</td>
<td>10</td>
</tr>
<tr>
<td>Square†</td>
<td>5000</td>
<td>1</td>
</tr>
<tr>
<td>Geometry†</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>Random†</td>
<td>–</td>
<td>5000</td>
</tr>
</tbody>
</table>

† Black-box attacks.
Understand objective surface in order to improve optimization.

2 Adversaries: “Objective Surface”

CCAT on SVHN

$L_\infty$ Perturbation in Adversarial Direction

Confidence

0 0.01 0.02 0.03 0.04

Adversarial Example
Understand **objective surface** in order to improve optimization.

- Fixed learning rates cause problems.
2 Adversaries: Backtracking

Understand objective surface in order to improve optimization with backtracking.

- Avoids oscillation and improves objective.
### Worst-Case Evaluation

Difference between per-attack results and per-example worst-case results:

<table>
<thead>
<tr>
<th></th>
<th>worst case</th>
<th>top-5 attacks/restarts out of 7 attacks with 84 restarts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN: RErr in % for $L_\infty$ with $\epsilon = 0.03$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT</td>
<td>56.0</td>
<td>52.1 52.0 51.9 51.6 51.4</td>
</tr>
<tr>
<td>CCAT</td>
<td>39.1</td>
<td>23.6 13.7 13.6 12.6 12.5</td>
</tr>
</tbody>
</table>

(Higher RErr means “stronger” attack(s).)

---

Confidence-Calibrated Adversarial Training – David Stutz
3 Worst-Case Evaluation

Difference between per-attack results and per-example worst-case results:

<table>
<thead>
<tr>
<th></th>
<th>worst case</th>
<th>top-5 attacks/restarts out of 7 attacks with 84 restarts</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN: RErr in % for $L_\infty$ with $\epsilon = 0.03$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AT</td>
<td>56.0</td>
<td>52.1, 52.0, 51.9, 51.6, 51.4</td>
</tr>
<tr>
<td>CCAT</td>
<td>39.1</td>
<td>23.6, 13.7, 13.6, 12.6, 12.5</td>
</tr>
</tbody>
</table>

(Higher RErr means “stronger” attack(s).)

- Attacking our CCAT requires many attacks/restarts.
Part 4

Results
SVHN: Generalization to Unseen Attacks

<table>
<thead>
<tr>
<th>( L_\infty )</th>
<th>( \varepsilon = 0.03 )</th>
<th>seen</th>
<th>RErr ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>56.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CCAT</td>
<td>39.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Lower RErr means “better” robustness.)
SVHN: Generalization to Unseen Attacks

<table>
<thead>
<tr>
<th>SVHN: RErr in % for $\tau@99%$ TPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_\infty$</td>
</tr>
<tr>
<td>$\epsilon = 0.03$</td>
</tr>
<tr>
<td>seen</td>
</tr>
<tr>
<td>RErr ↓</td>
</tr>
</tbody>
</table>

AT  
CCAT  
56.0  
39.1  

(Lower RErr means “better” robustness.)
SVHN: Generalization to Unseen Attacks

### SVHN: RErr in % for $\tau@99\%$ TPR

<table>
<thead>
<tr>
<th></th>
<th>$L_\infty$ (\epsilon = 0.03)</th>
<th>$L_\infty$ (\epsilon = 0.06)</th>
<th>$L_2$ (\epsilon = 2)</th>
<th>$L_1$ (\epsilon = 24)</th>
<th>$L_0$ (\epsilon = 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>seen</strong></td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
</tr>
<tr>
<td><strong>unseen</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>AT</strong></td>
<td>56.0</td>
<td>88.4</td>
<td>99.4</td>
<td>99.5</td>
<td>73.6</td>
</tr>
<tr>
<td><strong>CCAT</strong></td>
<td>39.1</td>
<td>53.1</td>
<td>29.0</td>
<td>31.7</td>
<td>3.5</td>
</tr>
</tbody>
</table>

(Lower RErr means “better” robustness.)
### CIFAR10: RErr in % for $\tau@99\%$ TPR

<table>
<thead>
<tr>
<th></th>
<th>$L_\infty$ $\epsilon = 0.03$</th>
<th>$L_\infty$ $\epsilon = 0.06$</th>
<th>$L_2$ $\epsilon = 2$</th>
<th>$L_1$ $\epsilon = 24$</th>
<th>$L_0$ $\epsilon = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>seen</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
<td>RErr ↓</td>
</tr>
<tr>
<td>AT</td>
<td>62.7</td>
<td>93.7</td>
<td>98.4</td>
<td>98.4</td>
<td>72.4</td>
</tr>
<tr>
<td>CCAT</td>
<td>67.9</td>
<td>92.0</td>
<td>51.8</td>
<td>58.5</td>
<td>20.3</td>
</tr>
</tbody>
</table>

(Lower RErr means “better” robustness.)
**“Unconventional” Attacks**

<table>
<thead>
<tr>
<th>CIFAR10: $\tau@99%$ TPR</th>
<th>adv. frames</th>
<th>distal</th>
<th>corrupted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>unseen</strong></td>
<td><strong>unseen</strong></td>
<td><strong>unseen</strong></td>
<td></td>
</tr>
<tr>
<td>RErr ↓</td>
<td>FPR ↓</td>
<td>CErr ↓</td>
<td></td>
</tr>
<tr>
<td>Normal</td>
<td>96.6</td>
<td>83.3</td>
<td>12.3</td>
</tr>
<tr>
<td>AT</td>
<td>78.7</td>
<td>75.0</td>
<td>16.2</td>
</tr>
<tr>
<td>CCAT</td>
<td>65.1</td>
<td>0</td>
<td>8.5</td>
</tr>
</tbody>
</table>

(FPR: fraction of non-rejected distal adversarial examples.)

(CErr: test error on corrupted examples after thresholding.)
## Improved Accuracy

<table>
<thead>
<tr>
<th></th>
<th>SVHN: Err in %</th>
<th>CIFAR10: Err in %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\tau = 0$</td>
<td>$\tau = 0$</td>
</tr>
<tr>
<td></td>
<td>99% TPR</td>
<td>99% TPR</td>
</tr>
<tr>
<td>Normal</td>
<td>3.6</td>
<td>8.3</td>
</tr>
<tr>
<td>AT</td>
<td>3.4</td>
<td>16.6</td>
</tr>
<tr>
<td>CCAT</td>
<td>2.9</td>
<td>10.1</td>
</tr>
</tbody>
</table>

(Err: test error before and after thresholding.)
Part 5

Conclusion
Robustness Evaluation

Checklist for reviews and experiments:

1. Reasonable metrics.
2. Multiple, adaptive attacks.
3. Worst-case evaluation.
Encourage low-confidence on adversarial examples:

- Robustness generalizes to unseen attacks;
- and accuracy improves.

- Explicit guidance how to behave off-manifold.
Questions?

More: davidstutz.de
Code coming soon!

References:

Appendix
“Power” Transition

How to choose $\lambda \propto \frac{1}{\|\delta\|_\infty}$ for:

6: $\tilde{y}_i = \lambda \text{one_hot}(y_i) + \frac{(1-\lambda)}{K} 1$ with $\lambda \propto \frac{1}{\|\delta\|_\infty}$

“Power” transition:

$$\lambda = 1 - (1 - \min(1, \frac{\|\delta\|_\infty}{\epsilon}))^\rho,$$

$\|\delta\|_\infty \leq \epsilon$

- Nearly exponential in confidence;
- avoids a bias towards the true label.
“Power” Transition

How to choose $\lambda \propto \frac{1}{\|\delta\|_\infty}$ for:

6: $\tilde{y}_i = \lambda \text{one_hot}(y_i) + \frac{(1-\lambda)}{K} 1$ with $\lambda \propto \frac{1}{\|\delta\|_\infty}$

$\lambda = (1 - \min(1, \|\delta\|_\infty / \epsilon))^\rho$

![Graph showing the relationship between Perturbation $\|\delta\|_\infty$ and Target Distribution $\tilde{y}$]
“Exponential” Transition

ow to choose $\lambda \propto \frac{1}{\|\delta\|_\infty}$ for:

6: $\tilde{y}_i = \lambda \text{one}_\text{hot}(y_i) + \frac{(1-\lambda)}{K}1$ with $\lambda \propto \frac{1}{\|\delta\|_\infty}$

Exponential transition:

$$\lambda = \exp(-\rho \|\delta\|_\infty), \quad \|\delta\|_\infty \leq \epsilon$$

- Exponential in confidence means linear in logits;
- keeps a bias towards the true label as $\lambda > 0$;
- $\rho$ depends on $\epsilon$, large $\rho$ required;
“Standard” Robust Test Error

= error on test examples that are “attacked”:

\[
\text{“Standard” RErr} = \frac{1}{N} \sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon} \mathbb{1}_{f(x_n + \delta) \neq y_n}
\]

\[
\implies \left\{ (x_n, y_n) \right\}_{n=1}^{N} \text{ test examples.}
\]
Considering the “Reject Option”

Adversarial Training (AT): 57.3% RErr

Ours (CCAT): 97.8% RErr

Confidence on Adversarial Examples
Confidence-Thresholded Robust Test Error

= error on test examples that are “attacked” and pass the confidence threshold $\tau$:

$$\text{RErr}(\tau) = \frac{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon, c(x_n+\delta) \geq \tau} \mathbb{1} f(x_n+\delta) \neq y_n}{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon} \mathbb{1} c(x_n+\delta) \geq \tau}$$

- $c(x_n) := \max_k f_k(x_n)$ confidence on $x_n$.
Special Cases

Why is the confidence-thresholded RErr non-trivial?

- Adversarial examples can have higher confidence than test examples.

![Confidence on Test Examples](chart1.png)

![Confidence on Adversarial Examples](chart2.png)
Confidence-thresholded RErr is approximated using our PGD attack and:

\[
\sum_{n=1}^{N} \max\{ \mathbb{1} f(x_n) \neq y_n \mathbb{1} c(x_n) \geq \tau, \mathbb{1} f(\tilde{x}_n) \neq y_n \mathbb{1} c(\tilde{x}_n) \geq \tau \}
\]

\[
\sum_{n=1}^{N} \max\{ \mathbb{1} c(x_n) \geq \tau, \mathbb{1} c(\tilde{x}_n) \geq \tau \}
\]

\[
\triangleright c(x_n) := \max_k f_k(x_n) \text{ confidence on } x_n;
\]
Choosing confidence threshold \( \tau \):

- independent of adversarial examples;
- avoid incorrectly rejecting (clean) test examples.
Adversaries: Iterations and Initialization

Use plenty of iterations and zero initialization:

<table>
<thead>
<tr>
<th>Optimization</th>
<th>backtracking+momentum</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>40</td>
<td>200</td>
</tr>
<tr>
<td>AT</td>
<td>38.4</td>
<td>46.2</td>
</tr>
<tr>
<td>CCAT</td>
<td>4.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

(Higher REErr means “stronger” attack.)

**SVHN:** REErr in % for $L_\infty$ with $\epsilon = 0.03$
Adversaries: Iterations and Initialization

Use plenty of iterations and zero initialization:

<table>
<thead>
<tr>
<th>Optimization</th>
<th>backtracking+momentum</th>
<th>—</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterations</td>
<td>40</td>
<td>200</td>
</tr>
<tr>
<td>AT</td>
<td>38.4</td>
<td>46.2</td>
</tr>
<tr>
<td>CCAT</td>
<td>4.0</td>
<td>5.0</td>
</tr>
</tbody>
</table>

(Higher RErr means “stronger” attack.)

- Our CCAT is (computationally) “harder” to attack.