Confidence-Calibrated Adversarial Training

Lessons for Evaluating Defenses

David Stutz

with Bernt Schiele, Matthias Hein

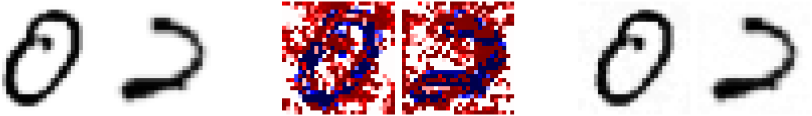
max planck institut informatik

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Part 0

Background

Adversarial Examples



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

Classifier

$$\underset{\delta}{\operatorname{argmax}} \; \mathcal{L}(f(x+\delta; w), y)$$

Cross-Entropy loss

$$\text{s.t. } \|\delta\|_p \leq \epsilon$$

Perceptual Similarity

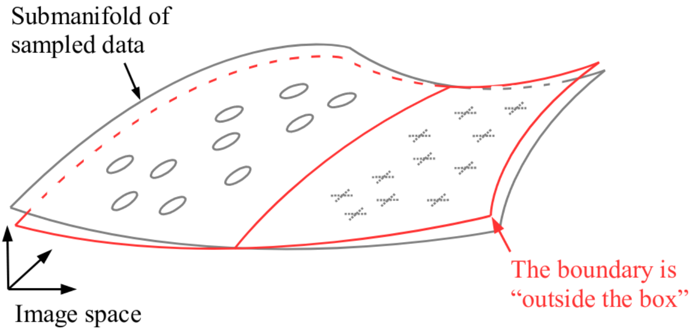
Common: $p = \infty$

Part 1

Where to Find Adversarial Examples

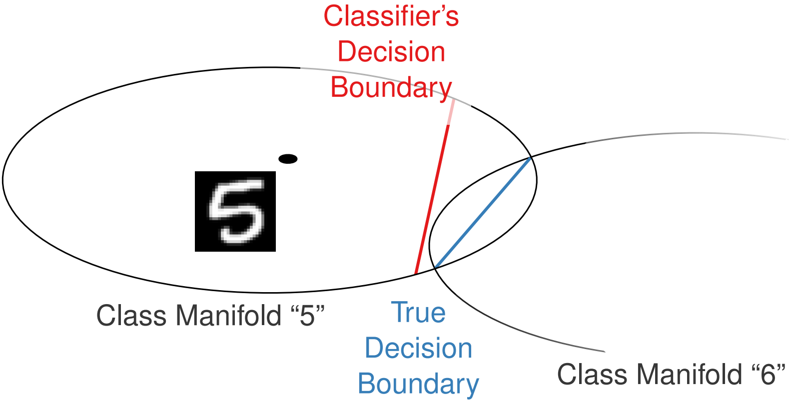
Manifold Assumption

Adversarial examples *leave* the data manifold.

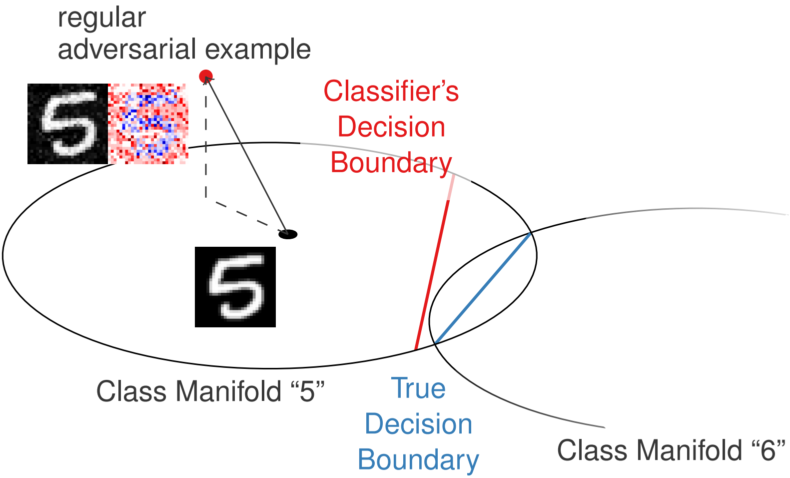


(Thanay and Griffin, 2016)

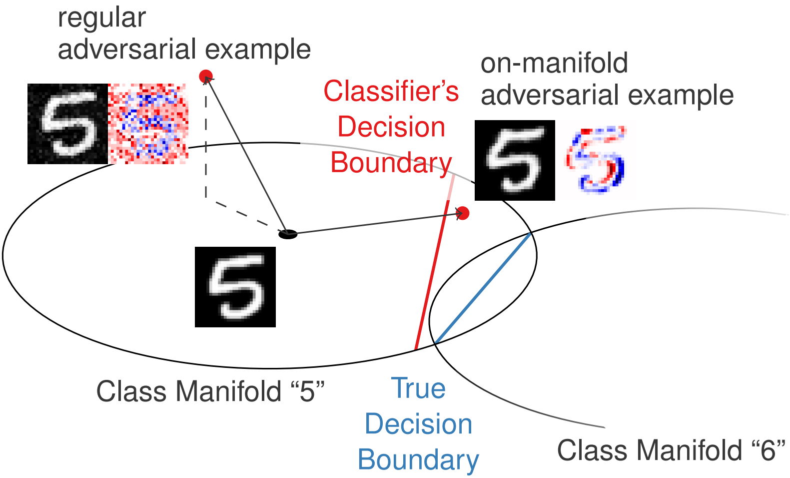
On- and Off-Manifold Adversarial Examples



On- and Off-Manifold Adversarial Examples

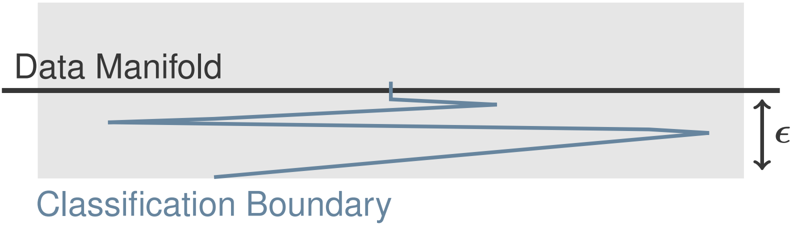


On- and Off-Manifold Adversarial Examples



Implications for Robustness

Vulnerability due to unpredictable behavior off-manifold:



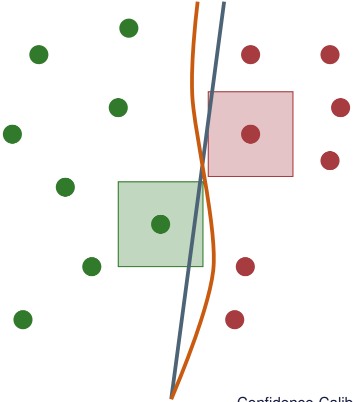
Part 2

Confidence Calibration of Adversarial Training

Revisiting Adversarial Training

Min-max robust optimization:

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



Adversarial Training: Pseudo Code

Adversarial Training (AT):

for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**
  for $i = 1, \ldots, B/2$ **do**
    {maximize cross-entropy loss:}
    $\tilde{x}_i := x_i + \operatorname{argmax}_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)$
    {enforce "label constancy" in ϵ-ball:}
    $\tilde{y}_i = y_i$
  update parameters using
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)$ {50% adversarial examples}
    $+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$ {50% clean examples}

Adversarial Training: Pseudo Code

Adversarial Training (AT):

for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**
  for $i = 1, \ldots, {}^B/_2$ **do**
    {maximize cross-entropy loss:}
    $\tilde{x}_i := x_i + \text{argmax}_{\|\delta\|\leq\epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)$
    {enforce "label constancy" in ϵ-ball:}
    $\tilde{y}_i = y_i$
  update parameters using
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)$ {50% adversarial examples}
    $+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$ {50% clean examples}

Adversarial Training: Pseudo Code

Adversarial Training (AT):

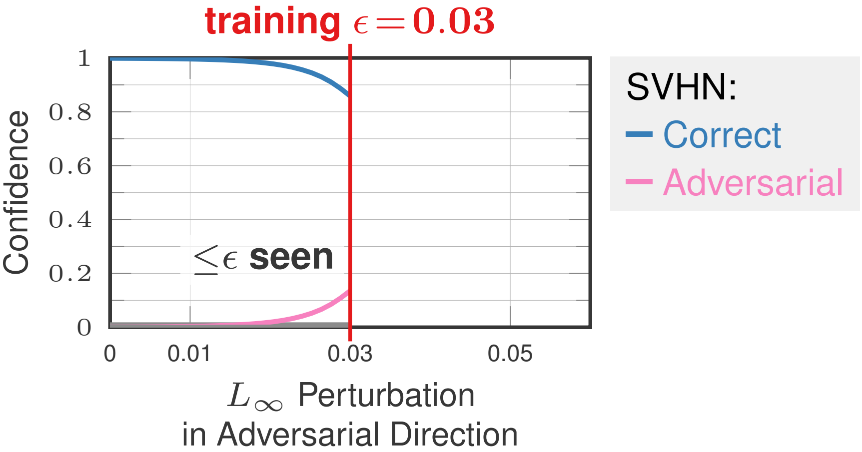
for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**
  for $i = 1, \ldots, {}^B\!/_2$ **do**
    {maximize cross-entropy loss:}
    $\tilde{x}_i := x_i + \operatorname{argmax}_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)$
    {enforce “label constancy” in ϵ-ball:}
    $\tilde{y}_i = y_i$
  update parameters using
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)$ {50% adversarial examples}
    $+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$ {50% clean examples}

Adversarial Training: Pseudo Code

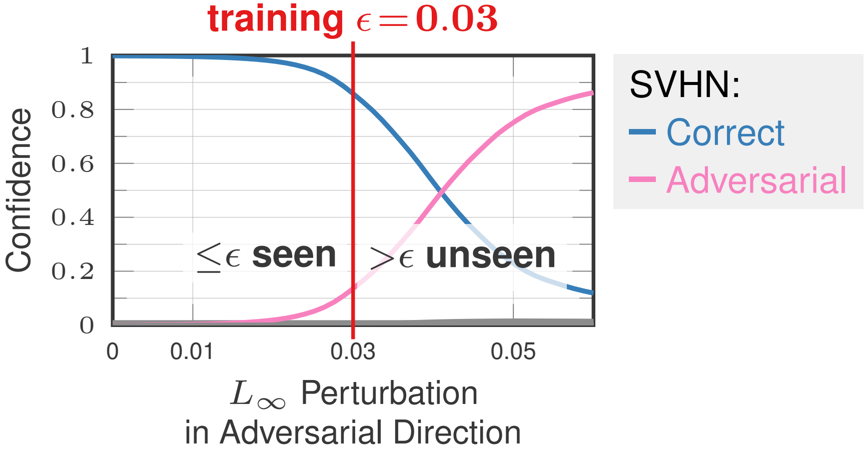
50%/50% Adversarial Training (AT):

for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**
  for $i = 1, \ldots, {}^B/_2$ **do**
    {maximize cross-entropy loss:}
    $\tilde{x}_i := x_i + \operatorname{argmax}_{\|\delta\| \leq \epsilon} \mathcal{L}(f(x_i + \delta; w), y_i)$
    {enforce “label constancy” in ϵ-ball:}
    $\tilde{y}_i = y_i$
  update parameters using
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)$ {50% adversarial examples}
    $+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$ {50% clean examples}

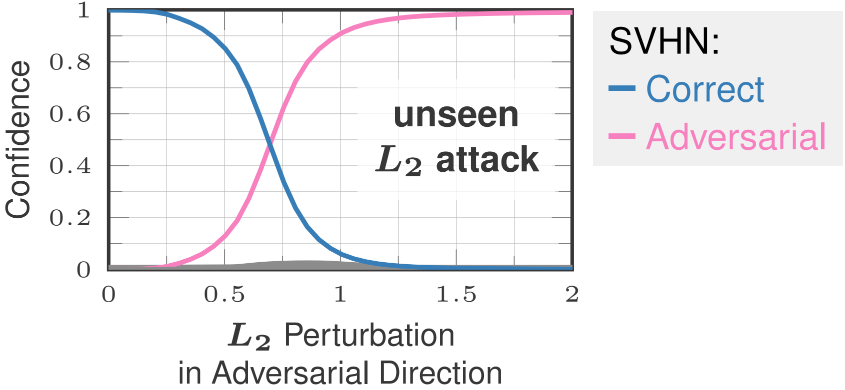
Robustness does *not* Generalize



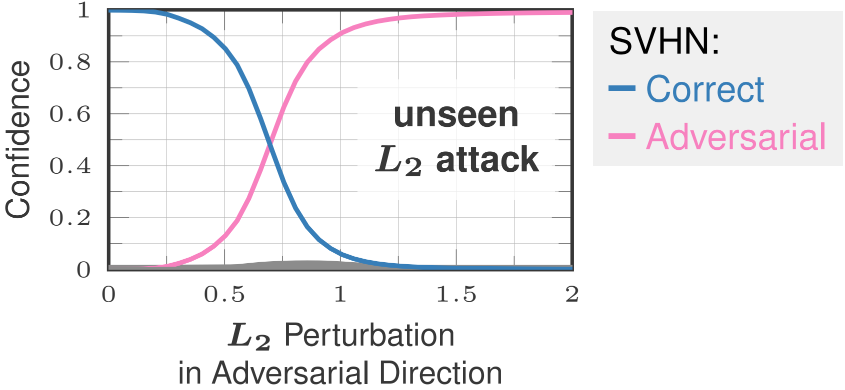
Robustness does *not* Generalize



Robustness does *not* Generalize

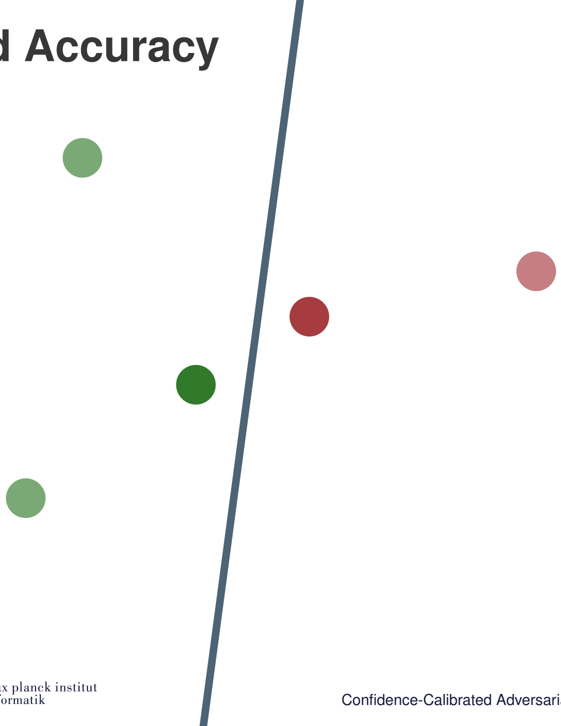


Robustness does *not* Generalize

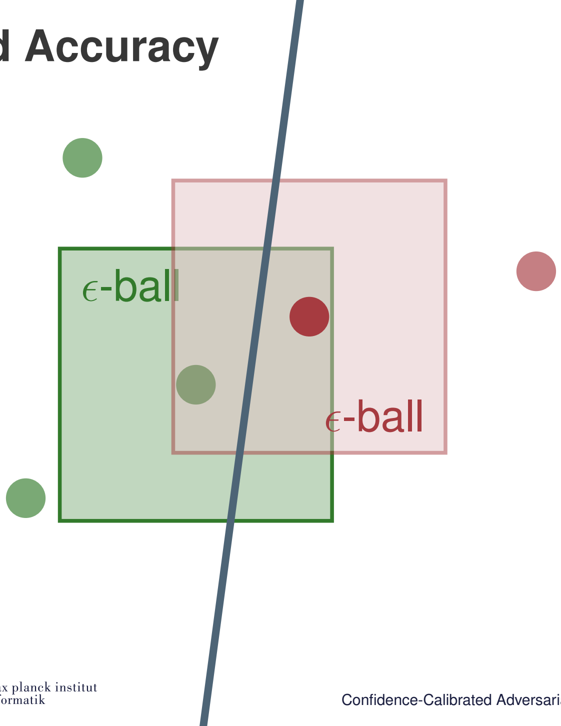


High confidence not meaningful beyond ϵ-ball.

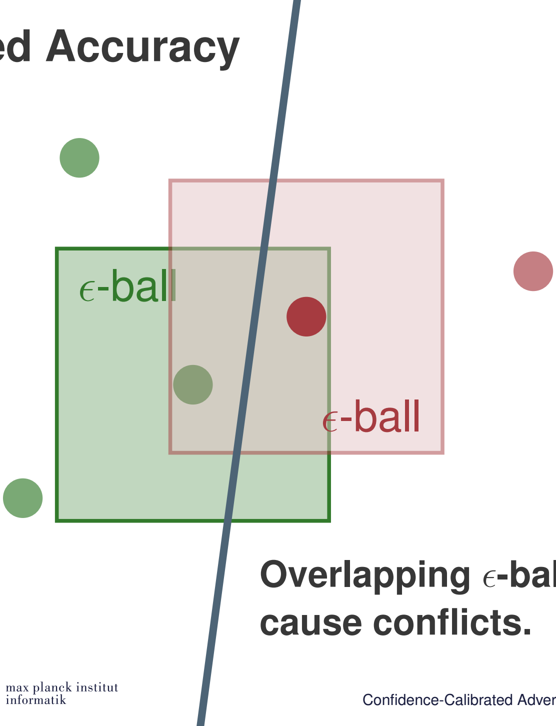
Reduced Accuracy



Reduced Accuracy

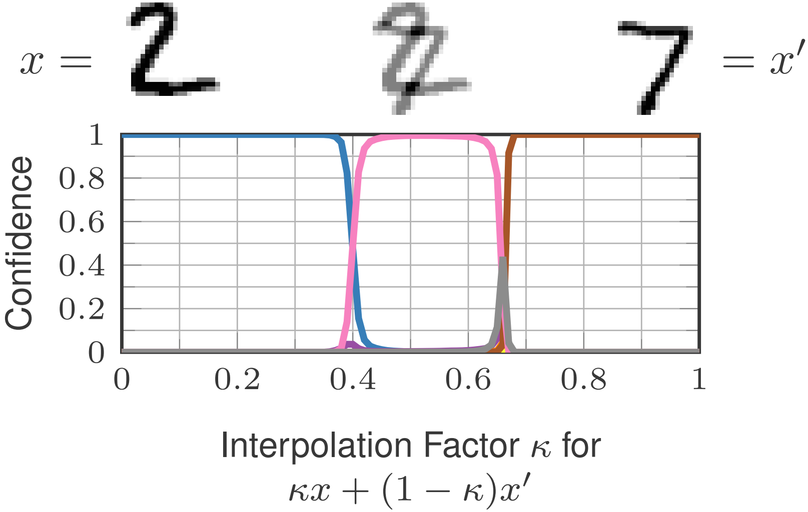


Reduced Accuracy



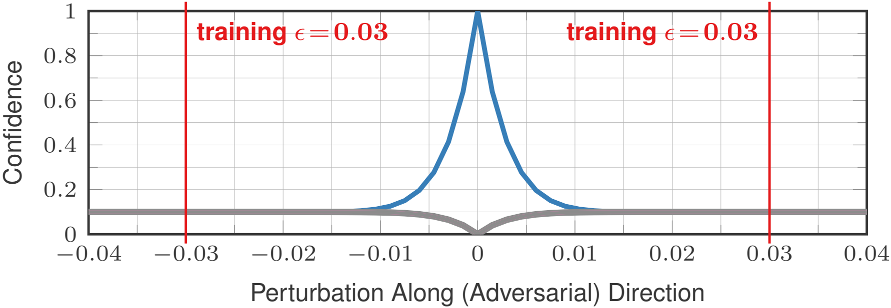
Overlapping ϵ-balls cause conflicts.

Reduced Accuracy: Illustration



Confidence-Calibrated Adversarial Training

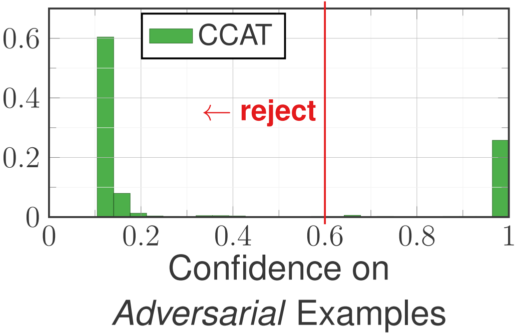
1 Encourage uniform distribution on adversarial examples within the ϵ-ball:



- Idea: low-confidence extrapolated beyond ϵ-ball.

Confidence-Calibrated Adversarial Training

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

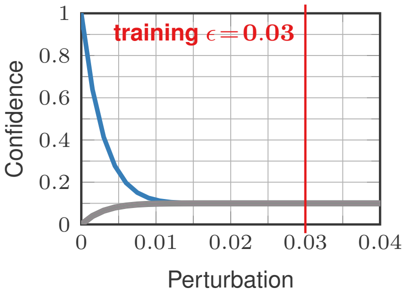


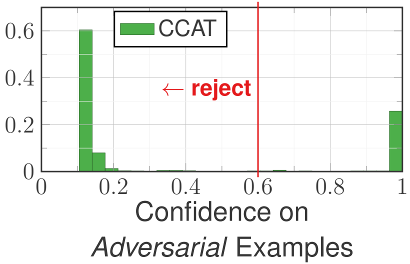
- Idea: adversarial examples receive low-confidence.

Confidence-Calibrated Adversarial Training

1 Encourage **low confidence on adversarial examples** within the ϵ-ball.

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





1 Training: Pseudo-Code

Confidence-Calibrated Adversarial Training (CCAT):

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

1 Training: Pseudo-Code

Confidence-Calibrated Adversarial Training (CCAT):

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

1 Training: Pseudo-Code

Confidence-Calibrated Adversarial Training (CCAT):

for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**
  for $i = 1, \ldots, {}^B/_2$ **do**
    {maximizes adversarial confidence:}
    $\tilde{x}_i := x_i + \operatorname{argmax}_{\|\delta\|_\infty \leq \epsilon} \max_{k \neq y_i} f_k(x_i + \delta; w)$
    {target distribution tends towards uniform:}
    $\tilde{y}_i = \lambda\,\text{one\_hot}(y_i) + \frac{(1-\lambda)}{K} 1$ with $\lambda \propto {}^1/_{\|\delta\|_\infty}$
  update parameters using
    $\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b)$
    $+ \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$

1 Training: Pseudo-Code

Confidence-Calibrated Adversarial Training (CCAT):

for batches $(x_1, y_1), \ldots, (x_B, y_B)$ **do**

  for $i = 1, \ldots, {}^B/_2$ **do**

    {maximizes adversarial confidence:}

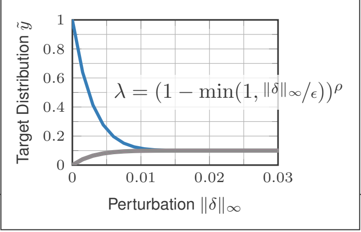
    $\tilde{x}_i := x_i + \text{ argmax }_{\|\delta\|_\infty \leq \epsilon} \max_{k \neq y_i} f_k(x_i + \delta; w)$

    {target distribution tends towards uniform:}

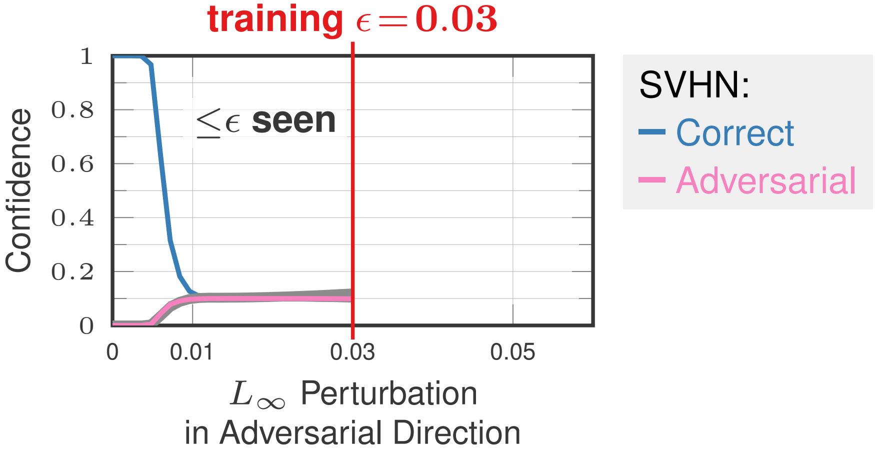
    $\tilde{y}_i = \lambda\,\text{one\_hot}(y_i) + \frac{(1-\lambda)}{K}1$ with $\lambda \propto {}^1/_{\|\delta\|_\infty}$

  update parameters using

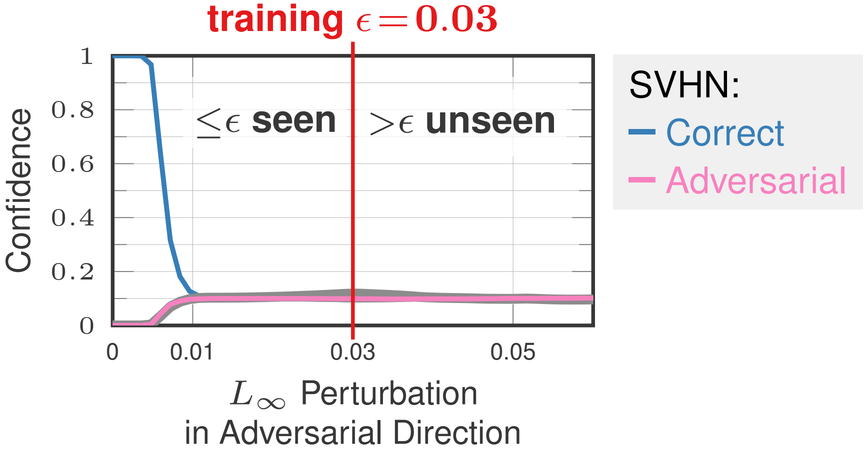
$$\sum_{b=1}^{B/2} \mathcal{L}(f(\tilde{x}_b), \tilde{y}_b) + \sum_{b=B/2}^{B} \mathcal{L}(f(x_b), y_b)$$



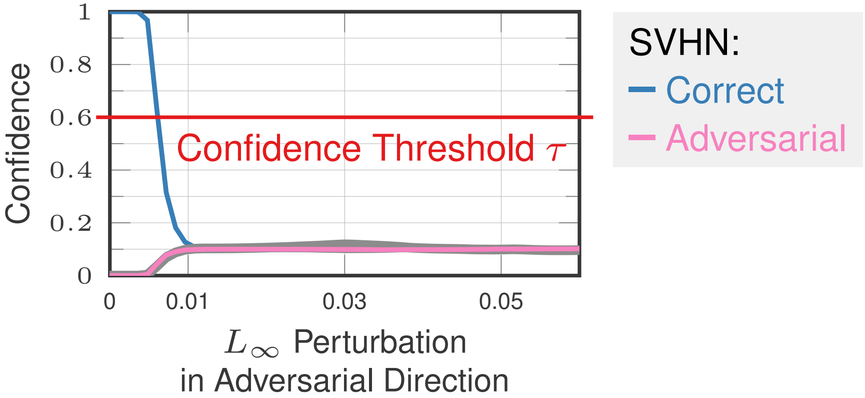
2 Confidence Thresholding: Robustness



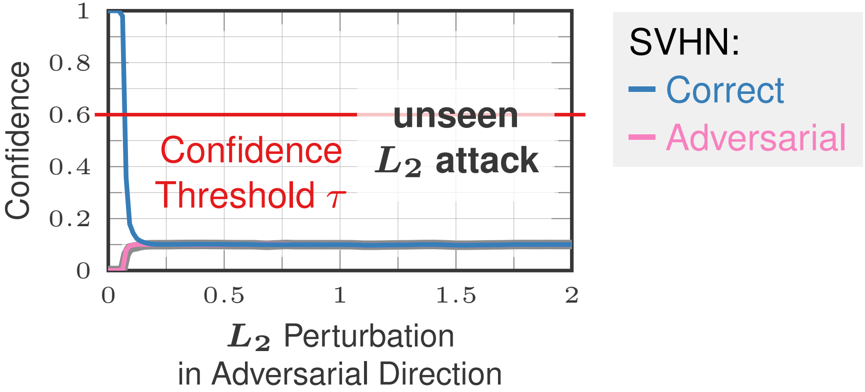
2 Confidence Thresholding: Robustness



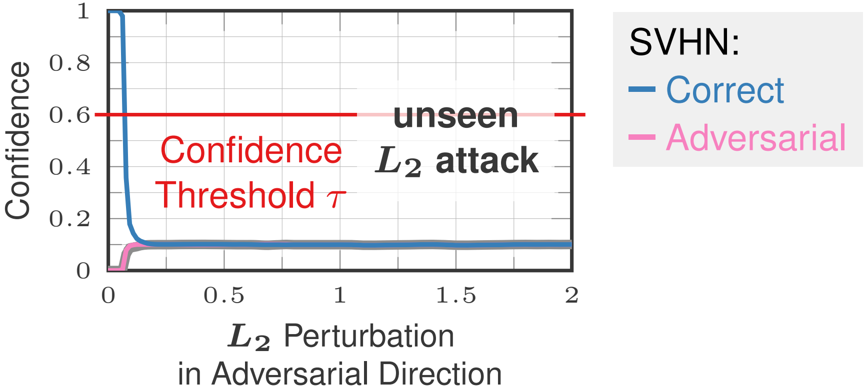
2 Confidence Thresholding: Robustness



2 Confidence Thresholding: Robustness

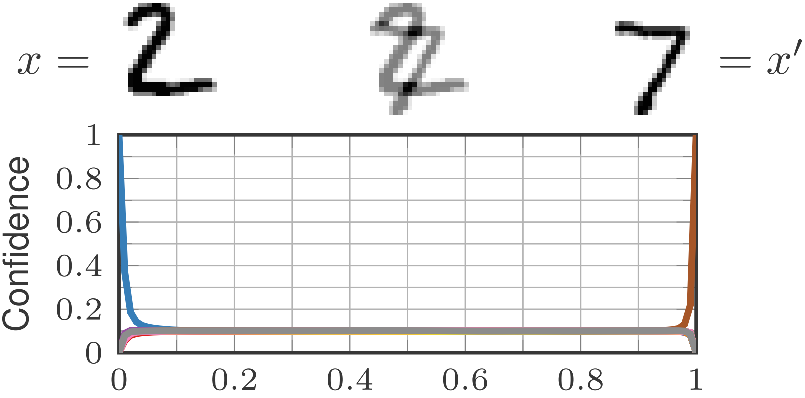


② Confidence Thresholding: Robustness



Uniform confidence extrapolates beyond ϵ-ball.

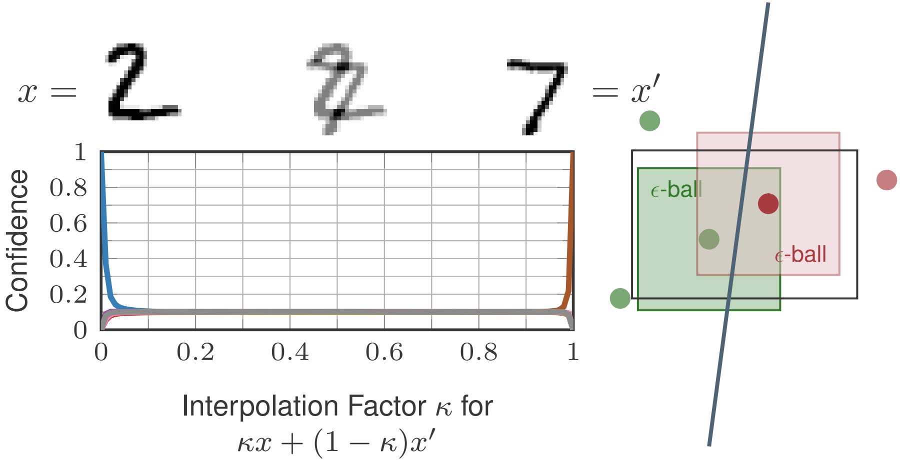
Improved Accuracy



Interpolation Factor κ for

$$\kappa x + (1 - \kappa) x'$$

Improved Accuracy



Overlapping ϵ-balls no problem.

Part 3

Lessons for Evaluation

Lessons for Evaluation

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

Lessons for Evaluation

1. Define fair evaluation metrics:
   - Reviewers *do not* like unnecessary new metrics.
2. Define proper adversar**ies**:
   - 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 adversar**ies**:
   - 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".

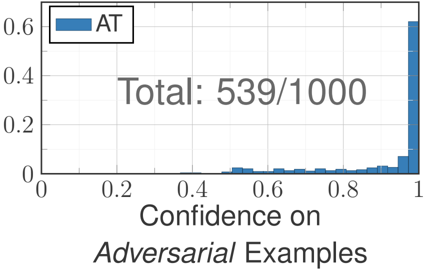
Adversarial Training (AT):
57.3% RErr

Ours (CCAT):
97.8% RErr

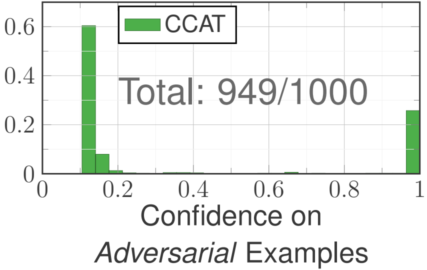
1 "Standard" Robust Test Error RErr

= error on test examples that are "attacked".

Adversarial Training (AT):
57.3% RErr



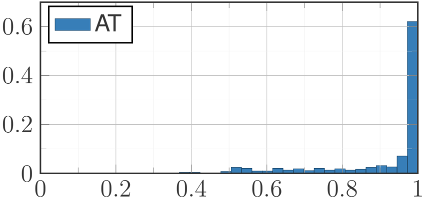
Ours (CCAT):
97.8% RErr



1 "Standard" Robust Test Error RErr

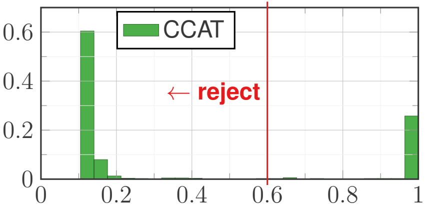
= error on test examples that are "attacked".

Adversarial Training (AT):
57.3% RErr



Confidence on *Adversarial* Examples

Ours (CCAT):
97.8% RErr



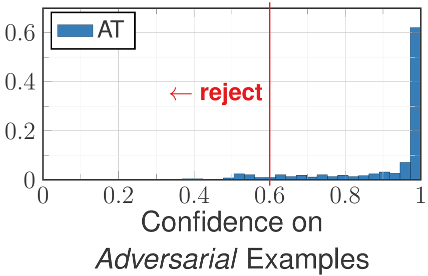
Confidence on *Adversarial* Examples

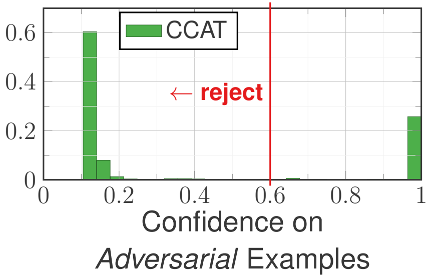
1 Confidence-Thresholded RErr

= error on test examples that are "attacked" and *pass* confidence thresholding.

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

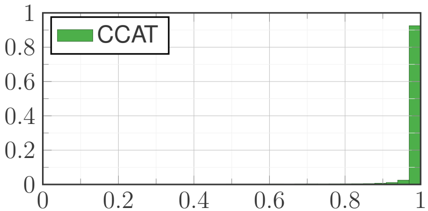
Ours (CCAT):
39.1% (−58.7%)



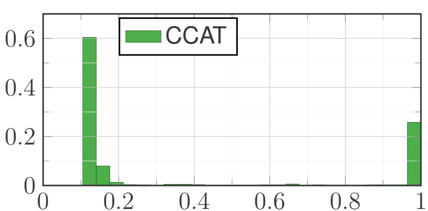


1 Confidence Threshold

-
-



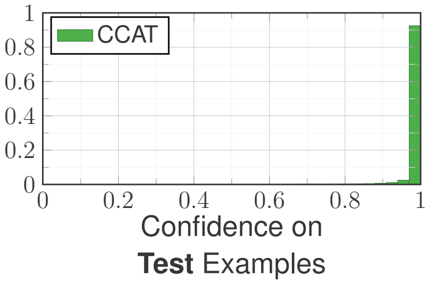
Confidence on
Test Examples

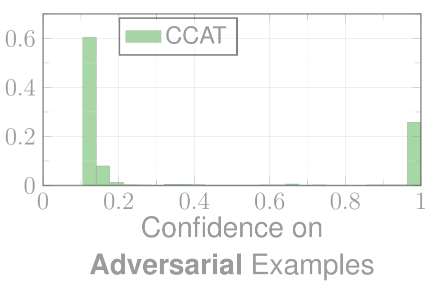


Confidence on
Adversarial Examples

1 Confidence Threshold

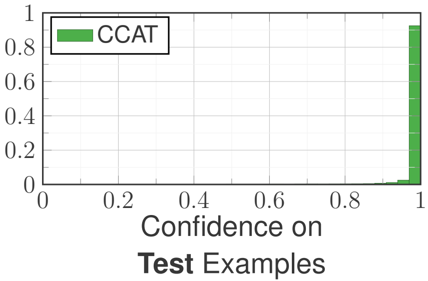
- independent of adversarial examples;
-

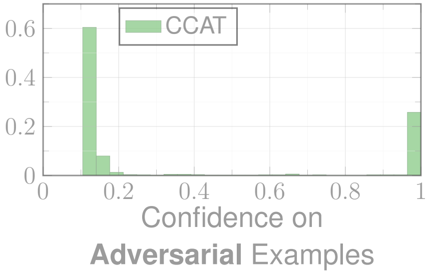




1 Confidence Threshold

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

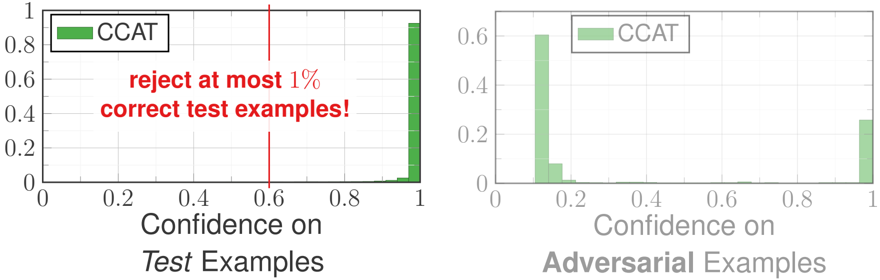




1 Confidence Threshold

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

Fix confidence threshold τ at 99%TPR.



2 Adversaries: Basics

"Adapted" objective:

Confidence in Class k

$$\underset{\|\delta\|_\infty \le \epsilon}{\operatorname{argmax}} \max_{k \neq y} f_k(x + \delta).$$

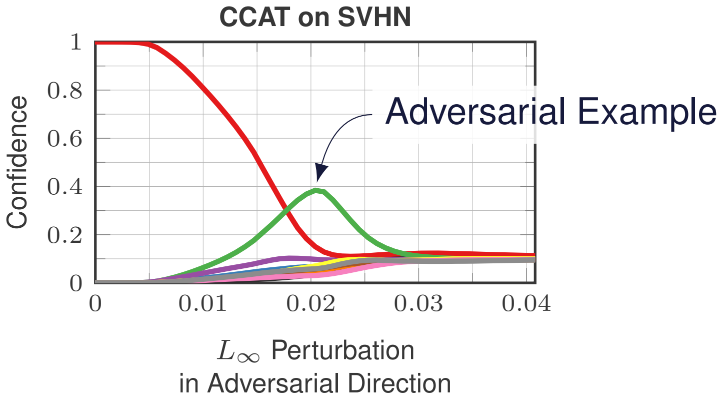
Applicable to many white- and black-box attacks:

Attack	Iterations	Restarts
PGD	200-1000	10-50
Query-Limited†	1000	11
Simple†	1000	10
Square†	5000	1
Geometry†	1000	1
Random†	–	5000

† Black-box attacks.

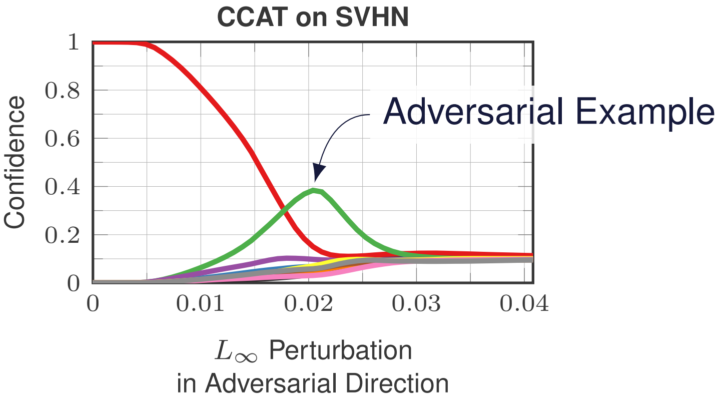
2 Adversaries: "Objective Surface"

Understand objective surface in order to improve optimization.



2 Adversaries: “Objective Surface”

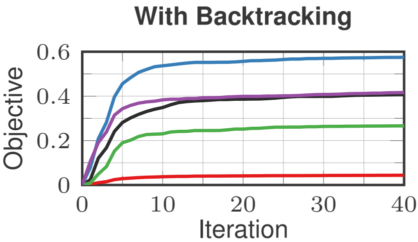
Understand objective surface in order to improve optimization.

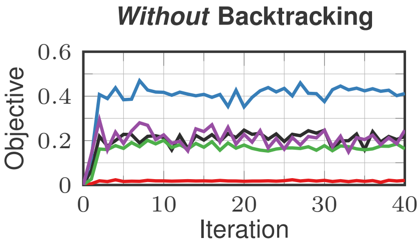


- Fixed learning rates cause problems.

2 Adversaries: Backtracking

Understand objective surface in order to **improve optimization with backtracking**.





- Avoids oscillation and improves objective.

3 Worst-Case Evaluation

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

SVHN: RErr in % for L_∞ with $\epsilon = 0.03$						
	worst case	top-5 attacks/restarts out of 7 attacks with 84 restarts				
AT	56.0	52.1	52.0	51.9	51.6	51.4
CCAT	39.1	23.6	13.7	13.6	12.6	12.5

(Higher RErr means "stronger" attack(s).)

3 Worst-Case Evaluation

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

SVHN: RErr in % for L_∞ with $\epsilon = 0.03$						
	worst case	top-5 attacks/restarts out of 7 attacks with 84 restarts				
AT	56.0	52.1	52.0	51.9	51.6	51.4
CCAT	39.1	23.6	13.7	13.6	12.6	12.5

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

- Attacking our CCAT requires many attacks/restarts.

Part 4

Results

SVHN: Generalization to Unseen Attacks

SVHN: RErr in % for $\tau@99\%$TPR					
	L_∞ $\epsilon = 0.03$				
	seen				
	RErr $\downarrow$				
AT	56.0				
CCAT	**39.1**				

(Lower RErr means “better” robustness.)

SVHN: Generalization to Unseen Attacks

SVHN: RErr in % for τ@99%TPR					
	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$
	seen	**unseen**	**unseen**	**unseen**	**unseen**
	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$
AT	56.0				
CCAT	**39.1**				

(Lower RErr means "better" robustness.)

SVHN: Generalization to Unseen Attacks

SVHN: RErr in % for $\tau@99\%\text{TPR}$					
	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$
	seen	**unseen**	**unseen**	**unseen**	**unseen**
	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$
AT	56.0	88.4	99.4	99.5	73.6
CCAT	**39.1**	**53.1**	**29.0**	**31.7**	**3.5**

(Lower RErr means "better" robustness.)

Cifar10: Generalization to Unseen Attacks

CIFAR10: RErr in % for τ@99%TPR					
	L_∞ $\epsilon = 0.03$	L_∞ $\epsilon = 0.06$	L_2 $\epsilon = 2$	L_1 $\epsilon = 24$	L_0 $\epsilon = 10$
	seen	**unseen**	**unseen**	**unseen**	**unseen**
	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$	RErr $\downarrow$
AT	**62.7**	93.7	98.4	98.4	72.4
CCAT	67.9	92.0	**51.8**	**58.5**	**20.3**

(Lower RErr means "better" robustness.)

“Unconventional” Attacks

CIFAR10: $\tau@99\%$TPR			
	adv. frames	distal	corrupted
	unseen	**unseen**	**unseen**
	RErr $\downarrow$	FPR$\downarrow$	CErr$\downarrow$
Normal	96.6	83.3	12.3
AT	78.7	75.0	16.2
CCAT	**65.1**	**0**	**8.5**

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

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

Improved Accuracy

	SVHN: Err in %		CIFAR10: Err in %	
	$\tau = 0$	99% TPR	$\tau = 0$	99% TPR
Normal	3.6	2.6	**8.3**	7.4
AT	3.4	2.5	16.6	15.5
CCAT	**2.9**	**2.1**	10.1	**6.7**

(Err: test error before and after thresholding.)

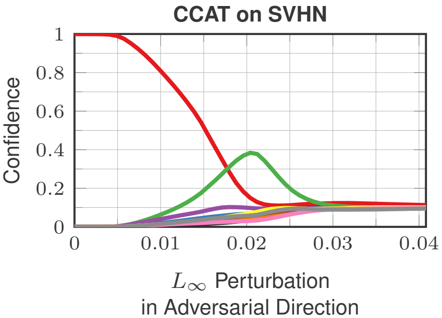
Part 5

Conclusion

Robustness Evaluation

Checklist for reviews and experiments:

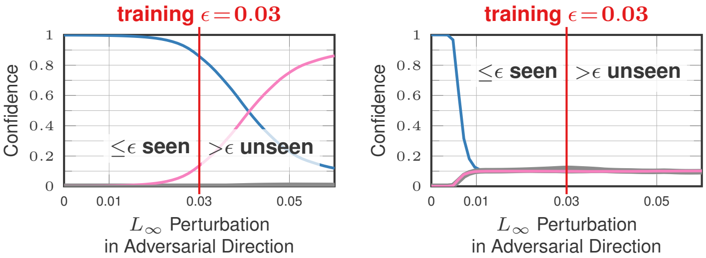
1. Reasonable metrics.
2. Multiple, adaptive attacks.
3. Worst-case evaluation.



Confidence-Calibrated Adversarial Training

Encourage low-confidence on adversarial examples:

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

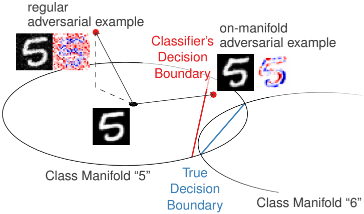


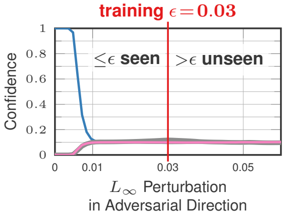
- Explicit guidance how to behave off-manifold.

Questions?

More: `davidstutz.de`

Code coming soon!





References:

- D. Stutz, M. Hein, B. Schiele. *Disentangling Adversarial Robustness and Generalization*. CVPR, 2019.
- D. Stutz, M. Hein, B. Schiele. *Confidence-Calibrated Adversarial Training: Generalizing to Unseen Attacks*. ArXiv, 2019.

Appendix

"Power" Transition

How to choose $\lambda \propto 1/\|\delta\|_\infty$ for:

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

"Power" transition:

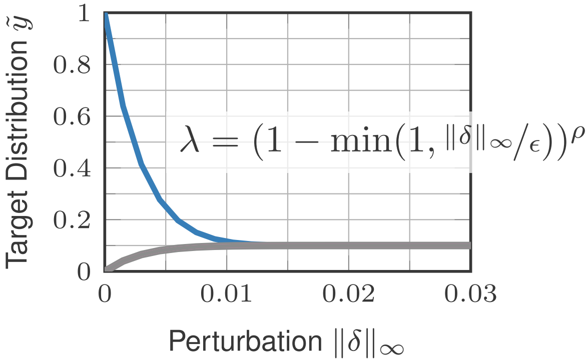
$$\lambda = 1 - (1 - \min(1, \frac{\|\delta\|_\infty}{\epsilon}))^\rho, \qquad \|\delta\|_\infty \leq \epsilon$$

- Nearly exponential in confidence;
- avoids a bias towards the true label.

"Power" Transition

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

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



“Exponential” Transition

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

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

Exponential transition:

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

- Exponential in confidence means linear in logits;
- keeps a bias towards the true label as $\lambda > 0$;
- ρ depends on ϵ, large ρ required;

“Standard” Robust Test Error

= error on test examples that are “attacked”:

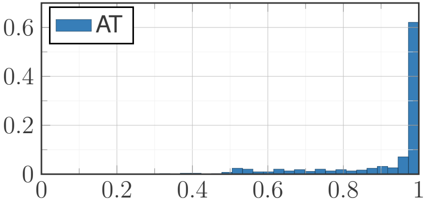
$$\text{“Standard” RErr} = \frac{1}{N}\sum_{n=1}^{N}\max_{\|\delta\|_p\leq\epsilon}\mathbb{1}_{f(x_n+\delta)\neq y_n}$$

- $\{(x_n, y_n)\}_{n=1}^N$ test examples.

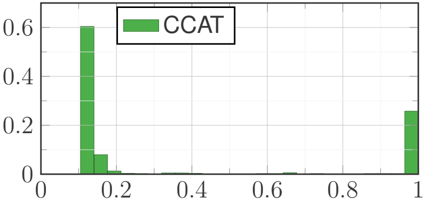
Considering the "Reject Option"

Adversarial Training (AT):
57.3% RErr

Ours (CCAT):
97.8% RErr



Confidence on *Adversarial* Examples



Confidence on *Adversarial* Examples

Confidence-Thresholded Robust Test Error

= error on test examples that are "attacked" *and* pass the confidence threshold τ:

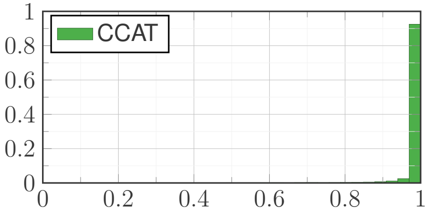
$$\text{RErr}(\tau) = \frac{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon, c(x_n+\delta) \geq \tau} \mathbb{1}_{f(x_n+\delta) \neq y_n}}{\sum_{n=1}^{N} \max_{\|\delta\|_p \leq \epsilon} \mathbb{1}_{c(x_n+\delta) \geq \tau}}$$

- $c(x_n) := \max_k f_k(x_n)$ confidence on x_n;

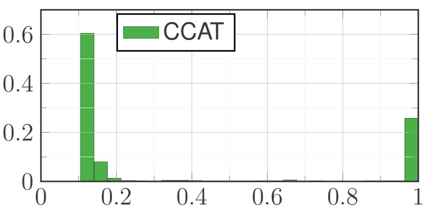
Special Cases

Why is the **confidence-thresholded RErr** non-trivial?

- Adversarial examples can have higher confidence than test examples.



Confidence on *Test* Examples



Confidence on *Adversarial* Examples

Implementation

Confidence-thresholded RErr is approximated using our PGD attack and:

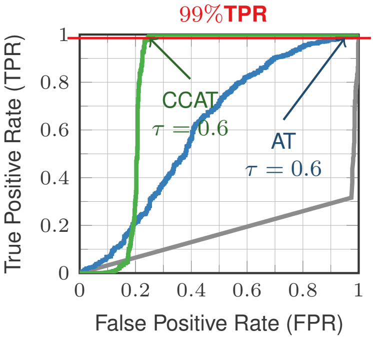
$$\frac{\sum_{n=1}^{N} \max\{\mathbb{1}_{f(x_n)\neq y_n}\mathbb{1}_{c(x_n)\geq\tau}, \mathbb{1}_{f(\tilde{x}_n)\neq y_n}\mathbb{1}_{c(\tilde{x}_n)\geq\tau}\}}{\sum_{n=1}^{N} \max\{\mathbb{1}_{c(x_n)\geq\tau}, \mathbb{1}_{c(\tilde{x}_n)\geq\tau}\}}$$

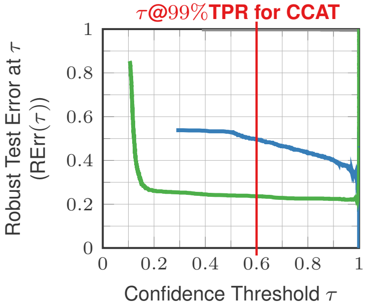
- $c(x_n) := \max_k f_k(x_n)$ confidence on x_n;

Confidence Threshold

Choosing confidence threshold τ:

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





Adversaries: Iterations and Initialization

Use plenty of iterations and zero initialization:

SVHN: RErr in % for L_∞ with $\epsilon = 0.03$						
Optimization	backtracking+momentum				–	
Iterations	40	200	1000	2000	60	300
AT	38.4	46.2	49.9	50.1	29.9	30.8
CCAT	4.0	5.0	22.8	23.3	2.6	2.6

(Higher RErr means “stronger” attack.)

Adversaries: Iterations and Initialization

Use plenty of iterations and zero initialization:

SVHN: RErr in % for L_∞ with $\epsilon = 0.03$						
Optimization	backtracking+momentum				–	
Iterations	40	200	1000	2000	60	300
AT	38.4	46.2	49.9	50.1	29.9	30.8
CCAT	4.0	5.0	22.8	23.3	2.6	2.6

(Higher RErr means "stronger" attack.)

- Our CCAT is (computationally) "harder" to attack.