Embedded
**Intelligence**

# GARD Task 1.4. - Phase I
# Preliminary Results

Embedded Intelligence - TA 1.1
January 5, 2021

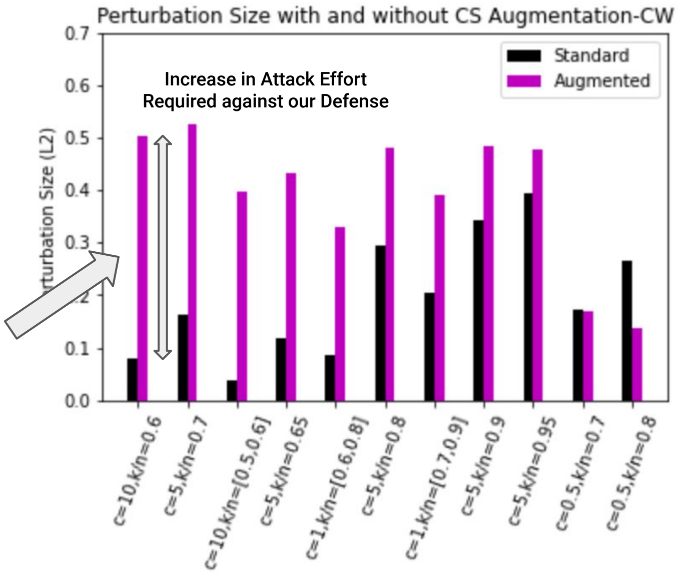
# Executive Summary

Embedded Intelligence

The Embedded Intelligence (EI) GARD research program is focused on creating a body of evidence to support or falsify the hypothesis that compressed sensing (CS) as a family of methodologies can be used as part of a layered Adversarial ML **defense** and **attack detection** system.

In Task 1.4., we show our proposed attack defense's effectiveness in:

- Increasing the risk for the attacker to reveal itself
  - We show we can force the attacker to use larger perturbations for a popular attack methodology
- Minimizing the overhead of implementing our CS-based defense
  - We show we can recover acceptable accuracy of the CS-defended classifiers under benign conditions, maintaining pressure on attacker



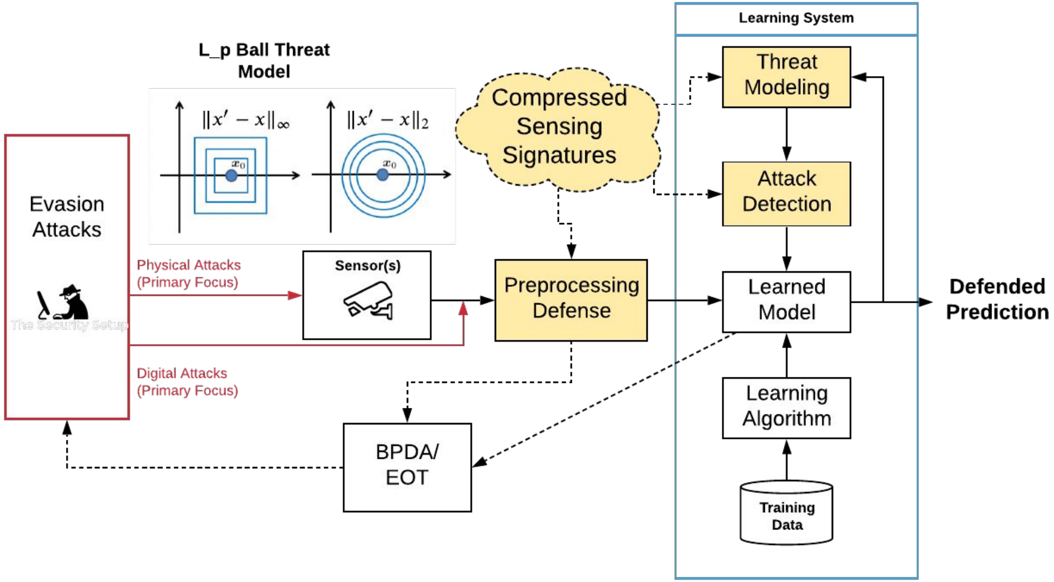
# Introduction

# Multi-component, Layered, Adversarial ML Defense

Embedded **Intelligence**

Our Phase 1 Focus

- Test time evasion attacks, white & black box settings
- Defending/detecting Lp ball and Patch threat models
- Focused on defending against ADAPTIVE attacks
- Developing Compressed Sensing as a useful family of defense methods

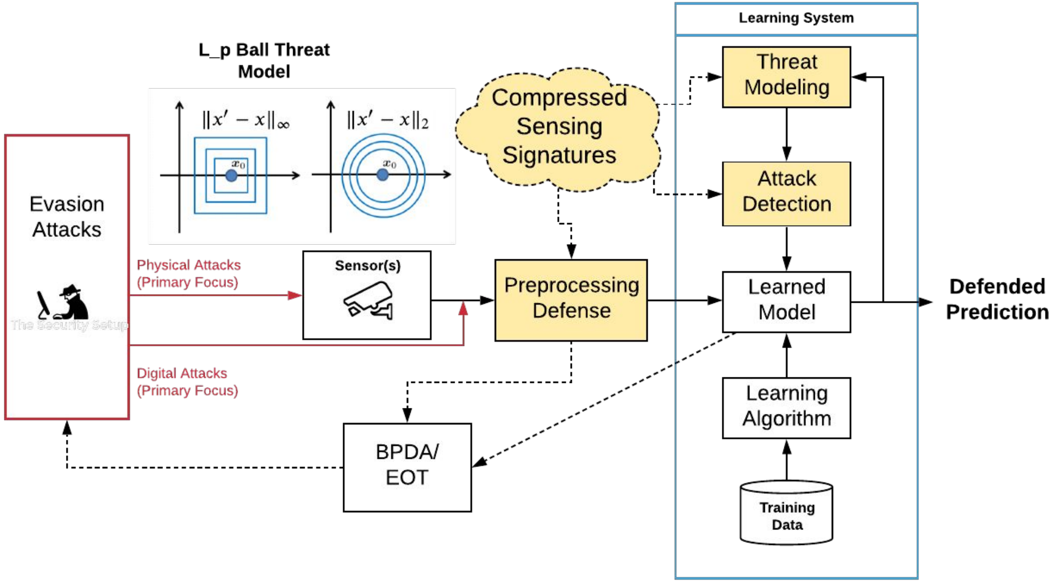


# Multi-component, Layered, Adversarial ML Defense

Embedded **Intelligence**

Assumptions about the world

- Attackers, given infinite resources, will always be successful
- Attackers do not want to be discovered / detected
- Larger perturbation attacks are generally more detectable and thus risky to the attacker

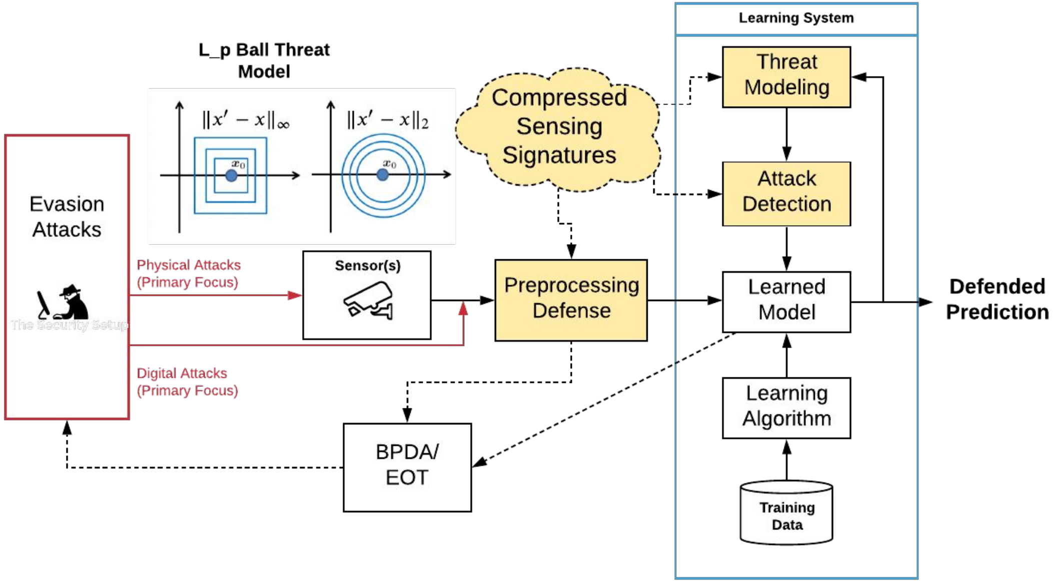


# Multi-component, Layered, Adversarial ML Defense

Embedded
Intelligence

**Economics Mindset**

- **Good multi-layered defense systems will make successful attacks “expensive”**
  - Computational cost
  - Human capital
  - Forcing attacker to use larger, more revealing perturbations




**In this report, we focus on whether**

1) **CS helps us force an attacker into using a larger perturbation**
2) **Acceptable system performance can be maintained under “non-attacked” conditions**

# Task 1.4. Goals

Embedded
**Intelligence**

| Task 1.4 Technical Goal | Task 1.4 Approach |
|---|---|
| Evaluate whether CS-based defense can force an attacker into using a larger perturbation | Contrary to conventional wisdom, we configure CS at the preprocessing layer to induce LARGE distortions on all incoming images. We also evaluate some types of stochasticity. |
| Evaluate whether acceptable system performance can be maintained in defended systems under “non-attacked” conditions | We use CS as a data augmentation step while training a network with the goal of recovering performance under benign conditions |

# Methods

Embedded
**Intelligence**

# Sophisticated Adversaries: Adaptive Attacks

Embedded Intelligence

Given white-box knowledge of a defended network and the defense methodologies implemented, attackers can *adapt* their attacks to render defended networks useless.

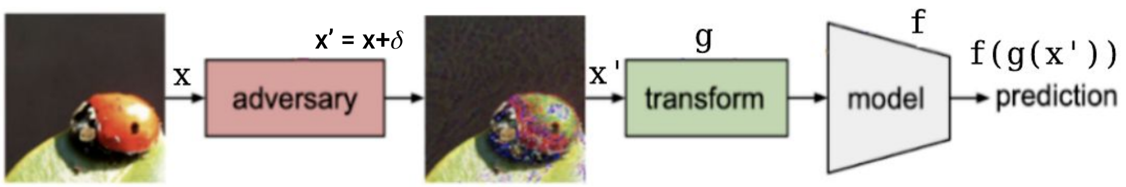
There are two very powerful and general methodologies for constructing adaptive attacks

1. Backward Pass Differentiable Approximation (BPDA)
    a. A strong counter to pre-processing based defenses using gradient shattering/masking
2. Expectation Over Transformation (EoT)
    a. A strong counter to defenses that impose a stochastic gradient

# Adaptive Attacks: BPDA

Embedded
Intelligence

**Testing:**




*General Case:*

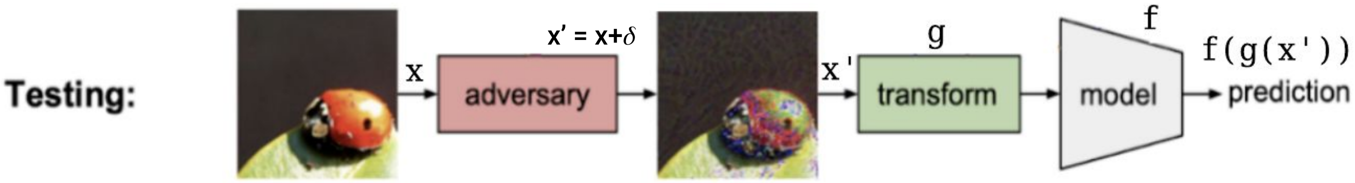
NN: $f(.) = f^{1 \ldots j}(.)$
$f^i(.)$ is non-differentiable $\longrightarrow$ find $g(x) \approx f^i(x)$ $\longrightarrow$ $\nabla_x f(x)$
forward pass: use $f^i(x)$
backward pass: replace $f^i(x)$ with $g(x)$

**A general & powerful adaptive attack**

**Lots of customization required, can be sophisticated to implement, requires significant Human Capital to pull this off**

# Adaptive Attacks: BPDA

Embedded
**Intelligence**




<u>*Special Case:*</u> *Assume that the defense "g" is implemented as a preprocessing layer, and that g(x) ≈ x* (e.g. g is a denoising method)

$$\text{classifier: } f(.) \longrightarrow \begin{array}{c}\text{construct } g(.)\\ \hat{f}(x) = f(g(x))\\ g(x) \approx x\end{array} \longrightarrow \begin{array}{c}\nabla_x g(x) \approx \nabla_x x = 1\\ \nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}\end{array}$$

**Easy to implement adaptive attack, and also generally applicable.**

**Much lower human capital required, as this version is implemented in common Adversarial ML software packages such as ART**

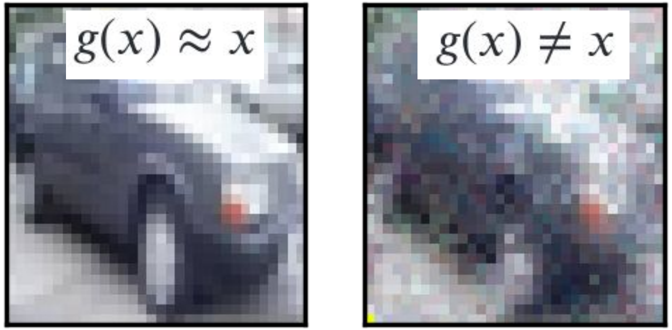
# Defending using Compressed Sensing

Embedded
**Intelligence**

Since the "Special Case" of BPDA is the most widely implemented, we will heavily distort x using CS as a testable defense strategy

**Is "Special Case" BPDA strong even when g(x) is not approximately x?**

classifier: $f(.)$ $\longrightarrow$ construct $g(.)$, $\hat{f}(x) = f(g(x))$, $g(x) \approx x$ $\longrightarrow$ $\nabla_x g(x) \approx \nabla_x x = 1$, $\nabla_x f(g(x))|_{x=\hat{x}} \approx \nabla_x f(x)|_{x=g(\hat{x})}$



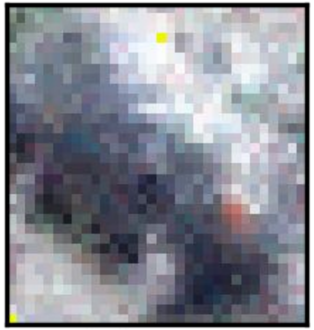
# Like CAPTCHA, except for ROBOTS

Embedded
**Intelligence**

**CAPTCHA for Cybersecurity**


Easy for human, Difficult for bot

**CS-based Adversarial ML Defense**



$g(x) \neq x$

**Easy for my bot, Hard for your bot**

# Compressed Sensing: A family of methodologies

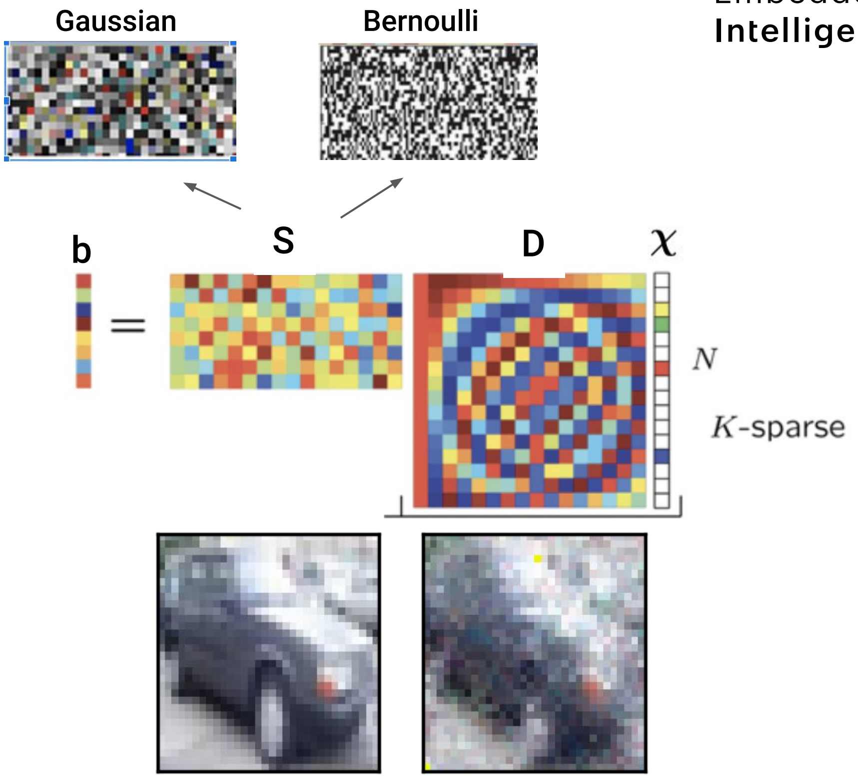
Embedded Intelligence

Compressed sensing has four main parameters that lead to transformations with different properties:

- k/n ratio: the ratio of the pixels used for image reconstruction
- Regularization parameter (c): sparsity level
- Random sensing matrix (S)
- Dictionary for representation (D)

CS parameters determine the level of distortion in the images (e.g. MSE).

CS parameters also determine the “amount” of stochasticity present in the preprocessing defenses.



# Compressed Sensing: A family of methodologies

Embedded **Intelligence**

$$\chi_{cs} = \{\chi \in \mathbb{R}^n | \min_\chi \|b - SD\chi\|_2 + c\|\chi\|_1\}$$

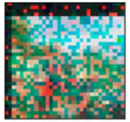
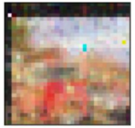
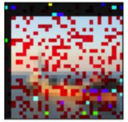
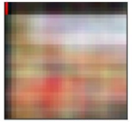
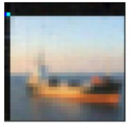
$$g(x) = x_{cs} = D\chi_{cs}$$

| | **Compressed Sensing Configuration (S,D,k/n,c)** | **Why?** |
|---|---|---|
| **Attack Detection** | Choose (S, D, k/n, c) s.t. $g(x+\boldsymbol{\delta}) \approx x$ | CS used as an estimator of $\boldsymbol{\delta}$ |
| **Defense** | Choose (S, D, k/n, c) s.t. $g(x+\boldsymbol{\delta}) \not\approx x$ | Adaptive attacks using BPDA assume $g(x) \approx x$, CS is used to confound the attacker |

# Configuring CS Parameters for Defense

Embedded
**Intelligence**

The below table demonstrates the examples of reconstructed images with different values of c and k/n. We can clearly observe that the choice of c and k/n determines the quality and distortion properties of the reconstructed images.

| | c=0.1 | | c=10 | | c=100 | |
|---|---|---|---|---|---|---|
| **k/n=0.5** |  mse 5.83552e-03 |  mse 8.33560e-02 |  mse 1.04438e-02 |  mse 7.31257e-03 |  mse 1.95172e-02 |  mse 1.74300e-02 |
| **k/n=0.7** |  mse 7.32935e-02 |  mse 3.84073e-03 |  mse 3.49069e-03 |  mse 2.40328e-03 |  mse 1.41184e-02 |  mse 1.27293e-02 |
| **k/n=0.9** |  mse 3.09770e-02 |  mse 4.51124e-04 |  mse 1.20807e-03 |  mse 7.01798e-04 |  mse 1.25262e-02 |  mse 9.10196e-03 |

**Original**

# Configuring CS Parameters for Defense

Embedded Intelligence

With low values of the sparsity penalty/parameter c, CS-based reconstructions over-fit the pixel mask used for the random sensing matrix. This causes unwanted artifacts.

# Configuring CS Parameters for Defense

Embedded
Intelligence

A good choice for (c, k/n) should have the following properties

- Leads to a high MSE value, applying stress to the BPDA assumption that g(x) ≈ x
- Maintains a recognizable structure of the original image (analogous to CAPTCHA)

| | c=0.1 | | c=10 | | c=100 | |
|---|---|---|---|---|---|---|
| k/n=0.5 | mse 5.83552e-03 | mse 8.33560e-02 | mse 1.04438e-02 | mse 7.31257e-03 | mse 1.95172e-02 | mse 1.74300e-02 |
| k/n=0.7 | mse 7.32935e-02 | mse 3.84073e-03 | mse 3.49069e-03 | mse 2.40328e-03 | mse 1.41184e-02 | mse 1.27293e-02 |
| k/n=0.9 | mse 3.09770e-02 | mse 4.51124e-04 | mse 1.20807e-03 | mse 7.01798e-04 | mse 1.25262e-02 | mse 9.10196e-03 |

Original

# Experimental Setup

Embedded Intelligence

## Task 1.4 Study of Compressed Sensing Defenses

- **Neural network training approach**
  - Standard training
  - “Learning the CAPTCHA”
    - Training with CS defense used as a data augmentation step
- **Datasets**
  - CIFAR-10
- **Threat Models**
  - Projected Gradient Descent (PGD) ($L^2$ / $L^{inf}$) *
  - Carlini-Wagner (C-W) ($L^2$ / $L^{inf}$) **
- **Defenses**
  - **Baseline Defenses for Comparison JPEG and Total Variance Minimization**
  - **Compressed Sensing Configurations**
    - **k/n ratio: the ratio of the pixels used for image reconstruction**
      - **Statically set between [0.5, 0.99]**
      - **Stochastically chosen from a predetermined set of values**
    - **Regularization parameter (c): sparsity level**
      - **Studied across 10 orders of magnitude**
    - **Random sensing matrix (S)**
      - **Fixed as random pixel mask**
    - **Dictionary for representation (D)**
      - **Fixed as Discrete Cosine Transform**

# Results

Embedded
**Intelligence**

# Summary of Results

Embedded Intelligence

- Compressed Sensing successfully imposes a “cost” on attackers
  - We observe an improved ability to defend against Carlini-Wagner (CW) adaptive attacks as compared to baseline methodologies
  - We observe an increased perturbation size by CW attacks when our defense is in place.
- “Learning the CAPTCHA” model training leads to superior defense
  - Training the neural network using CS as a data augmentation step rescues impaired accuracy inflicted by our CS-defense distortions to the input.
  - Further improvements to defense under attack scenarios are shown with CS data augmentation training of the model

# Baseline Methodologies for Comparison

Embedded **Intelligence**

## Baseline Methodologies

- JPEG Compression*
  - Parameter: Quality
- Total Variation Minimization*
  - Parameters:
    - k/n
    - Regularization parameter (c)

## Our Approach

- Compressed Sensing
  - Parameters:
    - k/n
    - Regularization parameter (c)

# Comparison with Baseline Defenses: Highlights

Embedded Intelligence

## Across JPEG, TVM, and our proposed CS defense, larger distortions are associated with better defense against CW adaptive attacks

All the CW attacks were performed in Adversarial Robustness Toolbox (ART)* with initial_const=0.01, max_iter=10, batch_size=64, learning_rate=0.01

| | k/n | Regularization Parameter | Reconstruction MSE | Average C-W Adaptive Attack Success |
|---|---|---|---|---|
| **JPEG Compression** | 0.99 (resolution) | - | 0.0001 | **100%** |
| **Total Variation Minimization** | 0.99 | 0.5(default) | 0.0035 | 80.65% ± 3.53 |
| | 0.99 | 0.03 (Athalye 2018) | 1.228e-39 | **100%** |
| **Compressed Sensing** | 0.99 | 5 (default) | 0.0015 | 77.33% ± 5.64 |

# Comparison with Baseline Defenses: Highlights

Embedded Intelligence

## Across JPEG, TVM, and our proposed CS defense, larger distortions are associated with better defense against CW adaptive attacks

All the CW attacks were performed in ART with

- initial_const=0.01, max_iter=10, batch_size=64, learning_rate=0.01

| | k/n | Regularization Parameter | Reconstruction MSE | Average C-W Adaptive Attack Success |
|---|---|---|---|---|
| **JPEG Compression** | 0.99 (resolution) | - | 0.0001 | **100%** |
| **Total Variation Minimization** | 0.99 | 0.5(default) | 0.0035 | 80.65% ± 3.53 |
| | 0.99 | 0.03 (Athalye 2018) | 1.228e-39 | **100%** |
| **Compressed Sensing** | 0.99 | 5 (default) | 0.0015 | 77.33% ± 5.64 |

**All defenses configured for g(x) ≈ x to confirm BPDA adaptive attack matches results reported in literature**

# Comparison with Baseline Defenses: Highlights

Embedded
Intelligence

## Across JPEG, TVM, and our proposed CS defense, larger distortions are associated with better defense against CW adaptive attacks

All the CW attacks were performed in ART with

- initial_const=0.01, max_iter=10, batch_size=64, learning_rate=0.01

| | k/n | Regularization Parameter | Reconstruction MSE | Average C-W Adaptive Attack Success |
|---|---|---|---|---|
| **JPEG Compression** | 0.99 (resolution) | - | 0.0001 | **100%** |
| **Total Variation Minimization** | 0.99 | 0.5(default) | 0.0035 | 80.65% ± 3.53 |
| | 0.99 | 0.03 (Athalye 2018) | 1.228e-39 | **100%** |
| **Compressed Sensing** | 0.99 | 5 (default) | 0.0015 | 77.33% ± 5.64 |

**All defenses configured for g(x) ≈ x to confirm BPDA adaptive attack matches results reported in Athalye, 2018**

# Comparison with Baseline Defenses: Highlights

Embedded Intelligence

## Across JPEG, TVM, and our proposed CS defense, larger distortions are associated with better defense against CW adaptive attacks

All the CW attacks were performed in ART with

- initial_const=0.01, max_iter=10, batch_size=64, learning_rate=0.01

| | k/n | Regularization Parameter | Reconstruction MSE | Average C-W Adaptive Attack Success |
|---|---|---|---|---|
| **JPEG Compression** | 0.99 (resolution) | - | 0.0001 | **100%** |
| **Total Variation Minimization** | 0.99 | 0.5(default) | 0.0035 | 80.65% ± 3.53 |
| | 0.99 | 0.03 (Athalye 2018) | 1.228e-39 | **100%** |
| **Compressed Sensing** | 0.99 | 5 (default) | 0.0015 | 77.33% ± 5.64 |

**Higher distortion (MSE) configurations show possible weakness in CW adaptive attacks (new result)**

# Comparison with Baseline Defenses: Full results

Embedded Intelligence

| | k/n | Regularization Parameter | Average Reconstruction MSE | Average C-W Adaptive Attack Success |
|---|---|---|---|---|
| **JPEG Compression** | 0.5 (default resolution) | - | 0.0015 | 95.08% |
| | 0.99 (resolution) | - | 0.0001 | 100% |
| **Total Variation Minimization** | 0.5 | 0.5(default) | 0.0043 | 67.75% ± 2.89 |
| | 0.99 | 0.5(default) | 0.0035 | 80.65% ± 3.53 |
| | 0.5 | 0.03(Athalye 2018) | 4.945e-41 | 100% |
| | 0.99 | 0.03 (Athalye 2018) | 1.228e-39 | 100% |
| **Compressed Sensing** | 0.5 | 5 (default) | 0.0085 | 61.31%±4.93 |
| | 0.99 | 5 (default) | 0.0015 | 77.33% ± 5.64 |
| | 0.5 | 10 (grid search) | 0.0087 | 52.58% ± 4.96 |
| | 0.99 | 10 (grid search) | 0.0027 | 75.16% ± 2.19 |

Largest distortion shows most promise for imposing cost on adaptive CW attacks

# Compressed Sensing Parameter Tuning and Data Augmentation

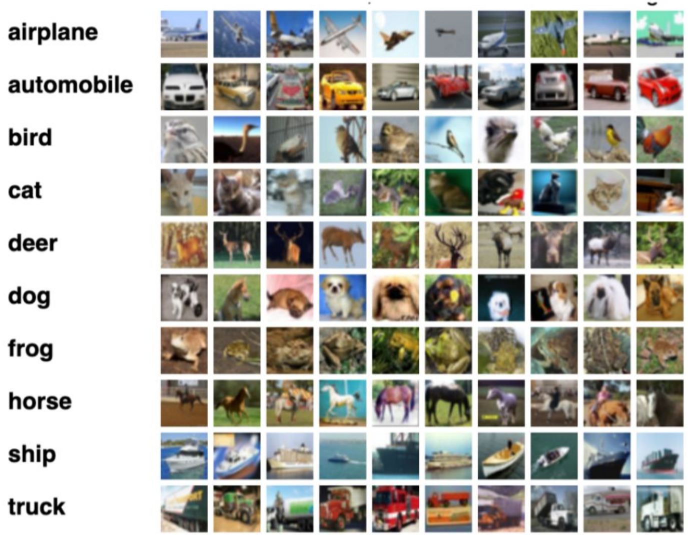
Embedded **Intelligence**

As MSE increases, the BPDA adaptive attack assumption of g(x) ≈ x breaks down, and attack success deteriorates

As we saw previously, different values of (c, k/n) distort the image in different ways

Optimal (c, k/n) values (with and without data augmentation) are **data dependent**.

Next: we performed a grid search to tune the CS defense for CIFAR-10.

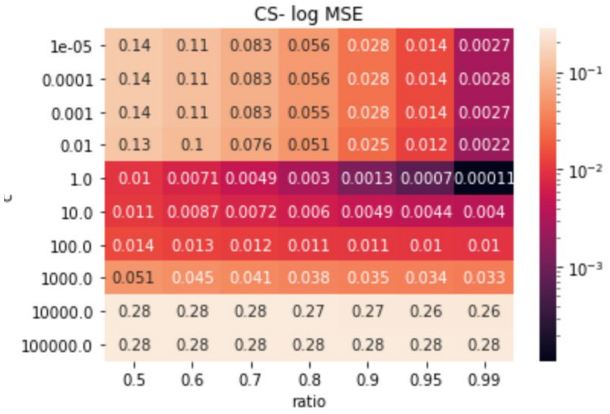


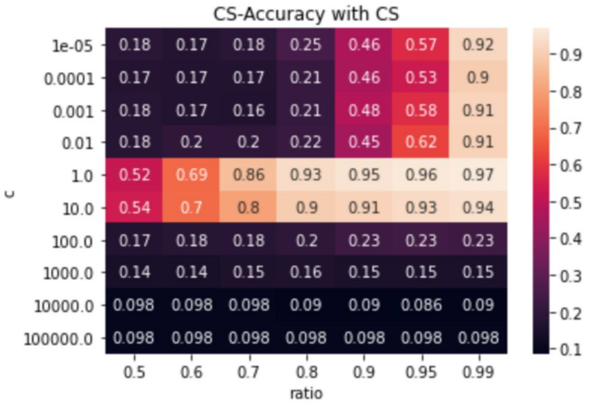

CIFAR-10 Dataset
Krizhevsky, A. and Hinton, G., 2009.
Learning multiple layers of features from tiny images.

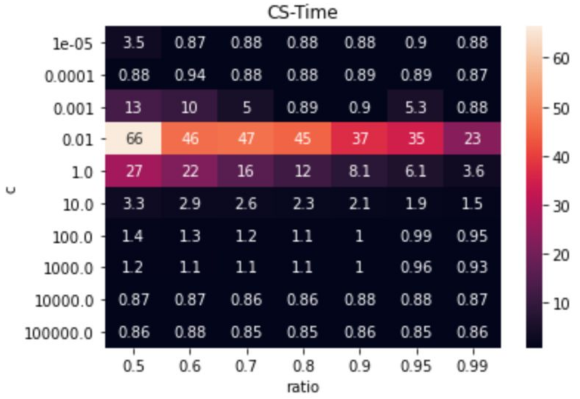
# CS  Parameter Tuning (no data augmentation)

Embedded
**Intelligence**

## Standard Training



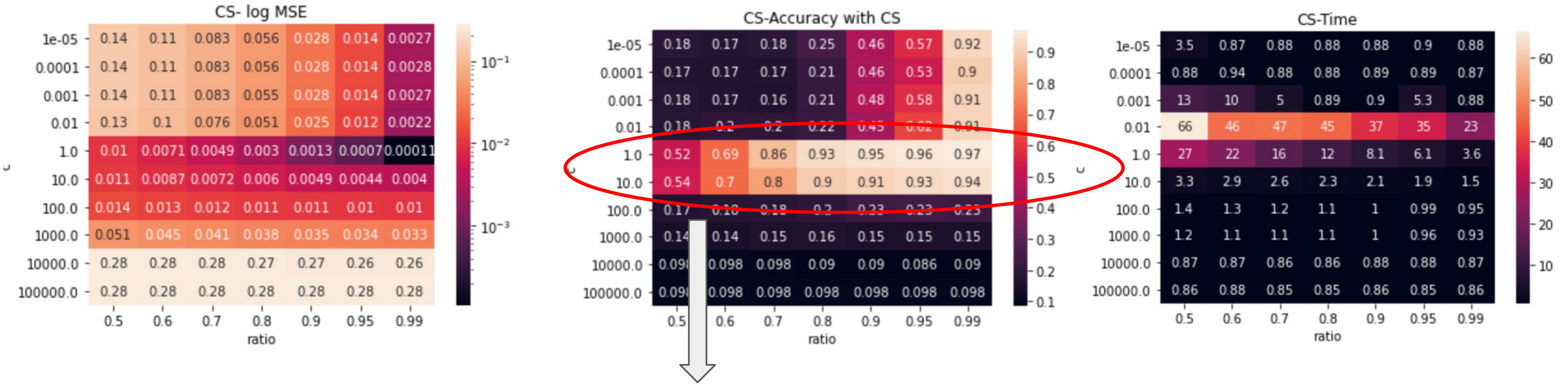



# CS Parameter Tuning (no data augmentation)

Embedded
**Intelligence**

Standard Training

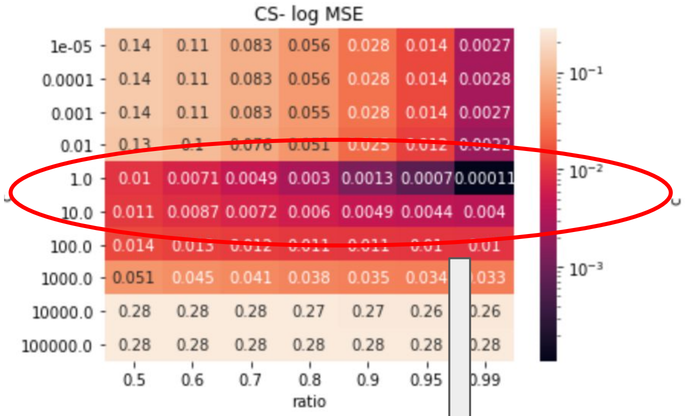


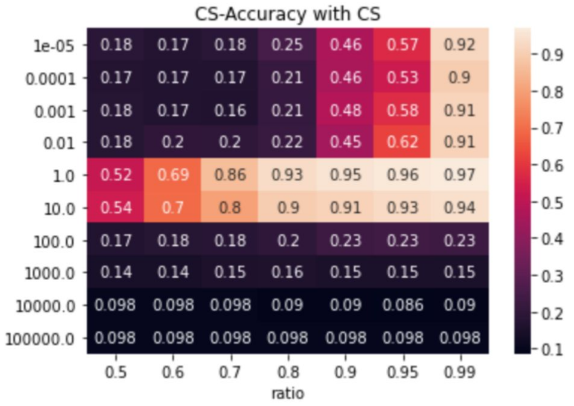

**Feasible choices of (c, k/n) based on baseline, unattacked, accuracies**

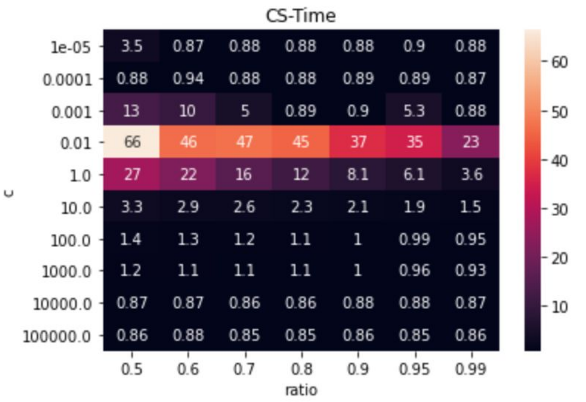
# CS Parameter Tuning (no data augmentation)



Standard Training



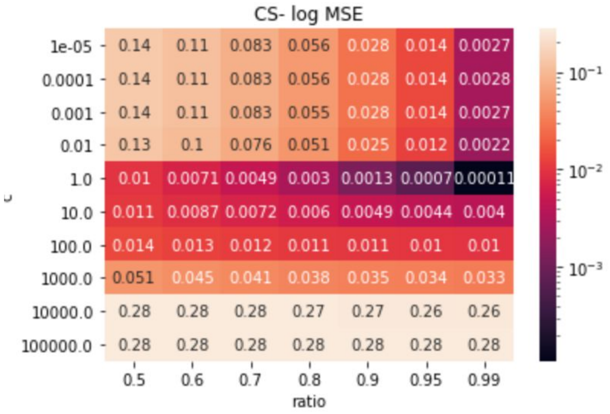



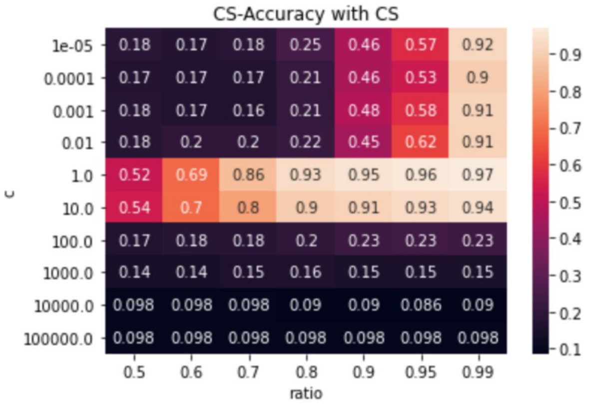

**Average MSE values of reconstructed images span 2 orders of magnitude in this region, offering flexibility for challenging BPDA-based adaptive attacks**

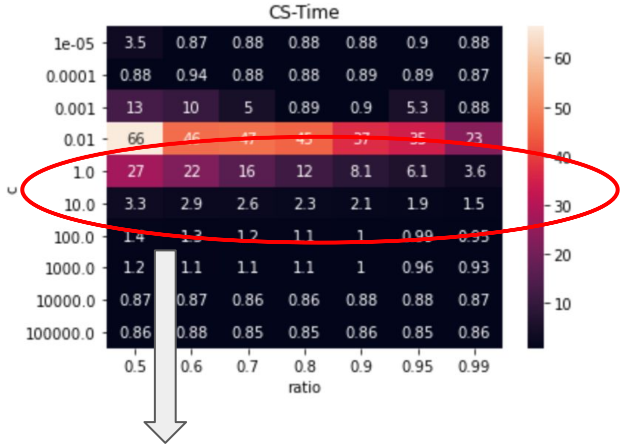
# CS Parameter Tuning (no data augmentation)

Embedded
Intelligence

## Standard Training



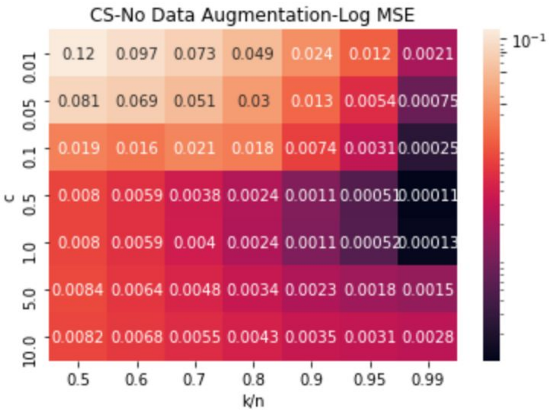

**Runtimes varies, which have implications on the use of CS in data augmentation at training time depending on (c, k/n)**

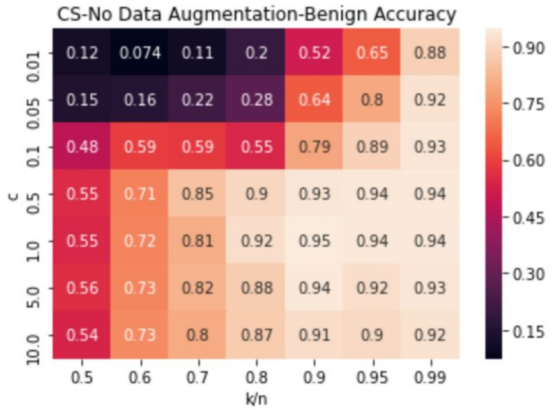
# CS  Parameter Tuning (no data augmentation)

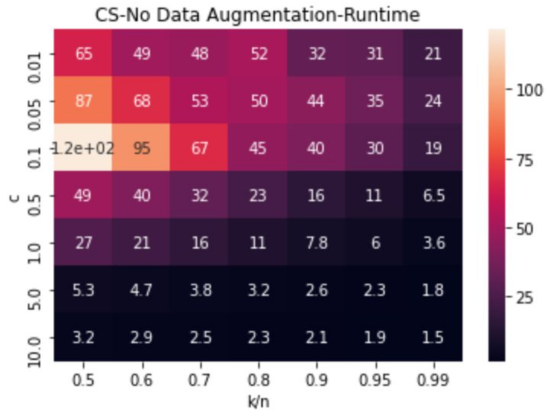
Embedded
**Intelligence**

## Key findings from grid search:

- Low accuracy when c<0.6
- Lower MSE/ higher accuracy for c∈[1,10]
- High runtime for c<1
- **c=10: lowest runtime, higher MSE and reasonable accuracy, promising for CS defense**

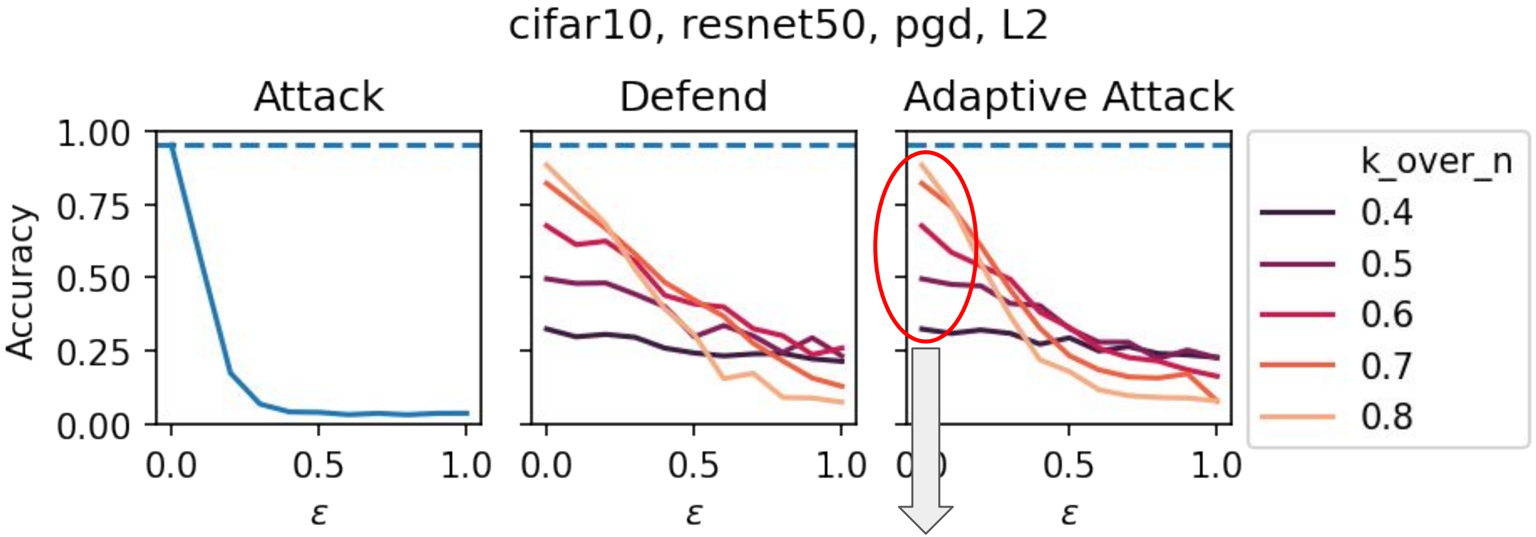


# CS Parameter Tuning (no data augmentation)

Embedded Intelligence



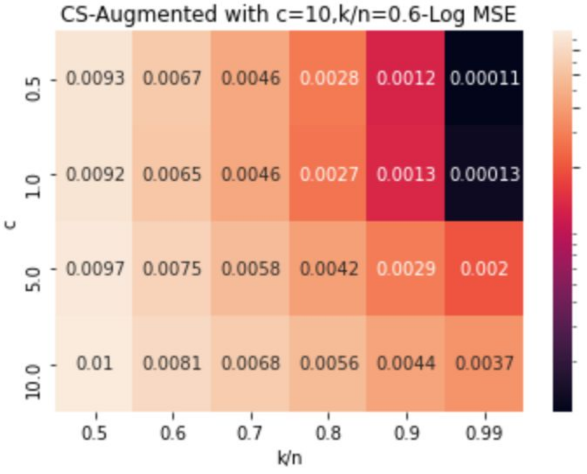

CS impedes attacker, but unattacked system performance unacceptably degrades with the defense in place

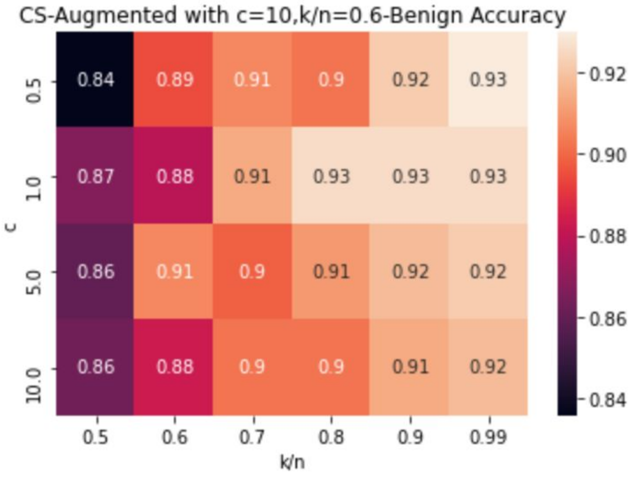
# Compressed Sensing with Data Augmentation Training

Embedded
Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired accuracy caused by the distortions the CS defense uses to mitigate the attack.**

Shown are heat maps from grid search for a model trained with c=10, k/n=0.6.



# Compressed Sensing with Data Augmentation Training

Embedded Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired benign accuracy inflicted by our CS defense.** The below table shows the benign accuracy with CS-data augmentation (c=10). CIFAR10-ResNet50

| c=10 | k/n=0.5 | k/n=0.6 | k/n=0.7 | k/n=0.8 | k/n=0.9 |
|---|---|---|---|---|---|
| **Standard (No Aug)** | 0.51 | 0.69 | 0.80 | 0.86 | 0.89 |
| **Aug with k/n=0.5** | 0.85 | | | | |
| **Aug with k/n=0.6** | | 0.86 | | | |
| **Aug with k/n=0.7** | | | 0.91 | | |
| **Aug with k/n=0.8** | | | | 0.91 | |
| **Aug with k/n=0.9** | | | | | 0.94 |
| **Aug with Stochastic k/n=[0.5,0.6]** | | | | | |

# Compressed Sensing with Data Augmentation Training

Embedded Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired benign accuracy inflicted by our CS defense.** The below table shows the benign accuracy with CS-data augmentation (c=10). CIFAR10-ResNet50

| c=10 | k/n=0.5 | k/n=0.6 | k/n=0.7 | k/n=0.8 | k/n=0.9 |
|---|---|---|---|---|---|
| **Standard (No Aug)** | 0.51 | 0.69 | 0.80 | 0.86 | 0.89 |
| **Aug with k/n=0.5** | 0.85 | | | | |
| **Aug with k/n=0.6** | | | | | |
| **Aug with k/n=0.7** | | | 0.91 | | |
| **Aug with k/n=0.8** | | | | 0.91 | |
| **Aug with k/n=0.9** | | | | | 0.94 |
| **Aug with Stochastic k/n=[0.5,0.6]** | | | | | |

Unacceptably low unattacked accuracy $g(x) \neq x$ in this region

(c, k/n) for which $g(x) \approx x$

# Compressed Sensing with Data Augmentation Training

Embedded Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired benign accuracy inflicted by our CS defense.** The below table shows the benign accuracy with CS-data augmentation (c=10). CIFAR10-ResNet50

| **c=10** | **k/n=0.5** | **k/n=0.6** | **k/n=0.7** | **k/n=0.8** | **k/n=0.9** |
|---|---|---|---|---|---|
| **Standard (No Aug)** | 0.51 | 0.69 | 0.80 | 0.86 | 0.89 |
| **Aug with k/n=0.5** | 0.85 | | | | |
| **Aug with k/n=0.6** | | 0.86 | | | |
| **Aug with k/n=0.7** | | | 0.91 | | |
| **Aug with k/n=0.8** | | | | 0.91 | |
| **Aug with k/n=0.9** | | | | | 0.94 |
| **Aug with Stochastic k/n=[0.5,0.6]** | | | | | |

**By training ResNet50 using our CS-defense as a data augmentation step (rather than without), significant performance recovery under unattacked conditions is observed (Including the $g(x) \neq x$ region)**

# Compressed Sensing with Data Augmentation Training

Embedded Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired benign accuracy inflicted by our CS defense.** The below table shows the benign accuracy with CS-data augmentation (c=10). CIFAR10-ResNet50

| c=10 | k/n=0.5 | k/n=0.6 | k/n=0.7 | k/n=0.8 | k/n=0.9 |
|---|---|---|---|---|---|
| **Standard (No Aug)** | 0.51 | 0.69 | 0.80 | 0.86 | 0.89 |
| **Aug with k/n=0.5** | 0.85 | 0.87 | 0.89 | 0.90 | 0.91 |
| **Aug with k/n=0.6** | 0.83 | 0.86 | 0.89 | 0.91 | 0.91 |
| **Aug with k/n=0.7** | 0.83 | 0.87 | 0.91 | 0.91 | 0.92 |
| **Aug with k/n=0.8** | 0.81 | 0.86 | 0.89 | | |
| **Aug with k/n=0.9** | 0.73 | 0.84 | 0.89 | | |
| **Aug with Stochastic k/n=[0.5,0.6]** | | | | | |

**Training with data augmentation using the "wrong" k/n value still enables some recovery of unattacked classification.**

**This implies a CS-defense (k/n) could be reconfigured in real-time without retraining**

# Compressed Sensing with Data Augmentation Training

Embedded Intelligence

Training with CS as part of data augmentation pipeline **rescues impaired benign accuracy inflicted by our CS defense.** The below table shows the benign accuracy with CS-data augmentation (c=10). CIFAR10-ResNet50

| c=10 | k/n=0.5 | k/n=0.6 | k/n=0.7 | k/n=0.8 | k/n=0.9 |
|---|---|---|---|---|---|
| **Standard** **(No Aug)** | 0.51 | 0.69 | 0.80 | 0.86 | 0.89 |
| **Aug with k/**[illegible] | [illegible] | [illegible] | [illegible]89 | 0.90 | 0.91 |
| **Aug with k/**[illegible] | [illegible] | [illegible] | [illegible]89 | 0.91 | 0.91 |
| **Aug with k/**[illegible] | [illegible] | [illegible] | [illegible]91 | 0.91 | 0.92 |
| **Aug with k/**[illegible] | [illegible] | [illegible] | [illegible]89 | 0.91 | 0.91 |
| **Aug with k/n=0.9** | 0.73 | 0.84 | 0.89 | 0.93 | 0.94 |
| **Aug with Stochastic k/n=[0.5,0.6]** | 0.87 | 0.88 | 0.90 | 0.91 | 0.90 |

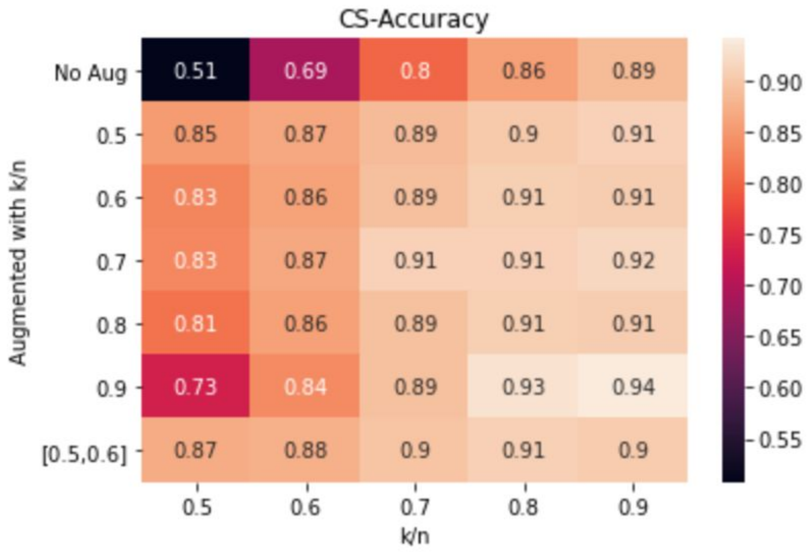
Training data augmentation using the stochastically selected k/n values shows impressive recovery of unattacked accuracy

This implies a CS-defense (k/n) could be reconfigured in real-time without retraining

# Compressed Sensing with Data Augmentation Training

Embedded
**Intelligence**

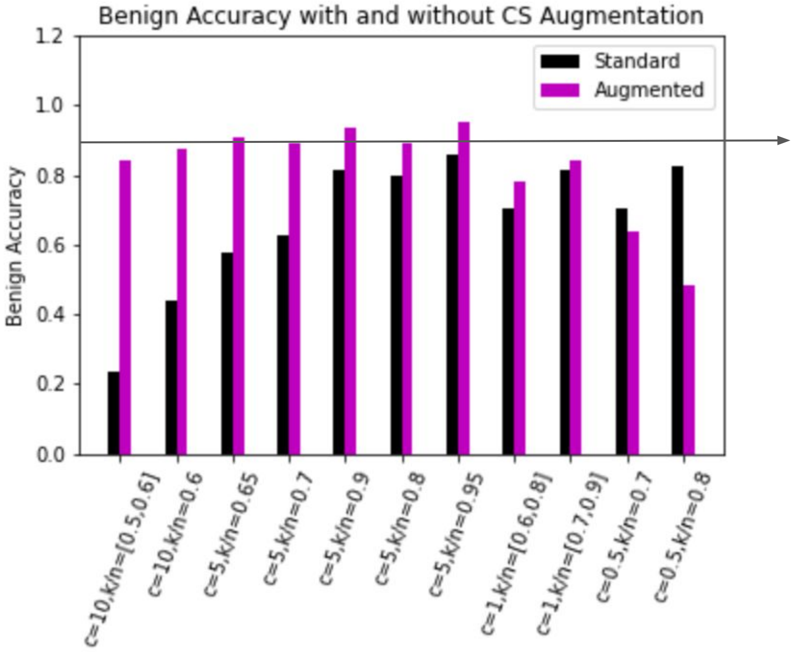
Training with CS as part of data augmentation pipeline rescues impaired benign accuracy inflicted by our CS defense. The below heatmap shows the benign accuracy with CS-data augmentation (c=10). CIFAR-10-ResNet50



# Internal Evaluation using Armory: Benign Accuracy

Embedded **Intelligence**

CS-data augmentation remarkably increases the benign accuracy of the neural networks. The effect is more significant when trained with c=10. This is aligned with our findings in Slide 33: Compressed Sensing Parameter Tuning (without Data Augmentation)



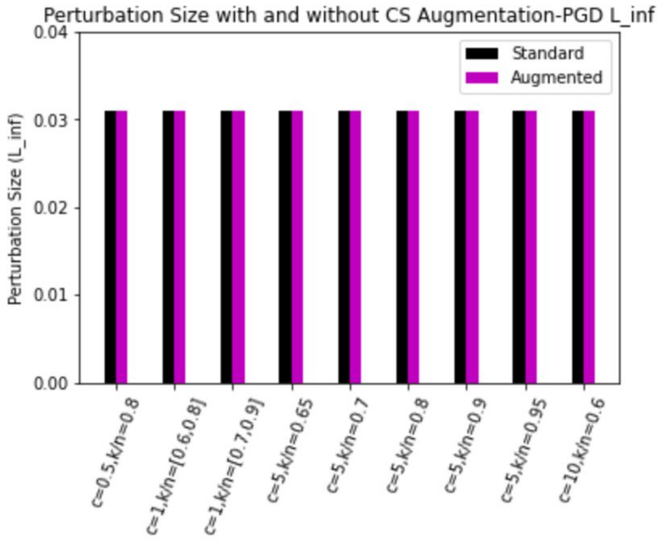

**90% accuracy threshold for acceptable benign/unattacked performance**

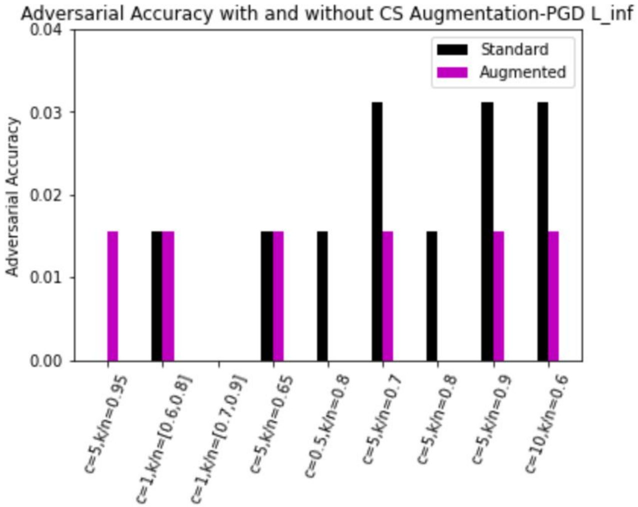
[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# PGD $L^{inf}$-Metric $L^{inf}$ - Eps=0.031 (Adaptive/BPDA)

Embedded Intelligence

This is where we left off in Eval 1. Note: Eps=0.31 is a VERY LARGE perturbation with respect to the infinity norm.



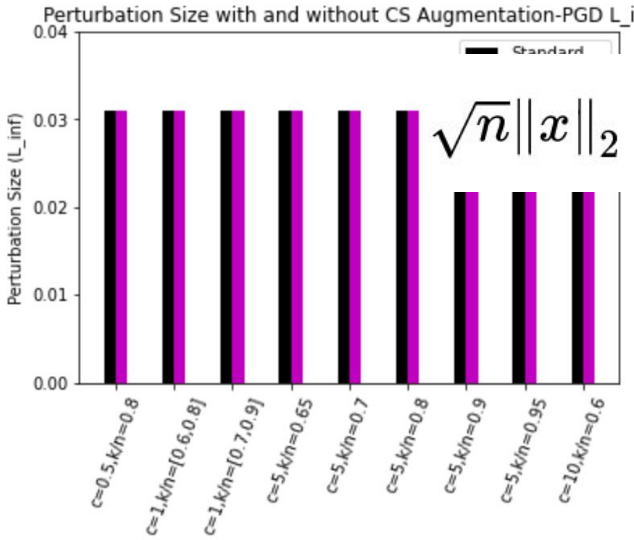




[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

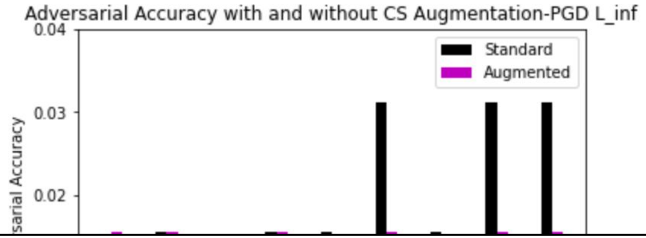
# PGD $L^{inf}$-Metric $L^{inf}$ - Eps=0.031 (Adaptive/BPDA)

Embedded
Intelligence

This is where we left off in Eval 1. Note: Eps=0.31 is a VERY LARGE perturbation with respect to the infinity norm.




$$\sqrt{n}\|x\|_2 \leq n\|x\|_\infty$$




From the above inequality relating L2 and Linf norms, and considering the dimension of the space n=32*32*3 for CIFAR-10, we can see that an Linf norm of 0.31 can be as large as a L2 norm of 9.7.

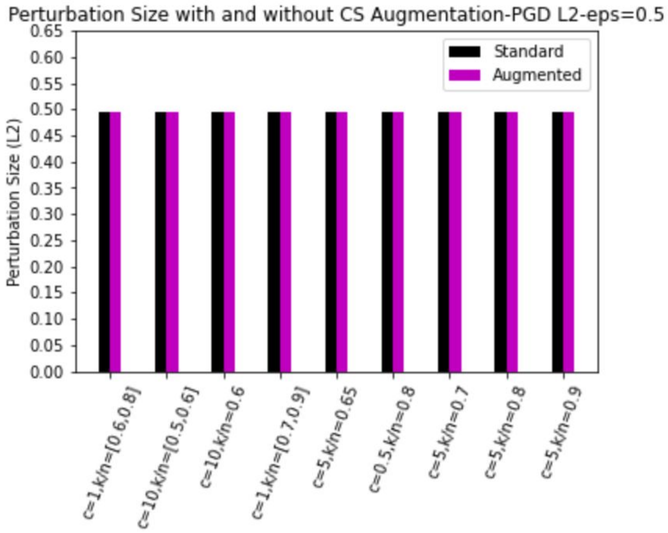
This is a very large perturbation.

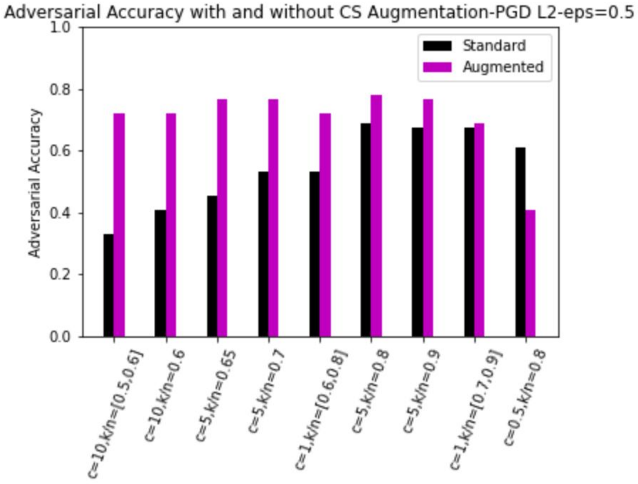
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# PGD $L^2$-Metric $L^2$ - Epsilon=0.5 (Large L2 Attack)

Embedded
**Intelligence**

We observe substantial increase in the adversarial accuracy when trained with c=10.



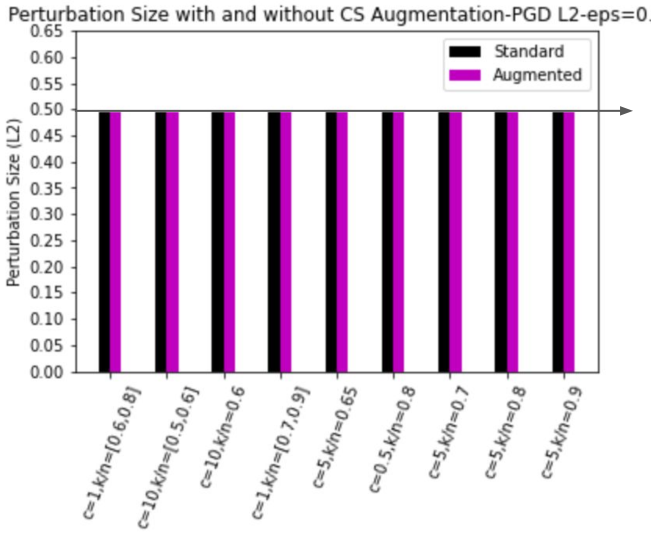




[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

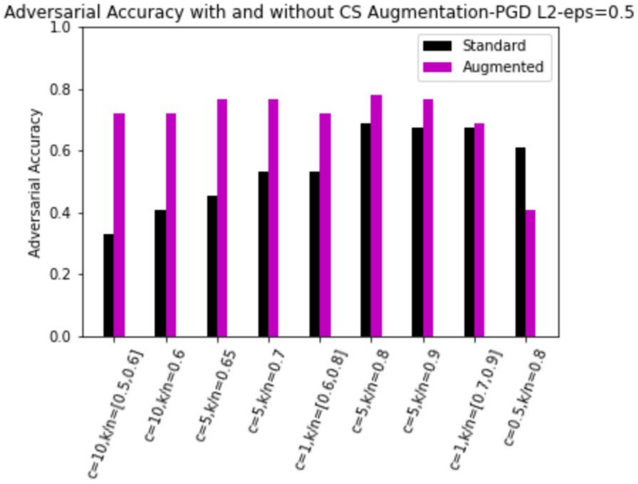
# PGD $L^2$-Metric $L^2$ - Epsilon=0.5 (Large L2 Attack)

Embedded **Intelligence**

We observe substantial increase in the adversarial accuracy when trained with c=10.




**PGD uses the entire perturbation budget.**

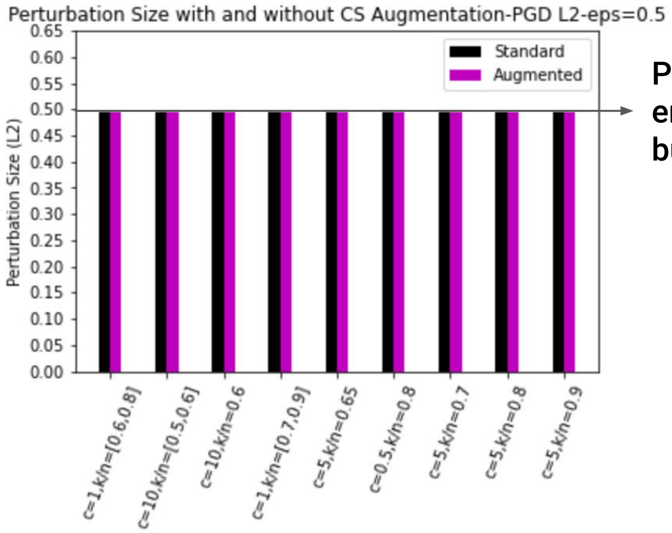



[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# PGD $L^2$-Metric $L^2$ - Epsilon=0.5 (Adaptive/BPDA)

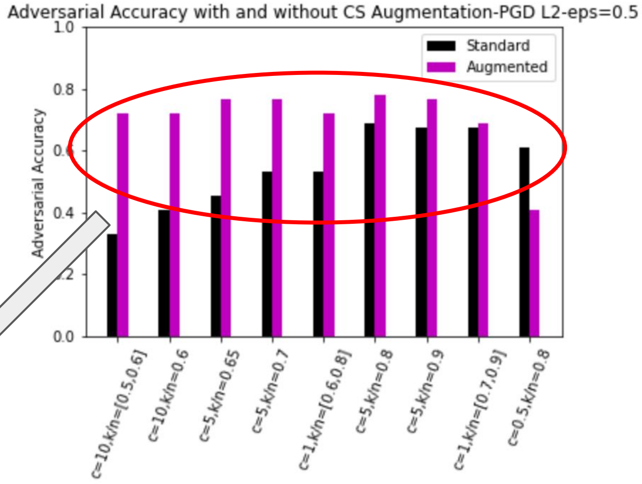
Embedded
**Intelligence**

We observe substantial increase in the adversarial accuracy when trained with c=10.




**PGD uses the entire perturbation budget.**

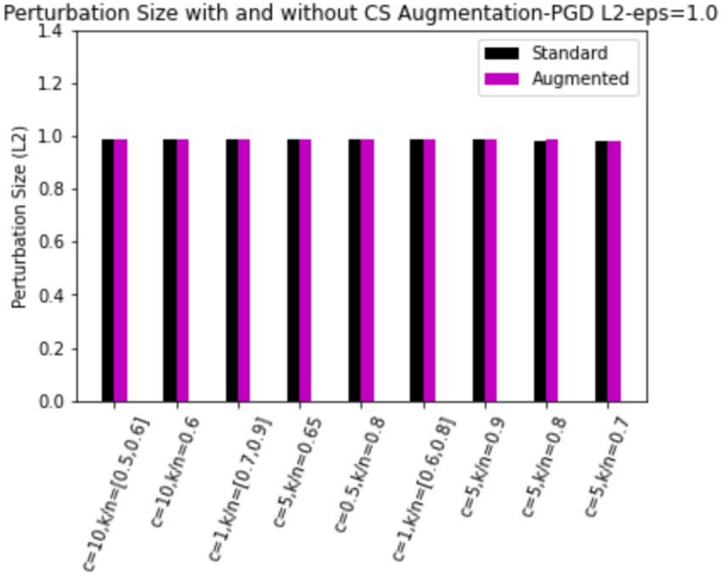
**Very significant recoveries of model performance while under attack**

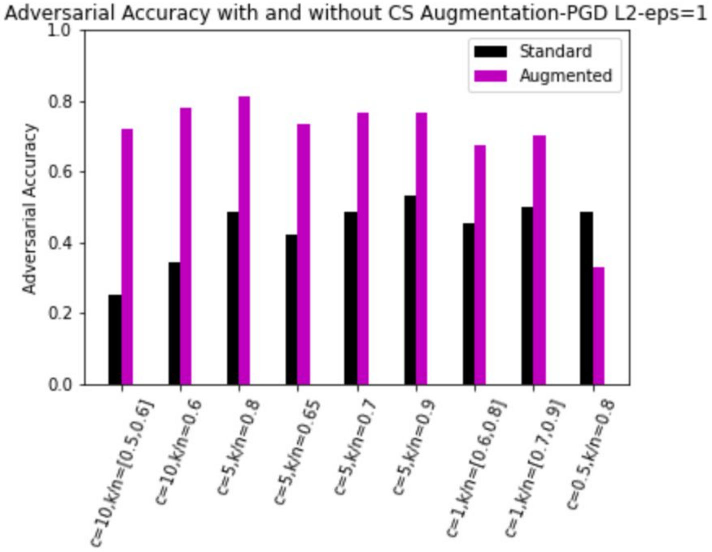



[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# PGD $L^2$-Metric $L^2$ - Epsilon=1.0 (Adaptive/BPDA)

Embedded
**Intelligence**




[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

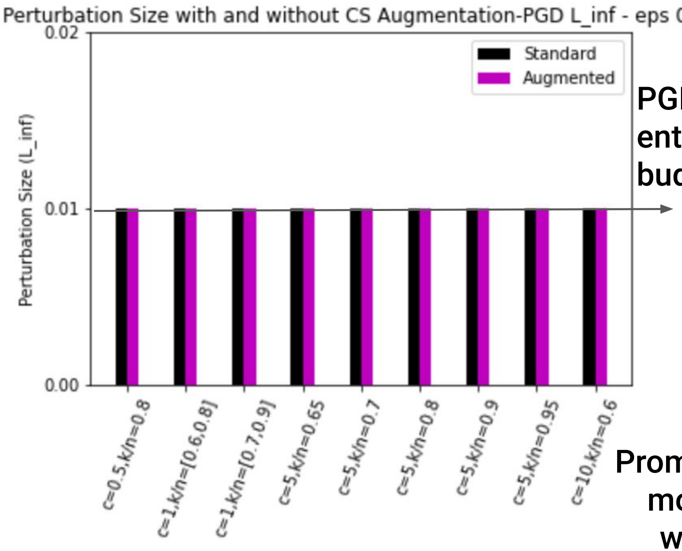
# PGD $L^2$-Metric $L^2$-Epsilon=0.5 and 1.0

Embedded Intelligence

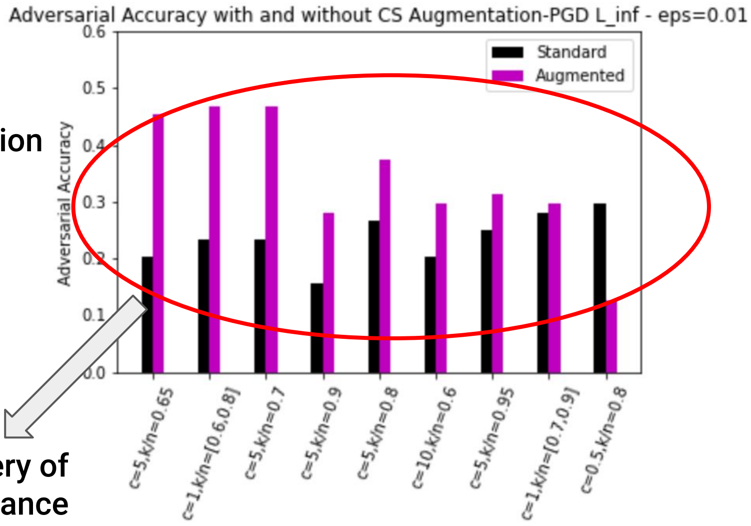
| c | k/n | Projected Gradient Descent $L^2$-eps=0.5 | | | Projected Gradient Descent $L^2$-eps=1.0 | | |
|---|---|---|---|---|---|---|---|
| | | Benign Accuracy | Adversarial Accuracy | Perturbation | Benign Accuracy | Adversarial Accuracy | Perturbation |
| 0.5 | 0.8 | 0.875/0.4844 | 0.6094/0.4063 | 0.4950/0.4955 | 0.8438/0.4844 | 0.4844/0.3281 | 0.9866/0.9865 |
| 1 | [0.6,0.65,0.7,0.75,0.8] | 0.6563/0.7969 | 0.5313/0.7188 | 0.4949/0.4959 | 0.6406/0.8281 | 0.4531/0.6719 | 0.9863/0.9860 |
| 1 | [0.7,0.75,0.8,0.85,0.9] | 0.7969/0.8438 | 0.6719/0.6875 | 0.4950/0.4957 | 0.7656/0.8594 | 0.5/0.7031 | 0.9860/0.9866 |
| 5 | 0.65 | 0.4531/0.9063 | 0.4531/0.7656 | 0.4948/0.4955 | 0.6094/0.8906 | 0.4219/0.7344 | 0.9858/0.9859 |
| 5 | 0.7 | 0.6875/0.9375 | 0.5313/0.7656 | 0.4951/0.4956 | 0.6719/0.8906 | 0.4844/0.7656 | 0.9858/0.9851 |
| 5 | 0.8 | 0.7813/0.8906 | 0.6875/0.7813 | 0.4951/0.4956 | 0.7813/0.8594 | 0.4844/0.8125 | 0.9863/0.9857 |
| 5 | 0.9 | 0.8438/0.963 | 0.6719/0.7656 | 0.4954/0.4957 | 0.7969/0.9375 | 0.5313/0.7656 | 0.9870/0.9865 |
| 10 | [0.5,0.52,0.54,0.56,0.58,0.6] | 0.3906/0.7969 | 0.3281/0.7188 | 0.4946/0.4954 | 0.3438/0.8438 | 0.25/0.7188 | 0.9851/0.9866 |
| 10 | 0.6 | 0.4375/0.8281 | 0.4063/0.7188 | 0.4949/0.4956 | 0.4375/0.8594 | 0.3438/0.7813 | 0.9853/0.9862 |

# PGD $L^{inf}$-Metric $L^{inf}$ -Epsilon=0.01

Embedded **Intelligence**




PGD uses the entire perturbation budget.




Promising recovery of model performance while under attack, continuing to investigate

[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# PGD $L^{inf}$-Metric $L^{inf}$ - Epsilon =0.01 and 0.031

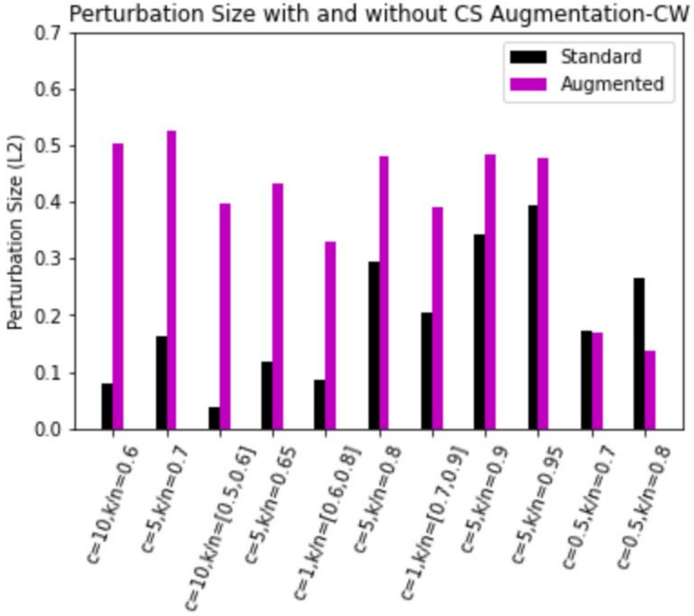
Embedded Intelligence

| c | k/n | Projected Gradient Descent $L^{inf}$-eps=0.01 | | | Projected Gradient Descent $L^{inf}$-eps=0.031 | | |
|---|---|---|---|---|---|---|---|
| | | Benign Accuracy | Adversarial Accuracy | Perturbation | Benign Accuracy | Adversarial Accuracy | Perturbation |
| 0.5 | 0.8 | 0.7969/0.5156 | 0.2344/0.125 | 0.01/0.01 | 0.7813/0.5313 | 0.0156/0 | 0.031/0.031 |
| 1 | [0.6,0.65,0.7,0.75,0.8] | 0.7031/0.875 | 0.2813/0.4688 | 0.01/0.01 | 0.6563/0.7656 | 0.0156/0.0156 | 0.031/0.031 |
| 1 | [0.7,0.75,0.8,0.85,0.9] | 0.8281/0.875 | 0.2031/0.2969 | 0.01/0.01 | 0.7969/0.8438 | 0/0 | 0.031/0.031 |
| 5 | 0.65 | 0.625/0.8906 | 0.2344/0.4531 | 0.01/0.01 | 0.4688/0.9063 | 0.0156/0.0156 | 0.031/0.031 |
| 5 | 0.7 | 0.75/0.9063 | 0.2656/0.4688 | 0.01/0.01 | 0.6875/0.9063 | 0.0313/0.0156 | 0.031/0.031 |
| 5 | 0.8 | 0.7813/0.8906 | 0.25/0.375 | 0.01/0.01 | 0.8125/0.8594 | 0.0156/0 | 0.031/0.031 |
| 5 | 0.9 | 0.8438/0.9375 | 0.2031/0.2813 | 0.01/0.01 | 0.8594/0.9375 | 0.0313/0.0156 | 0.031/0.031 |
| 5 | 0.95 | 0.875/0.9375 | 0.1563/0.3125 | 0.01/0.01 | 0.8594/0.9688 | 0/0.0156 | 0.031/0.031 |
| 10 | 0.6 | 0.4688/0.4219 | 0.2969/0.2969 | 0.01/0.01 | 0.4688/0.8438 | 0.0313/0.0156 | 0.031/0.031 |

# CW $L^2$-Metric $L^2$ (Adaptive/BPDA)

Embedded Intelligence

CS-data augmentation increases the perturbation size of CW attack. The effect is more significant when trained with c=10. Since CW is an unbounded attack, this is the ultimate goal and could force an attack to risk being vulnerable.




[0.5,0.6]=[0.5,0.52,0.54,0.56,0.58,0.6]
[0.6,0.8]=[0.6,0.65,0.7,0.75,0.8]
[0.7,0.9]=[0.7,0.75,0.8,0.85,0.9]

# CW $L^2$-Metric $L^2$ (Adaptive/BPDA)

Embedded Intelligence

| C | k/n | Carlini-Wagner $L^2$-Metric $L^2$ | | |
|---|---|---|---|---|
| | | Benign Accuracy | Adversarial Accuracy | Perturbation |
| 0.5 | 0.7 | 0.703/0.641 | 0.578/0.453 | 0.171/0.171 |
| 0.5 | 0.8 | 0.828/0.484 | 0.563/0.328 | 0.265/0.136 |
| 1 | [0.6,0.65,0.7,0.75,0.8] | 0.703/0.781 | 0.5/0.5 | 0.084/0.331 |
| 1 | [0.7,0.75,0.8,0.85,0.9] | 0.813/0.844 | 0.5/0.391 | 0.206/0.392 |
| 5 | 0.65 | 0.578/0.906 | 0.406/0.375 | 0.119/0.433 |
| 5 | 0.7 | 0.625/0.891 | 0.5/0.422 | 0.162/0.528 |
| 5 | 0.8 | 0.797/0.891 | 0.375/0.266 | 0.294/0.481 |
| 5 | 0.9 | 0.813/0.938 | 0.313/0.25 | 0.344/0.486 |
| 5 | 0.95 | 0.859/0.953 | 0.219/0.172 | 0.396/0.479 |
| 10 | [0.5,0.52,0.54,0.56,0.58,0.6] | 0.234/0.844 | 0.438/0.469 | 0.039/0.396 |
| 10 | 0.6 | 0.438/0.875 | 0.313/0.453 | 0.079/0.504 |

This is a 500% increase in the perturbation size used by the  in the best CS defense configuration.

We will continue to investigate this.

# Discussion

Embedded
**Intelligence**

# Task 1.4 Goals & Results

Embedded **Intelligence**

| Task 1.4 Technical Goal | Task 1.4 Results |
|---|---|
| Evaluate whether CS-based defense can force an attacker into using a larger perturbation | We had very promising results in forcing adaptive attacks (CW & PGD) to use larger perturbations by pushing BPDA outside of the assumption of $g(x) \approx x$ |
| Evaluate whether acceptable system performance can be maintained in defended systems under "non-attacked" conditions | By modifying the training procedure to include Compressed Sensing as the defense, we were able to recover acceptable benign/unattacked accuracy |

# Important Questions to Ask Next

- We've shown CS defense can force an attacker to increase the perturbation size used. What is the downstream benefit of that on an attack detection system, if at all?
- Can we devise better adaptive attacks against CS using the full form BPDA which includes searching for approximations to CS that are differentiable?
- If "more" stochasticity is leveraged in CS, will that add further "costs" to attackers by forcing them to use Expectation-over-Transformation in addition to BPDA to construct adaptive attacks?
- Do these results generalize to other datasets and signal types?

# Appendices

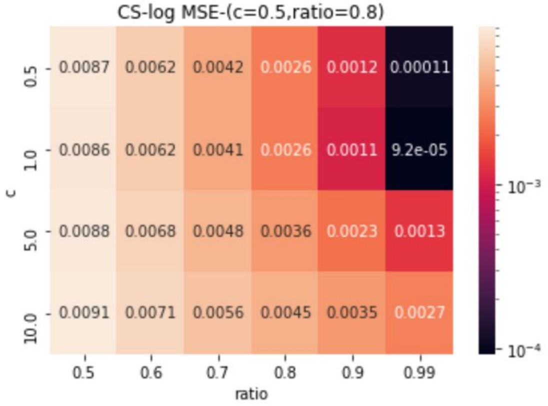
Embedded
**Intelligence**

# Appendix I: CS Parameter Tuning without and with Data Augmentation Training

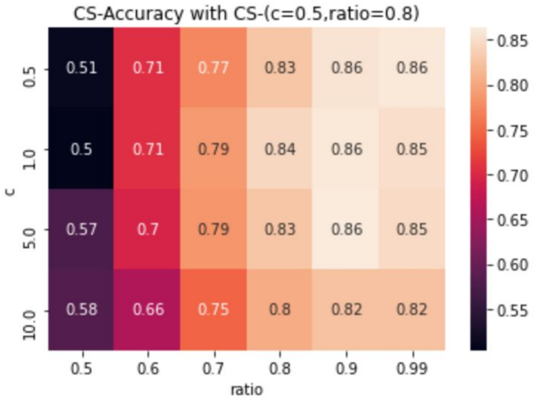
Embedded
Intelligence

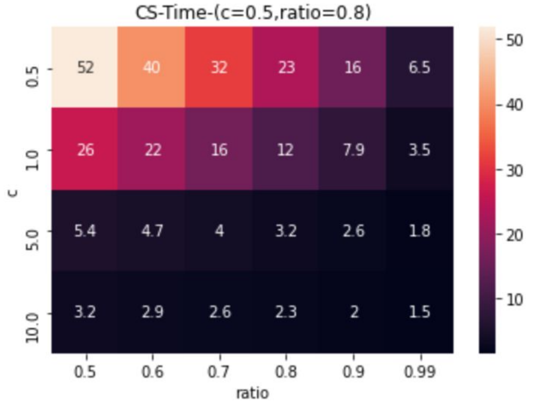
# CS Parameter Tuning with Data Augmentation Training

Embedded
**Intelligence**




c=0.5,k/n=0.8

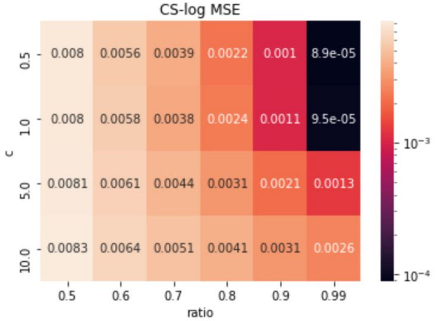


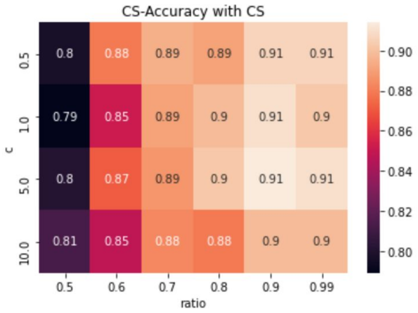



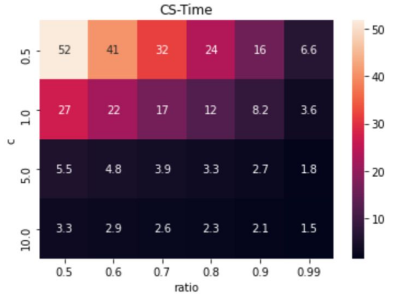
# CS Parameter Tuning with Data Augmentation Training

Embedded
**Intelligence**

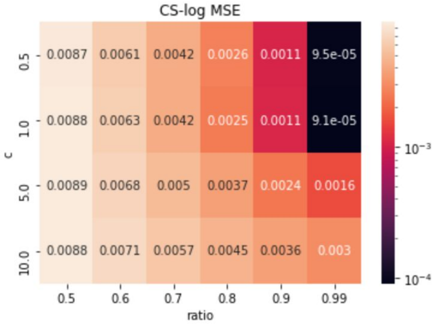
## c=1,k/n=0.6 to 0.8

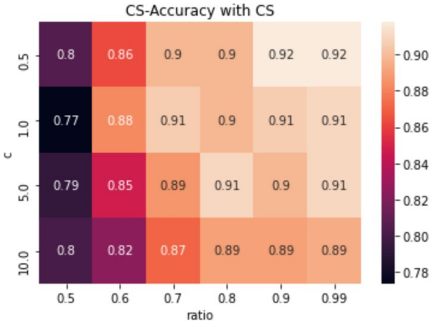


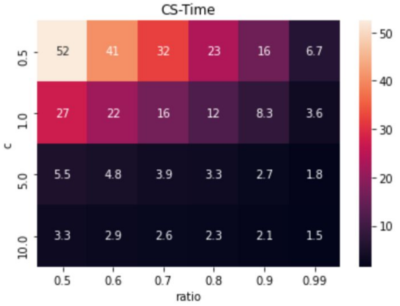




c=1,k/n=0.7 to 0.9



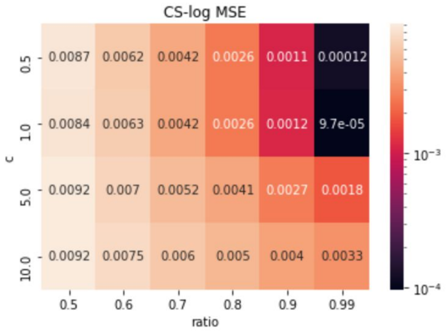



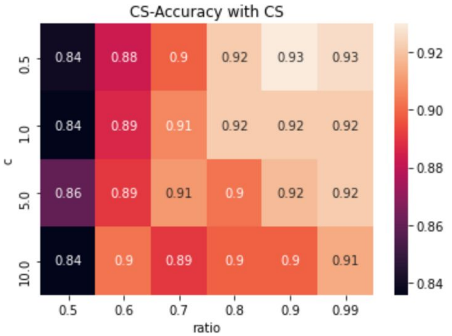



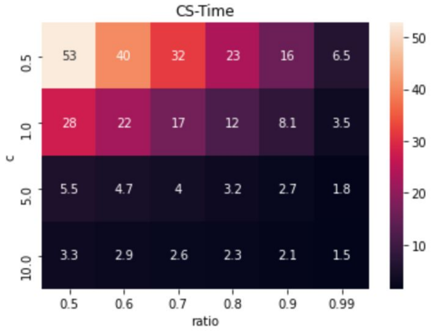
# CS Parameter Tuning with Data Augmentation Training

Embedded
**Intelligence**

## c=5,k/n=0.65




## c=5,k/n=0.8