Machine Learning Systems

Lecture 12: Security of ML Systems (Adversarial ML)

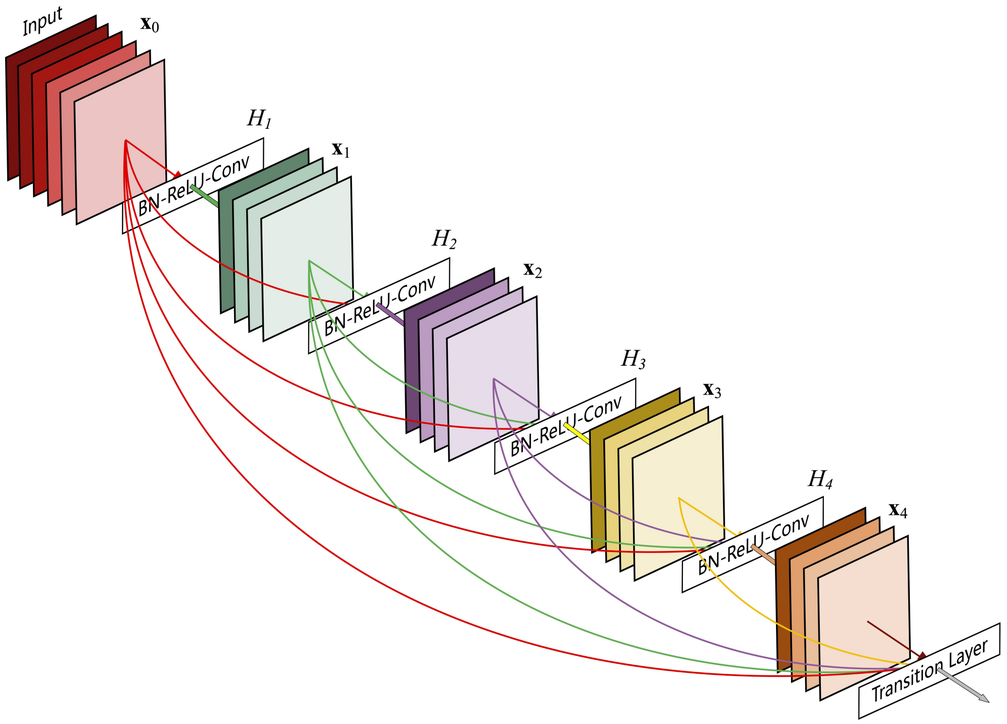
Pooyan Jamshidi

So what this talk is about?

The Security of

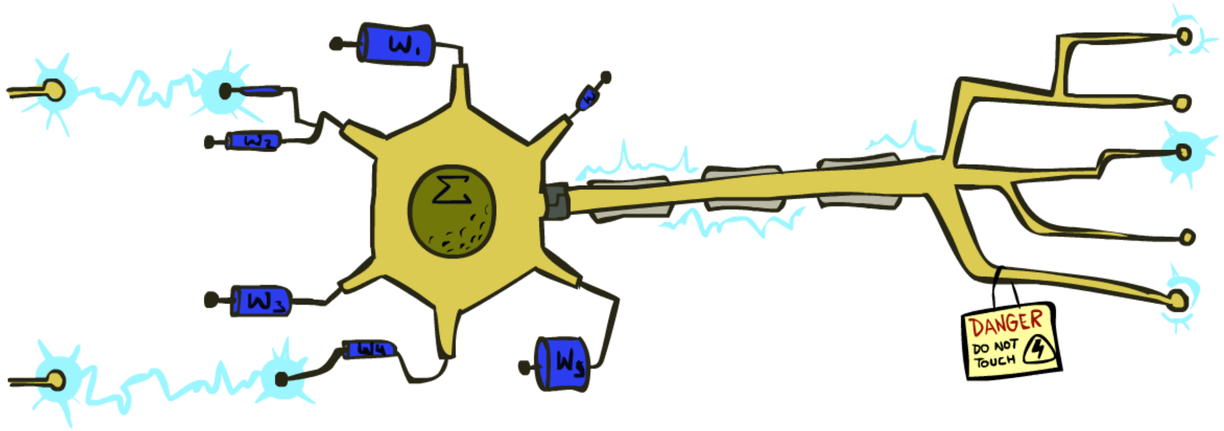
~~Machine~~ Learning

Deep

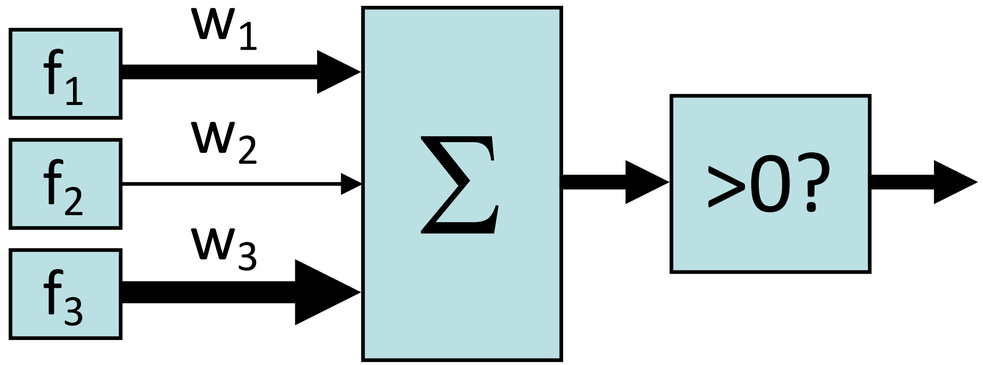


A quick recap about
Supervised Machine Learning

Linear Classifiers

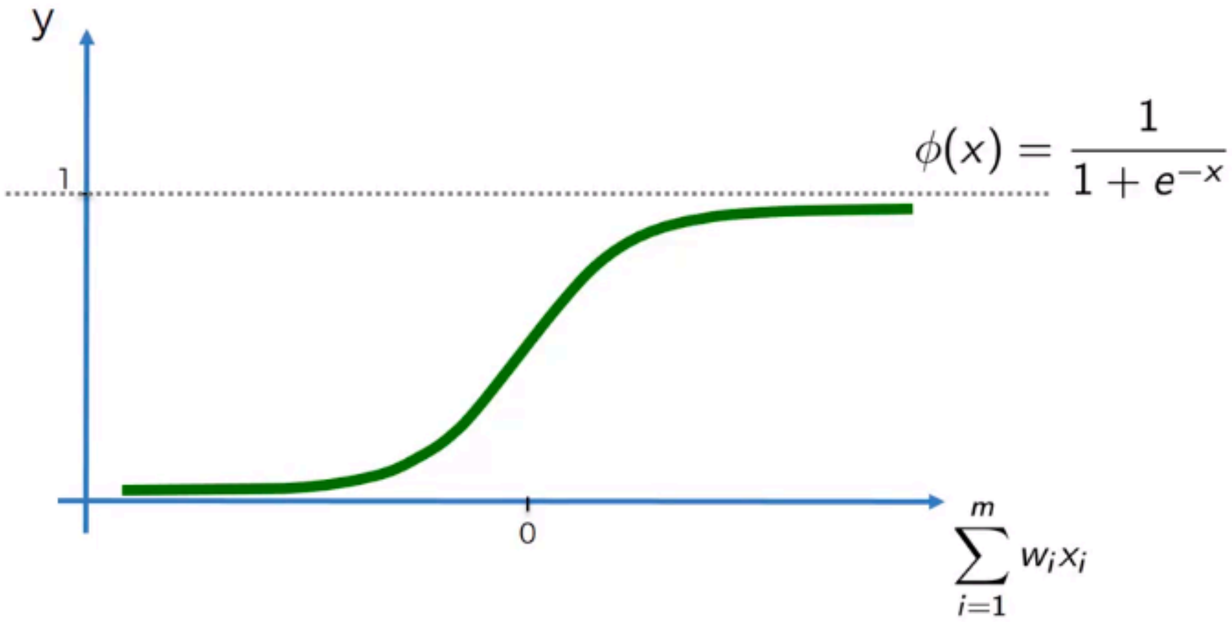


$$activation_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$



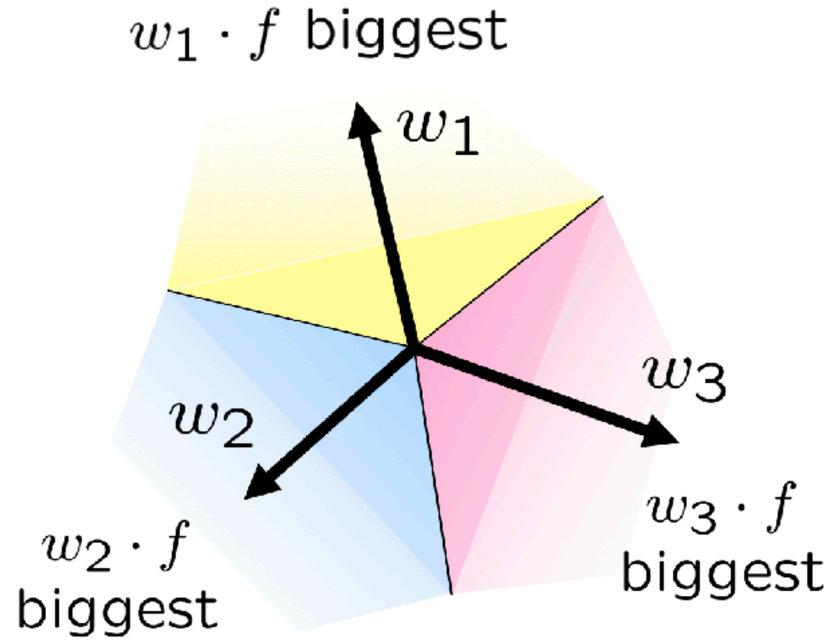
How to get probabilistic decisions?

- Activation: $z = w \cdot f(x)$
- If z very positive -> want probability going to 1
- If z very negative -> want probability going to 0



Multiclass Logistic Regression

- Multi-class linear classification
  - A weight vector for each class: w_y
  - Score (activation) of a class y: $w_y \cdot f(x)$
  - Prediction w/highest score wins: $y = argmax_y w_y \cdot f(x)$



- How to make the scores into probabilities?

$$z_1, z_2, z_3 \rightarrow \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}, \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}, \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

original activations

softmax activations

Best w?

- Maximum likelihood estimation:

$$\max_w \quad ll(w) = \max_w \quad \sum_i \log P(y^{(i)}|x^{(i)}; w)$$

- With:

$$P(y^{(i)}|x^{(i)}; w) = \frac{e^{w_{y^{(i)}} \cdot f(x^{(i)})}}{\sum_y e^{w_y \cdot f(x^{(i)})}}$$

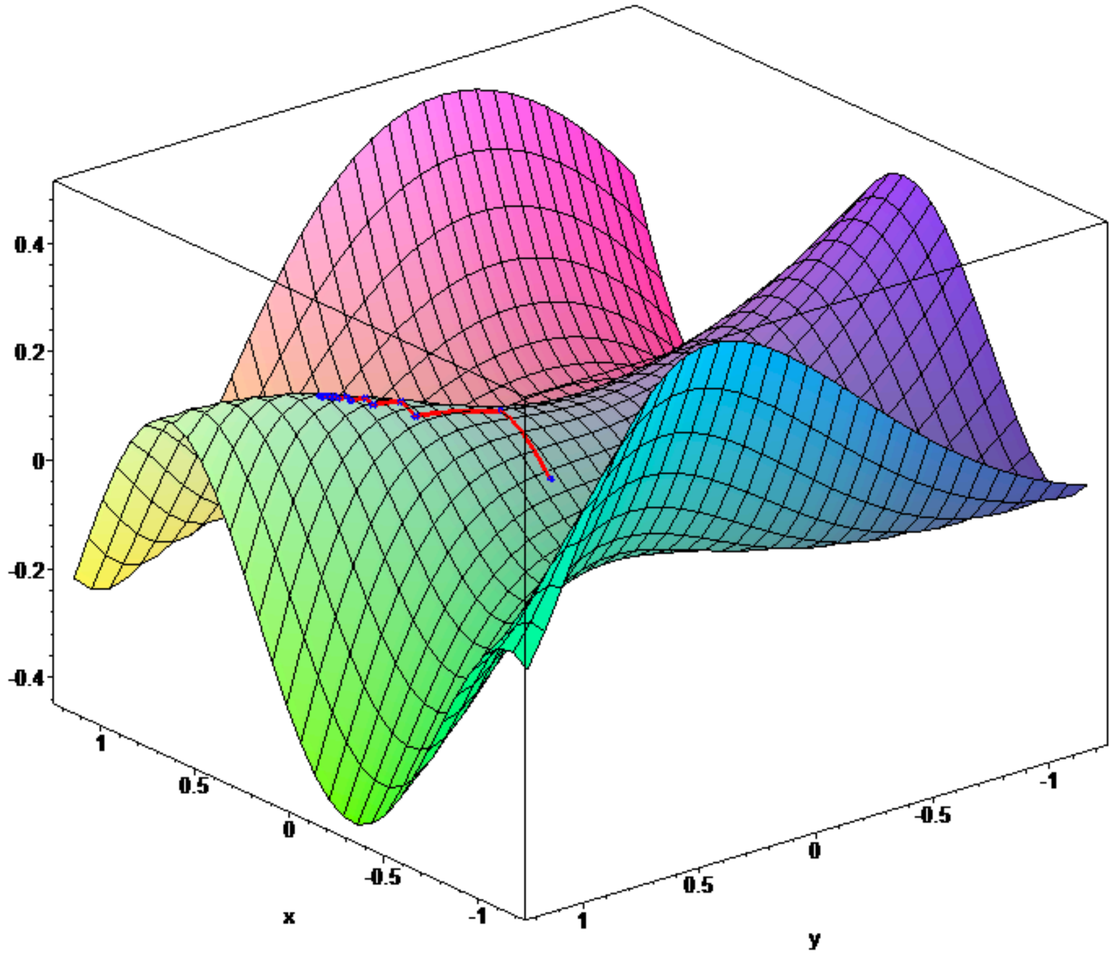
How do we solve the optimization problem?

$$\max_w \quad ll(w) = \max_w \quad \underbrace{\sum_i \log P(y^{(i)}|x^{(i)}; w)}_{g(w)}$$

Gradient Ascent

$$w \leftarrow w + \alpha * \nabla g(w)$$

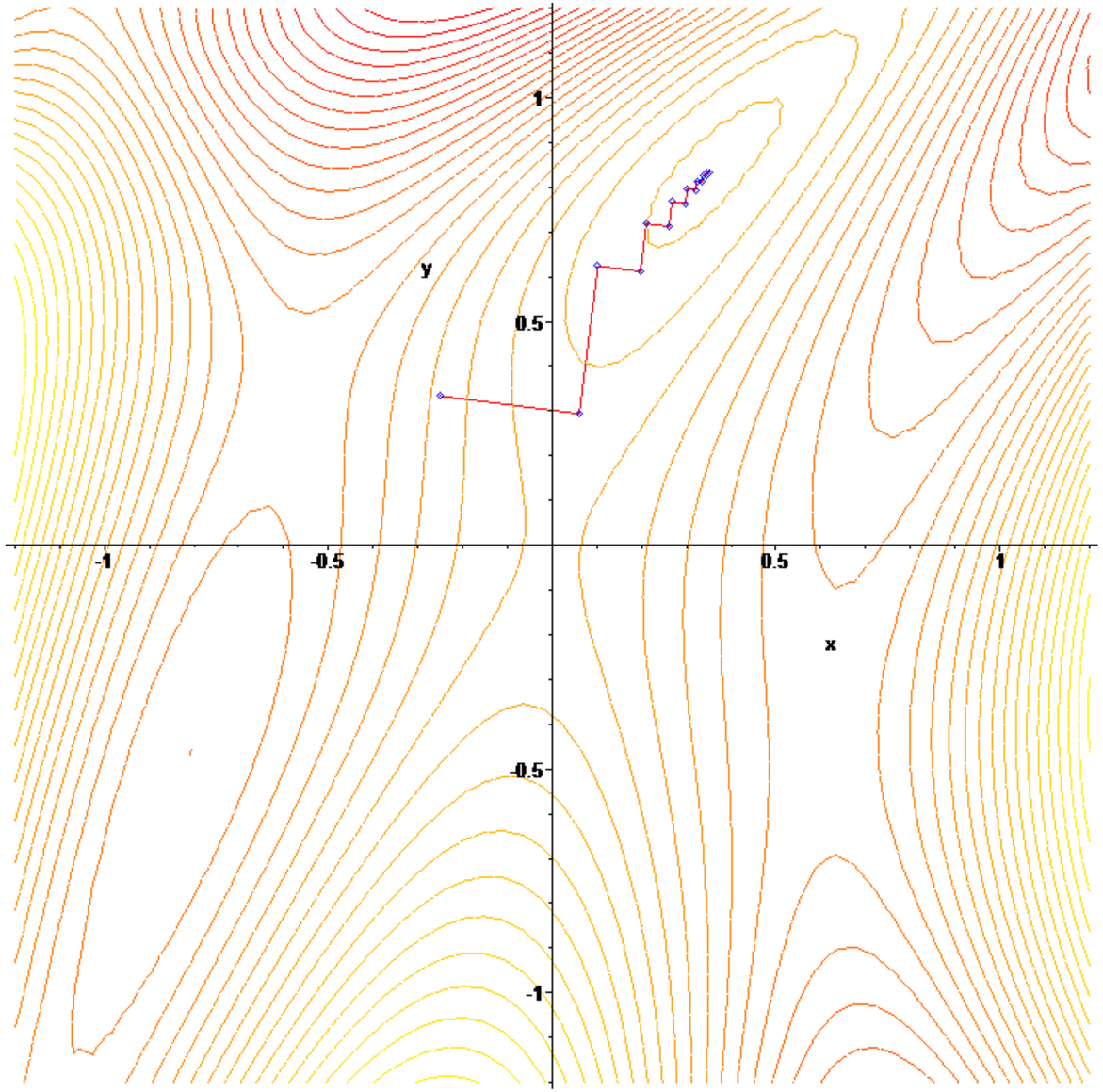
$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial w_1} \\ \frac{\partial g}{\partial w_2} \\ \dots \\ \frac{\partial g}{\partial w_n} \end{bmatrix}$$



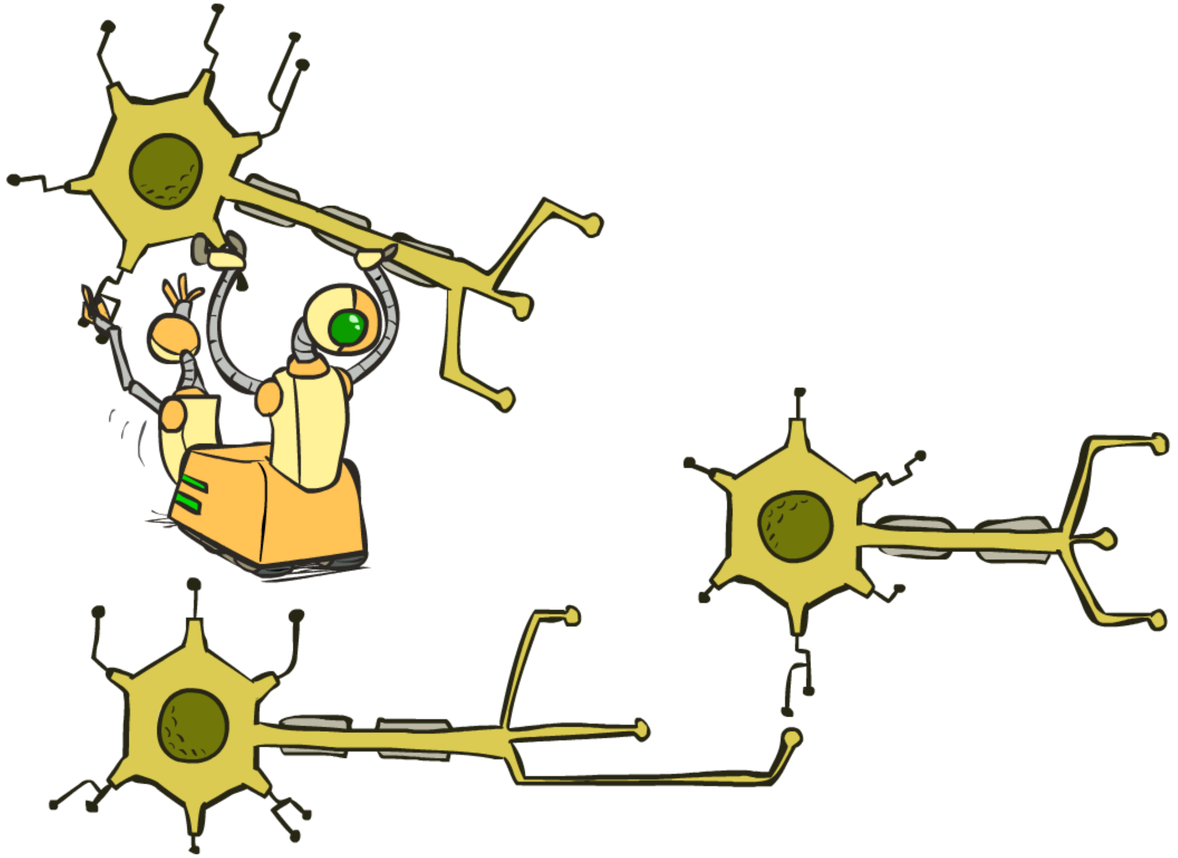
Gradient Ascent

$$w \leftarrow w + \alpha * \nabla g(w)$$

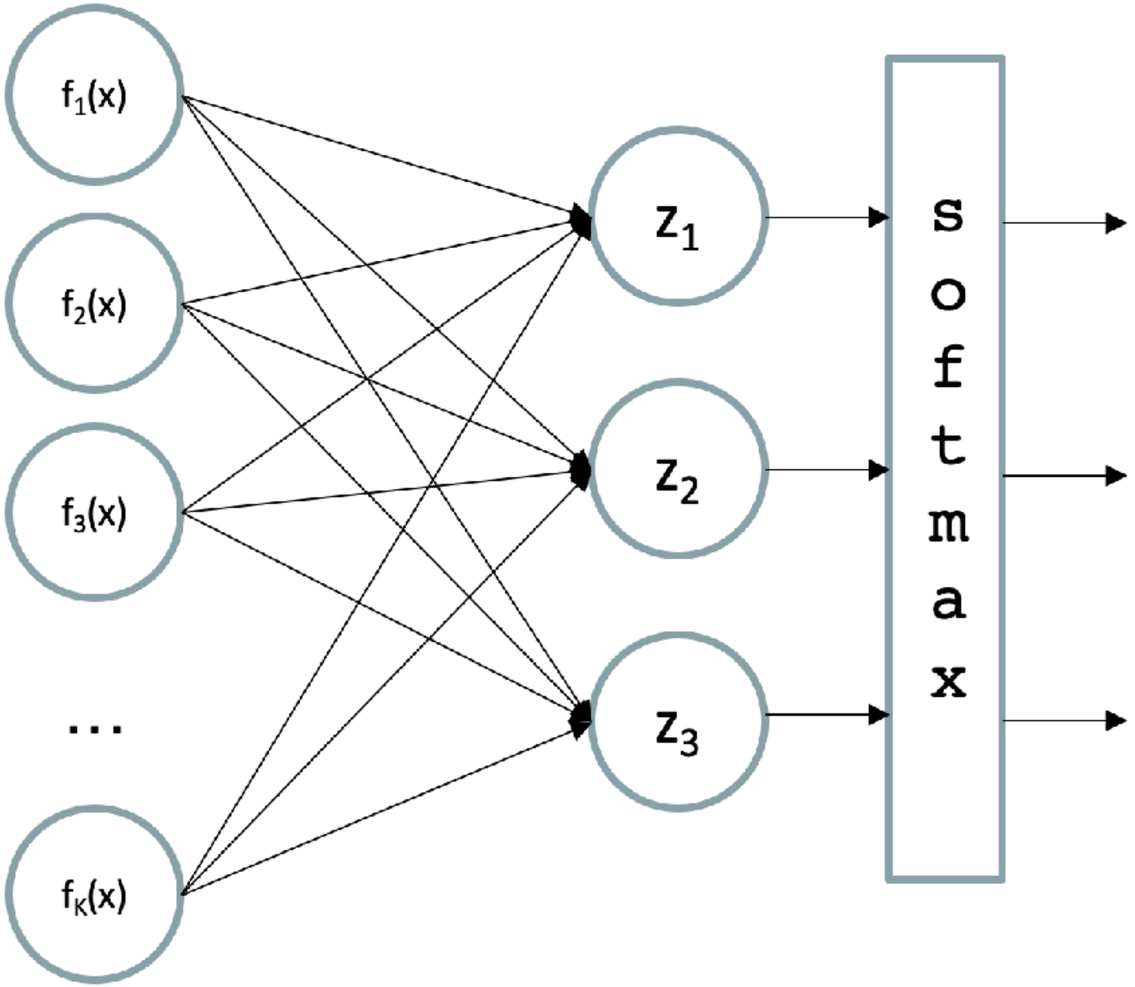
$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial w_1} \\ \frac{\partial g}{\partial w_2} \\ \dots \\ \frac{\partial g}{\partial w_n} \end{bmatrix}$$



Neural Networks



Multi-class Logistic Regression is a special case of neural network

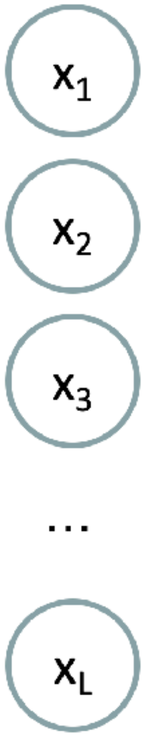


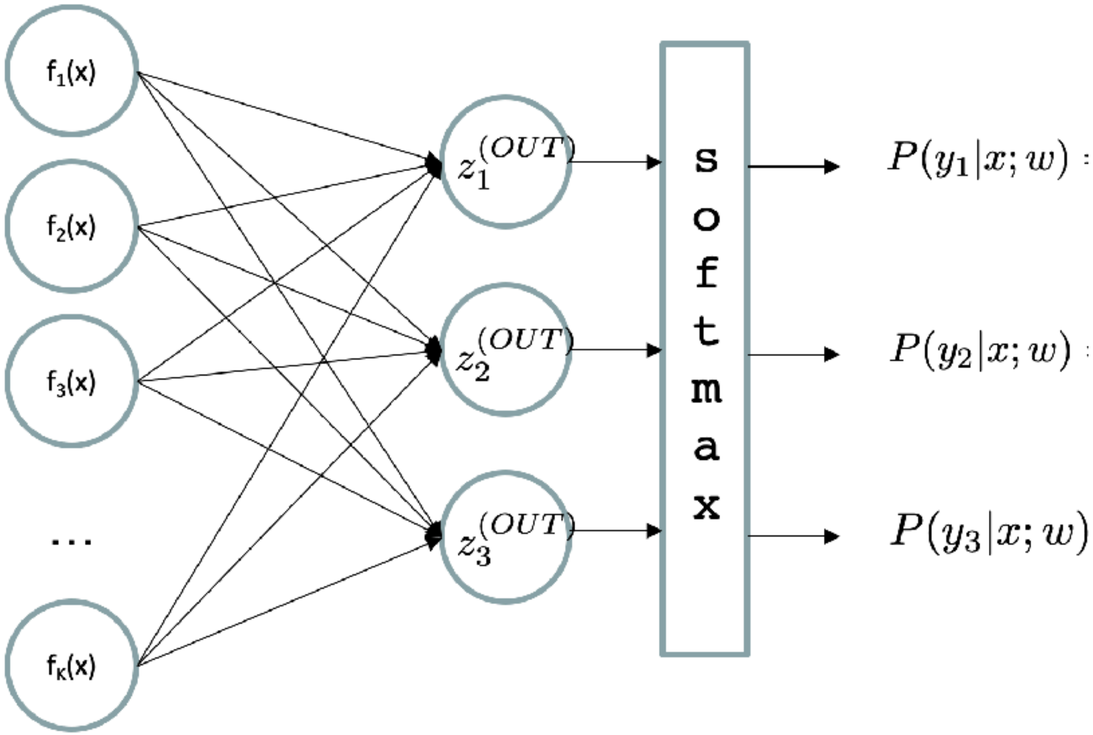
$$P(y_1|x;w) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$P(y_2|x;w) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

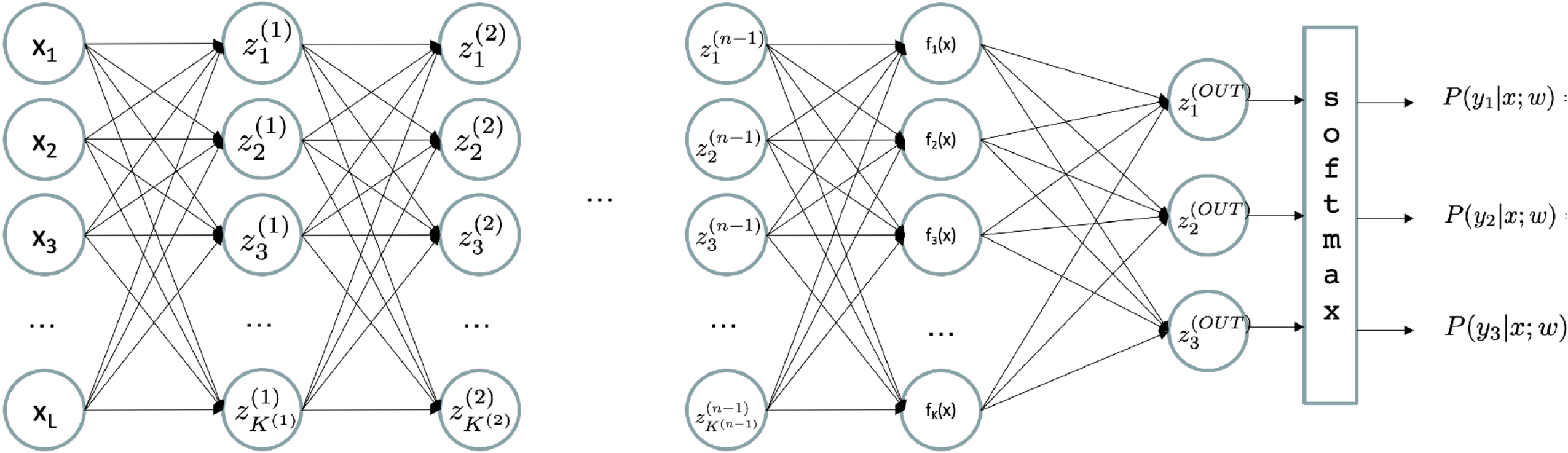
$$P(y_3|x;w) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

Deep Neural Network = Also learn the features!

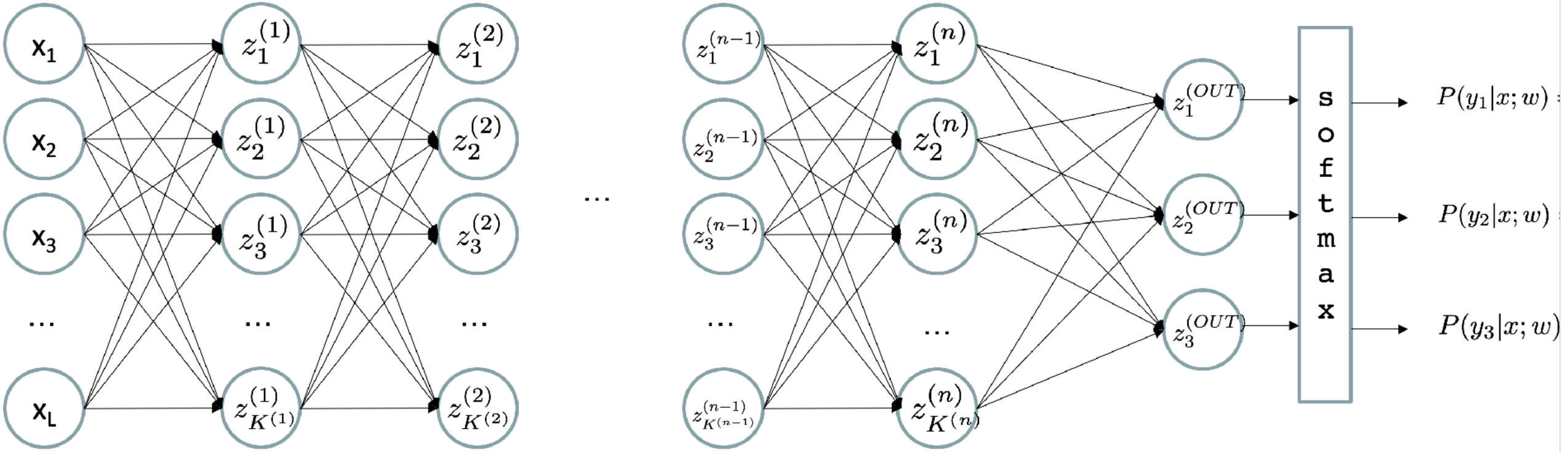




Deep Neural Network = Also learn the features!



Deep Neural Network = Also learn the features!



$$z_i^{(k)} = g(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)})$$

g = nonlinear activation function

Training the deep neural network is just like logistic regression

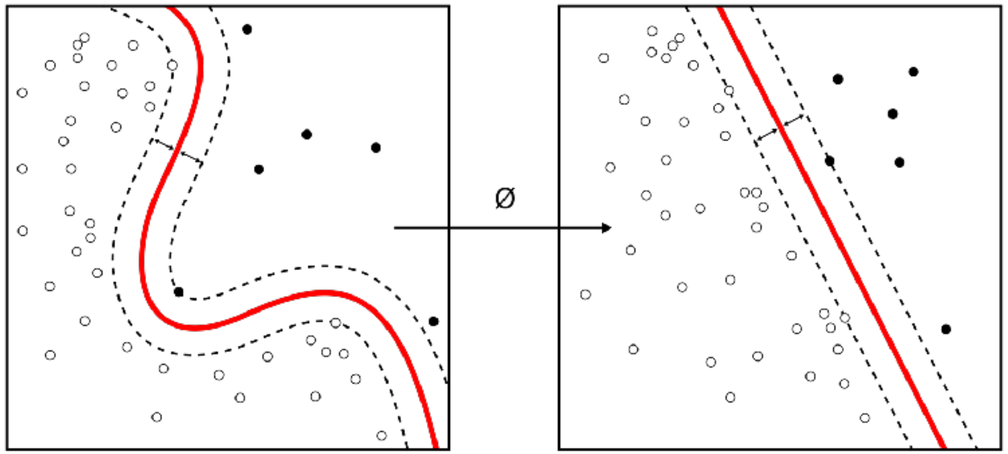
$$\max_w \quad ll(w) = \max_w \quad \sum_i \log P(y^{(i)}|x^{(i)}; w)$$

→ just run gradient ascent

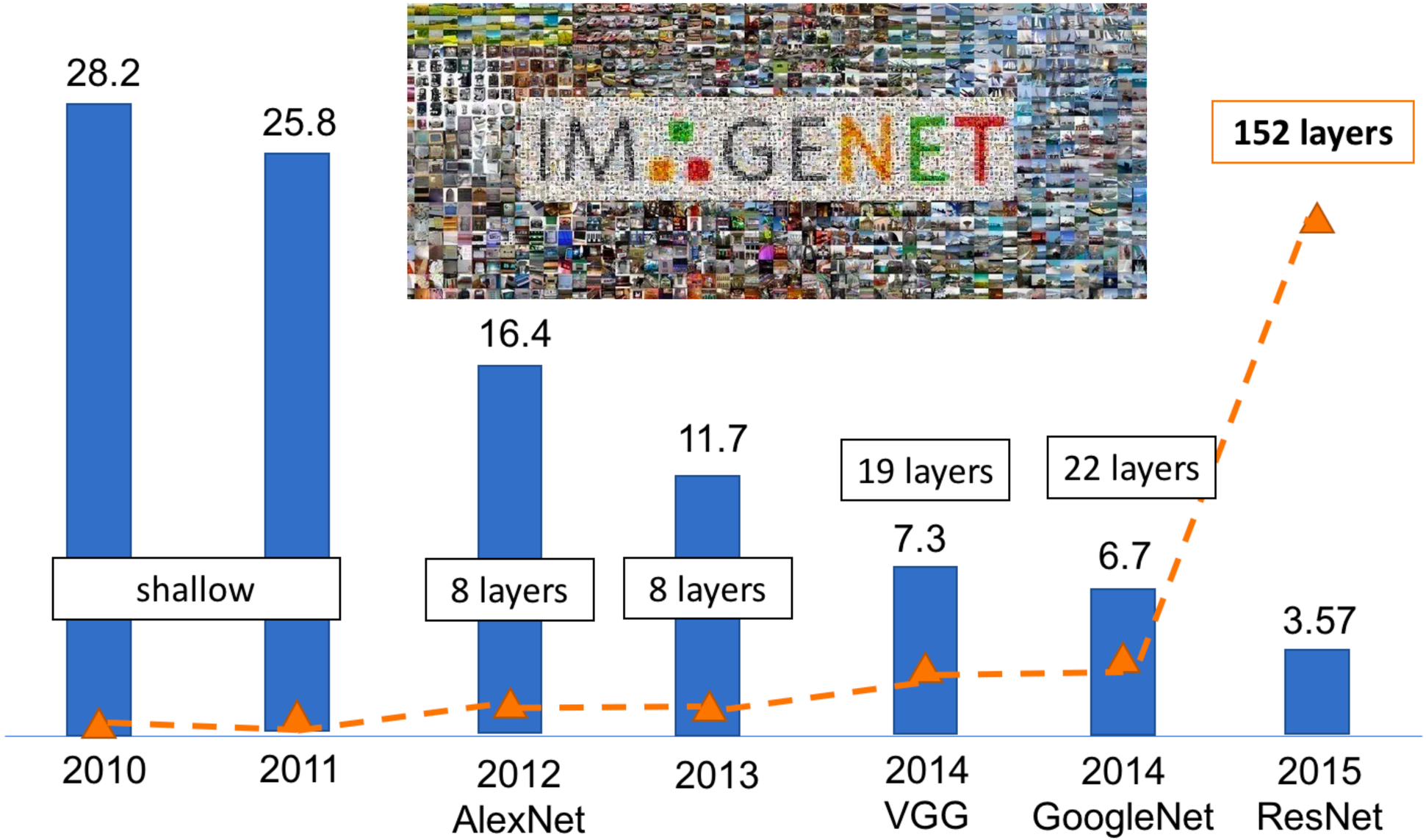
\+ stop when log likelihood of hold-out data starts to decrease

ML developed a rich theory to guide us here

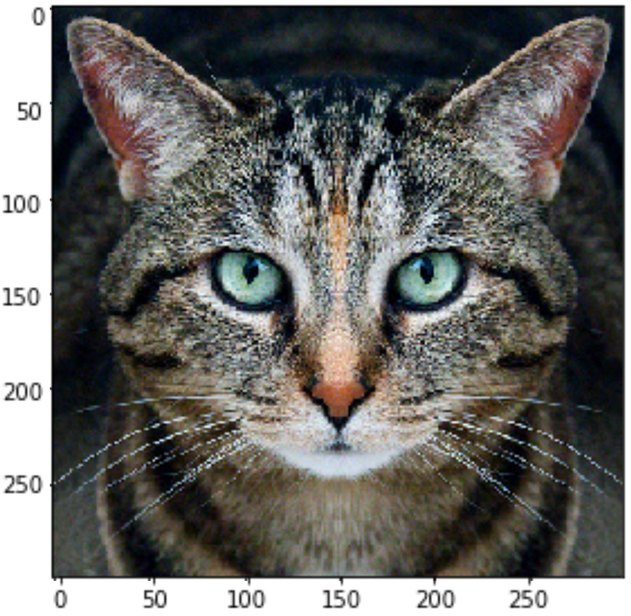
(and this was its only goal)

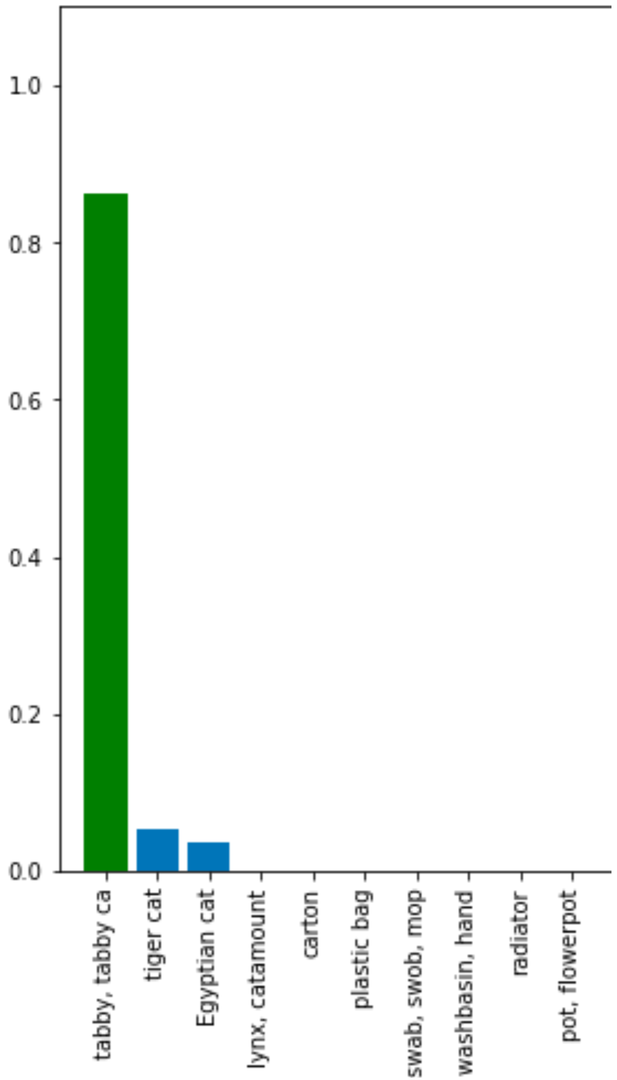


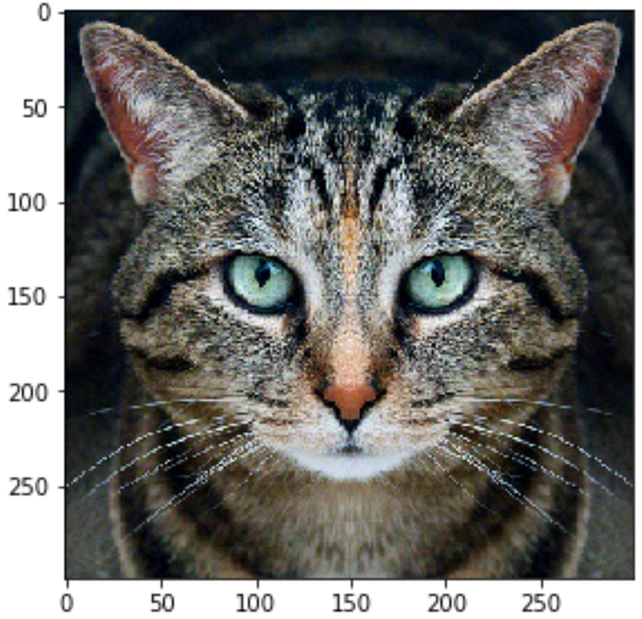
Machine Learning: The Success Story

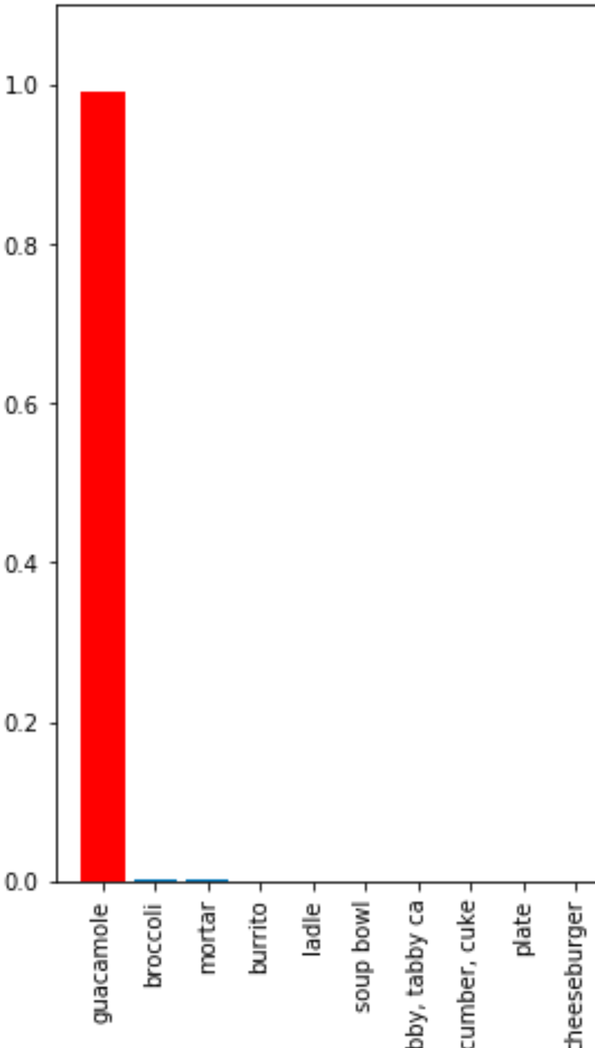


Deep neural networks can be easily fooled









Neural networks can be tricked

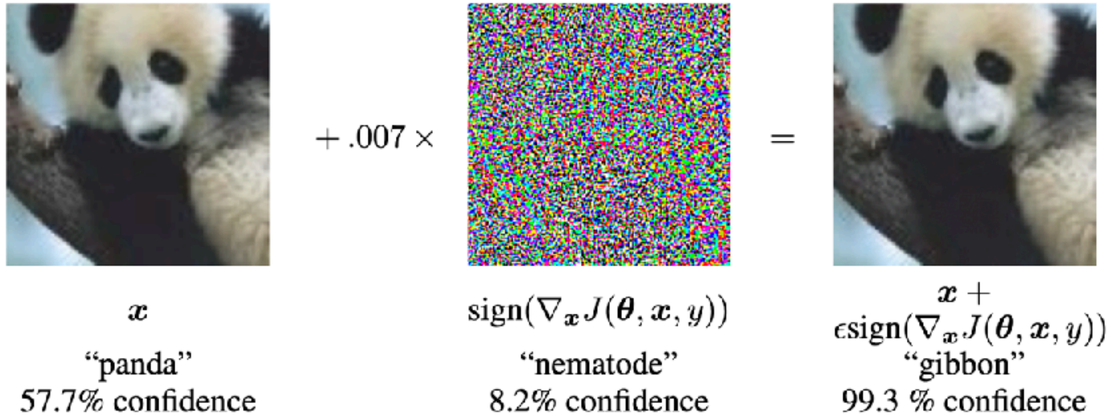
89% tabby cat

Adversarial
Perturbation

98% guacamole

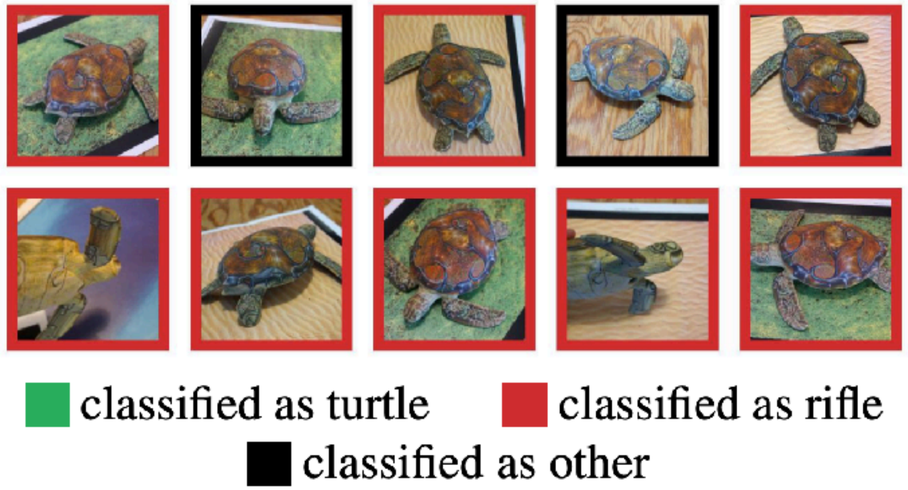
Yes, neural networks can be tricked that easily

Adversarial Examples



[Goodfellow et al. 2014]: Imperceptible noise can fool DNN classifiers

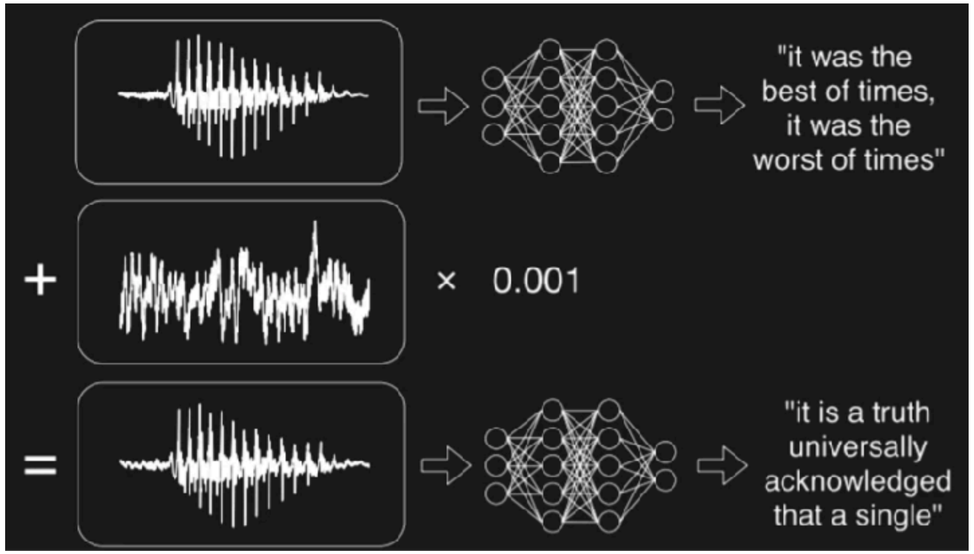
[Engstrom, Tran, Tsipras, Schmidt, Madry 2018]: Rotation + Translation can fool classifiers



[Athalye, Engstrom, Ilyas, Kwok 2017]: 3D-printed model classified as rifle from most viewpoints

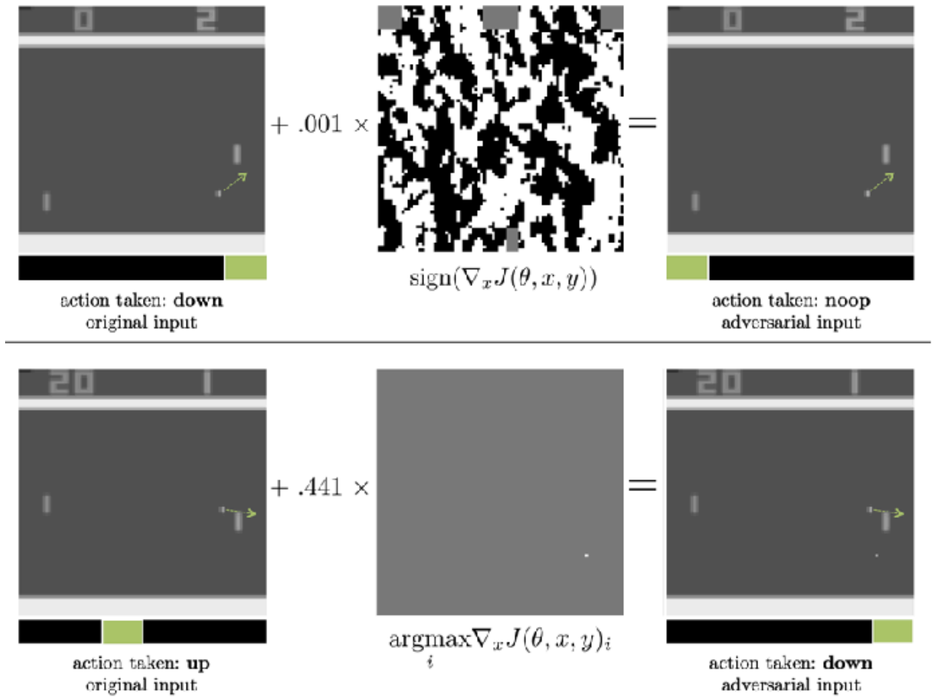
Adversarial Examples (Security)

[Sharif et al. 2016]: Glasses the fool face classifiers



[Carlini et al. 2016]: Voice commands that are imperceptible by humans

Adversarial Examples (RL, NLP)



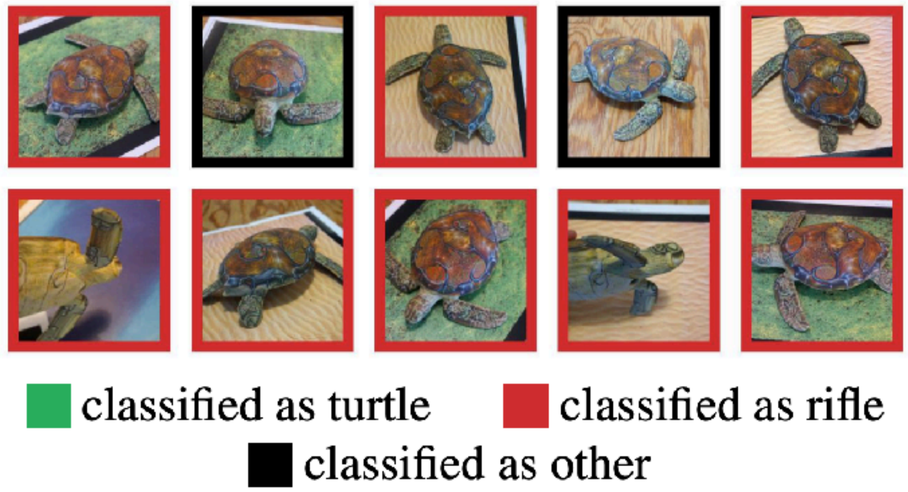
[Huang et al. 2017]: Small input changes can decrease RL performance

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

[Jia Liang 2017]: Irrelevant sentences confused reading comprehension systems

Should we be worried?

Probably not here!



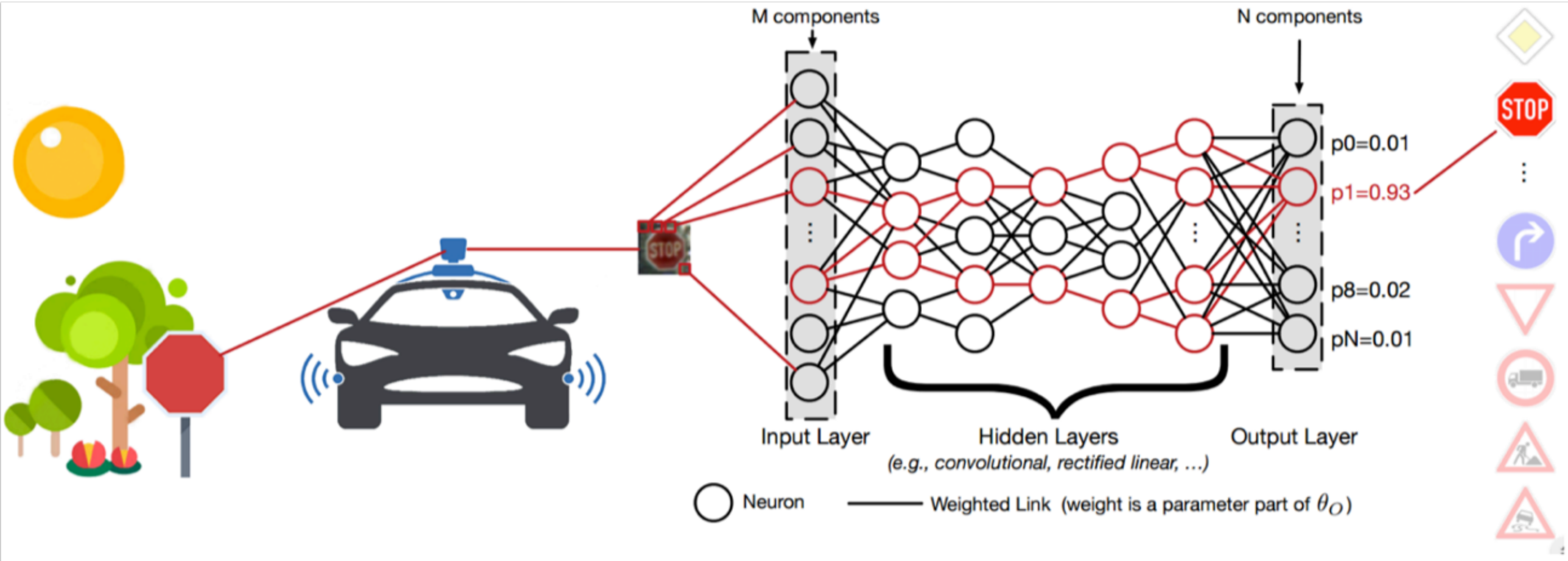
[Athalye, Engstrom, Ilyas, Kwok 2017]:
3D-printed model classified as rifle from most viewpoints

But we should be worried here!

[Pei et al. 2017]: DeepXplore: Automated Whitebox Testing of Deep Learning Systems

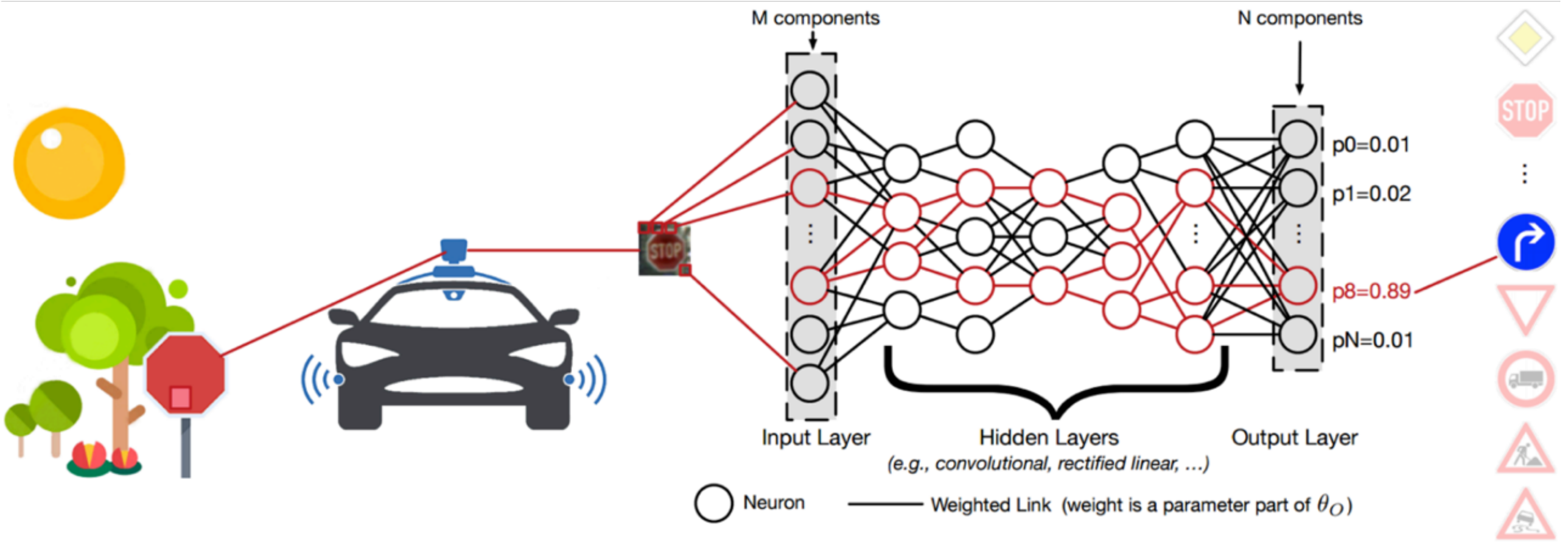
[Tian et al. 2017]: DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars

Should we be worried?



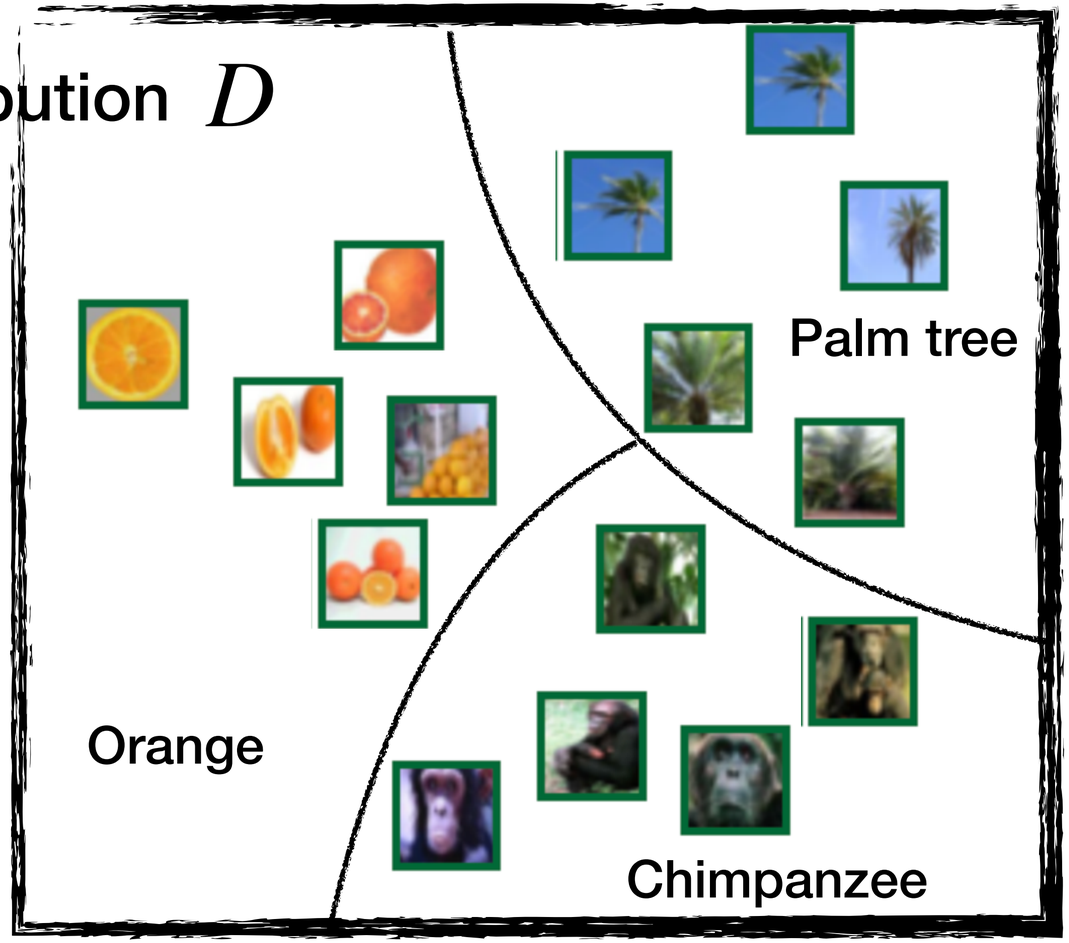
Source: [Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami, 2016]

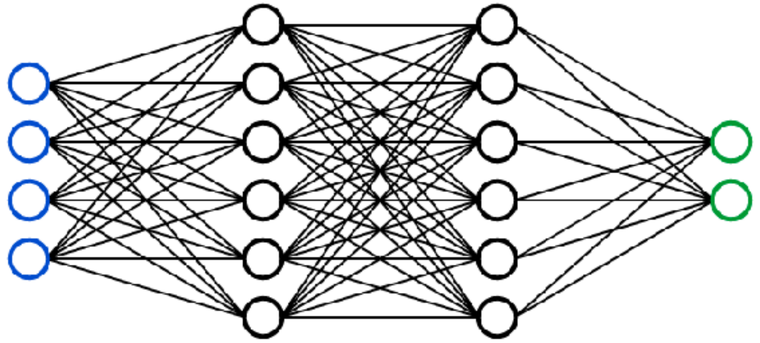
Should we be worried?



Source: [Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, Ananthram Swami, 2016]

Where Do Adversarial Examples Come From?





θ f_θ

$(x, y) =$  , Orange

$f_{\theta_1}(x, y) =$ **palm tree** ✗

$f_{\theta_2}(x, y) =$ **orange** ✓

Goal of ML:

Find θ^* such that

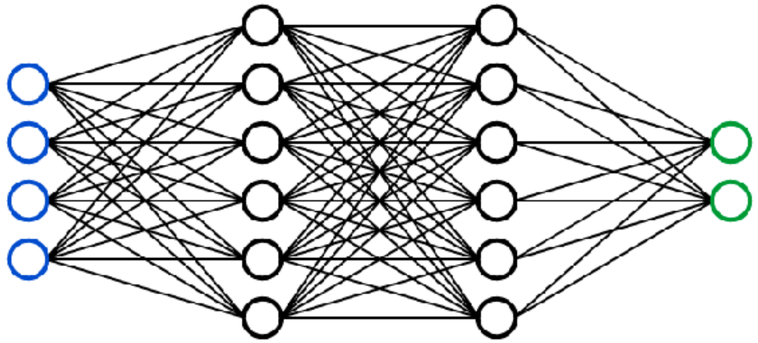
$\mathbb{E}_{(x,y)\sim D}\mathcal{L}(\theta^*, x, y)$ **Is small**

Where Do Adversarial Examples Come From?

$$min_{\theta}\mathcal{L}(\theta, x, y)$$

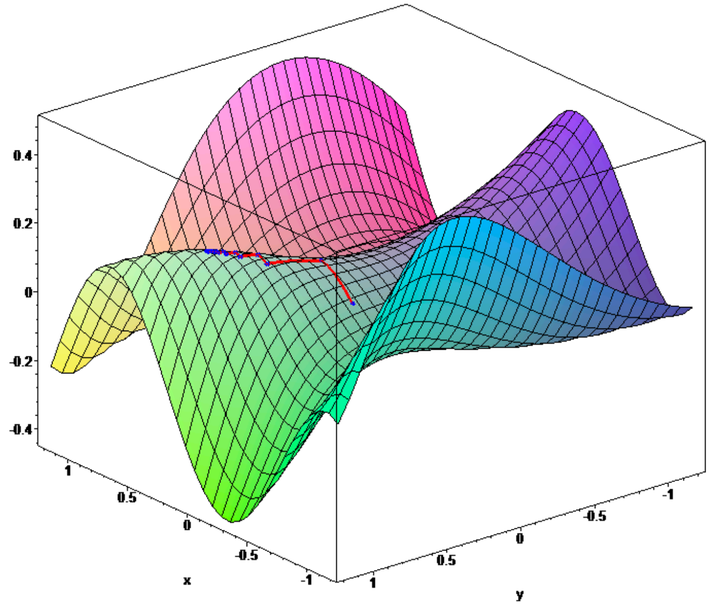
$$max_{\delta}\mathcal{L}(\theta, x+\delta, y)$$

$$||\delta||_p \leq \epsilon$$



θ

Gradient Descent
to find good parameters



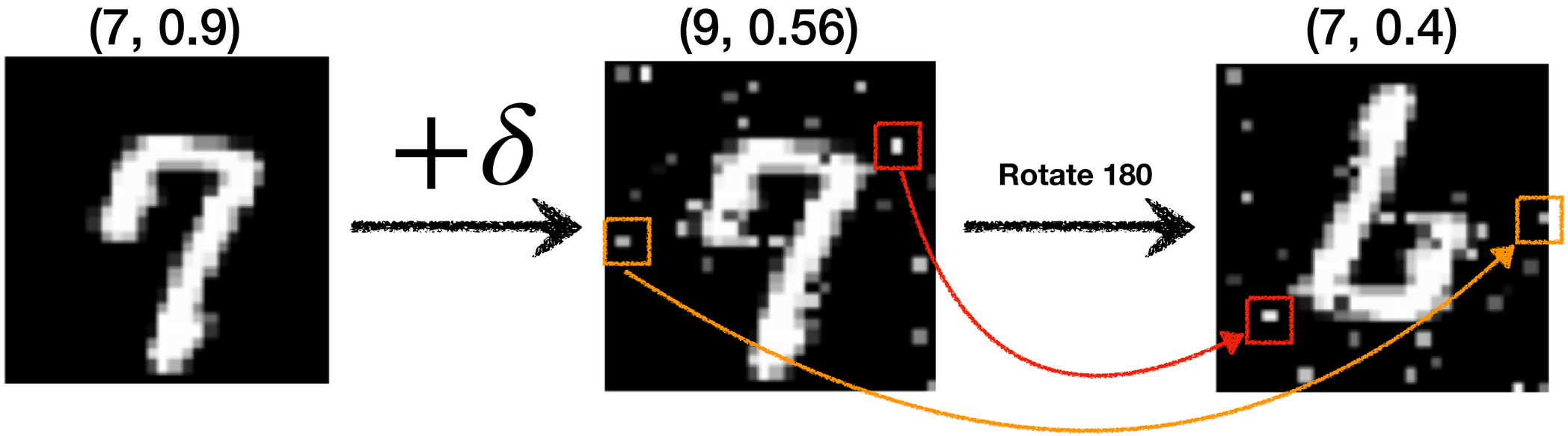
"Adversarial vulnerability is a direct result of our models' sensitivity to well-generalizing features in the data."

[Ilyas et al. 2019]: Adversarial Examples Are Not Bugs, They Are Features

ATHENA:

A Framework for Defending Machine Learning Systems Against Adversarial Attacks

Key idea behind our approach: Input transformation

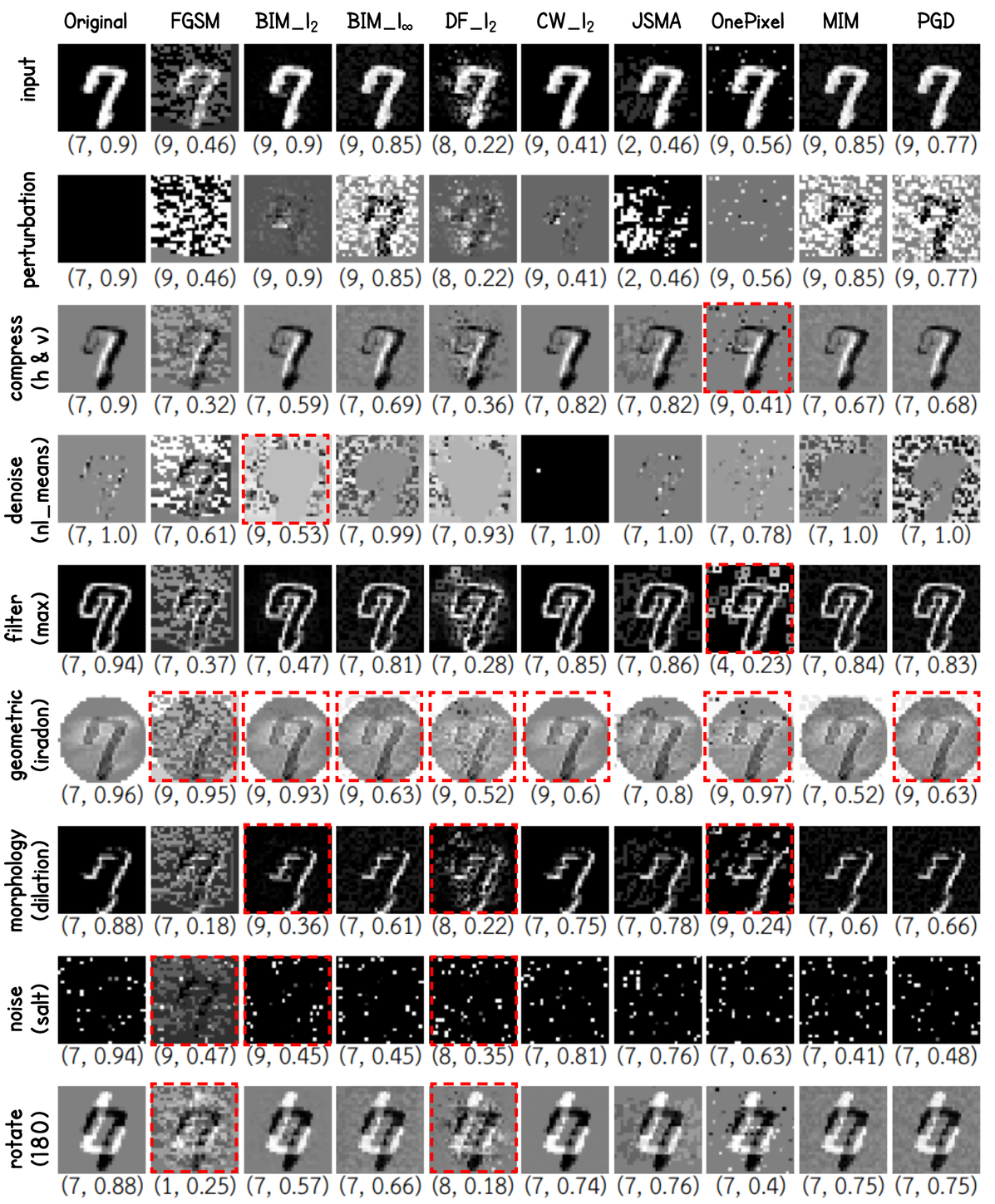


Insights

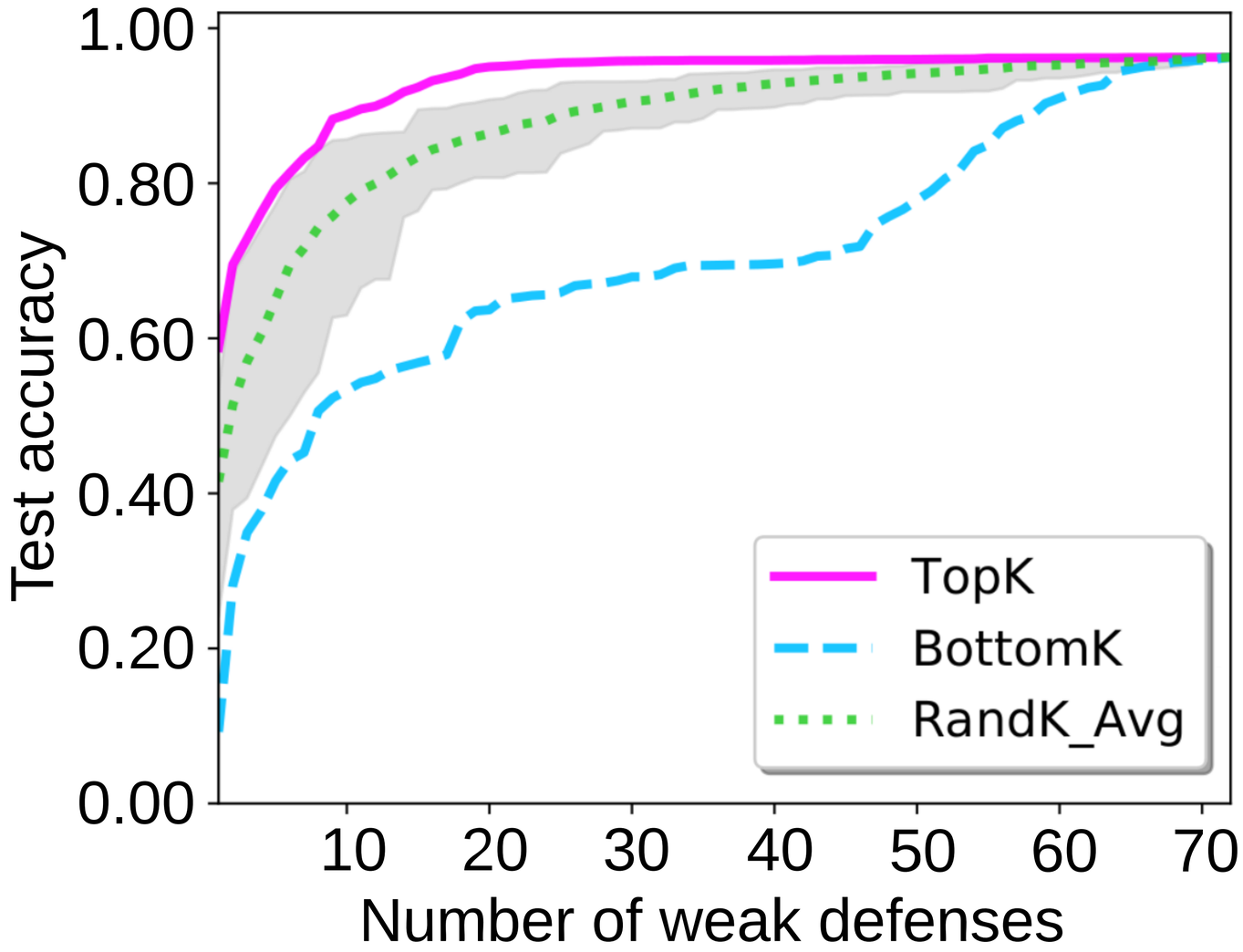
- Effectiveness of WDs *varies*
- WDs *complement* each other

-> A defense based on *ensemble of WDs* can be independent of particular type of adversarial attack

WD: Weak Defense

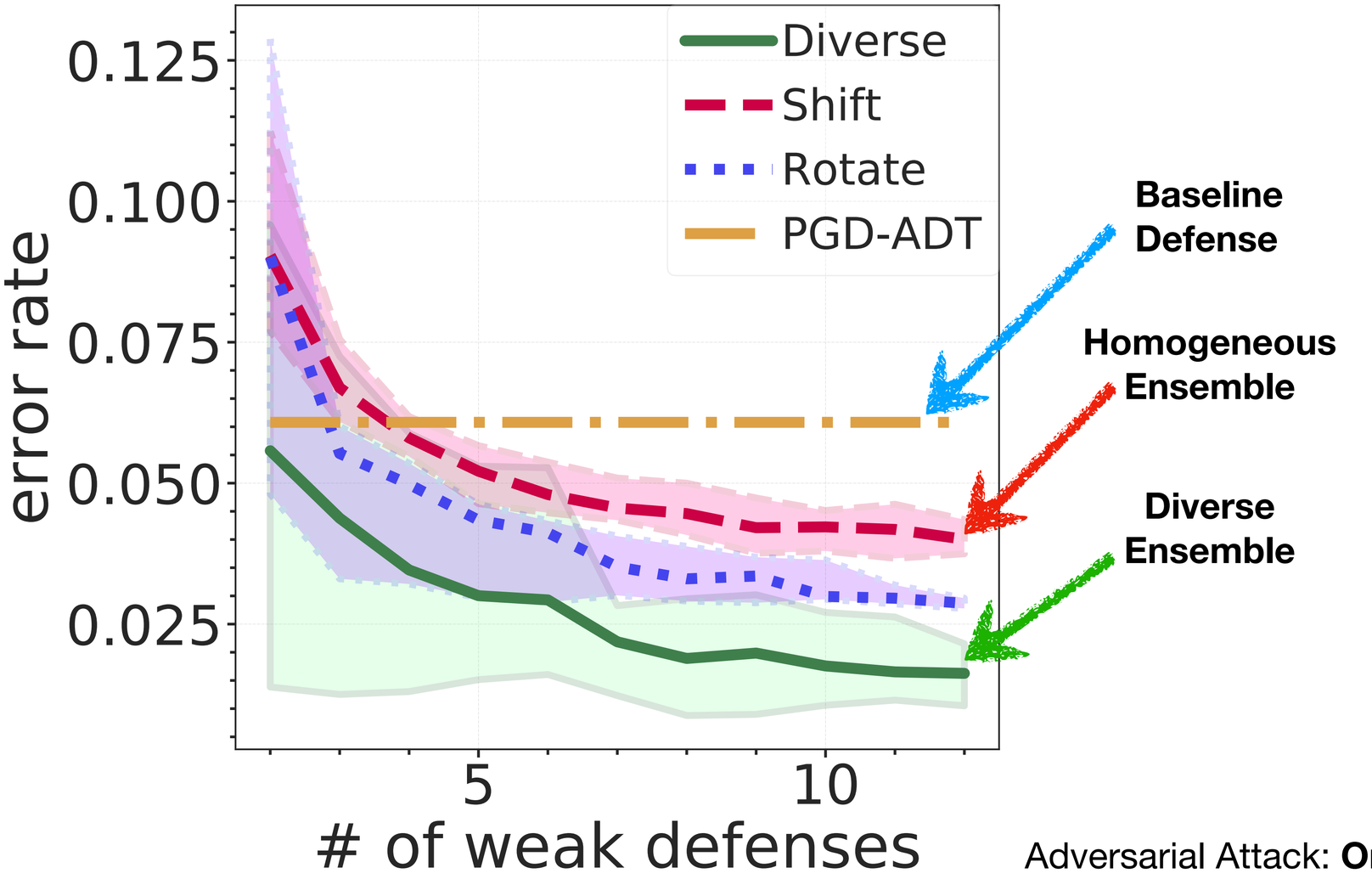


Quality and quantity of weak defenses matter



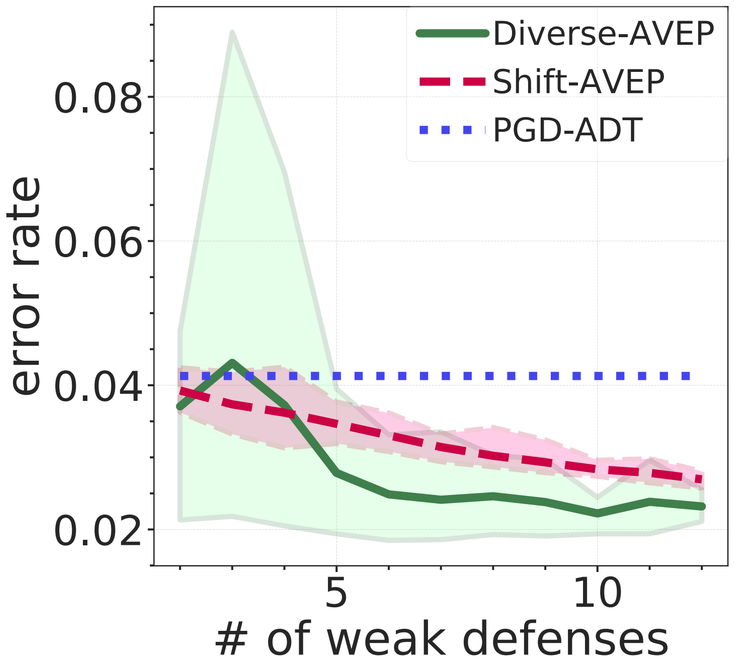
Adversarial Attack: DeepFool

Diversity of weak defenses matters

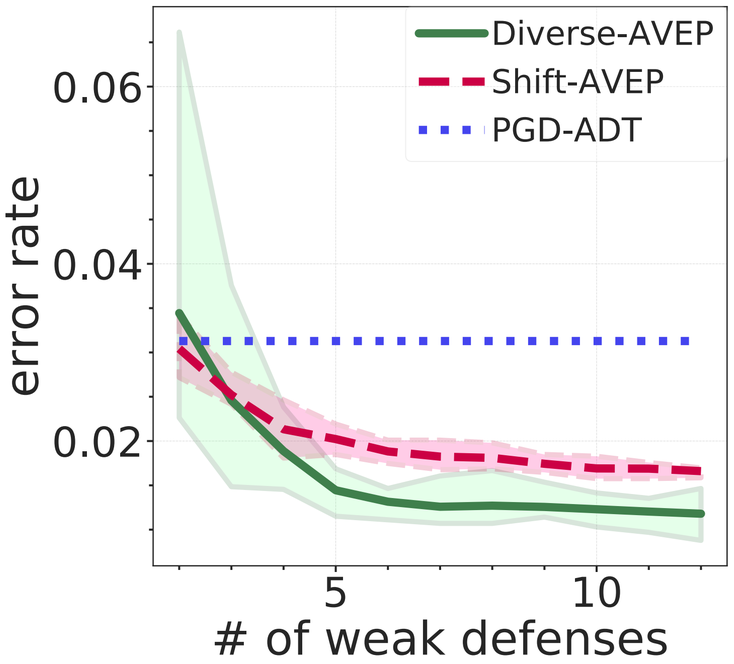


Adversarial Attack: **One-Pixel**
Error Rate Undefended: 0.5588
PGD-ADT: Adversarial Training

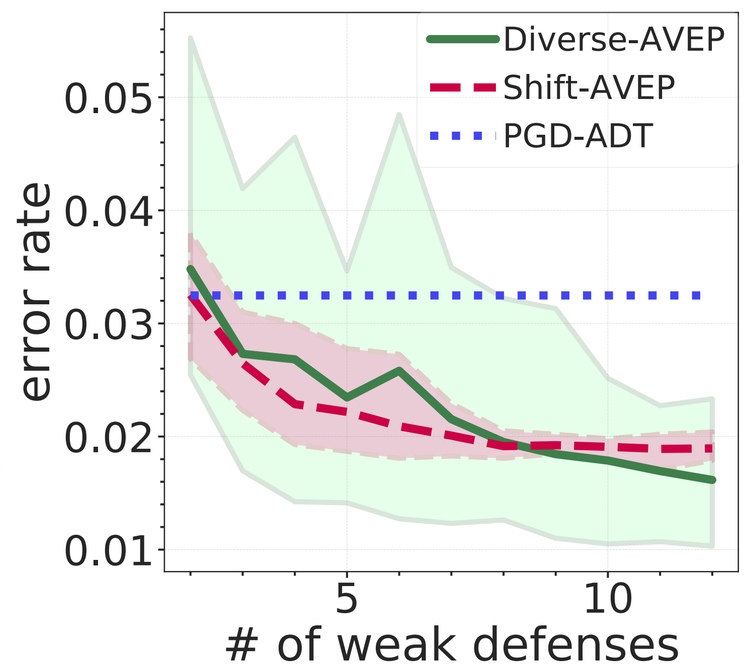
Diversity of weak defenses matters



Adversarial Attack: **BIM_I2**
Error Rate Undefended: 0.92



Adversarial Attack: **MIM**
Error Rate Undefended: 0.94

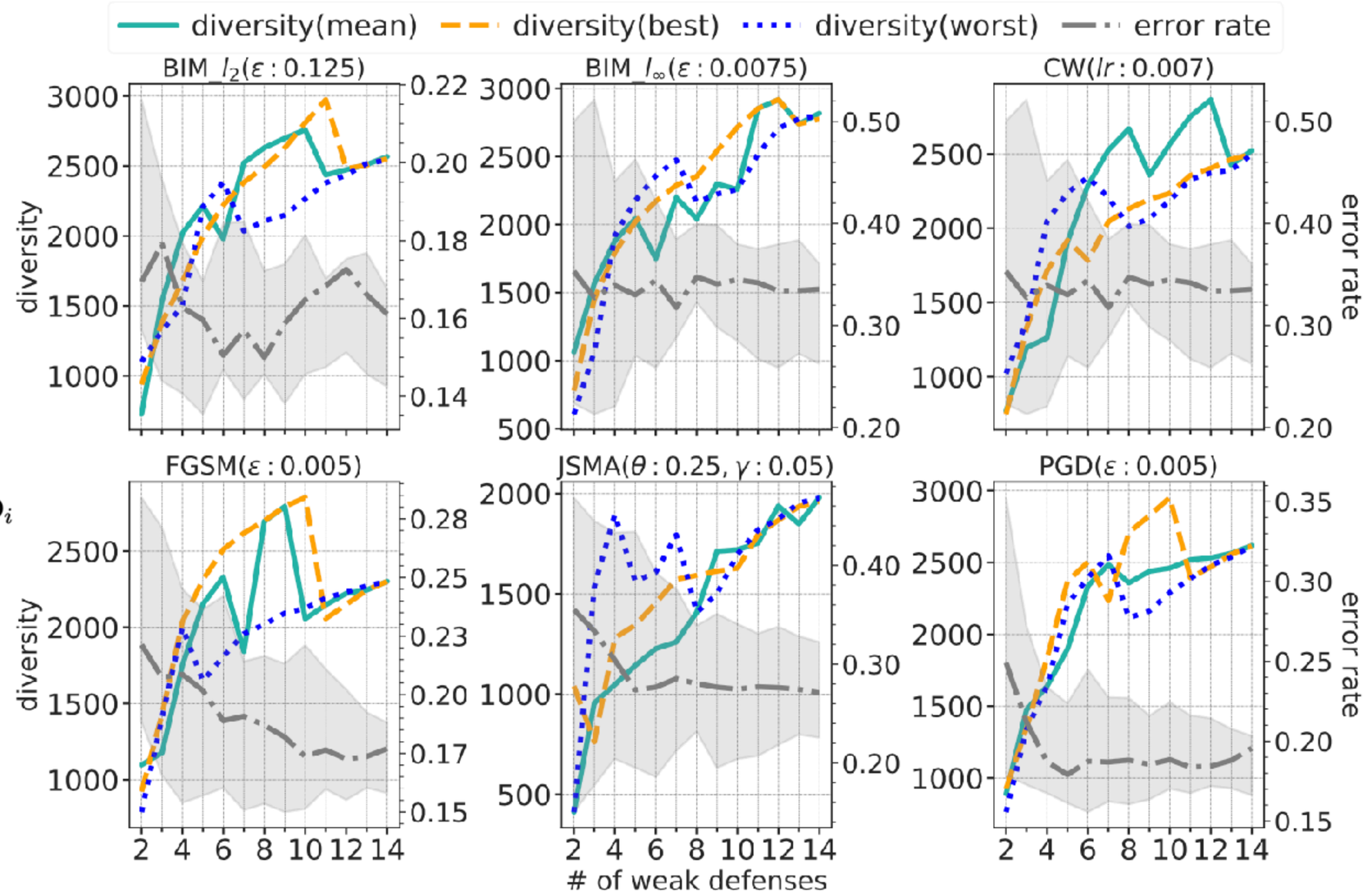


Adversarial Attack: **PGD**
Error Rate Undefended: 0.96

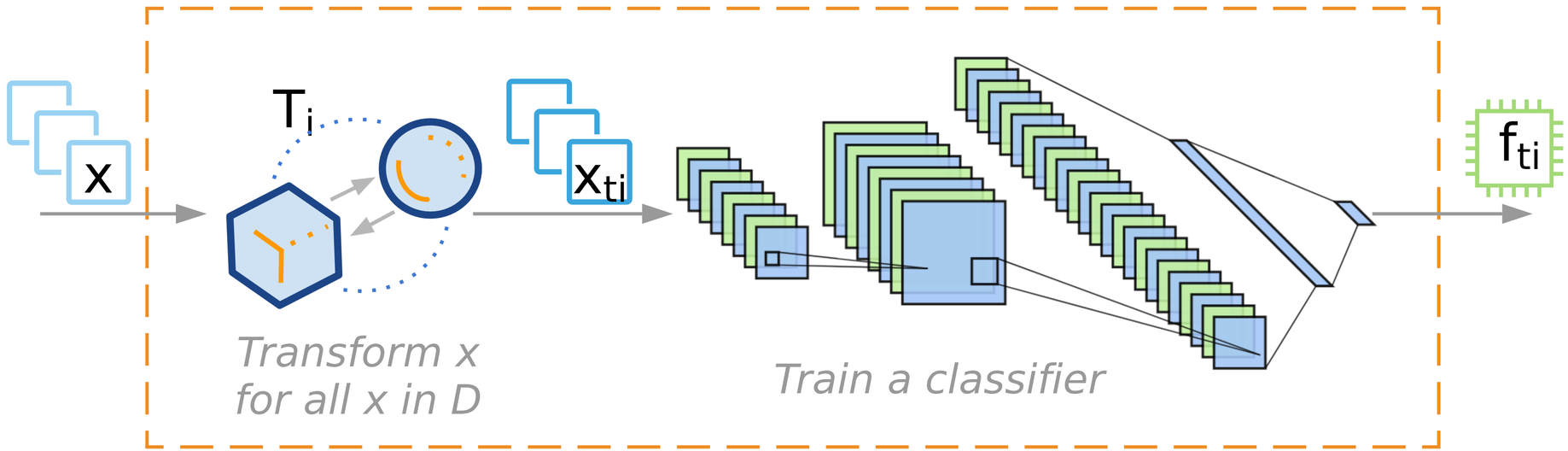
Diversity of weak defenses matters

$$\psi = (\min_{i \in \{1,\ldots,K\}} |S_i|) - (|\bigcap_i S_i|)$$

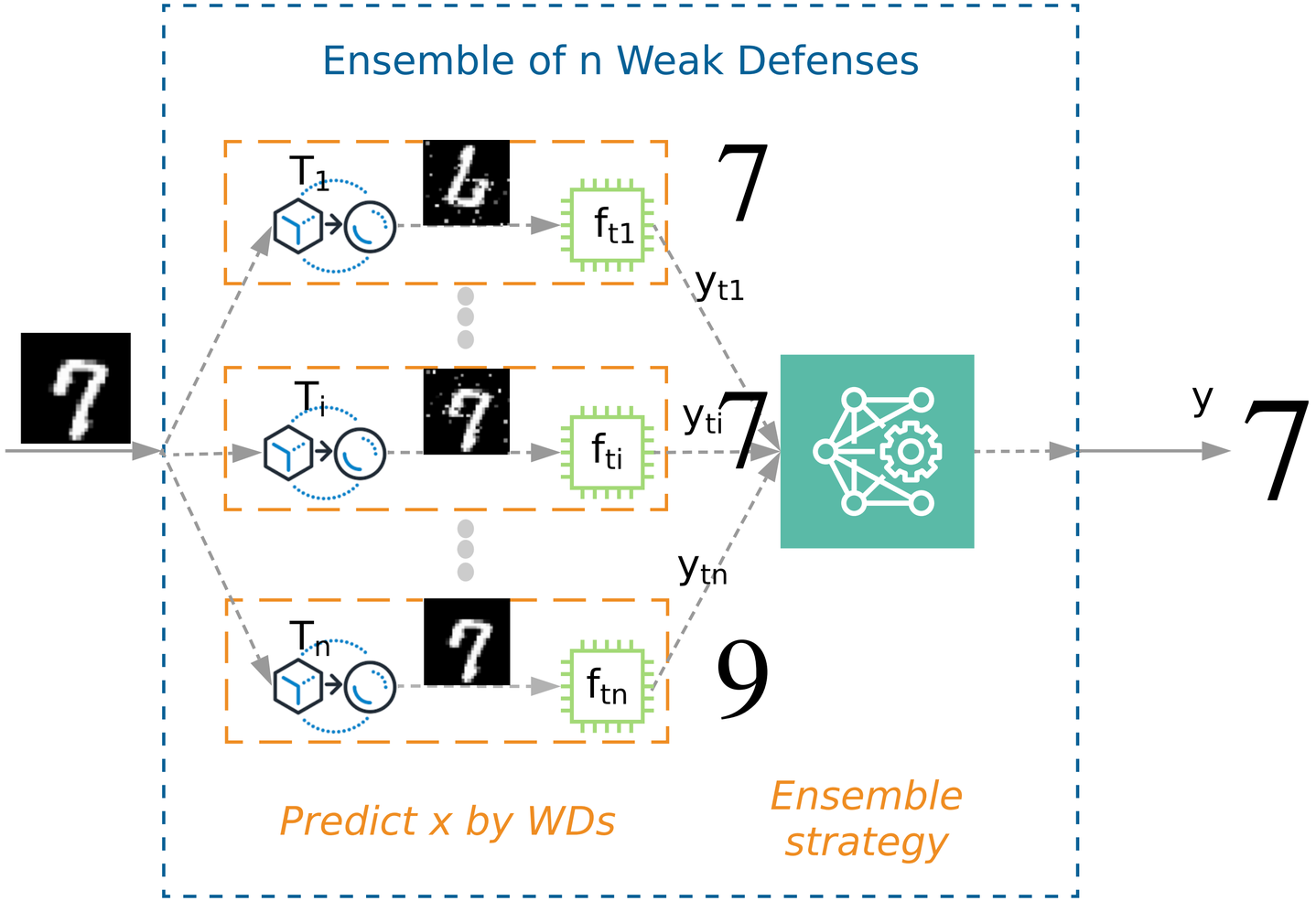
S_i is the set of examples correctly predicted by WD_i



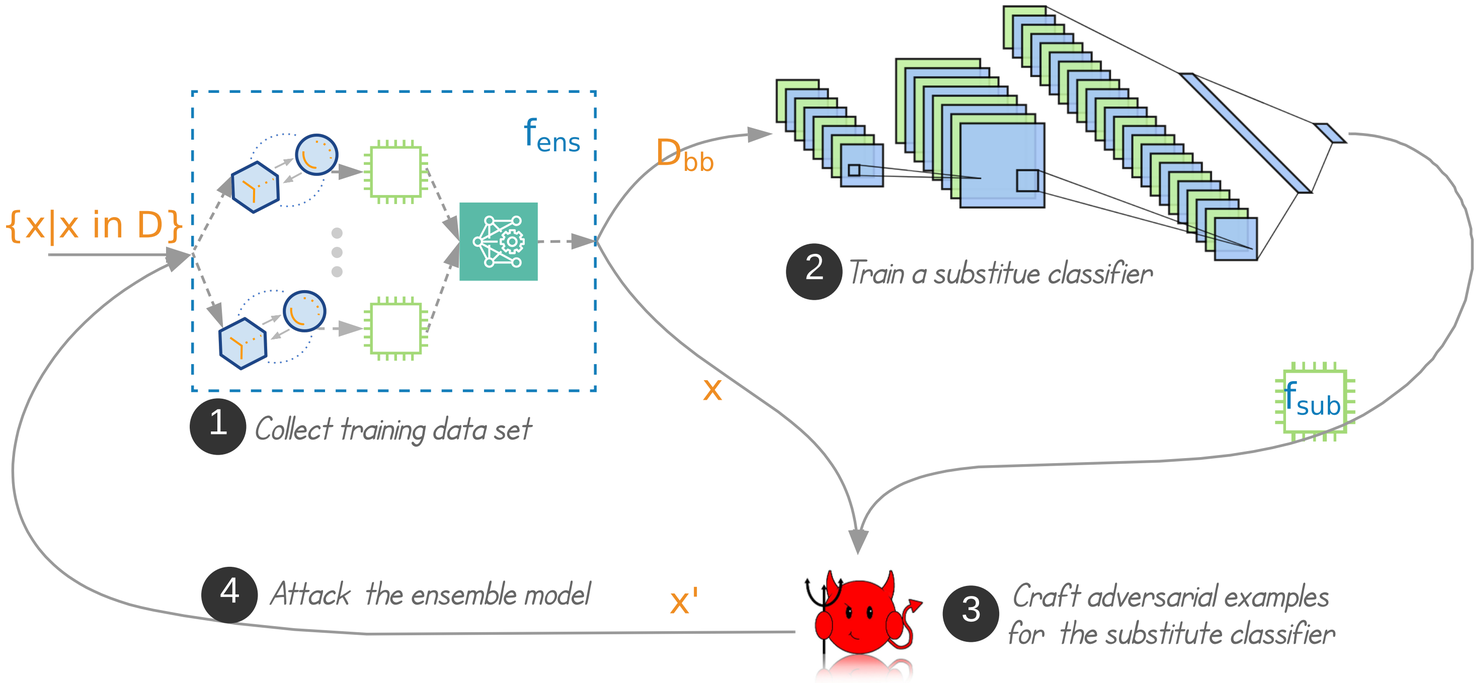
Each weak defense is essentially a model trained on a particular type of transformation



ATHENA produces the final output based on agreement between weak defenses at deployment time



Evaluation

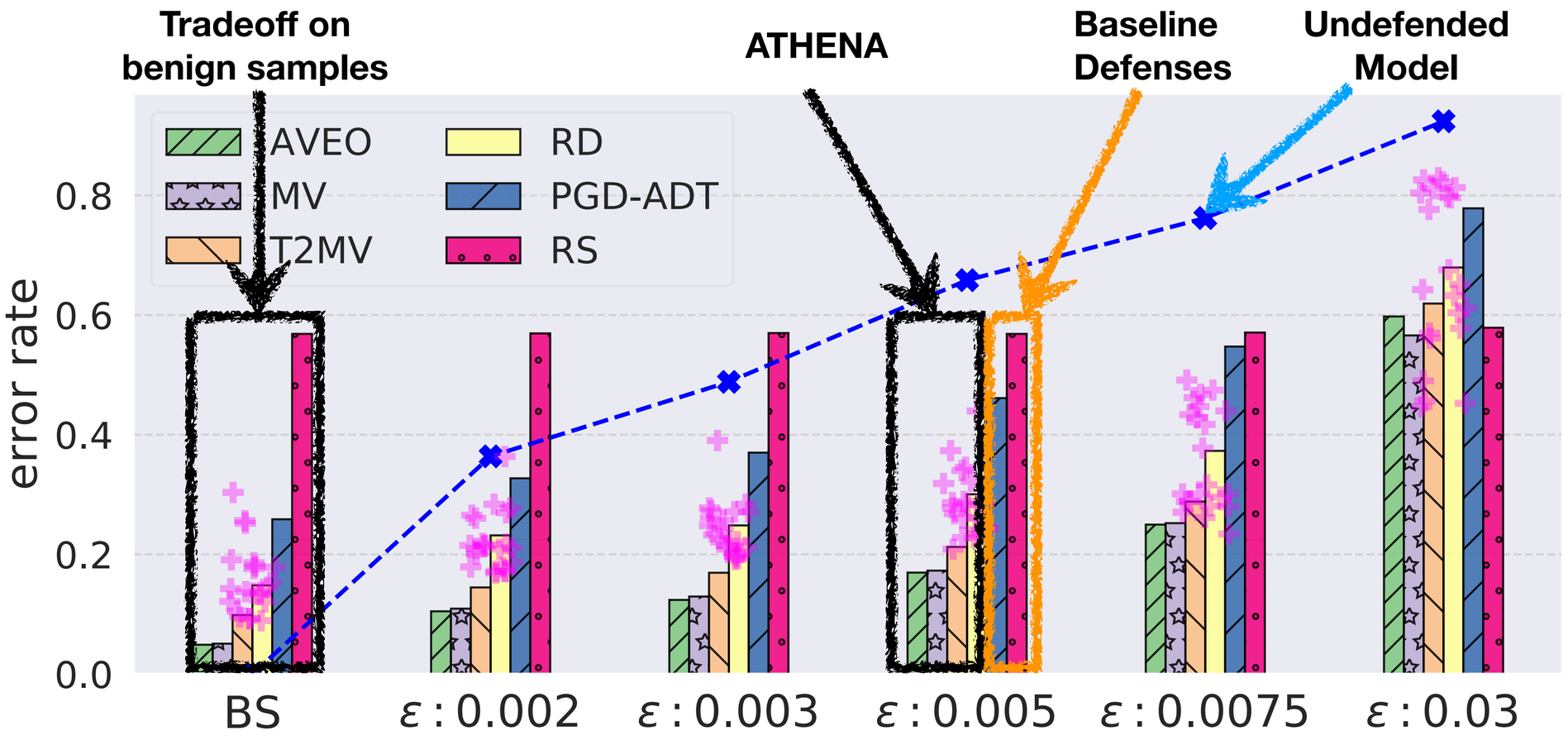

{x|x in D}
f_{ens}
D_{bb}
1 Collect training data set
2 Train a substitue classifier
f_{sub}
x
x'
3 Craft adversarial examples for the substitute classifier
4 Attack the ensemble model

Threat model: What we can assume about the knowledge of the adversary and its strength

Knows the parameters of

	Target Classifier	Existence of Defense	Weak Defenses	Ensemble Strategy
Zero-knowledge	✓	✗	✗	✗
Blackbox	✗	✓	✗	✗
Greybox	✓	✓	✓	✗
Whitebox	✓	✓	✓	✓

Although the effectiveness of each weak defense varies, ATHENA is able to decrease the error rate effectively

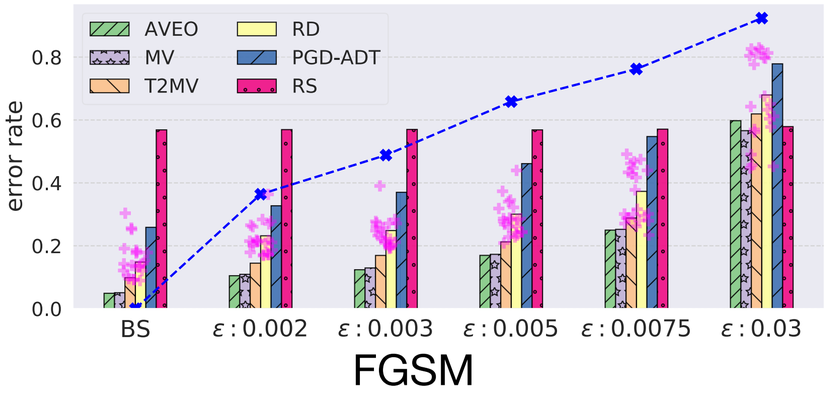


ATHENA (ensemble strategy):
- MV: Majority Voting
- T2MV: Top-2 MV
- AVEO: Average of Output
- RD: Random Defense

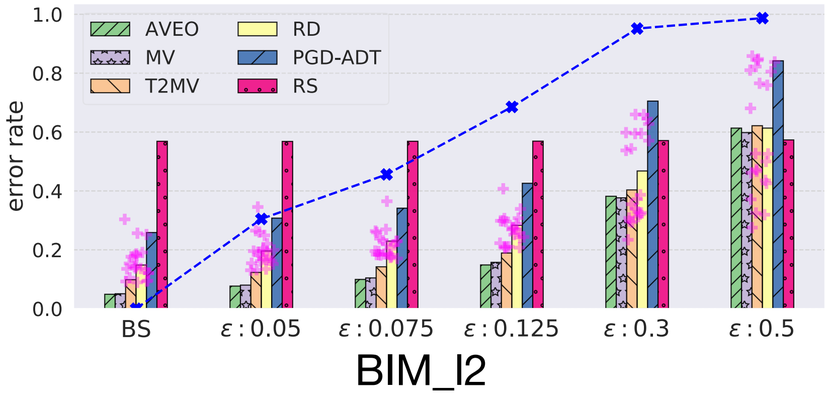
Baseline Defense:
PGD-ADT: Adversarial Training
RS: Randomized Smoothing

Adversarial Attack: FGSM
Model: 28×10 Wide ResNet
Dataset: CIFAR100

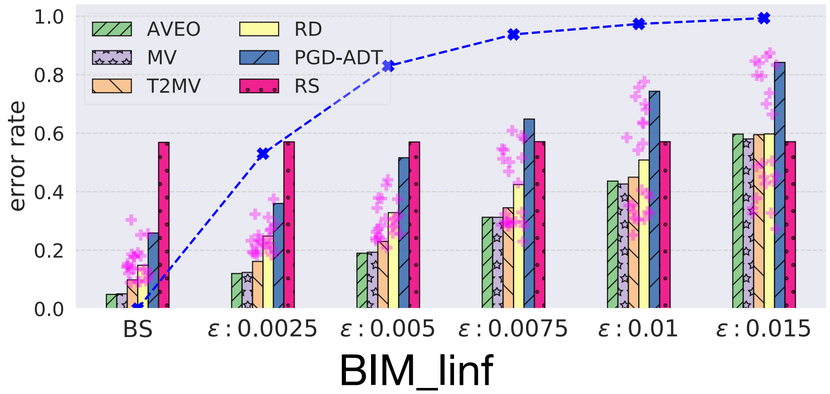
Although the effectiveness of each weak defense varies, ATHENA is able to decrease the error rate effectively



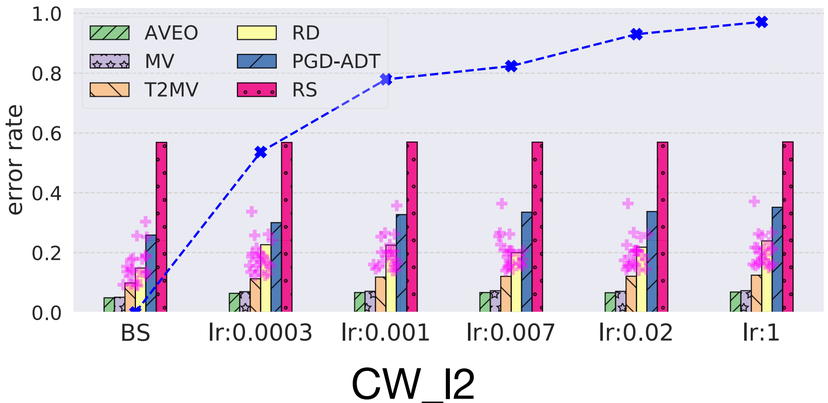
FGSM



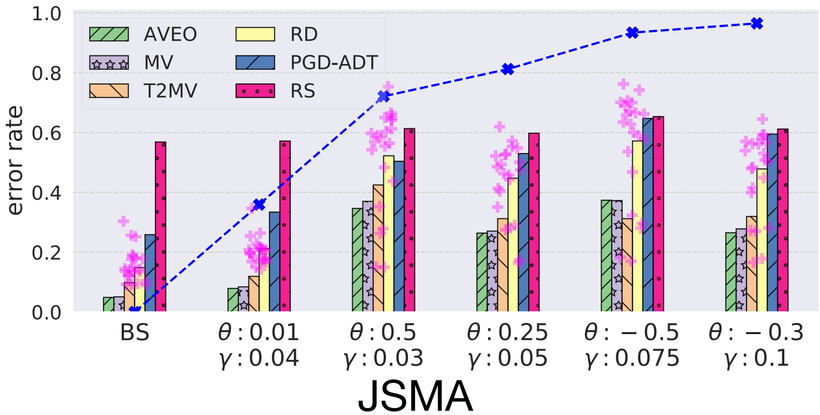
BIM_l2



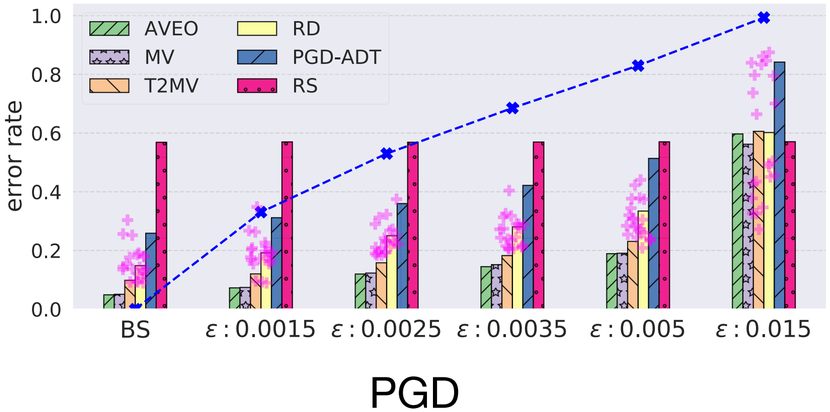
BIM_linf



CW_l2



JSMA



PGD

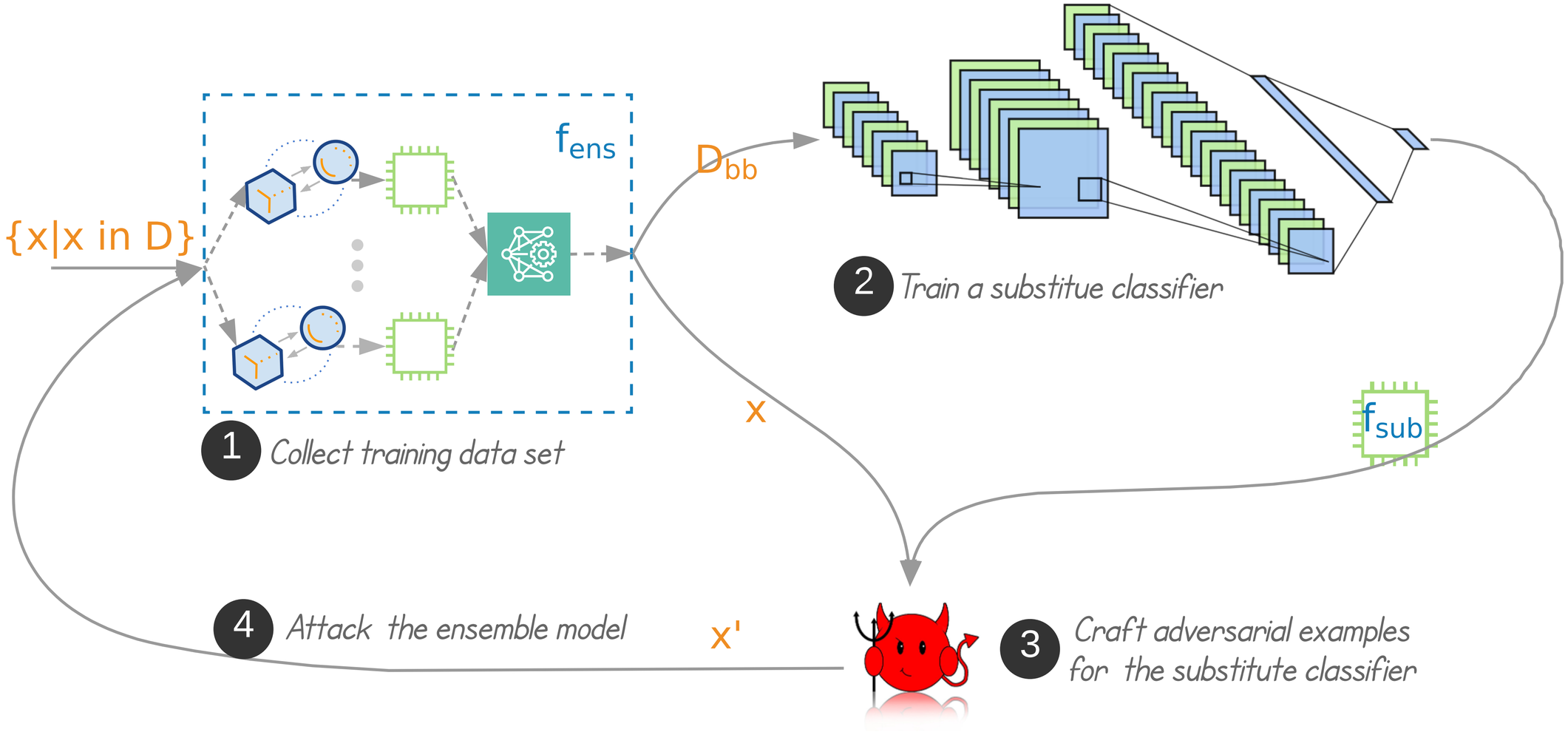
Model: 28×10 Wide ResNet
Dataset: CIFAR100

Threat model

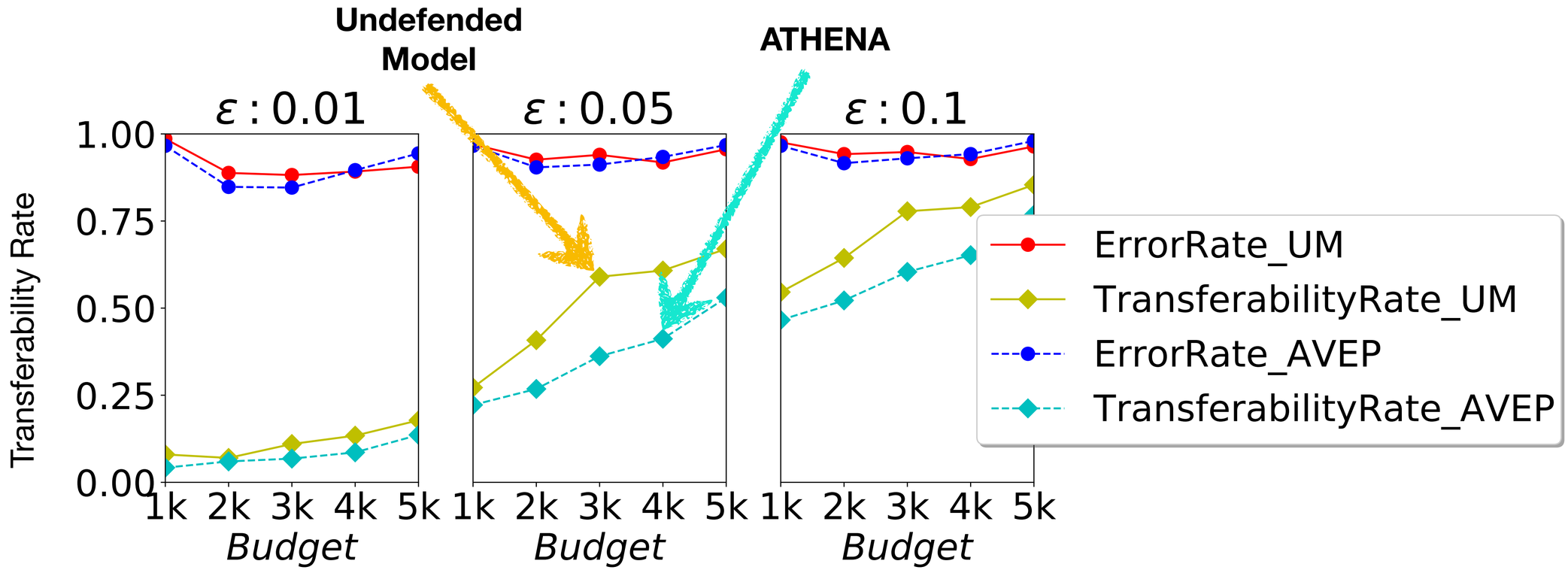
Knows the parameters of

	Target Classifier	Existence of Defense	Weak Defenses	Ensemble Strategy
Zero-knowledge	✓	✗	✗	✗
Blackbox	✗	✓	✗	✗
Greybox	✓	✓	✓	✗
Whitebox	✓	✓	✓	✓

Blackbox attack: The transferability-based approach

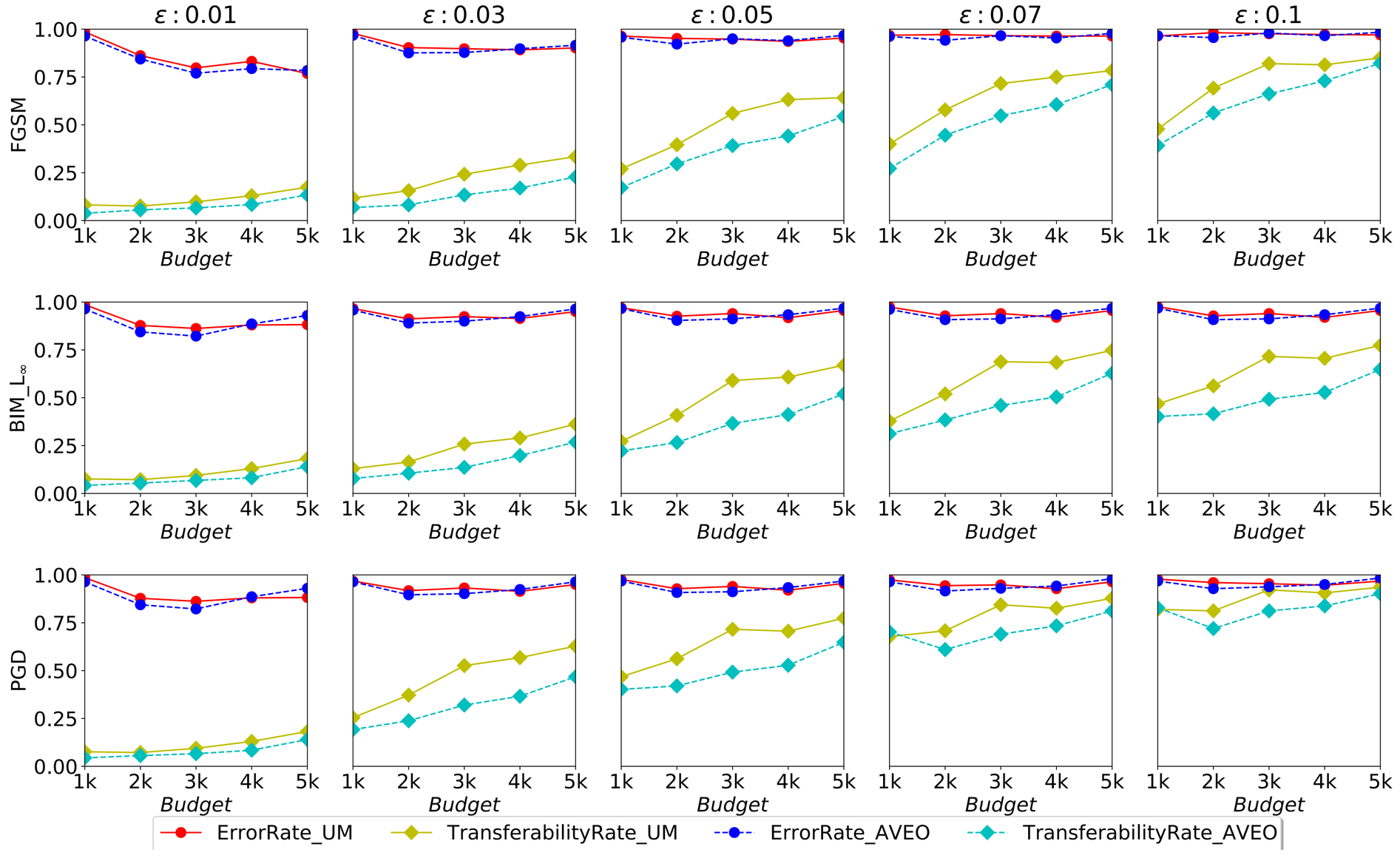


ATHENA lowered the “transferability” of adversarial examples from the surrogate model to the target model



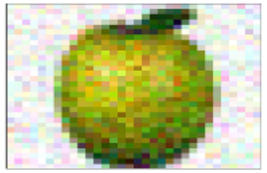
Adversarial Attack: BIM_linf
Model: 28×10 Wide ResNet
Dataset: CIFAR100

ATHENA lowered the transferability of adversarial examples from the surrogate model to the target model

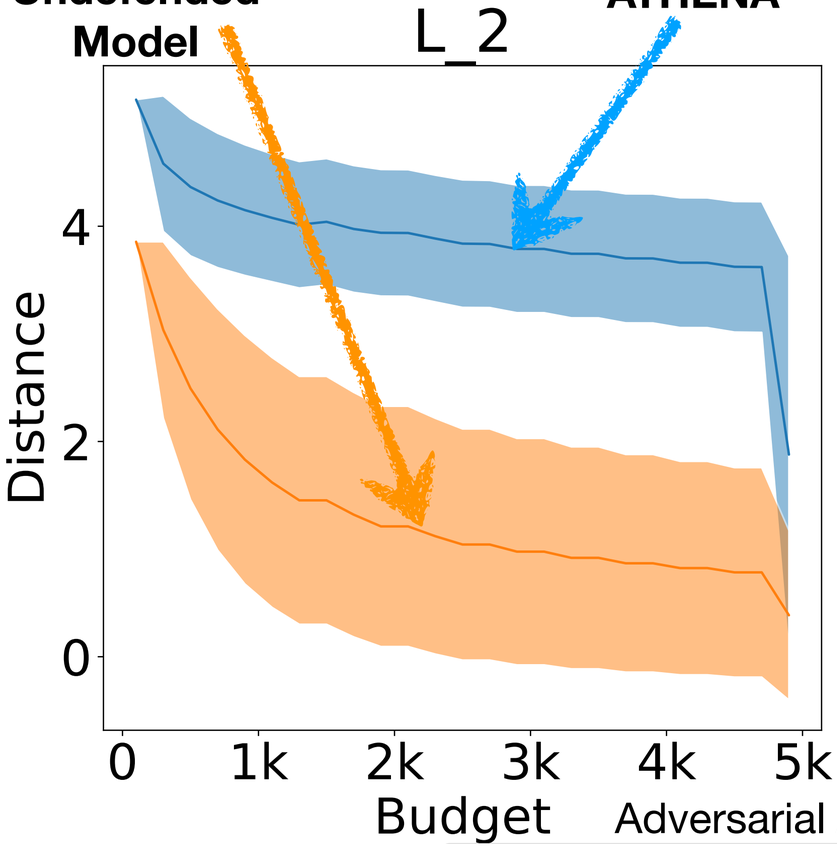


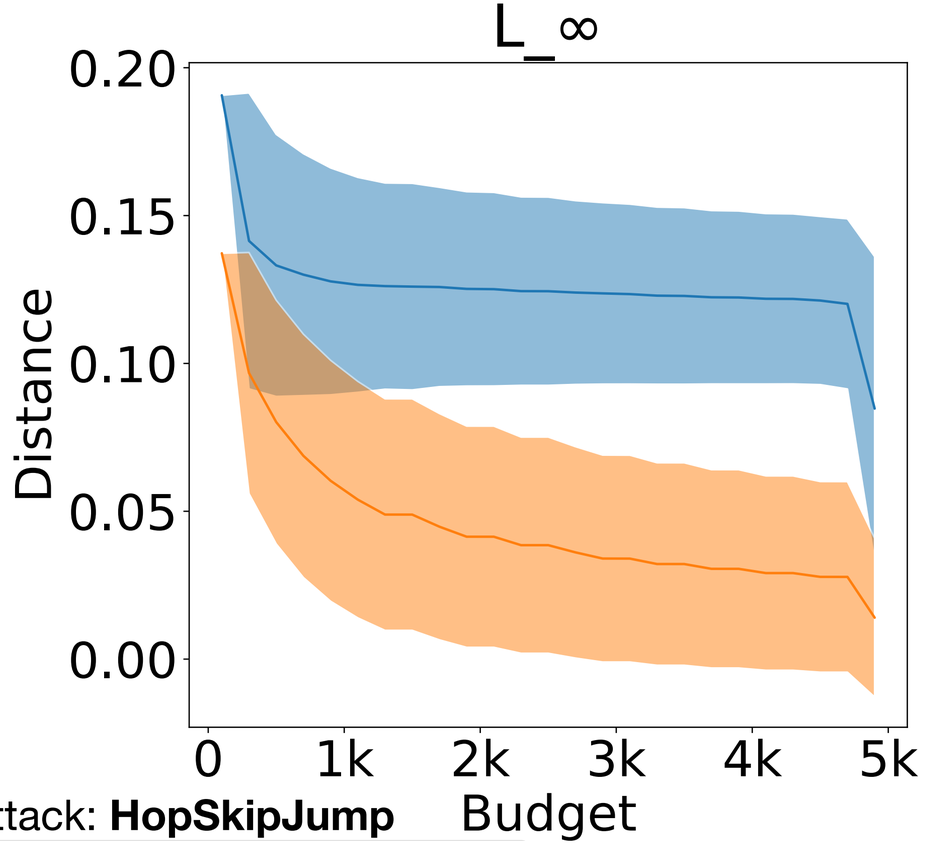
ATHENA forces the “optimization-based” blackbox attack to generate adversarial examples with larger perturbation

Undefended Model



ATHENA





Adversarial Attack: **HopSkipJump**

Model: 28×10 Wide ResNet

Dataset: CIFAR100

Threat model

Knows the parameters of

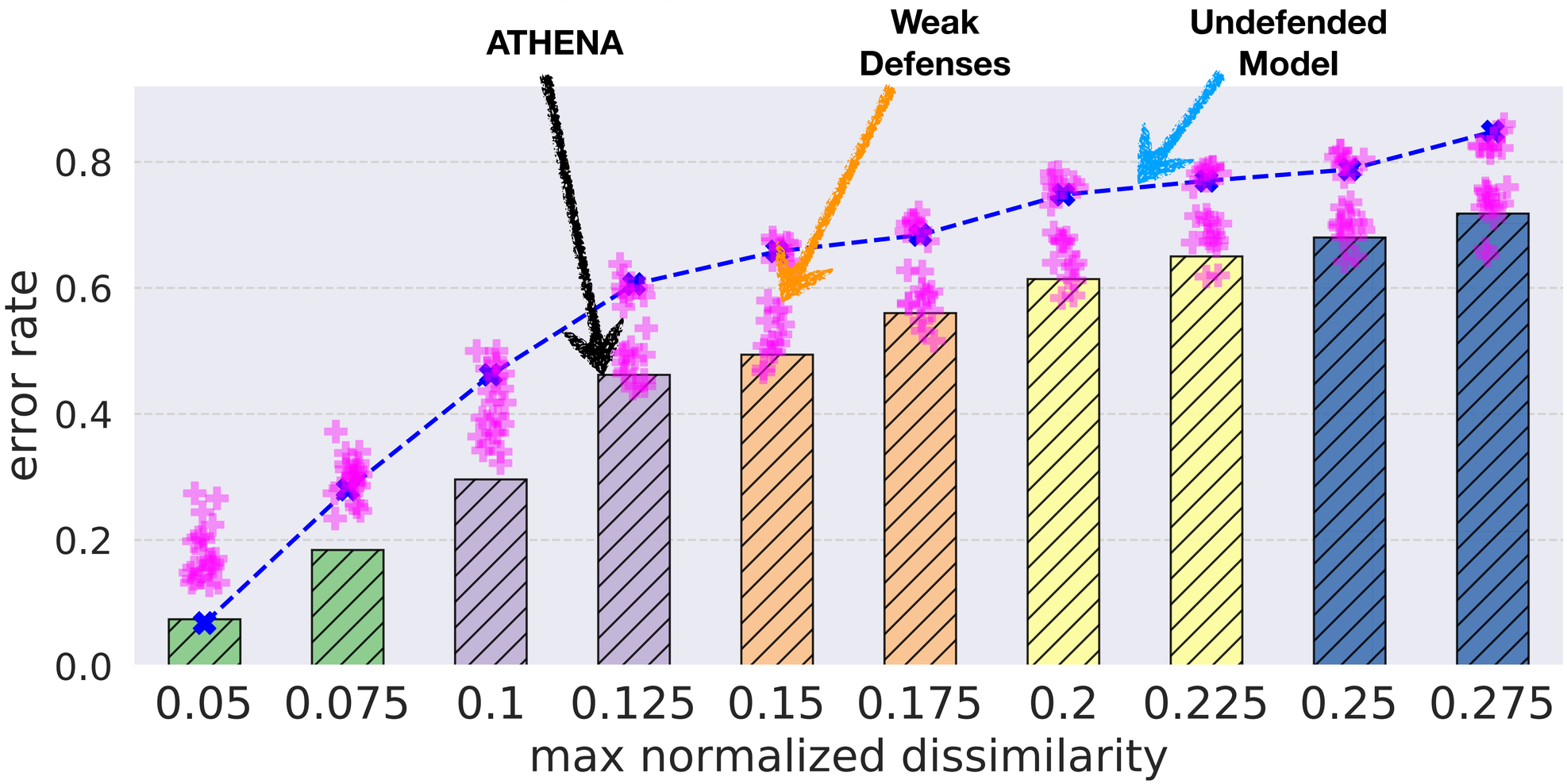
	Target Classifier	Existence of Defense	Weak Defenses	Ensemble Strategy
Zero-knowledge	✓	✗	✗	✗
Blackbox	✗	✓	✗	✗
Greybox	✓	✓	✓	✗
Whitebox	✓	✓	✓	✓

Greedy White-box Attack

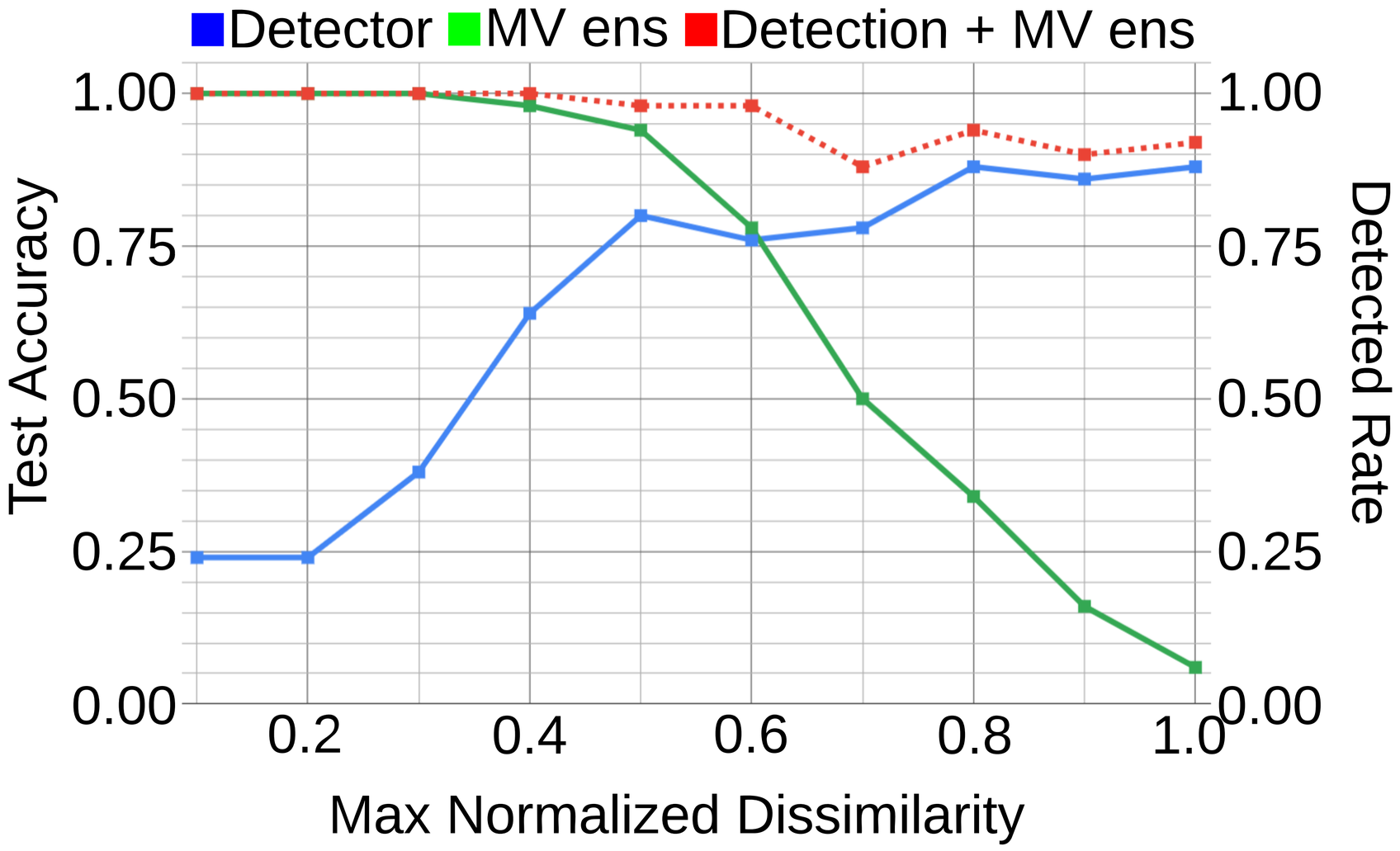
Algorithm 1: Crafting white-box AEs (Greedy)

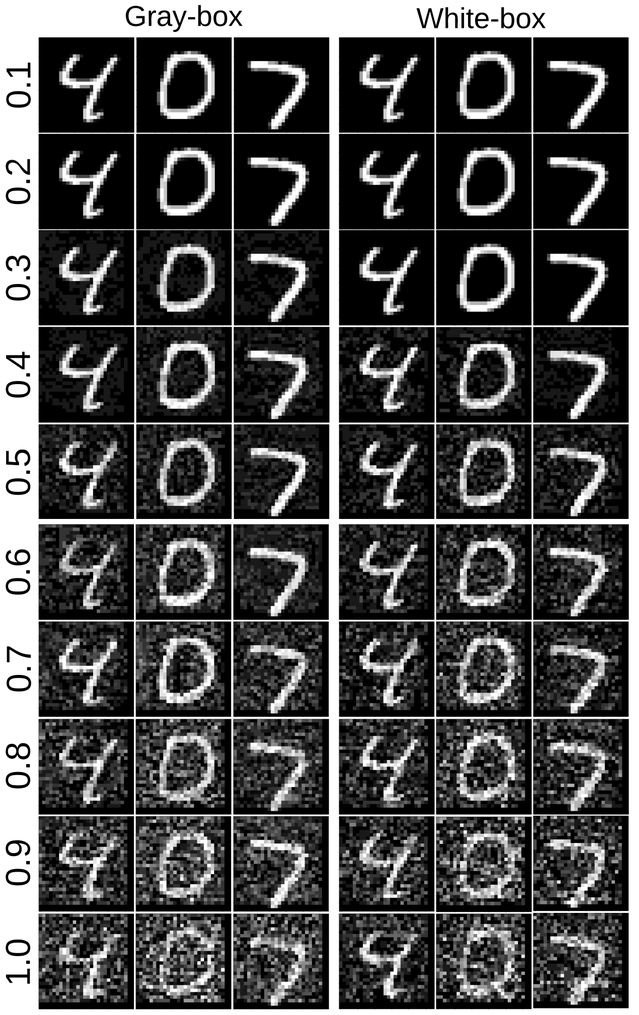
```
   input     : x, y, attacker, N, max_dissimilarity
1  F_fooled ← {};
2  F_cand ← all weak defenses;
3  x' ← x;
4  while size(F_fooled) < N do
5      f_target ← pickTarget(F_cand, strategy);
6      // getPerturbation(x) returns ||x − x'||_2
7      perturbation ← attacker.getPerturbation(f_target, x');
8      x'_tmp ← x' + perturbation
9      // dissimilairity(x', x) returns the normalized l_2
           dissimilarities between x' and x
10     if dissimilarity(x'_tmp, x) > max_dissimilarity then
11         break;
12     end
13     for f_{t_i} in F_cand do
14         if y ≠ f_{t_i}(x'_tmp) then
15             addModel(F_fooled, f_{t_i});
16             removeModel(F_cand, f_{t_i});
17         end
18     end
19     x' ← x'_tmp;
20 end
21 return x';
```

A strong adaptive white-box adversary may be able to successfully bypass the defense

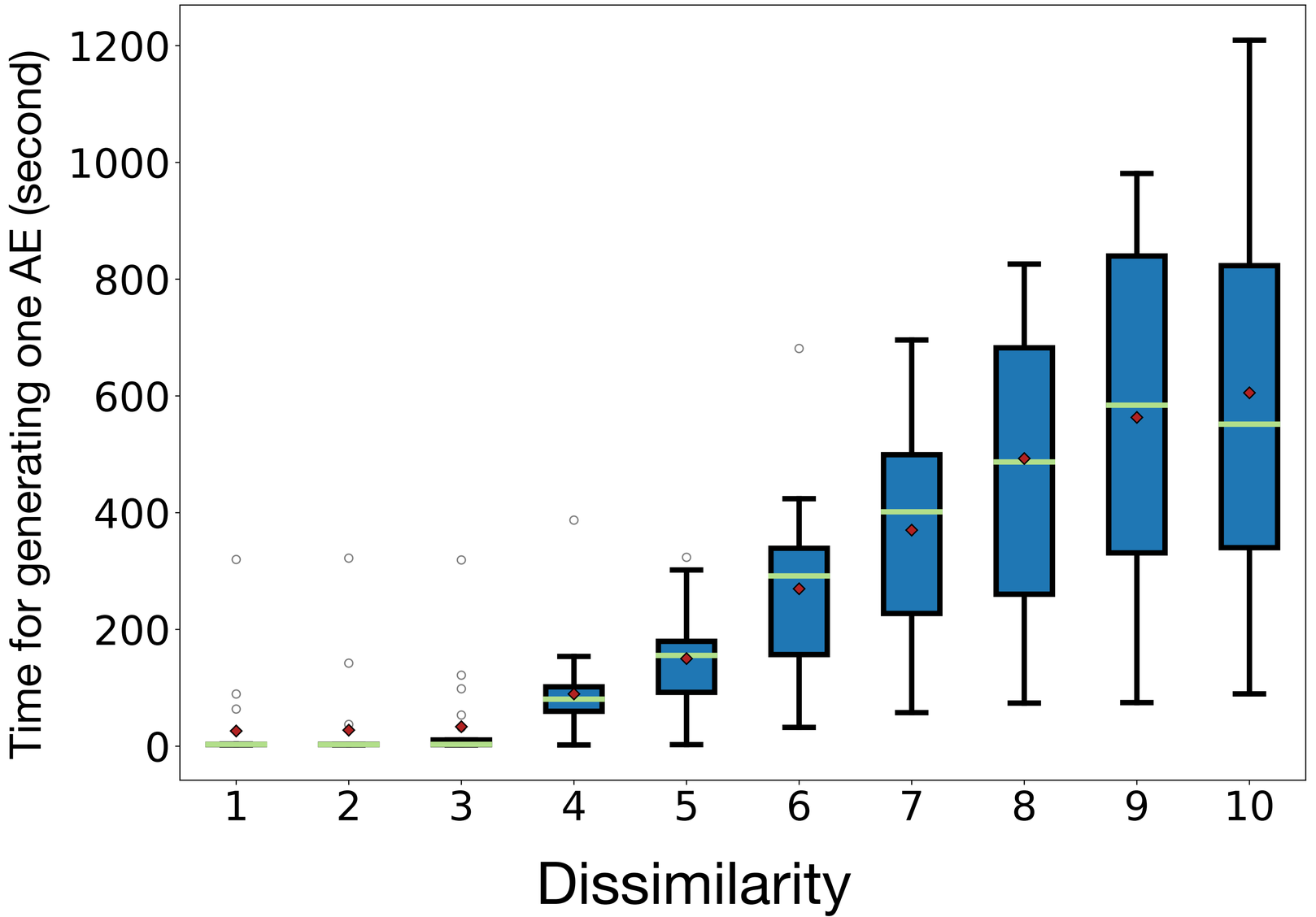


However, it becomes very easy to “detect” such attacks, so a defense+detection would be robust





Also, it comes with a high cost



An Adaptive Adversary for ATHENA

Standard Adversarial Attack

$$max_{\delta} \mathcal{L}(\theta, x+\delta, y)$$

$$||\delta||_p \leq \epsilon$$

EOT (Expectation Over Transformation)

$$x' = \underset{x'}{\text{argmax}}\, \mathbb{E}_{t\sim T}[\log f(y_t|t(x')]$$

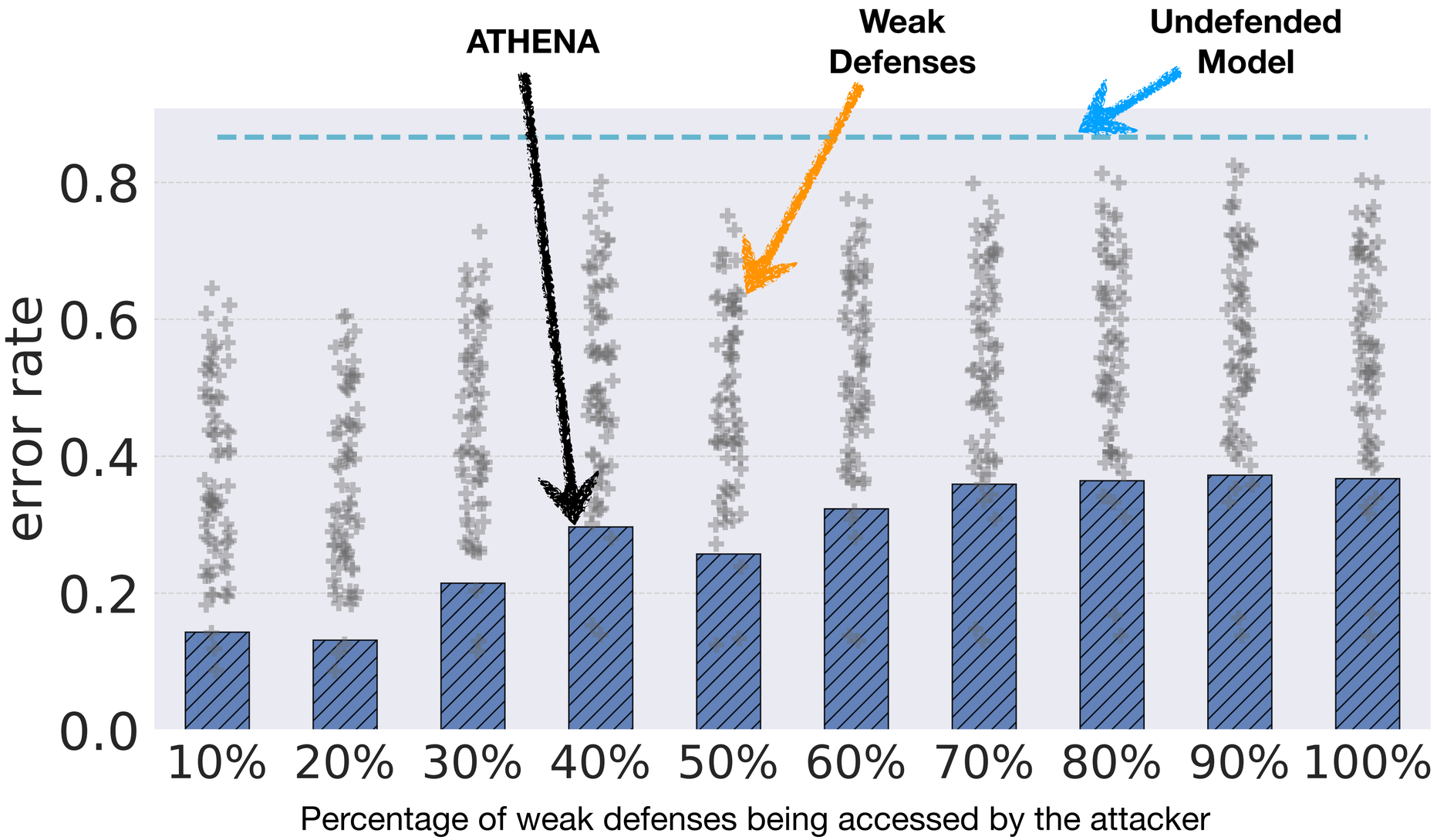
$$\text{s.t. } \mathbb{E}[d(t(x'), t(x)] < \epsilon, x \in [0,1]^d,$$

Extending EOT for Ensembles

$$\underset{x'}{\text{argmax}}\, \frac{1}{K} \Sigma_{i=1}^{K} \mathbb{E}_{t\sim T}[\log f_i(y_t|t(x')]$$

$$\text{s.t. } \mathbb{E}_{t\sim T}[d(t(x'), t(x))] < \epsilon, x \in [0,1]^d,$$

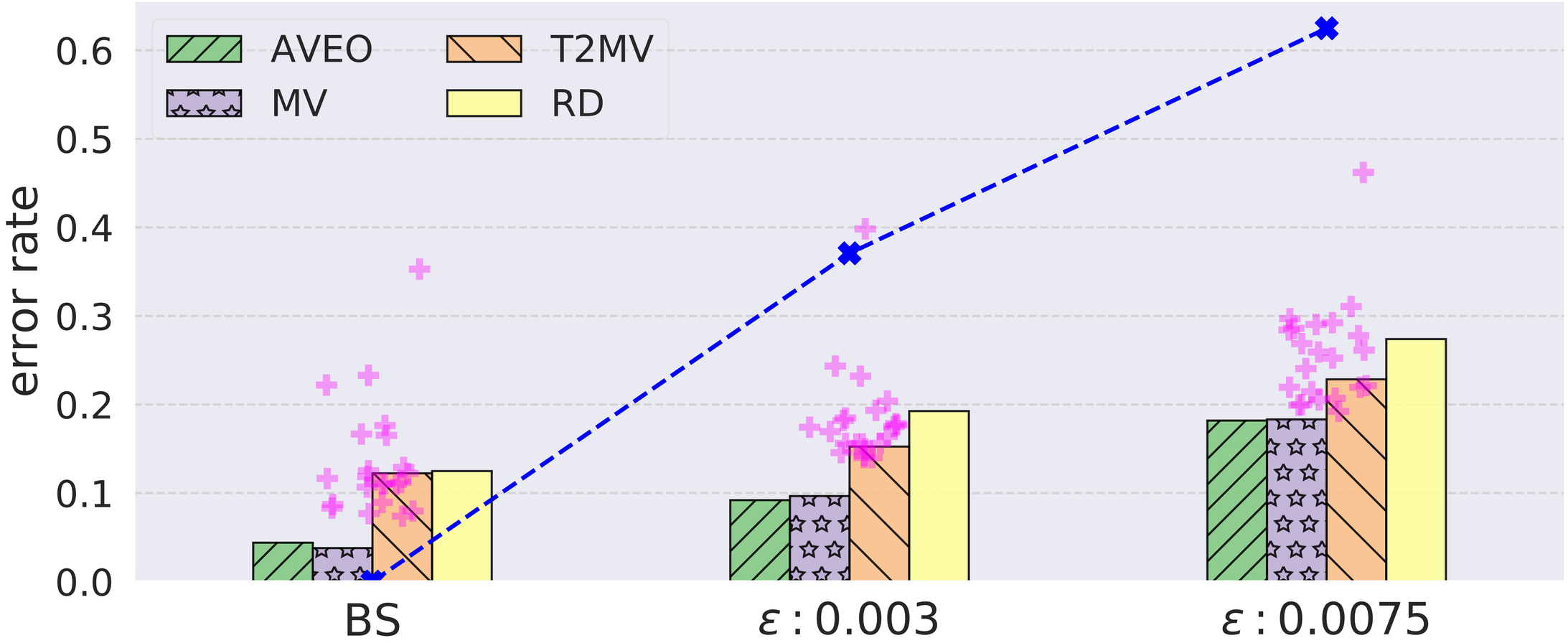
As the adaptive attacker knows more about ATHENA, it can launch more successful attacks



Is ATHENA a general defense?

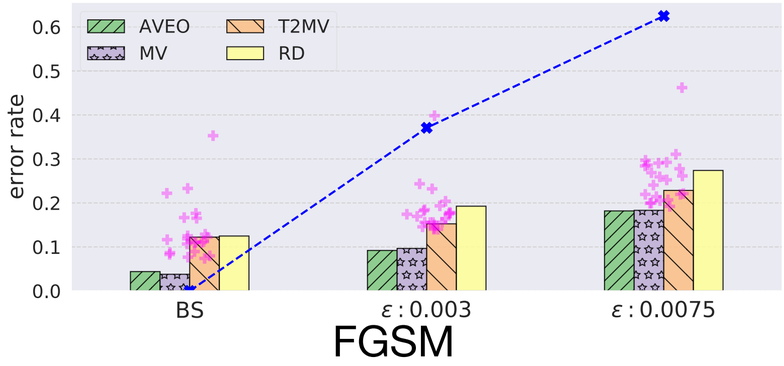
Will it work with different types of machine learning models?

ATHENA performs similarly well with other types of machine learning models (DNNs, SVMs, RF)

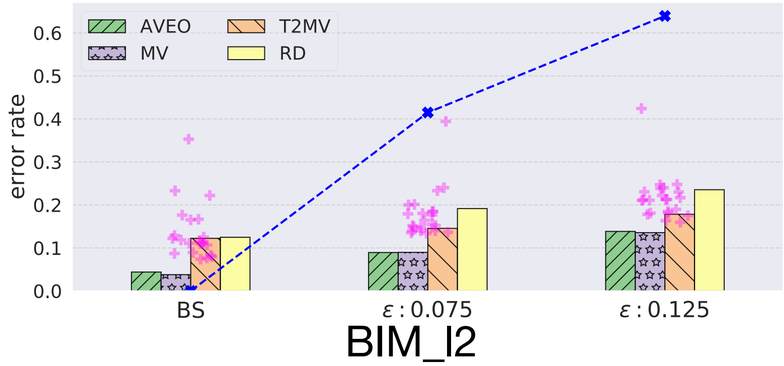


Adversarial Attack: FGSM
Model: ResNet Shake-Shake reg
Dataset: CIFAR100

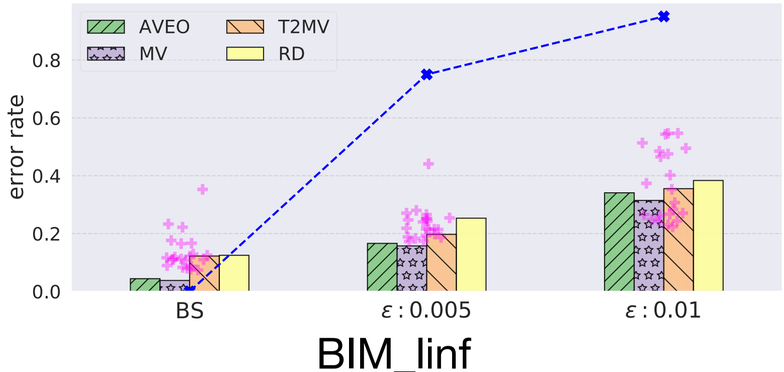
ATHENA is effective similarly with other types of models



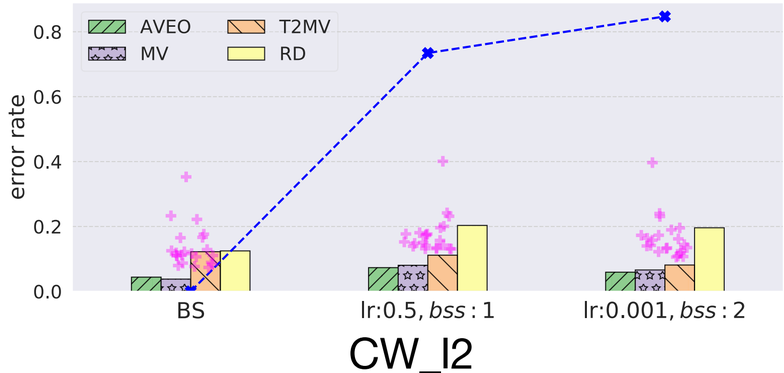
FGSM



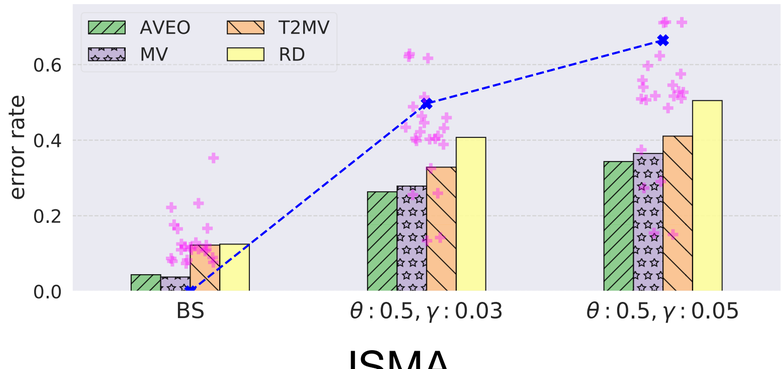
BIM_l2



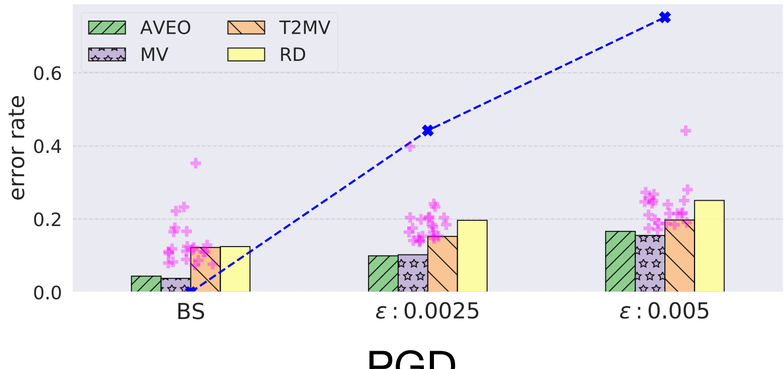
BIM_linf



CW_l2



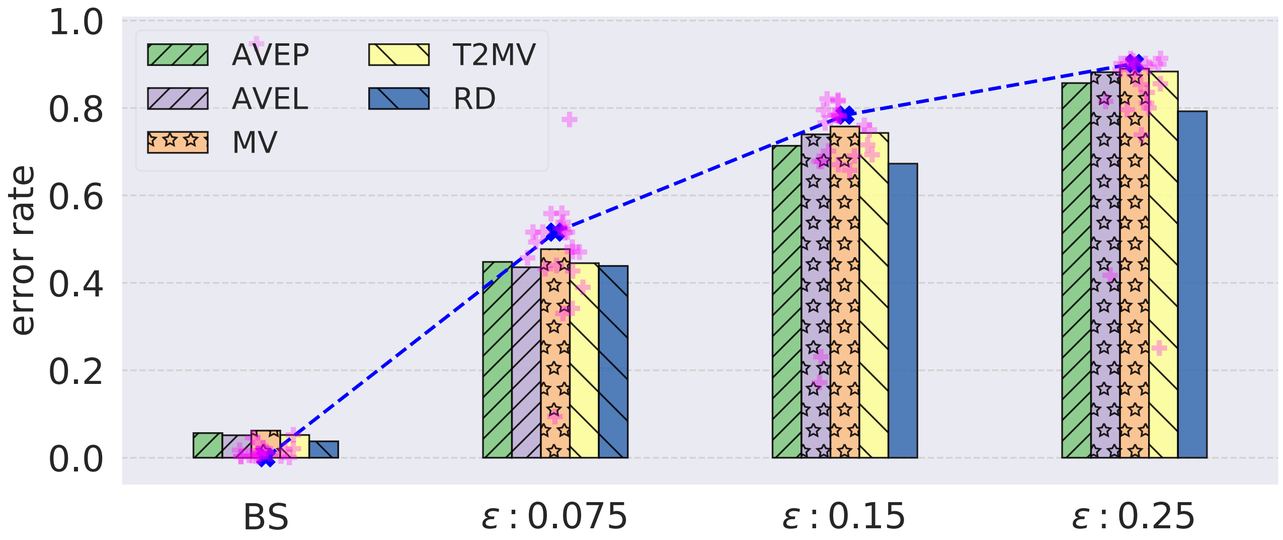
JSMA



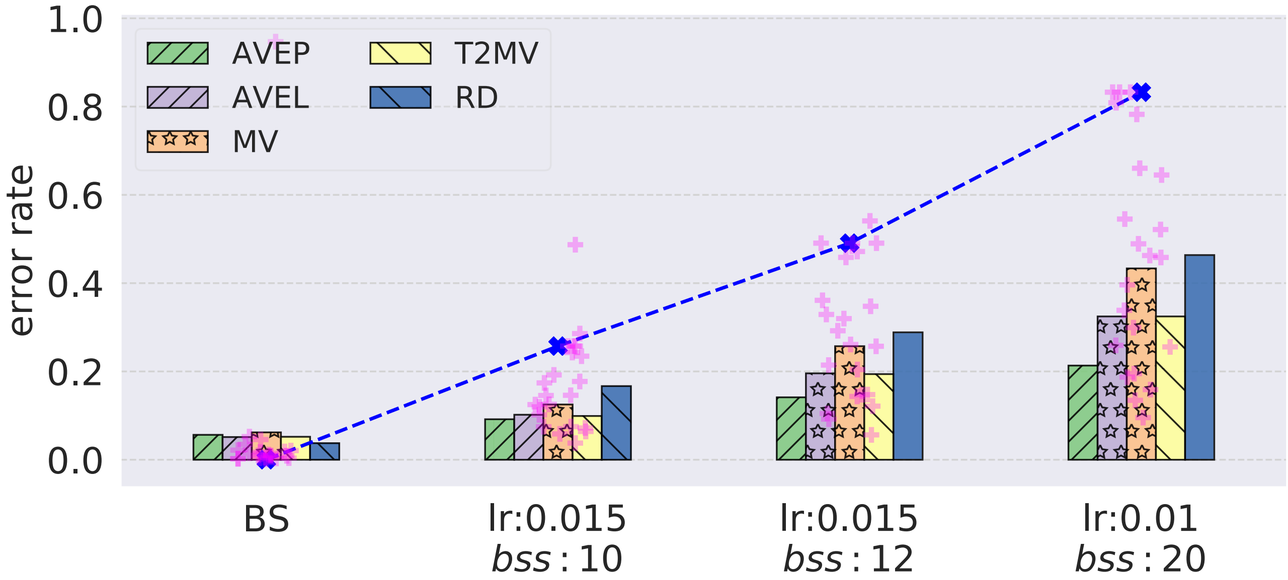
PGD

Model: ResNet Shake-Shake reg.
Dataset: CIFAR100

However, the effectiveness of defense may vary depending on the type of models



Adversarial Attack: FGSM
Model: SVM
Dataset: MNIST

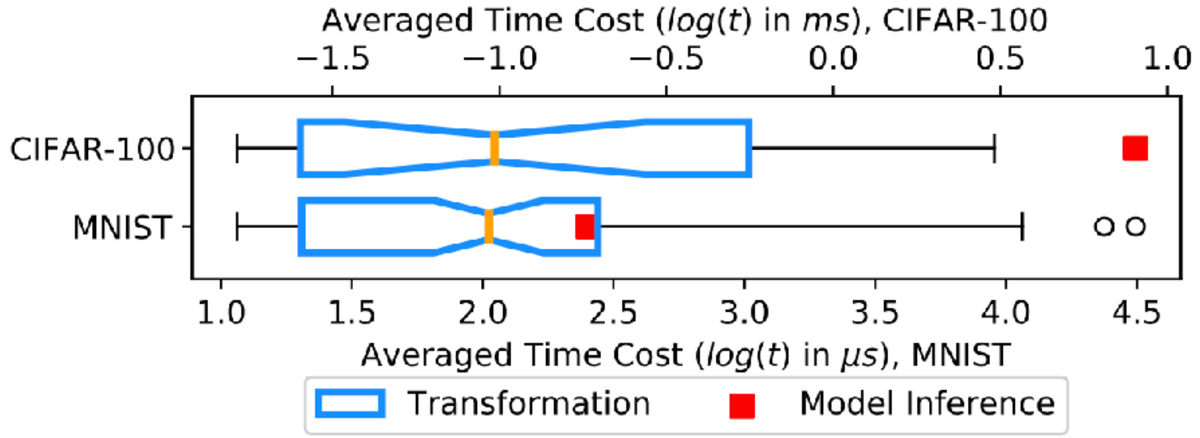


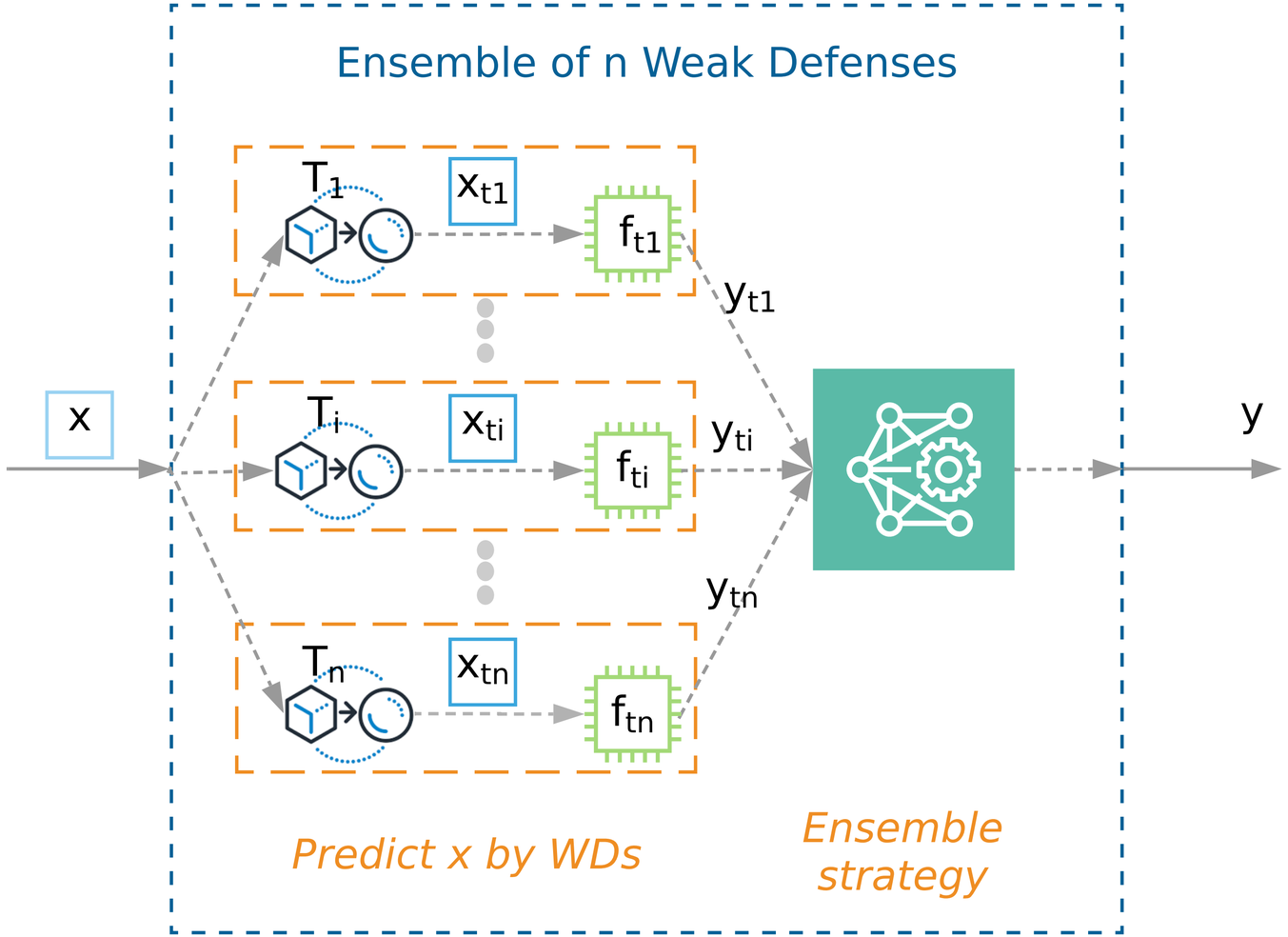
Adversarial Attack: CW_l2
Model: SVM
Dataset: MNIST

What is the overhead of ATHENA?

- Memory
- Inference Time

The memory overhead of ATHENA is linear with number of WDs, the inference time is on par with model inference





ATHENA is:

- Flexible
- Extensible
- General
- Moderate overhead

Website

https://softsys4ai.github.io/athena/

30GB experimental data

ATHENA

A Framework based on Diverse Weak Defenses for Building Adversarial Defense

By

Ying Meng, Jianhai Su, Jason M O'Kane, and Pooyan Jamshidi

AISys

arXiv Preprint	Code GitHub	CSCE 585 Project	HelloWorld Tutorial