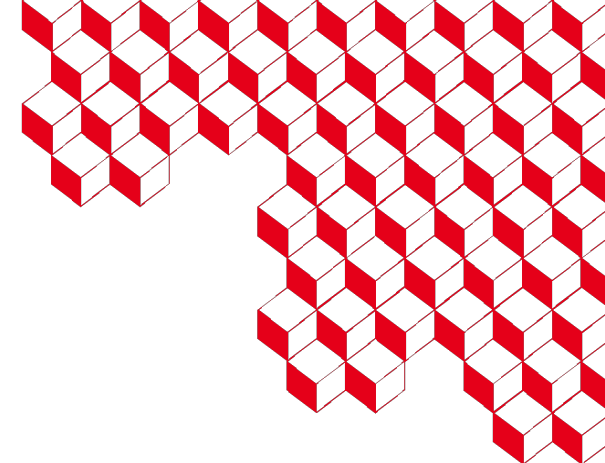




**PhD Defense, 19/12/2025**



# **Over-Approximating Neural Networks for Verification, Robustness, and Explainability**

Serge DURAND (CEA). PhD Candidate ED-STIC, Paris-Saclay

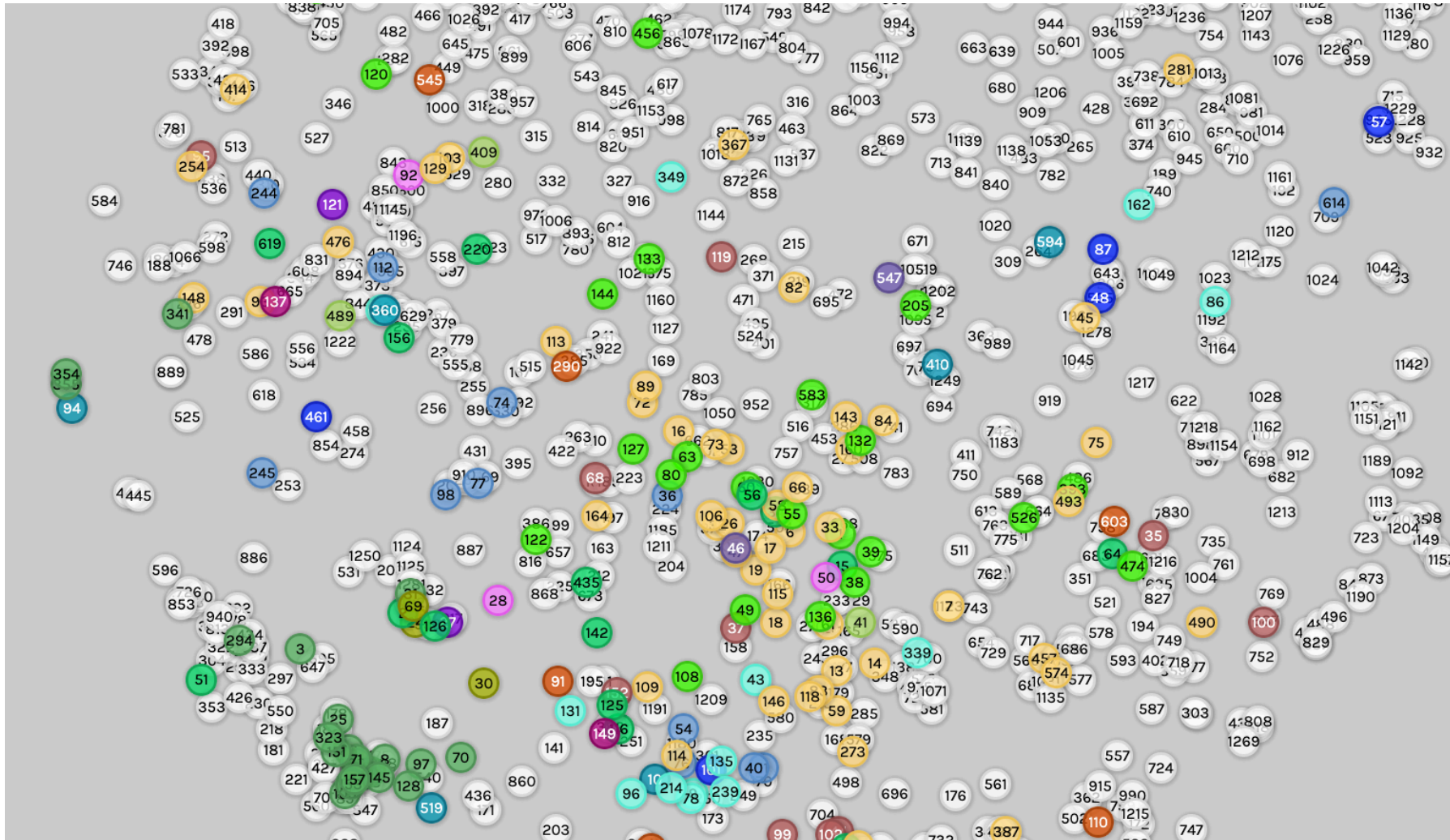
Co-supervisor: Zakaria CHIHANI (DILS/LSL)

Co-supervisor: Caterina URBAN (ENS/Inria ANTIQUE)

Director: François TERRIER (DRT/LIST)

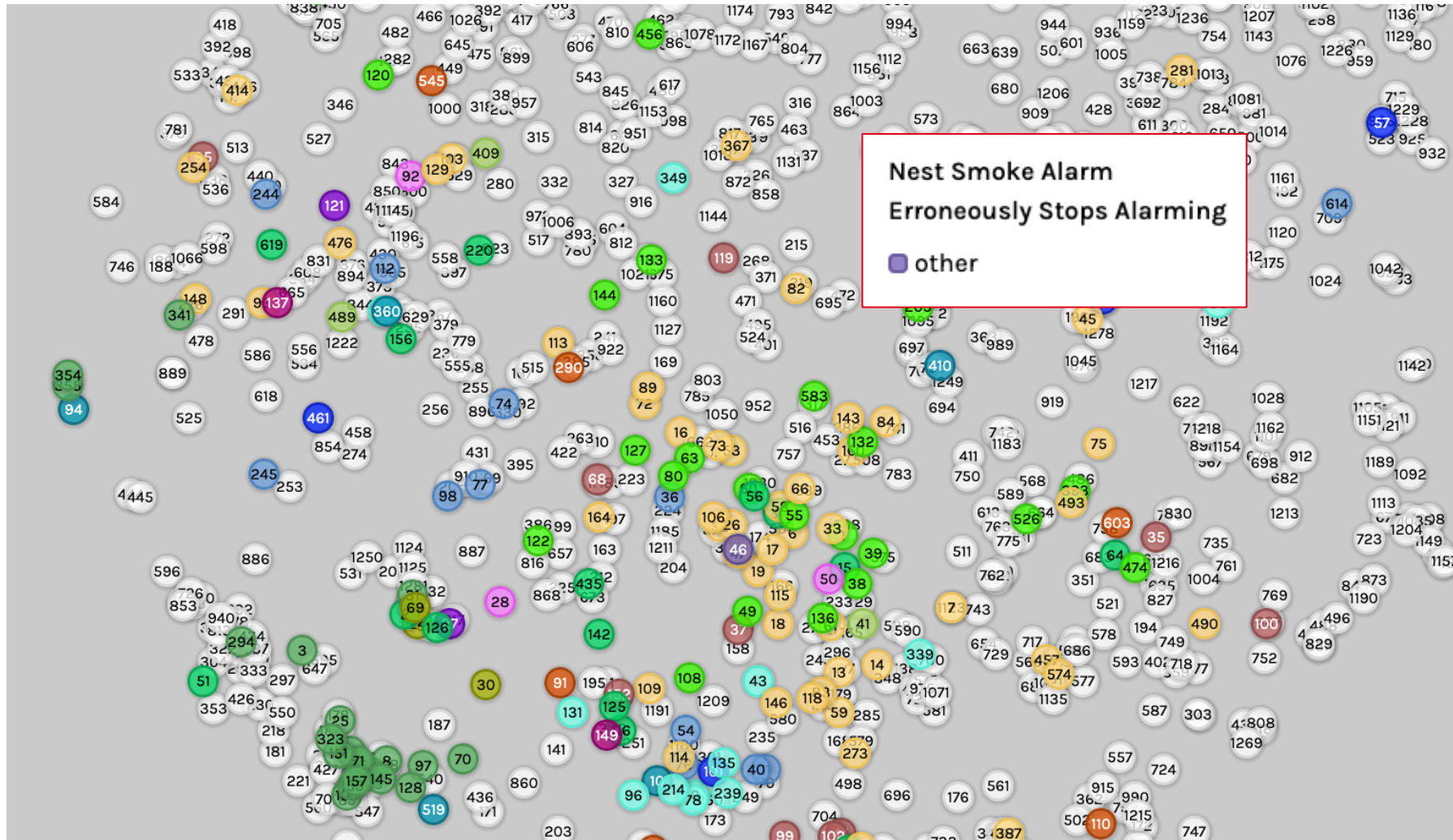


# The need for trustworthy AI



More than 1200 incidents reported!

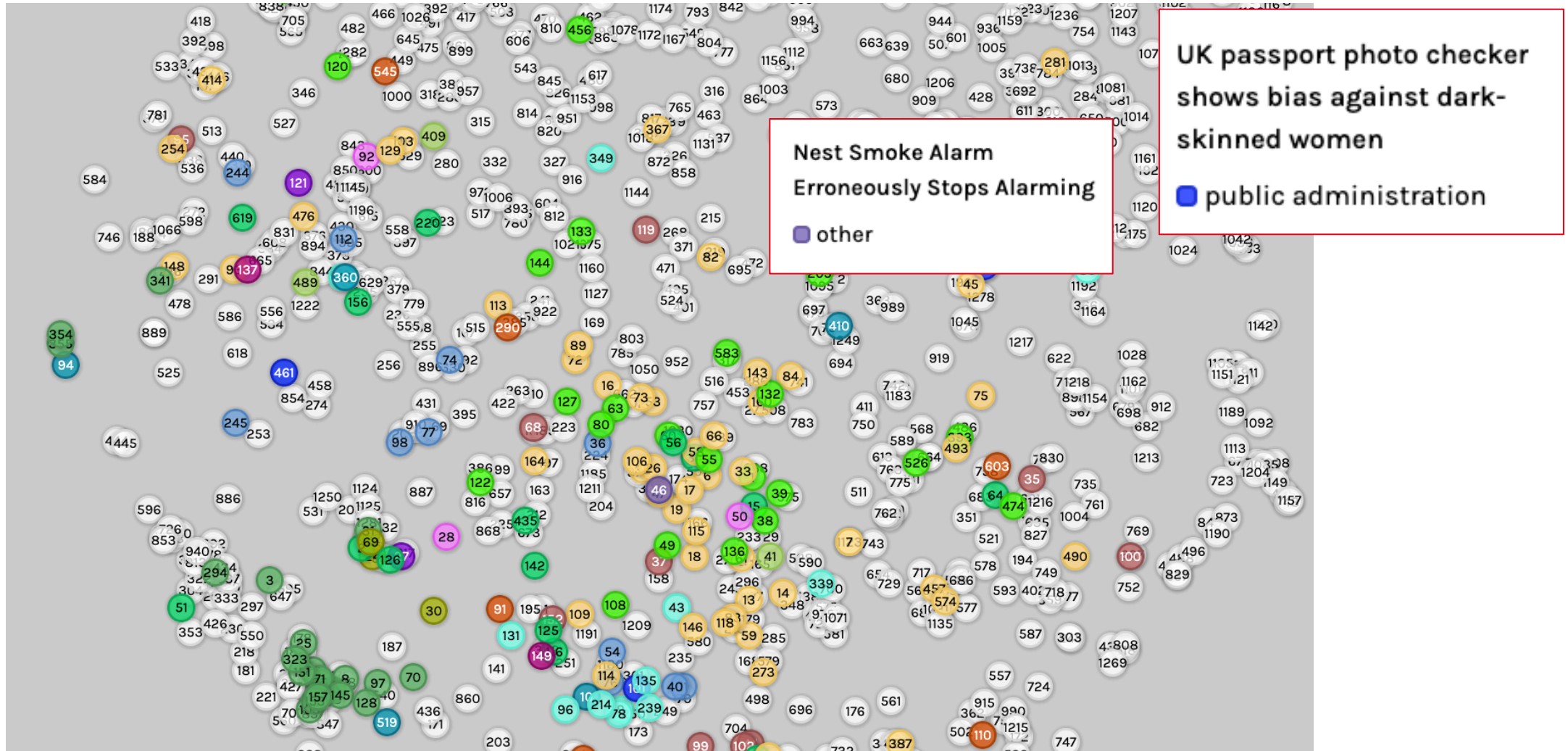
# The need for trustworthy AI



More than 1200 incidents reported!



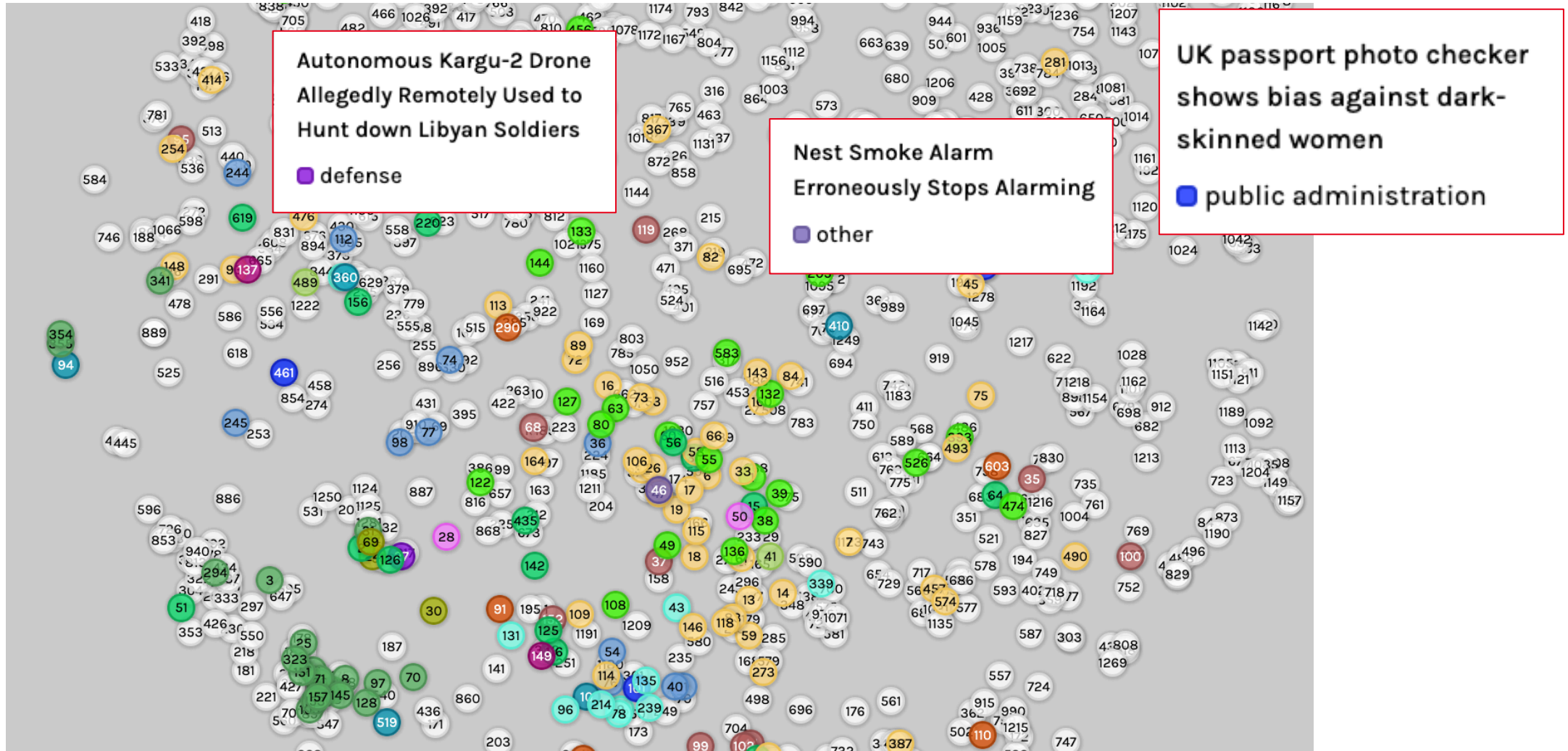
# The need for trustworthy AI



More than 1200 incidents reported!

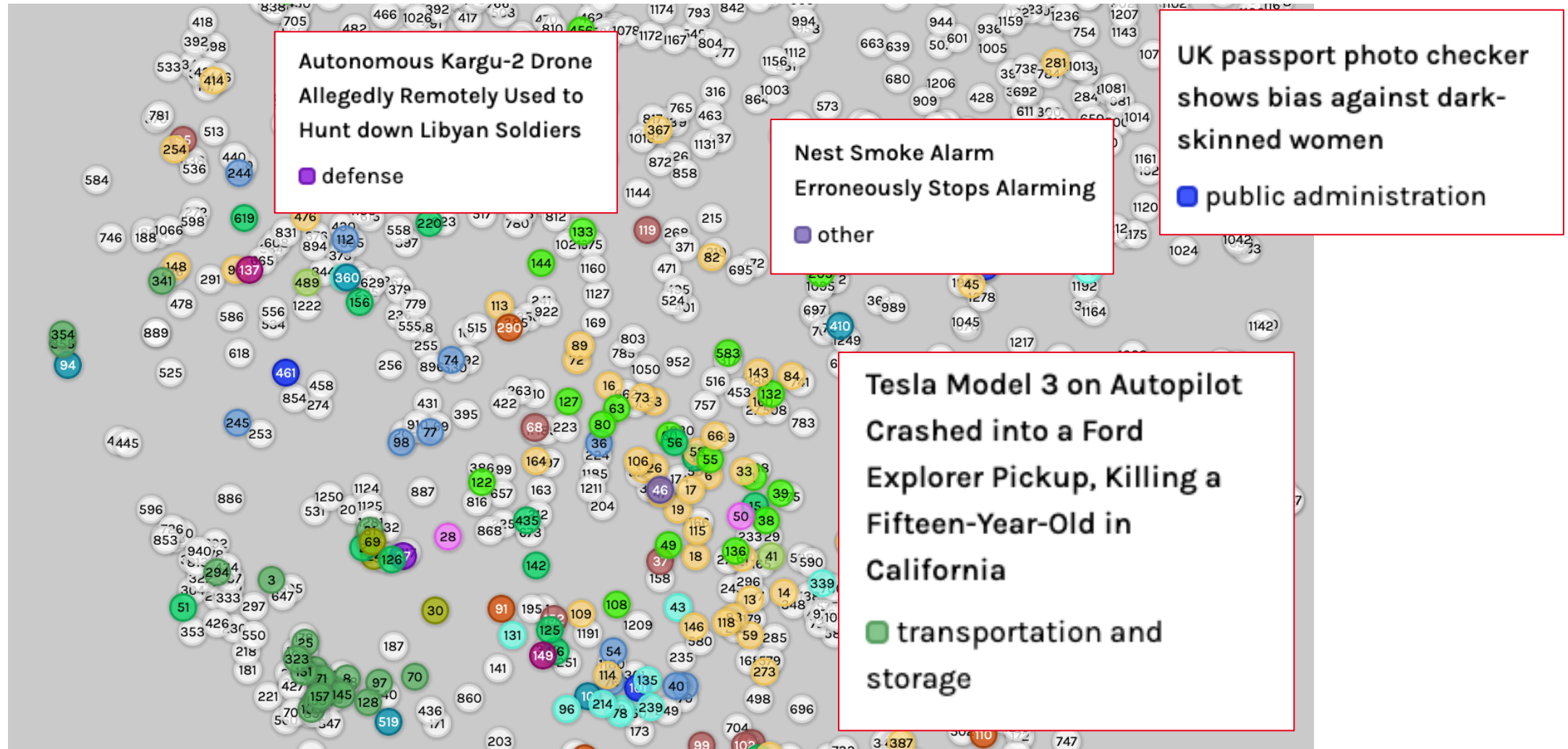


# The need for trustworthy AI



More than 1200 incidents reported!

# The need for trustworthy AI



More than 1200 incidents reported!

# Aspects of Trustworthiness

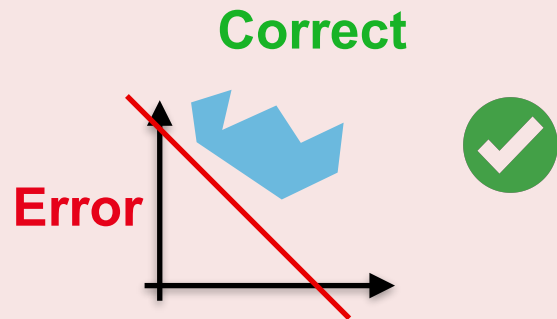




# Aspects of Trustworthiness

## Verification

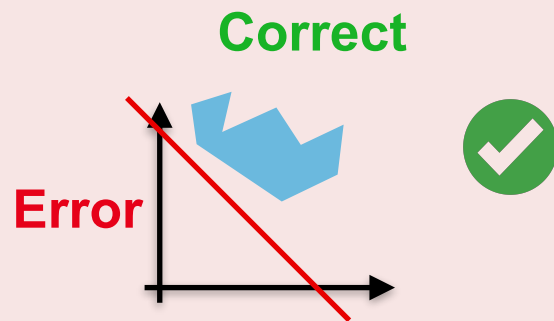
Proving the model has the desired behavior



# Aspects of Trustworthiness

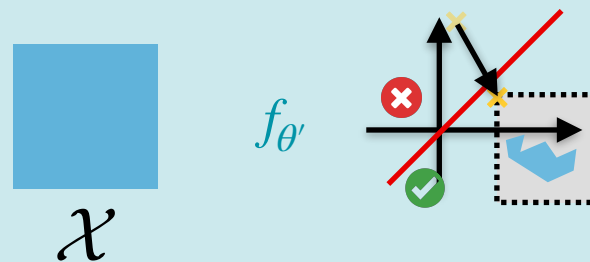
## Verification

Proving the model has the desired behavior



## Robust training

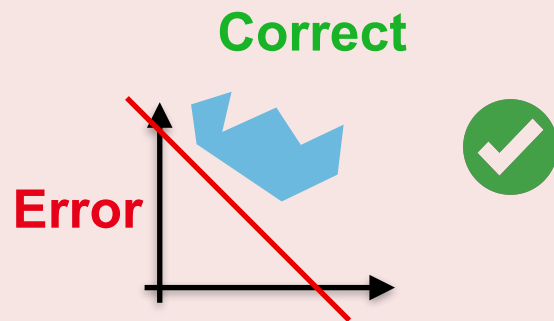
Enforcing the stable prediction on similar inputs at training time



# Aspects of Trustworthiness

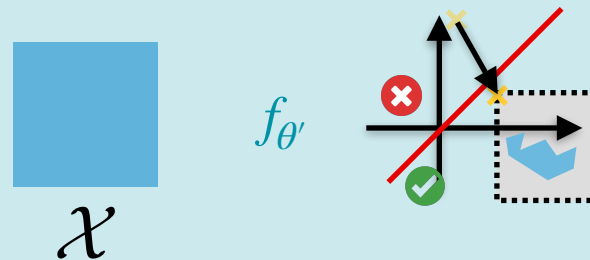
## Verification

Proving the model has the desired behavior



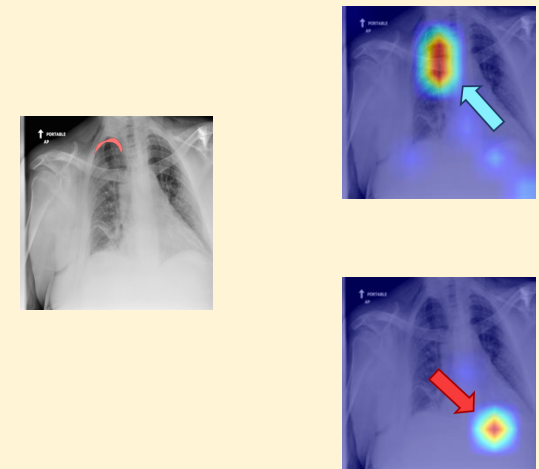
## Robust training

Enforcing the stable prediction on similar inputs at training time



## Explainability

Highlight the reason behind the decision of the model

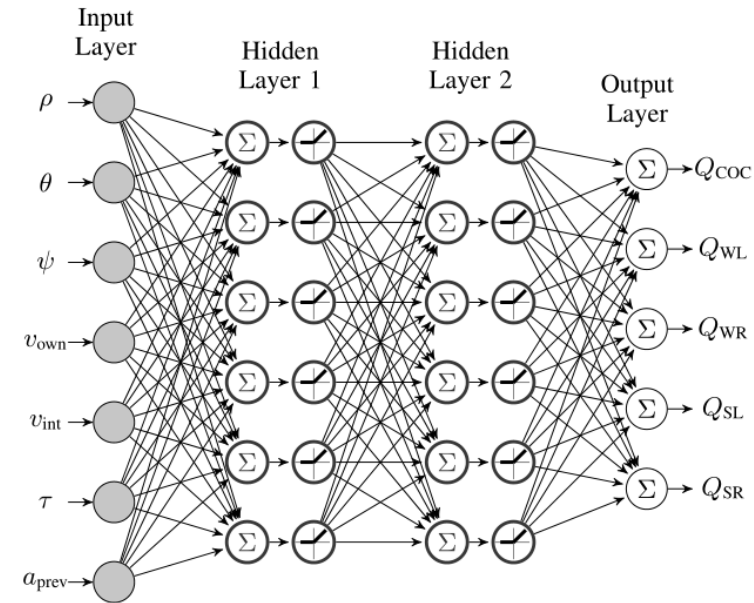
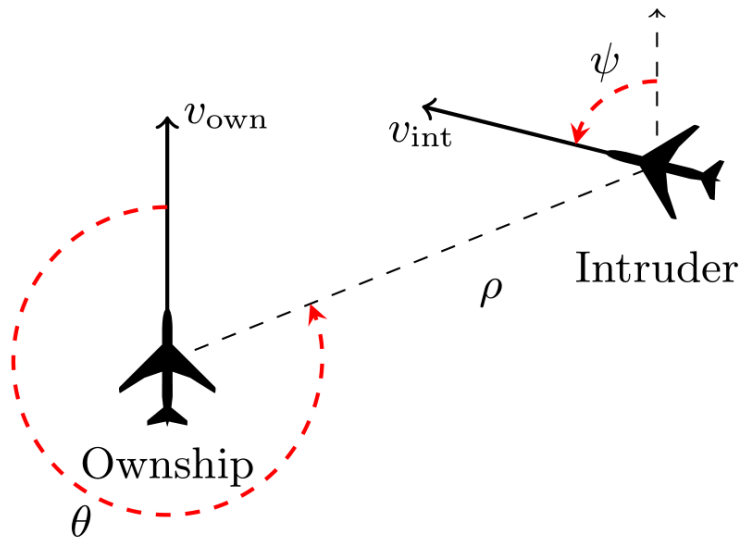




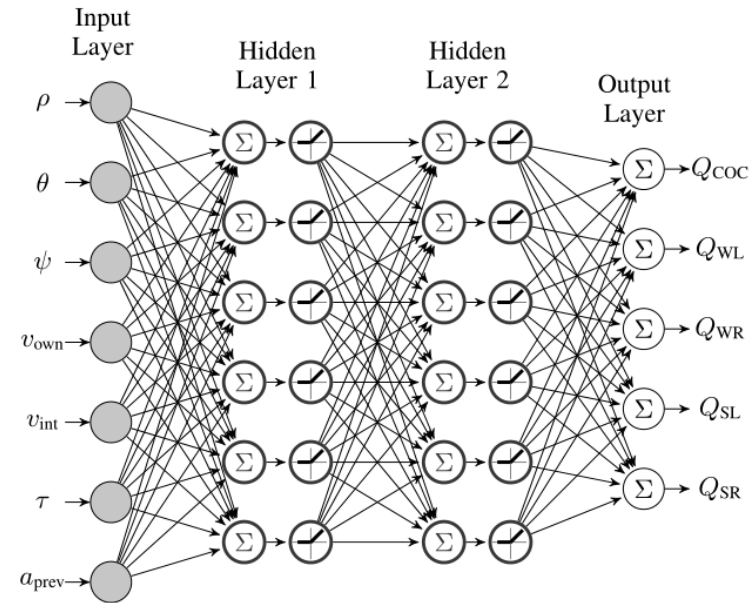
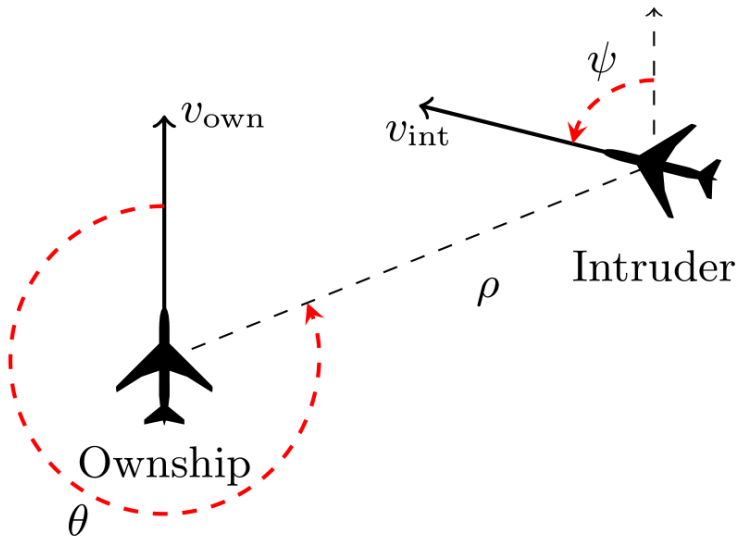


# 1 ■ Verifying Neural Networks

# Global Safety



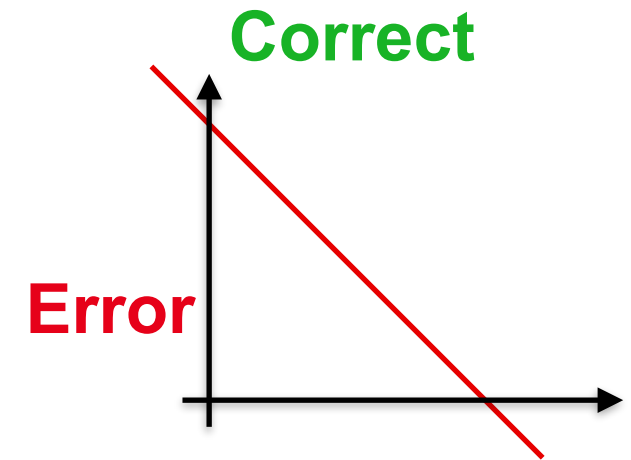
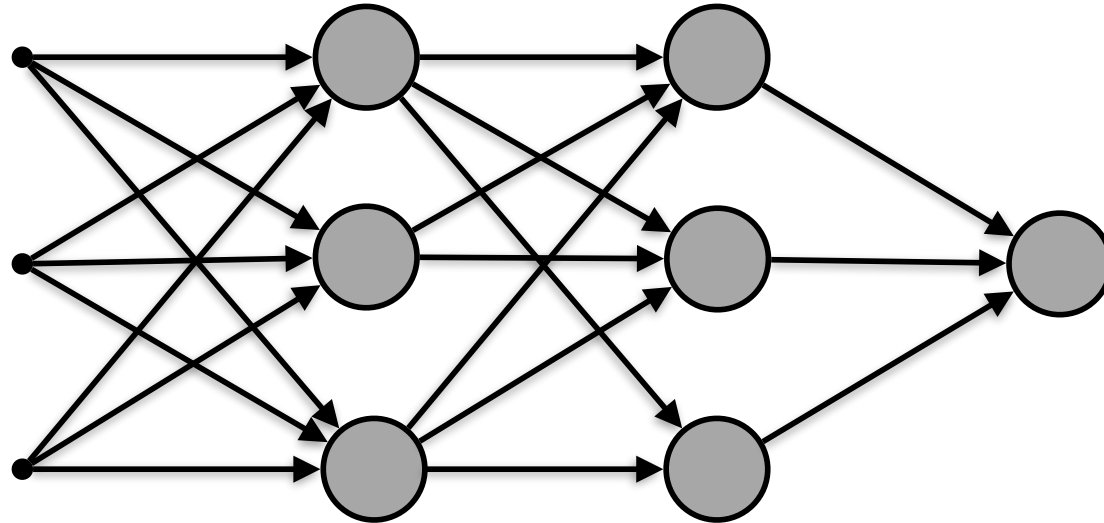
# Global Safety



**Property:** If the intruder is directly ahead and is moving towards the ownship, the Clear of Conflict command should not be advised

# Verification of Neural Networks

## Input-output properties

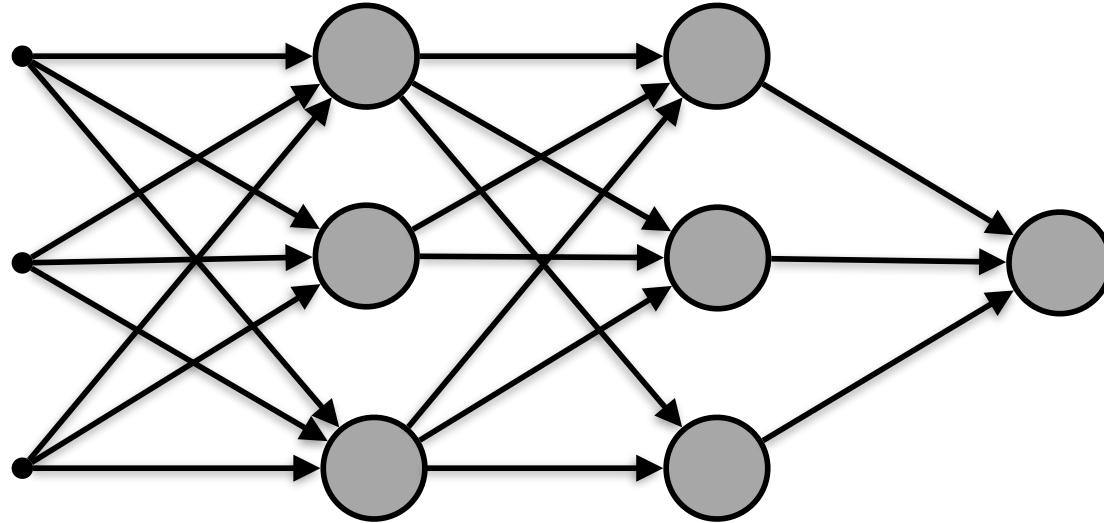


$\mathcal{X}$

$f_{\theta}$

$f_{\theta}(\text{blue square})?$

# Exact Verification

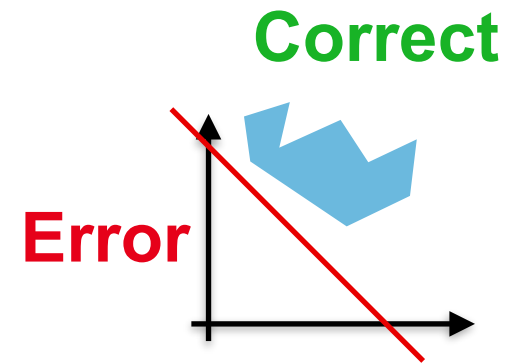
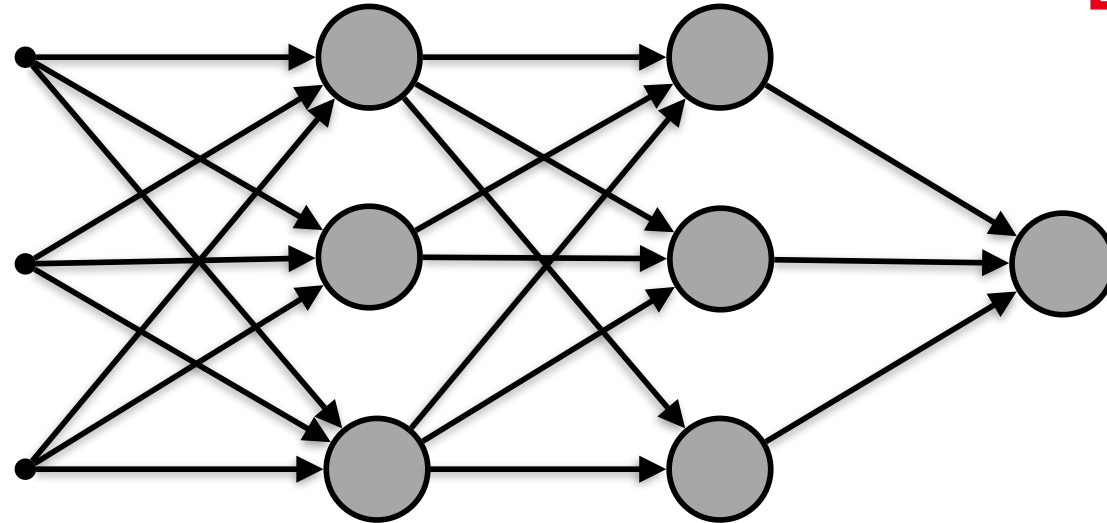


$$f_{\theta}(\text{blue square}) = \text{blue arrow}$$

 $\mathcal{X}$  $f_{\theta}$



# Exact Verification



Safe :)



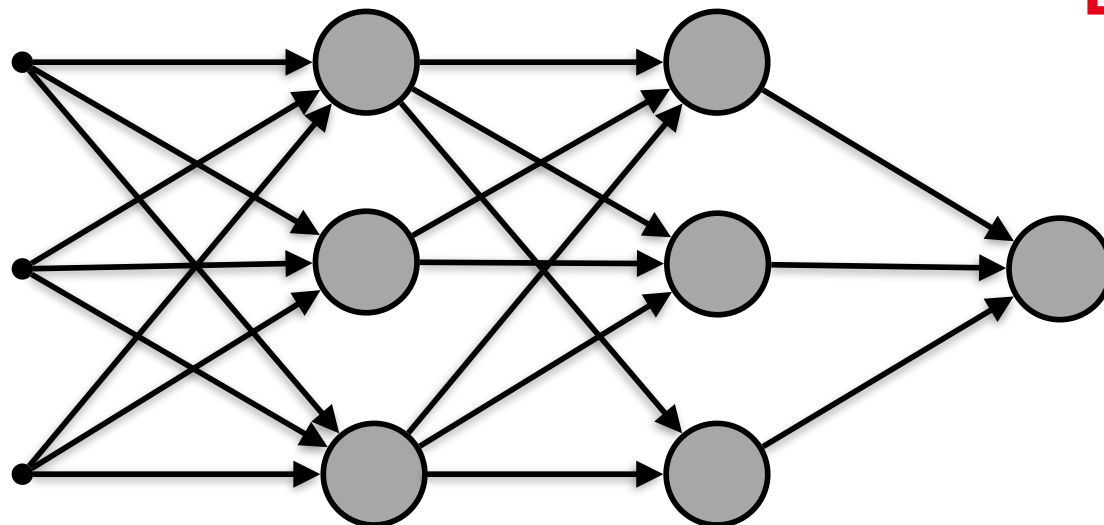
$$f_{\theta}(\text{blue square}) = \text{blue arrow}$$

 $\mathcal{X}$  $f_{\theta}$

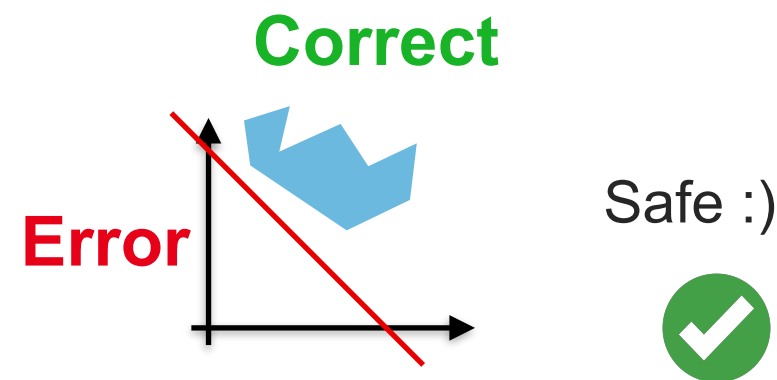
# Exact Verification



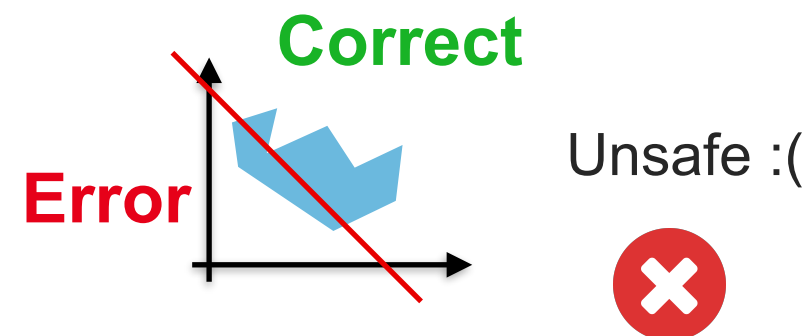
$\mathcal{X}$



$f_\theta$



$$f_\theta(\text{blue square}) = \text{blue shape}$$

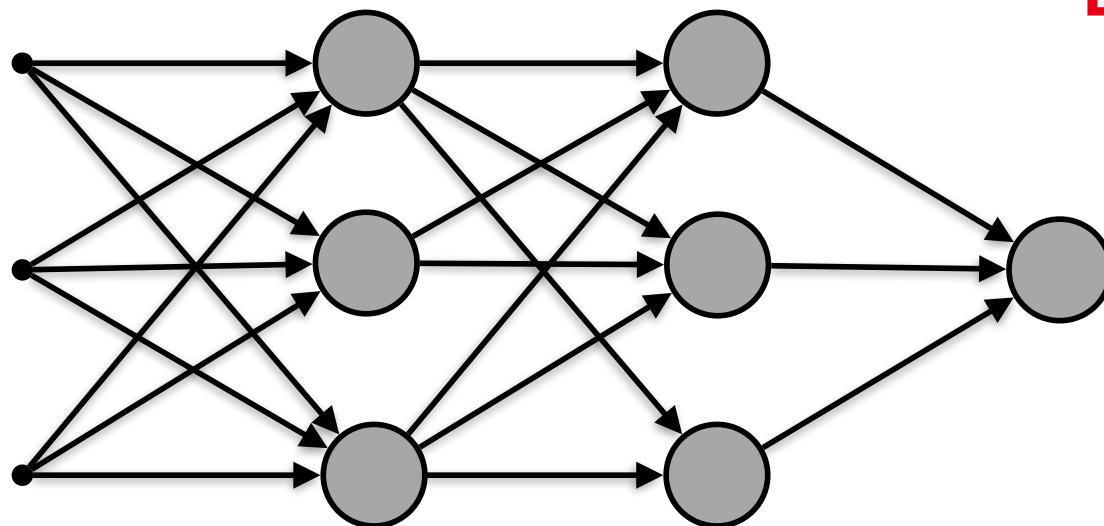


# Exact Verification

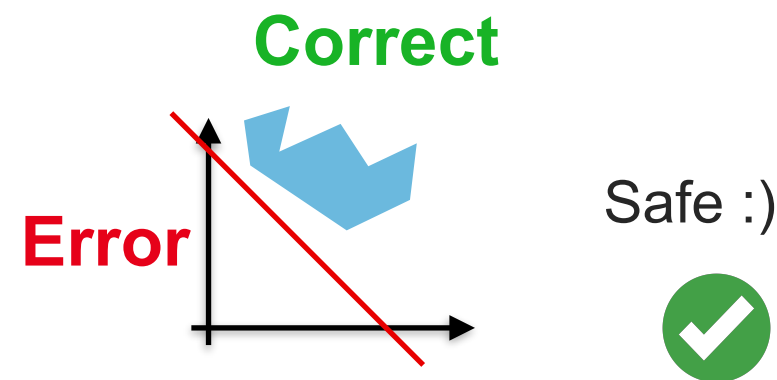
NP-Complete [Katz2017]



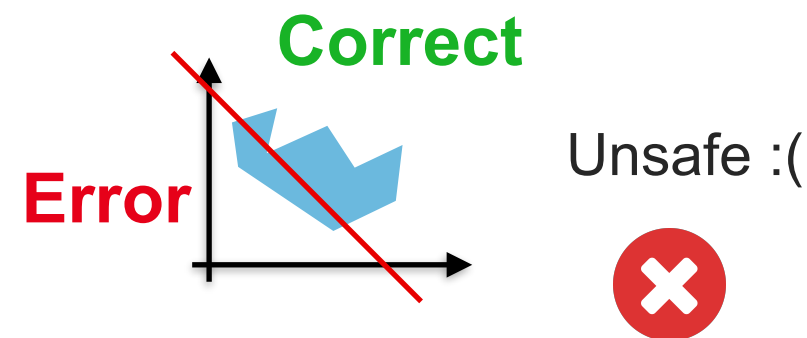
$\mathcal{X}$



$f_\theta$



$$f_\theta(\text{blue square}) = \text{blue shape}$$



# Over-approximating Neural Networks

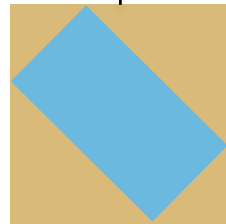


Input space

# Over-approximating Neural Networks

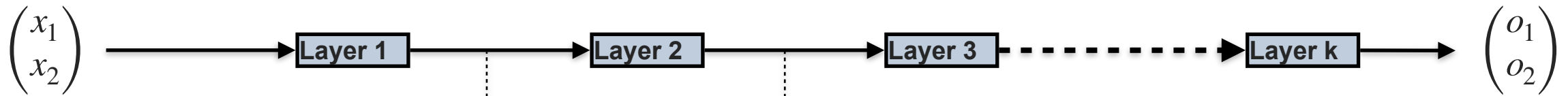


Input space

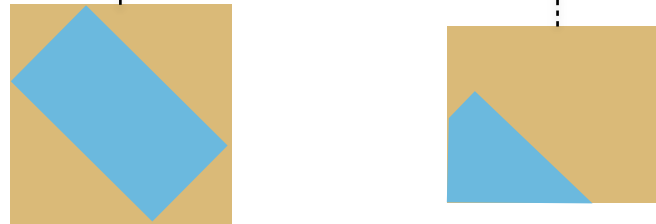


Exact reachable space and its over-approximation

# Over-approximating Neural Networks



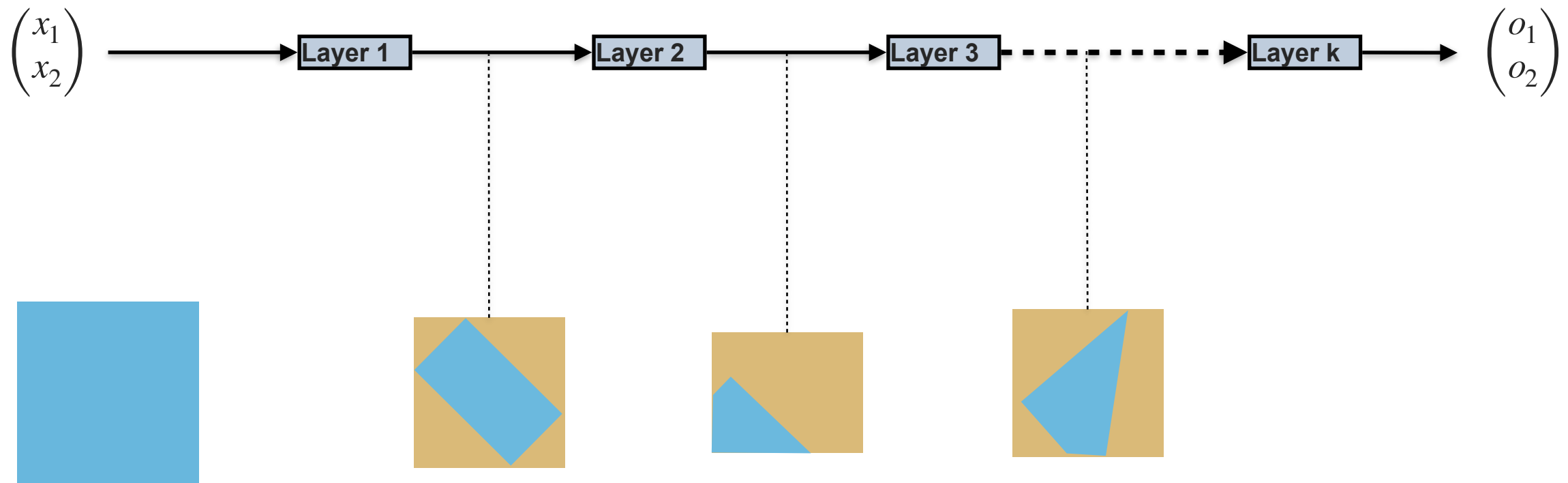
Input space



Exact reachable space and its over-approximation



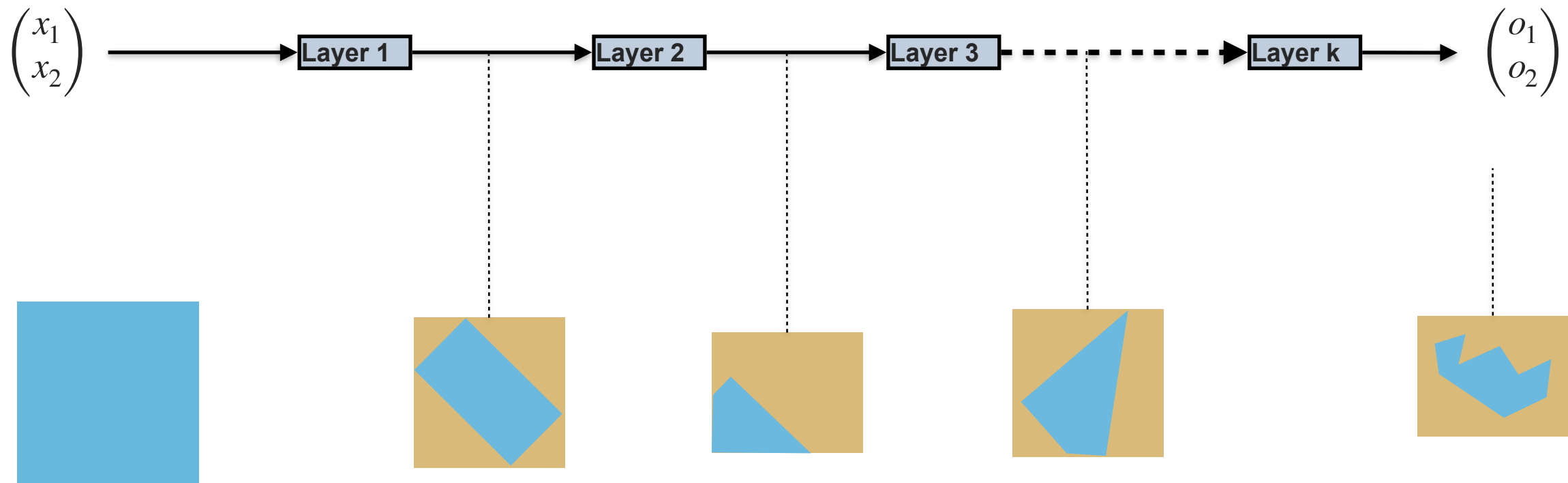
# Over-approximating Neural Networks



Input space

Exact reachable space and its  
over-approximation

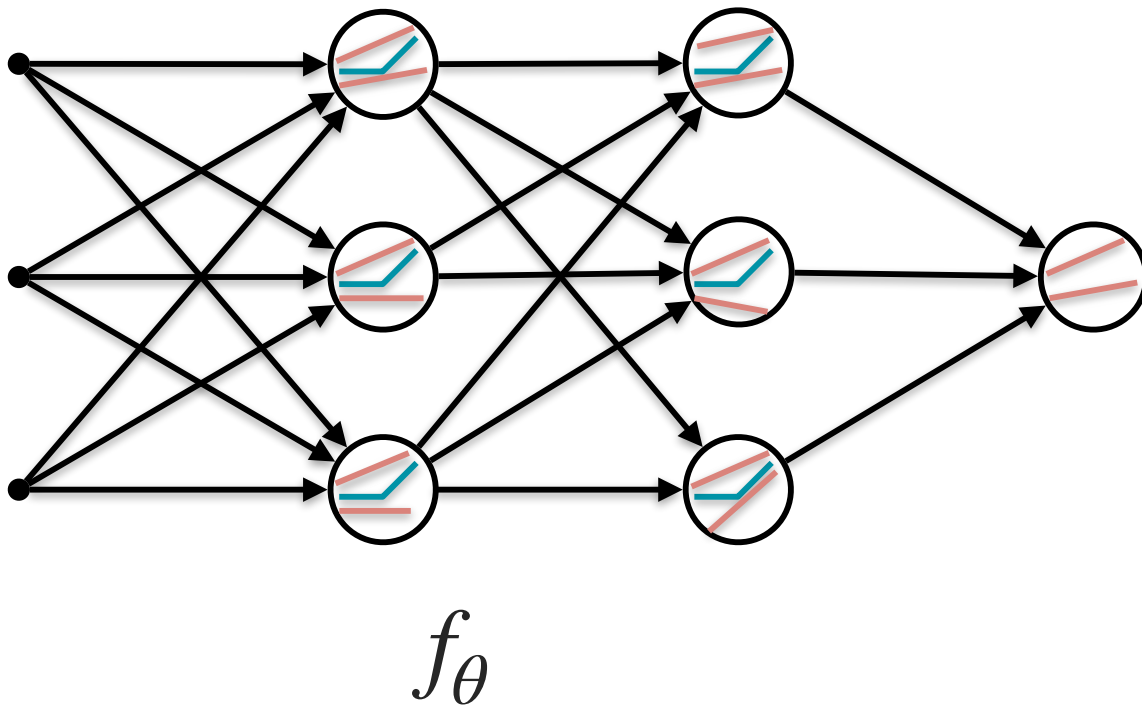
# Over-approximating Neural Networks



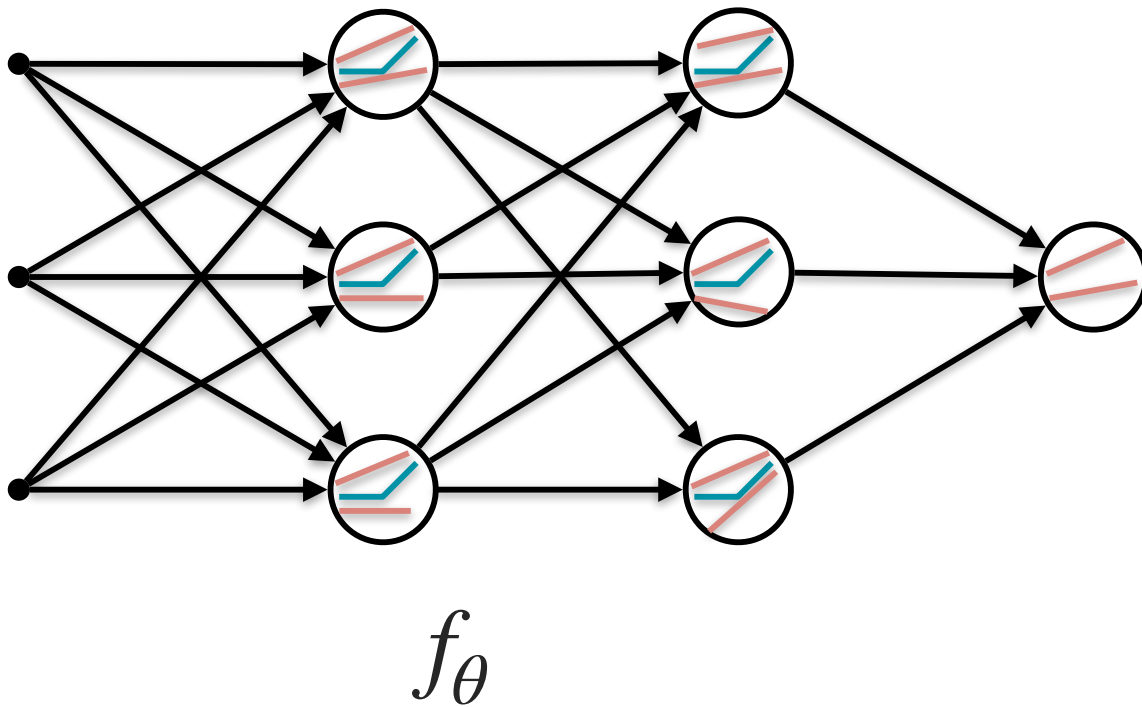
# Linear Approximations



# Linear Approximations

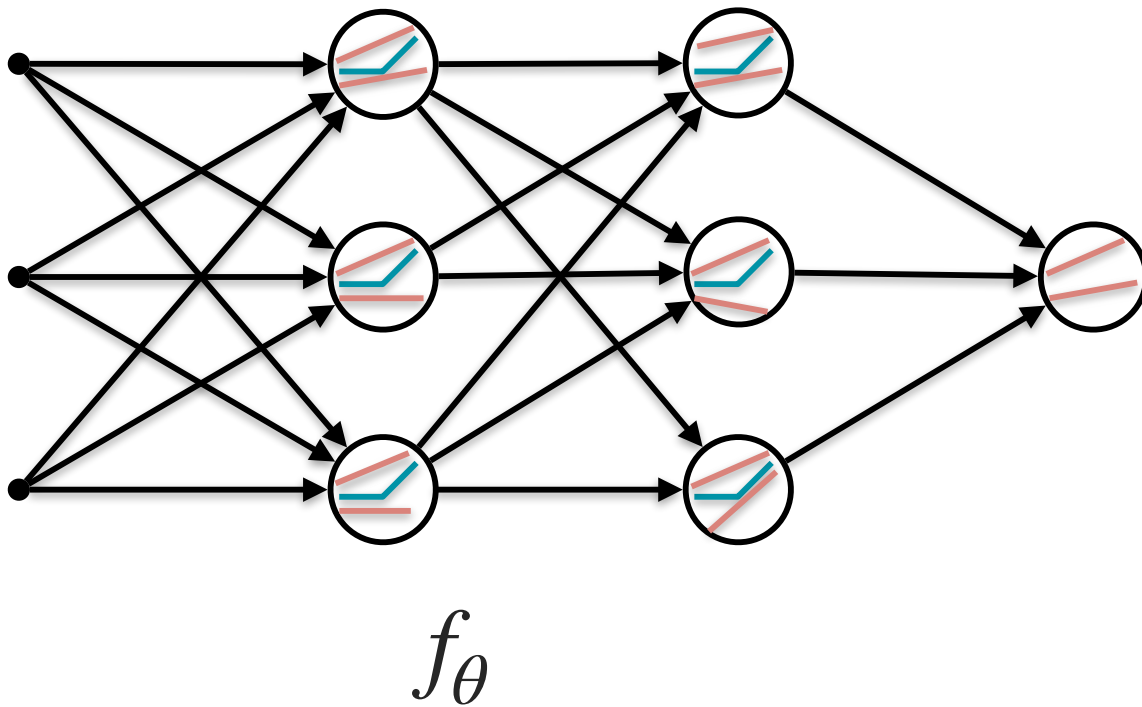


# Linear Approximations



$$\forall x \in \mathcal{X}, L(x) \leq f_\theta(x) \leq U(x)$$

# Linear Approximations



$$\forall x \in \mathcal{X}, L(x) \leq f_\theta(x) \leq U(x)$$

$$L(x) = \sum_{i=1}^n \underline{\alpha}_i x_i$$

$$U(x) = \sum_{i=1}^n \overline{\alpha}_i x_i$$

# Over-approximations for verification

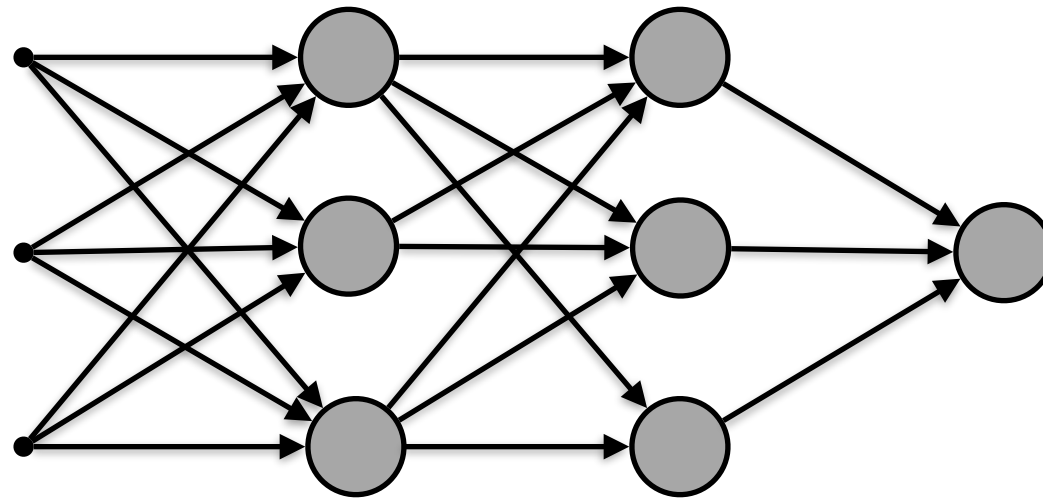




# Over-approximations for verification



$\mathcal{X}$



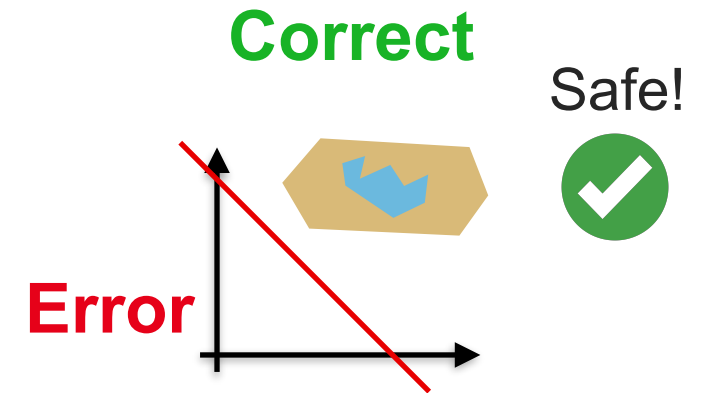
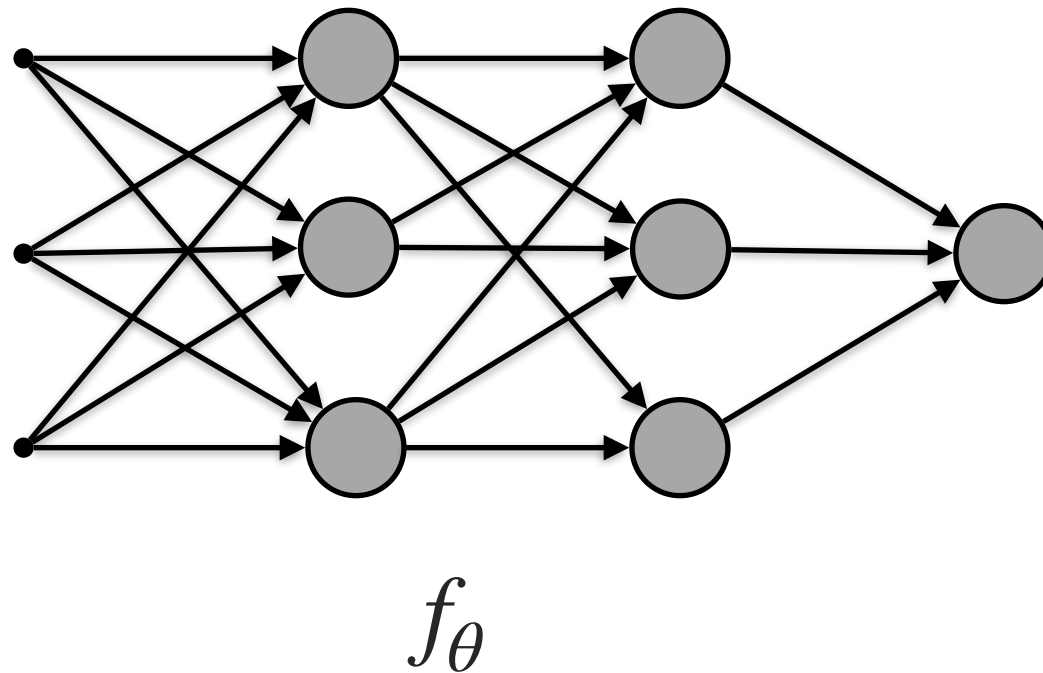
$f_\theta$

$$f_\theta(\text{blue square}) \subset \text{yellow hexagon}$$

# Over-approximations for verification

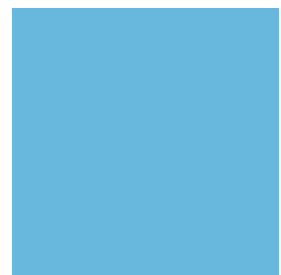


$\mathcal{X}$

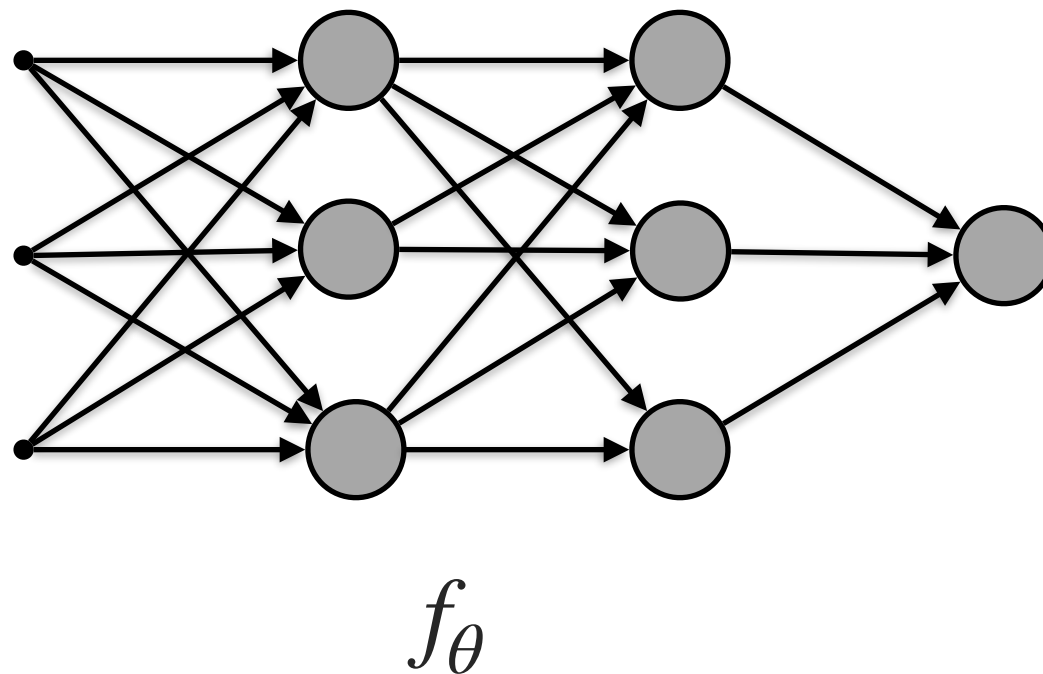


$$f_{\theta}(\text{blue square}) \subset \text{yellow hexagon}$$

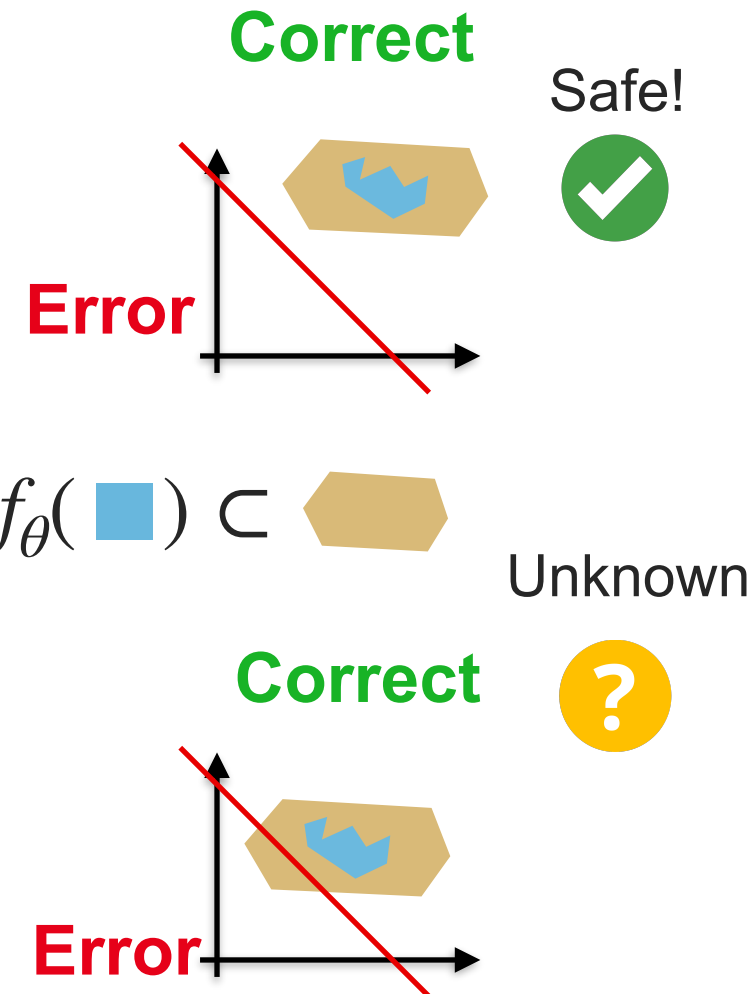
# Over-approximations for verification



$\mathcal{X}$



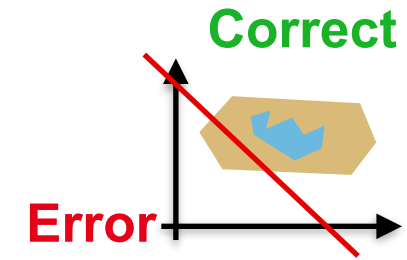
$f_\theta$



# Input partitioning

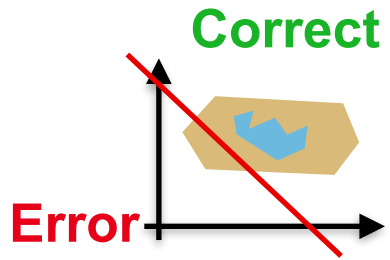


# Input partitioning



False Alarm ?

# Input partitioning



$\mathcal{X}$



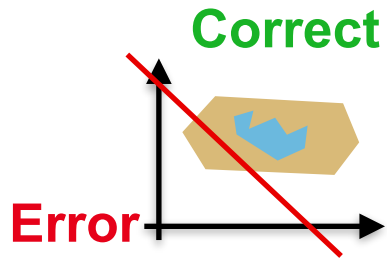
$\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$



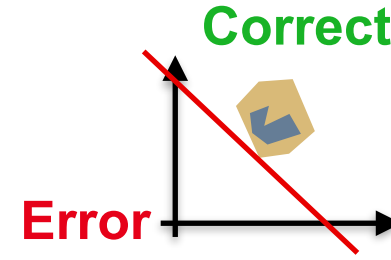
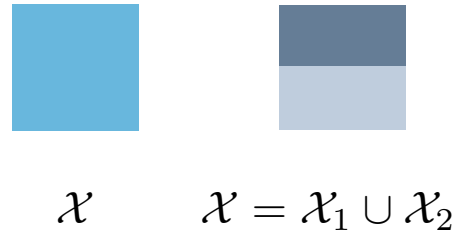
False Alarm ?

1) Partition the input space

# Input partitioning



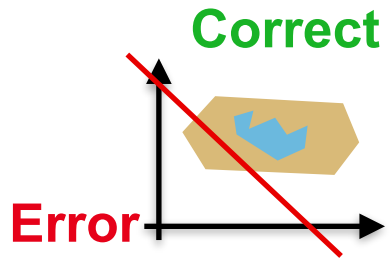
False Alarm ?



2 ) Solve two verification problems

1) Partition the input space

# Input partitioning



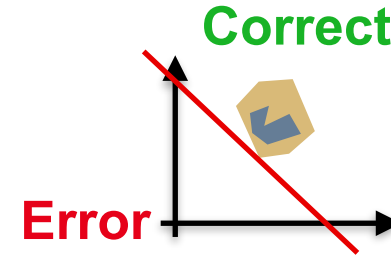
False Alarm ?



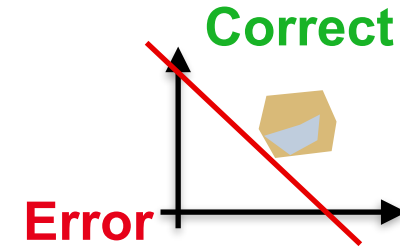
$\mathcal{X}$



$\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$



2 ) Solve two verification problems



1) Partition the input space



# Iterative input partitioning



# Iterative input partitioning

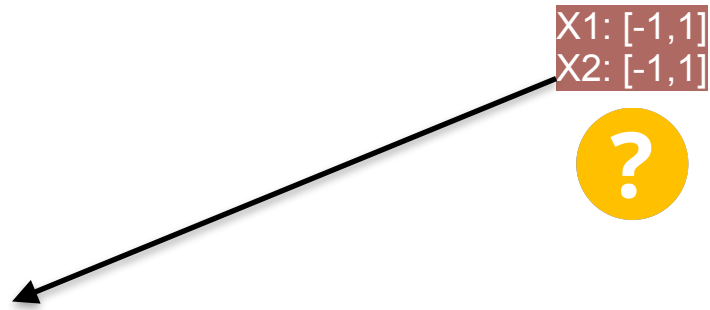
X1: [-1,1]  
X2: [-1,1]

# Iterative input partitioning

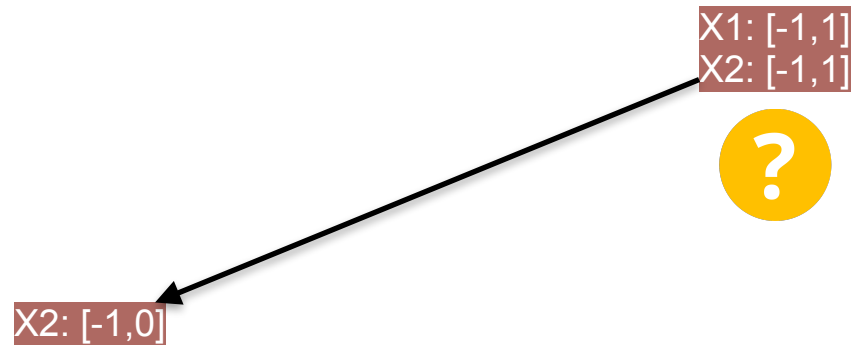
X1: [-1,1]  
X2: [-1,1]



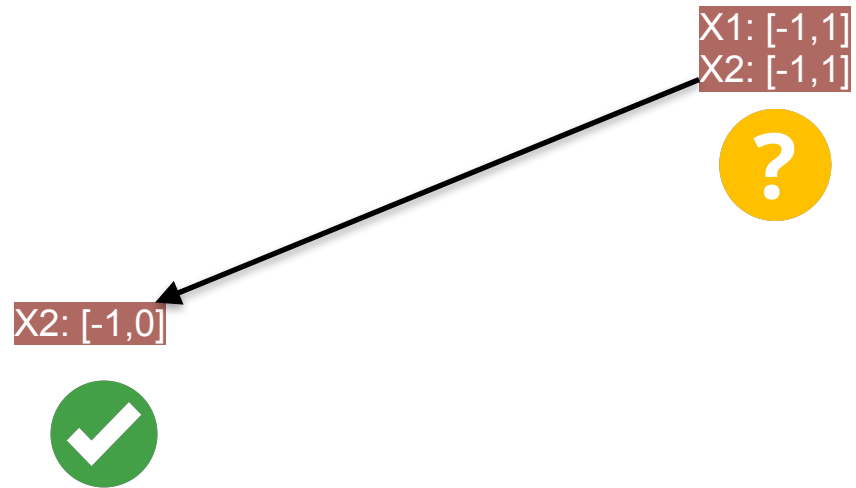
# Iterative input partitioning



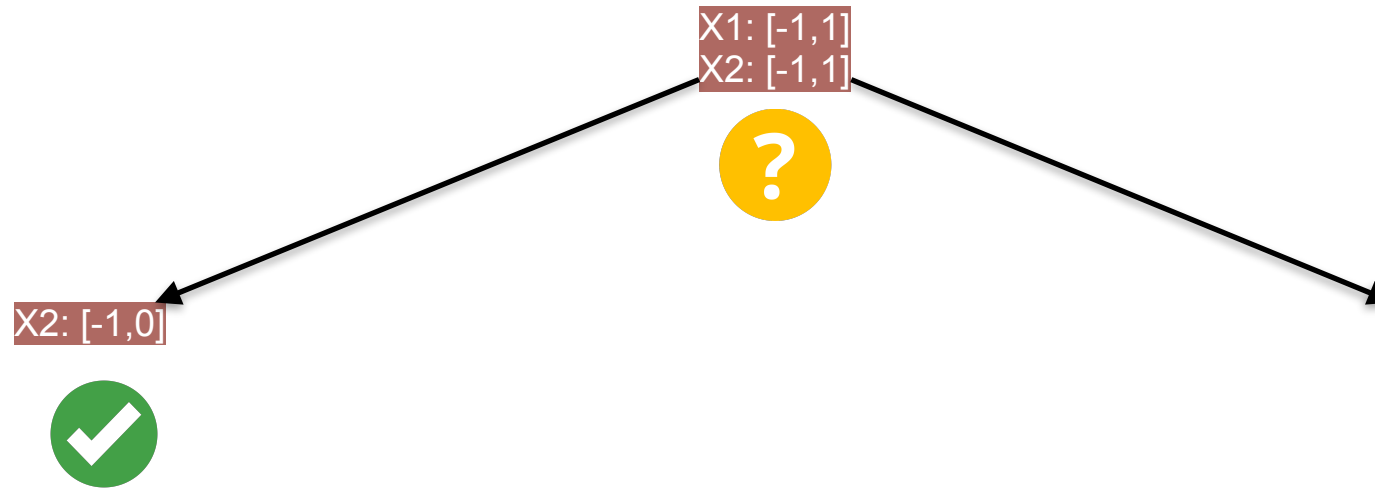
# Iterative input partitioning



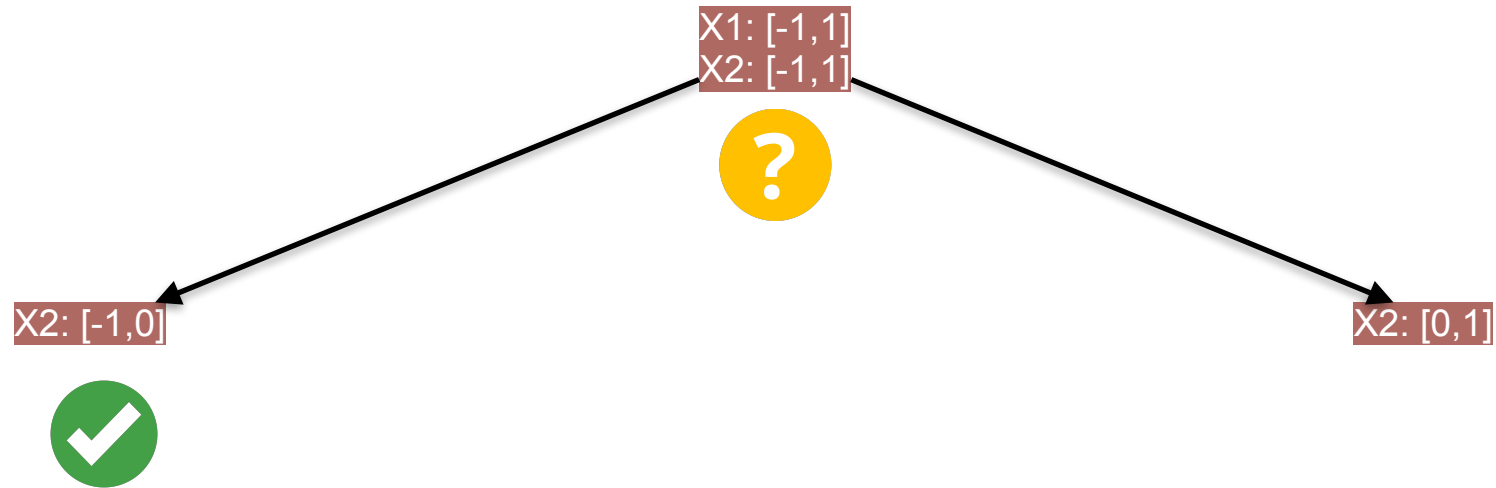
# Iterative input partitioning



# Iterative input partitioning

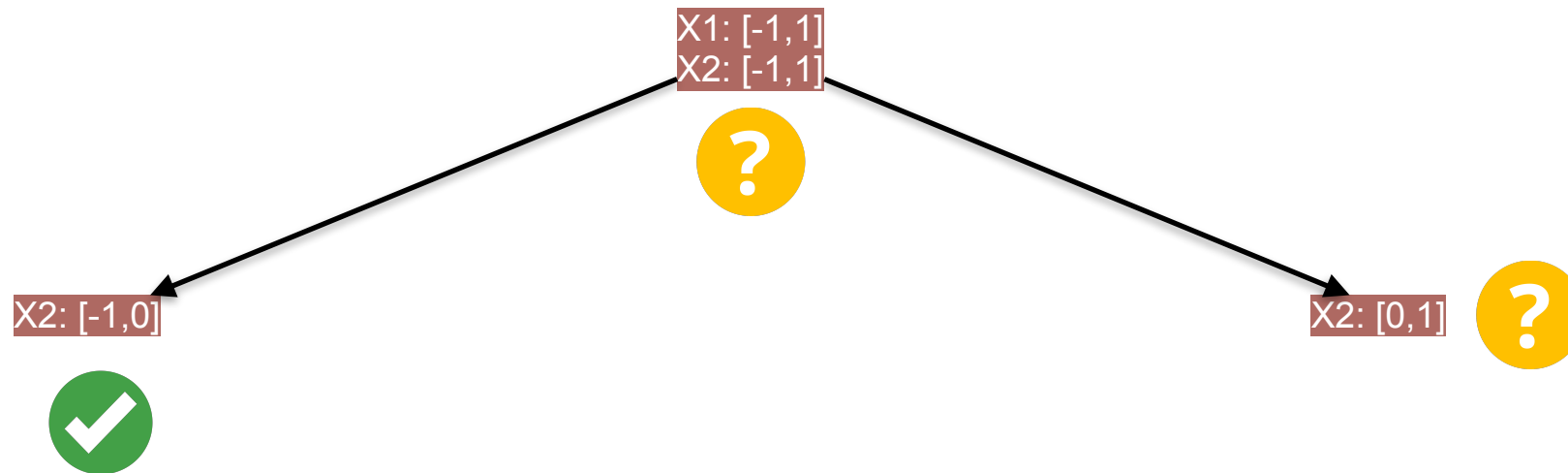


# Iterative input partitioning

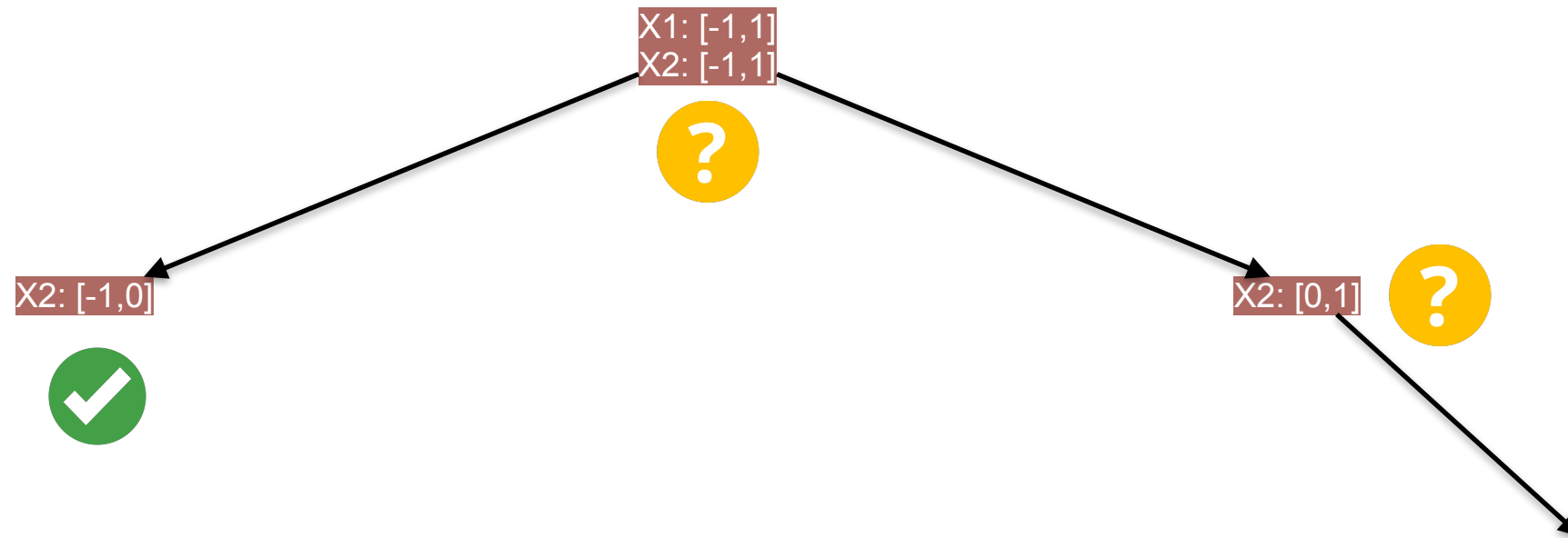




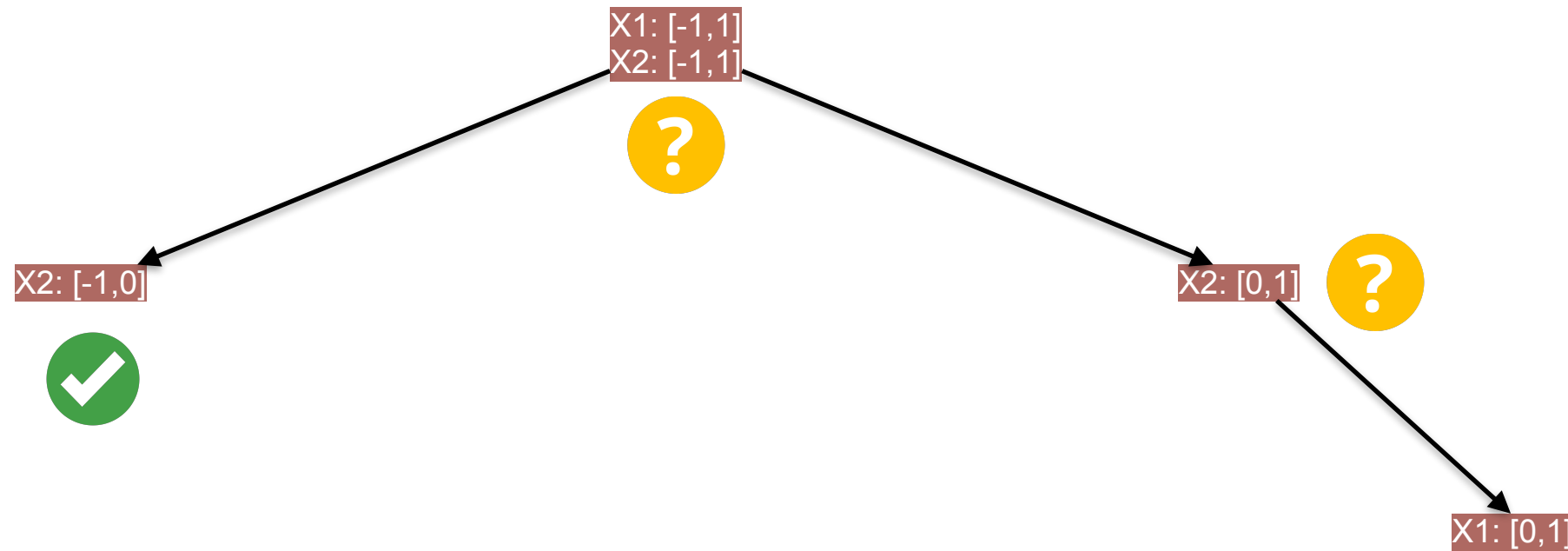
# Iterative input partitioning



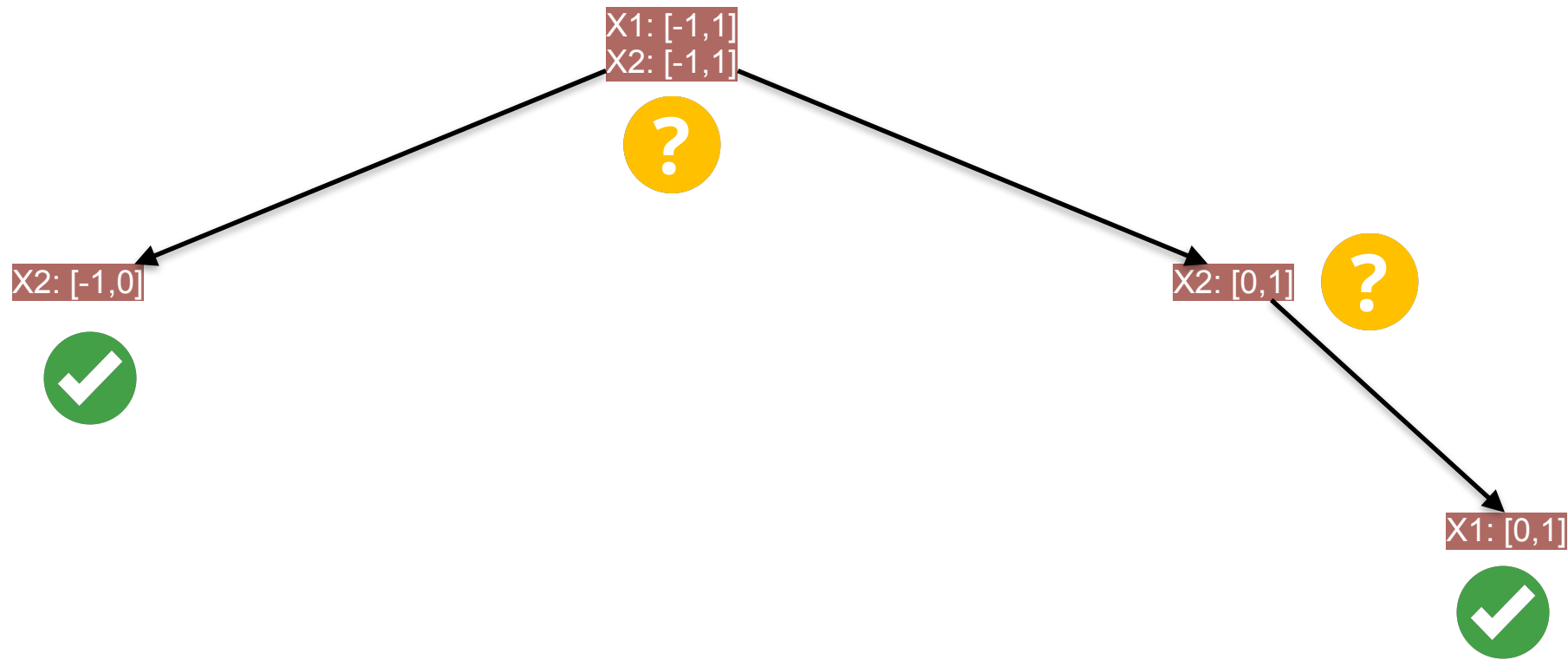
# Iterative input partitioning



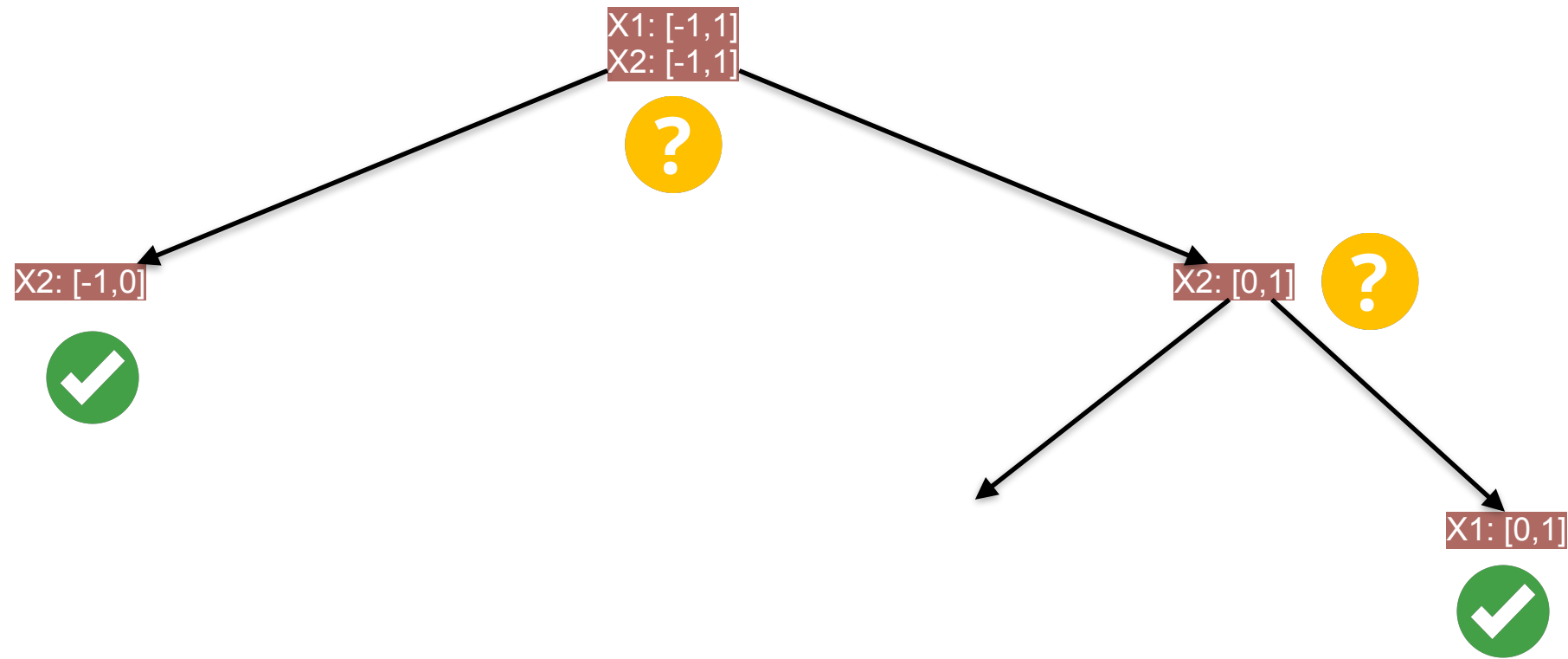
# Iterative input partitioning



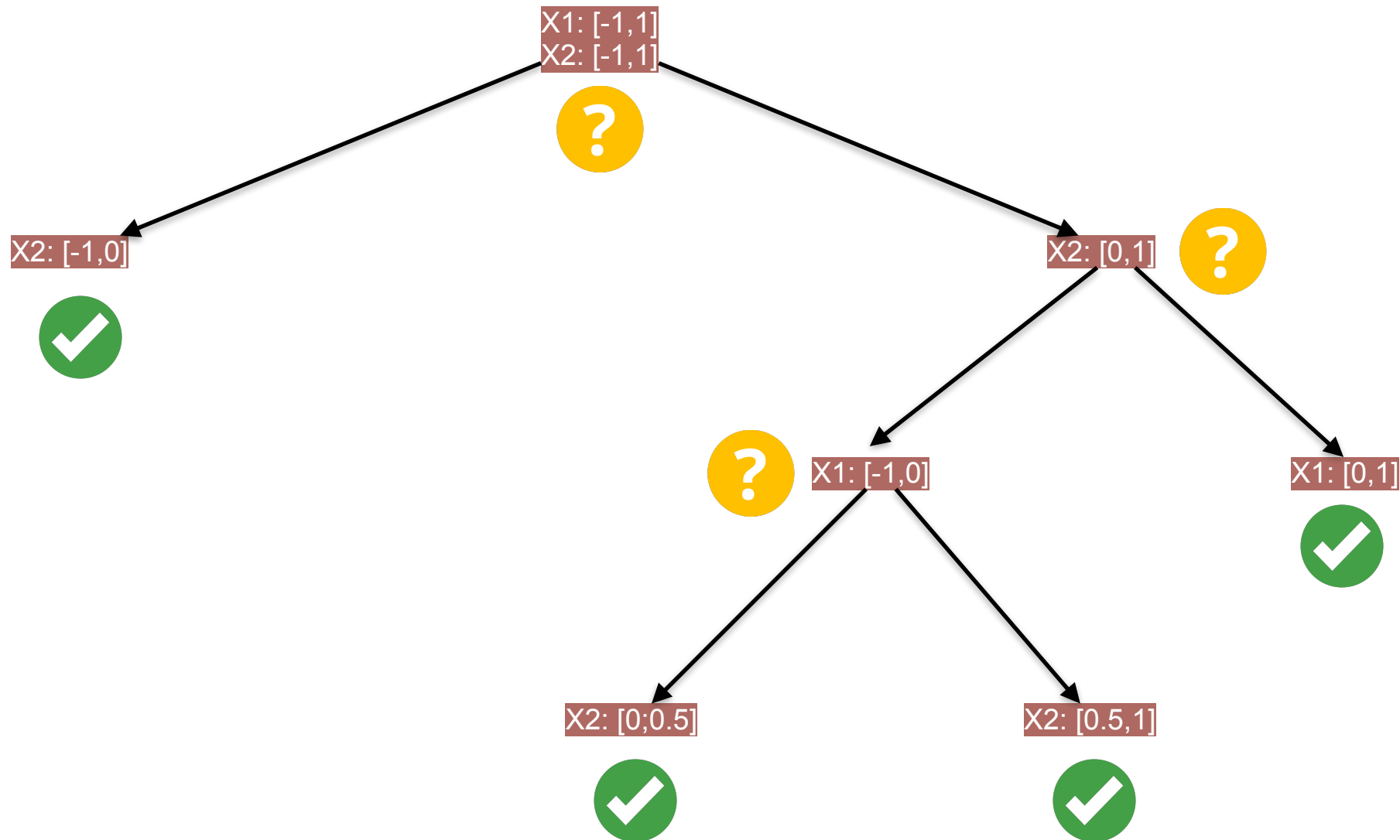
# Iterative input partitioning



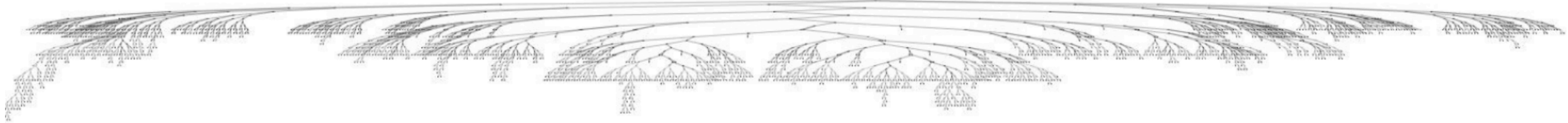
# Iterative input partitioning



# Iterative input partitioning



# “Real life” example



- Industrial Use Case
- 3133 subproblems to solve!

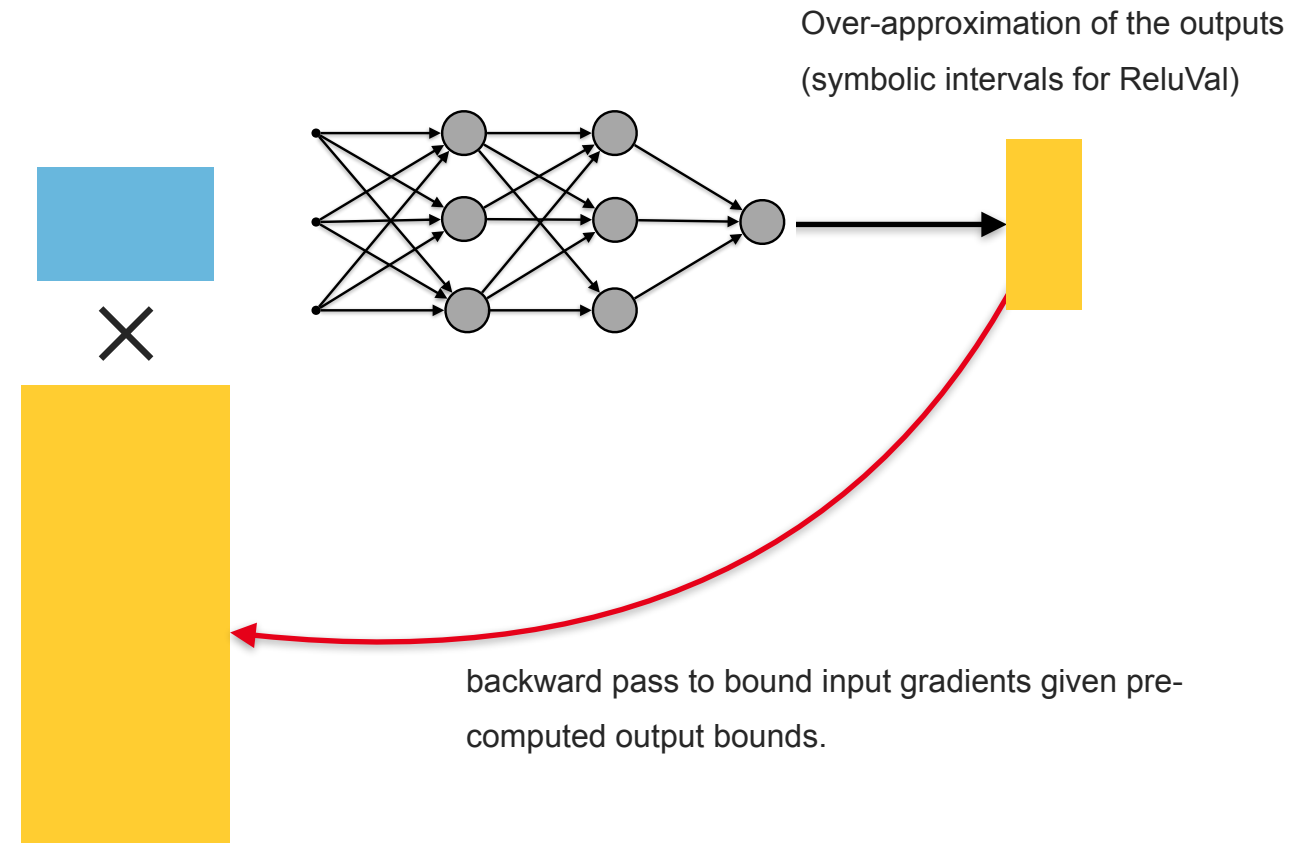
# Existing heuristics:

- Random
- Largest interval
- Gradient Smear [ReluVal, Wang 2018]



# Existing heuristics:

- Random
- Largest interval
- Gradient Smear [ReluVal, Wang 2018]

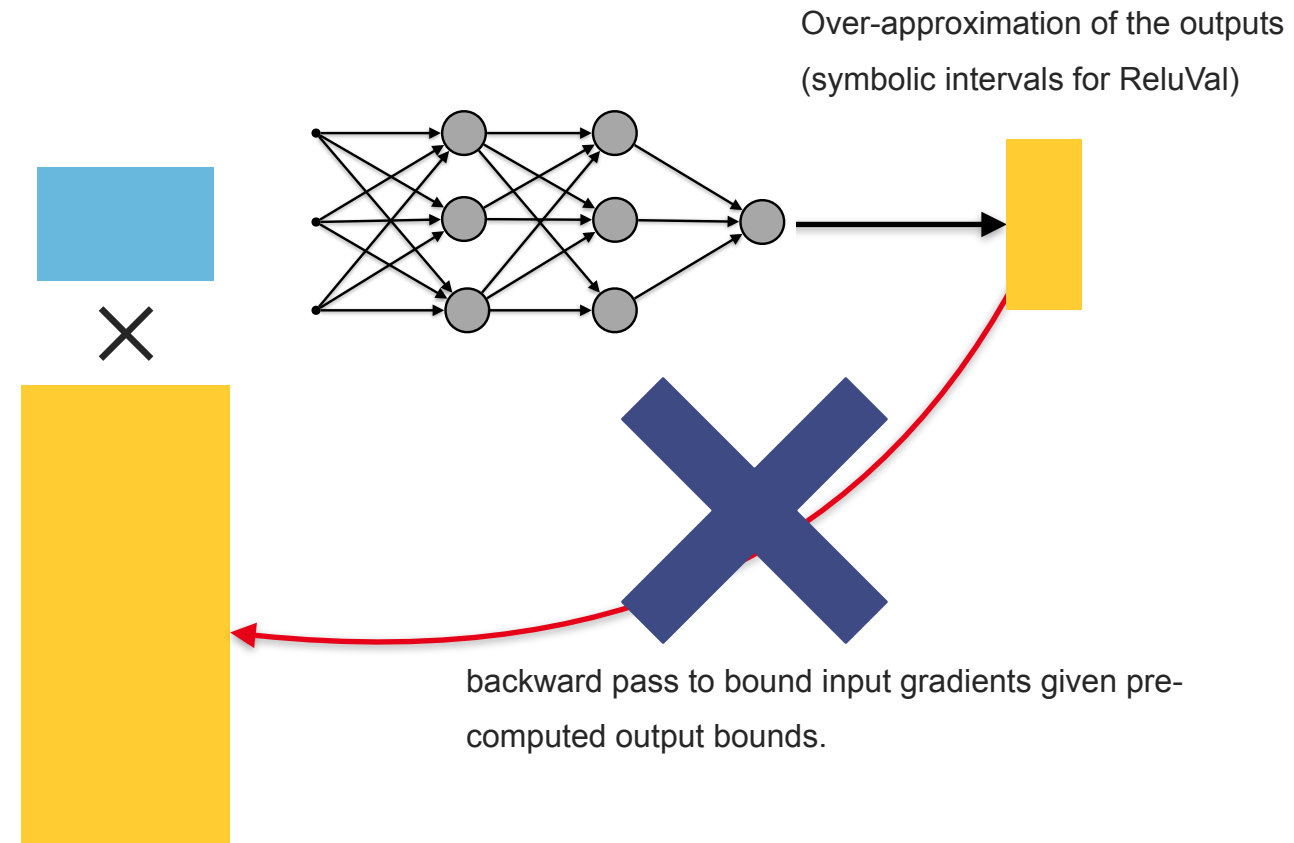


Final score: largest upper bound times input width

# Existing heuristics:

- Random
- Largest interval
- Gradient Smear [ReluVal, Wang 2018]

**Our proposal: reuse information already computed during the over-approximation**



Final score: largest upper bound times input width

# ReCIPH: (Relational coefficient for input partitioning heuristic)



# ReCIPH: (Relational coefficient for input partitioning heuristic)

$$\forall x \in \mathcal{X}, L(x) \leq f_{\theta}(x) \leq U(x) \qquad L(x) = \sum_{i=1}^n \underline{\alpha}_i x_i \qquad U(x) = \sum_{i=1}^n \overline{\alpha}_i x_i$$

# ReCIPH: (Relational coefficient for input partitioning heuristic)

$$\forall x \in \mathcal{X}, L(x) \leq f_{\theta}(x) \leq U(x) \quad L(x) = \sum_{i=1}^n \underline{\alpha}_i x_i \quad U(x) = \sum_{i=1}^n \overline{\alpha}_i x_i$$

- ReCIPH score for the input  $x_i = \frac{|\underline{\alpha}_i| + |\overline{\alpha}_i|}{2} \cdot (u_i - l_i)$

# ReCIPH: (Relational coefficient for input partitioning heuristic)

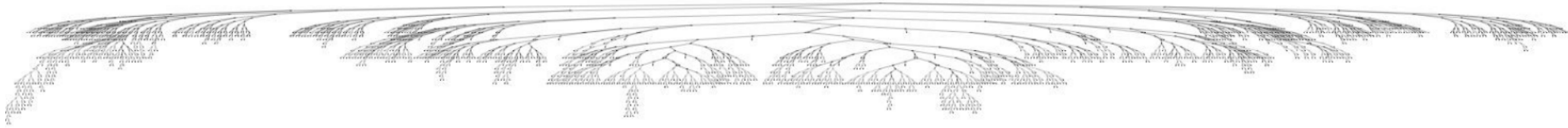
$$\forall x \in \mathcal{X}, L(x) \leq f_{\theta}(x) \leq U(x) \quad L(x) = \sum_{i=1}^n \underline{\alpha}_i x_i \quad U(x) = \sum_{i=1}^n \overline{\alpha}_i x_i$$

- ReCIPH score for the input  $x_i = \frac{|\underline{\alpha}_i| + |\overline{\alpha}_i|}{2} \cdot (u_i - l_i)$
- Variable to split first: highest ReCIPH score

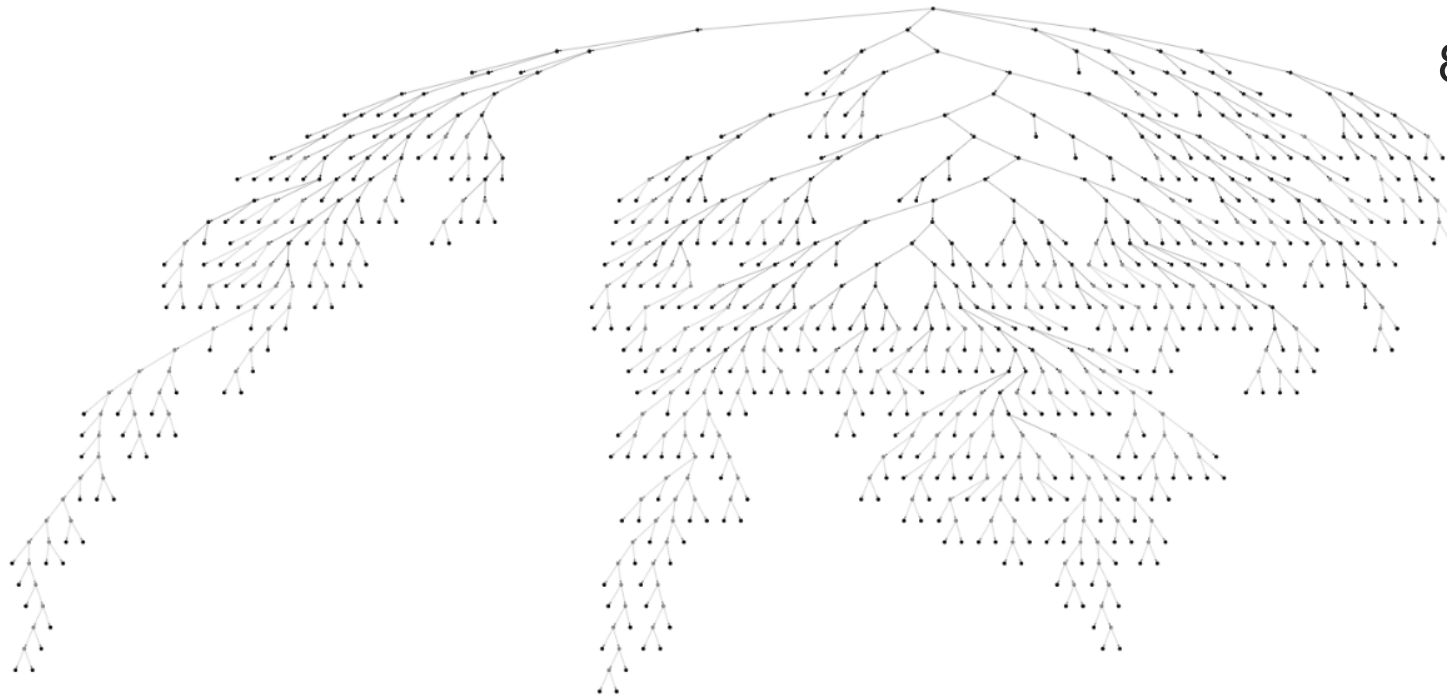
# Impact on our use case



3113 subproblems



879 subproblems



# ACAS Xu benchmark

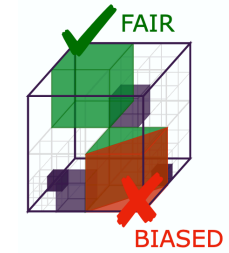
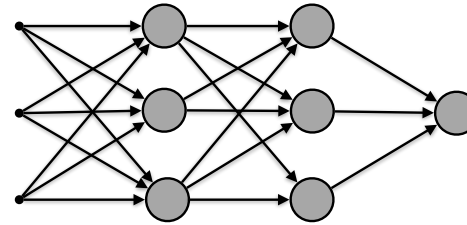
Reported metric: **number of subproblems**

| Properties | Width   | ReCIPH        | Gradient Smear |
|------------|---------|---------------|----------------|
| 1          | 68 567  | <b>15 289</b> | 16 521         |
| 3          | 319 306 | <b>16 140</b> | 209 924        |
| 4          | 25 906  | <b>1 828</b>  | 2 206          |
| 5          | 142 637 | <b>7 023</b>  | 29 169         |
| 9          | 3 055   | <b>2 395</b>  | 3 877          |
| 10         | 1 331   | 335           | <b>217</b>     |



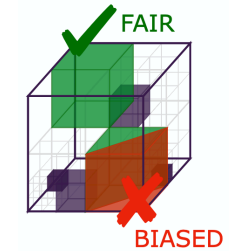
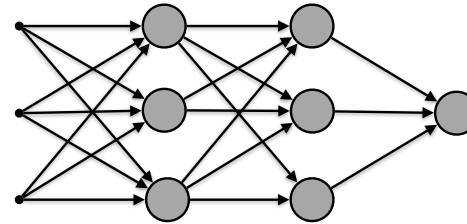
# Wider applicability

- LIBRA (Fairness analysis)



# Wider applicability

- LIBRA (Fairness analysis)



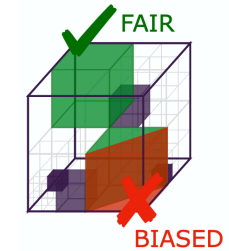
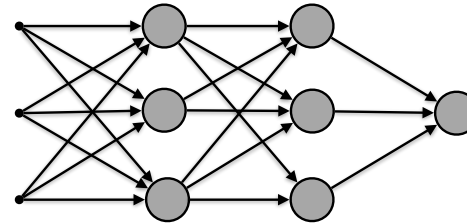
- PyRAT : 6/12 benchmarks in VNN-COMP 2024 where PyRAT finished 2 / 8 tools



# PyRAT

# Wider applicability

- LIBRA (Fairness analysis)



- PyRAT : 6/12 benchmarks in VNN-COMP 2024 where PyRAT finished 2 / 8 tools



**PyRAT**

- Verification remains a hard challenge!

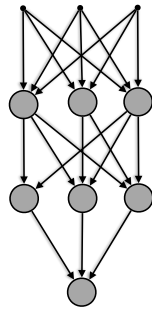


# **2 ■ Over-Approximations for Empirical Robustness**

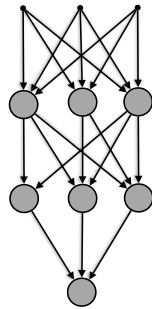
# Local Robustness



# Local Robustness

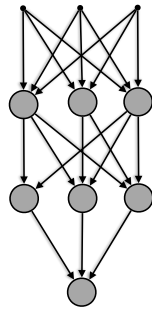


# Local Robustness



Pandas (57.7% confidence)

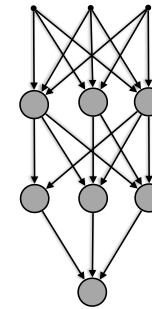
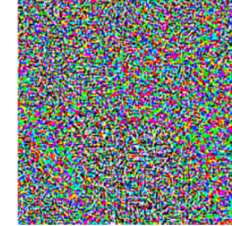
# Local Robustness



Pandas (57.7% confidence)

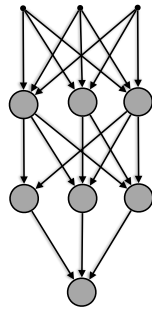


$+ .007 \times$





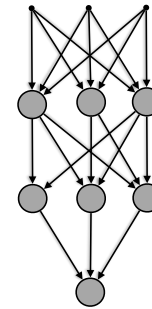
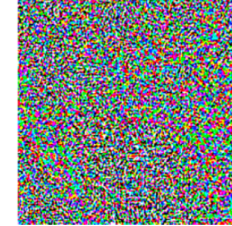
# Local Robustness



Pandas (57.7% confidence)

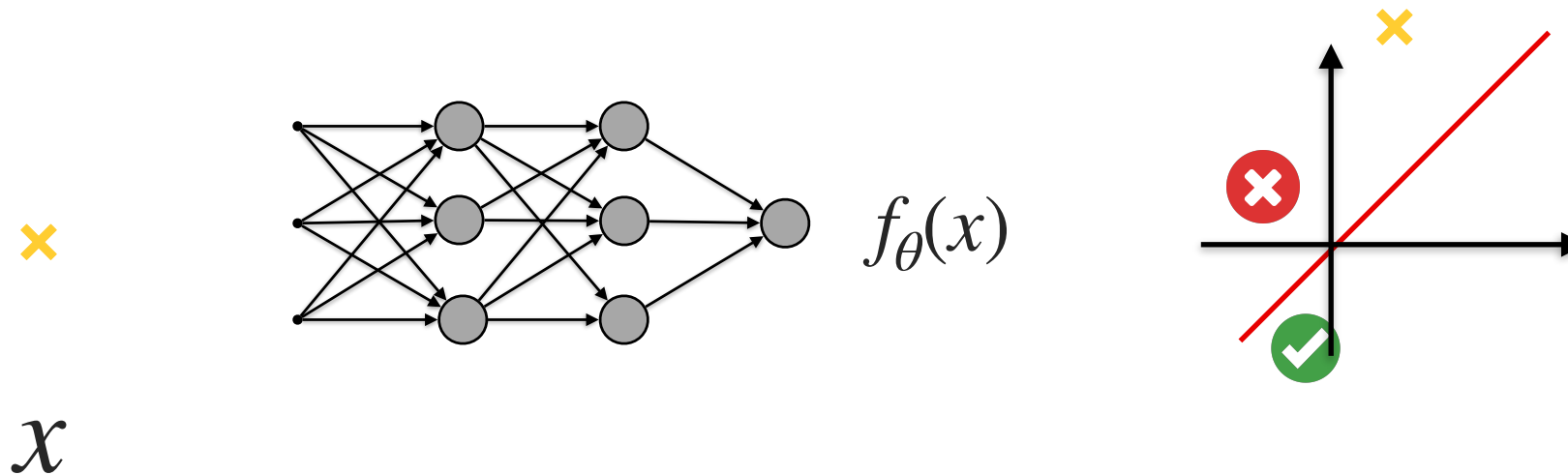


$+ .007 \times$



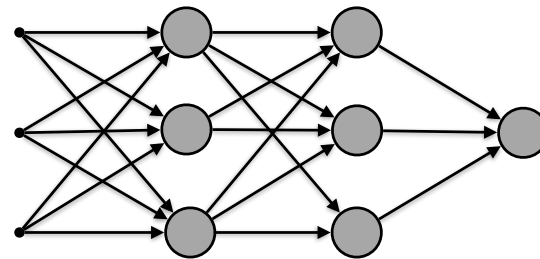
Gibbon (99.3% confidence)

# Neural Network Supervised Training

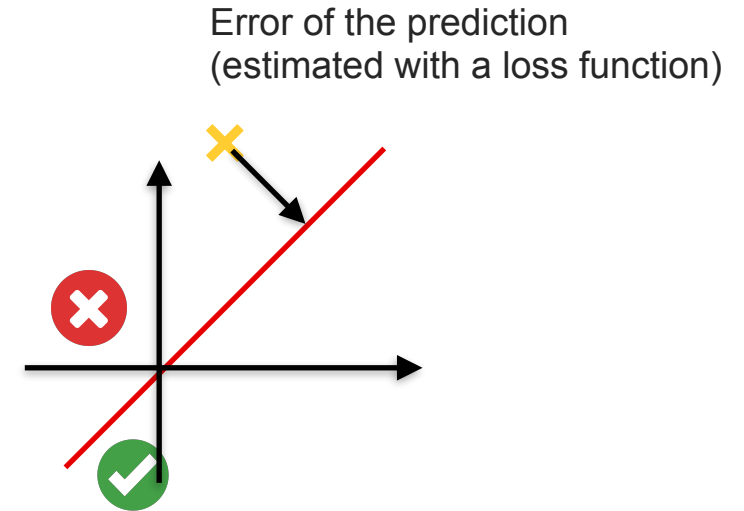


# Neural Network Supervised Training

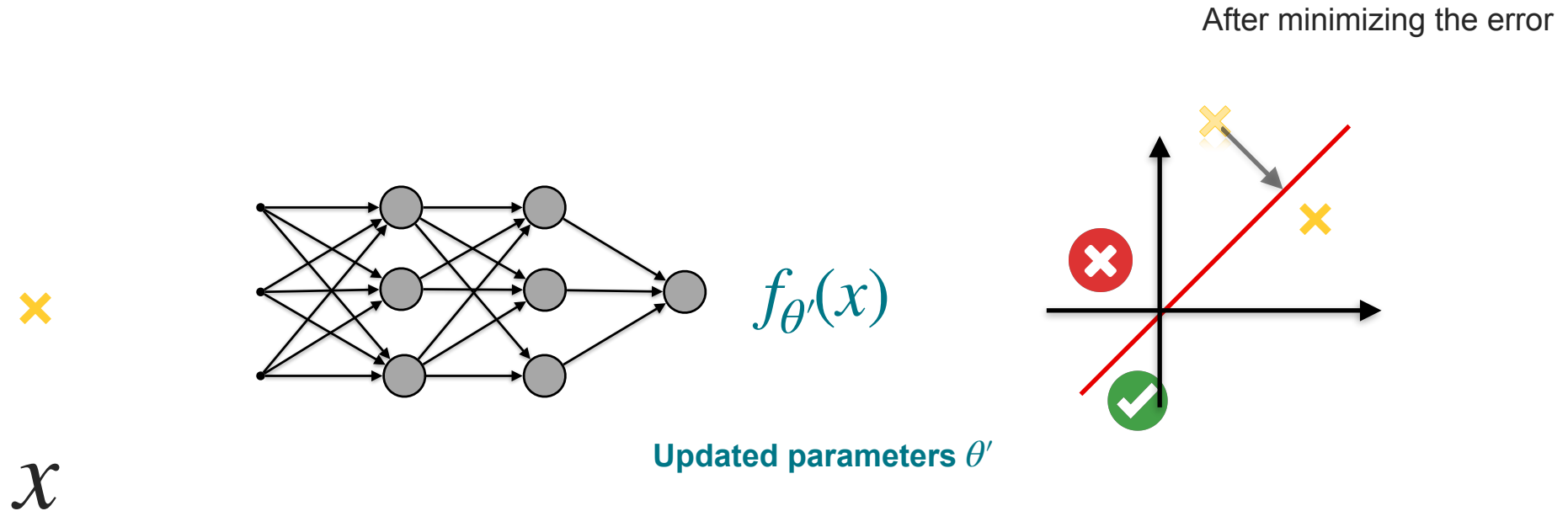
$x$



$f_{\theta}(x)$



# Neural Network Supervised Training

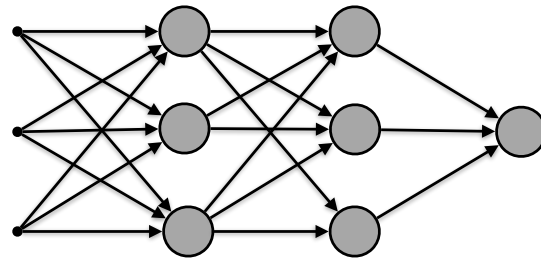


# Robust Training

## Certified Training (CT)

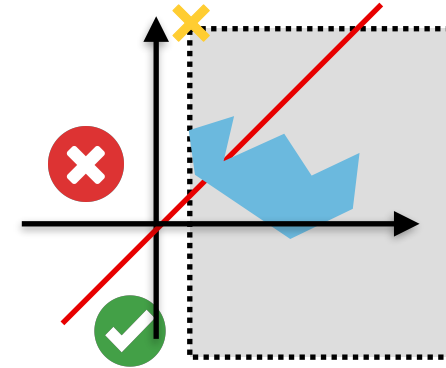


$\mathcal{X}$



$f_{\theta}(\text{blue square})$

Worst case of *the over-approximation of the reachable space*

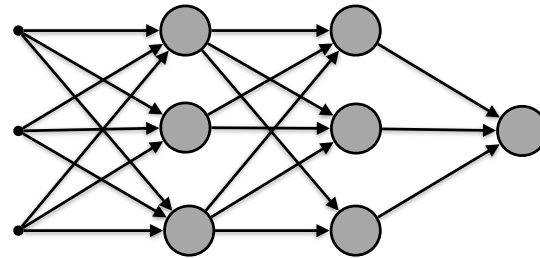


# Robust Training

## Certified Training (CT)

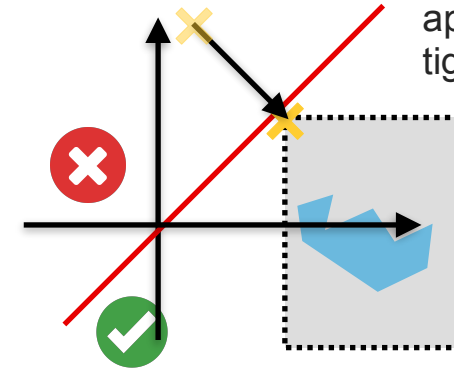


$\mathcal{X}$



$f_{\theta'}(\blacksquare)$

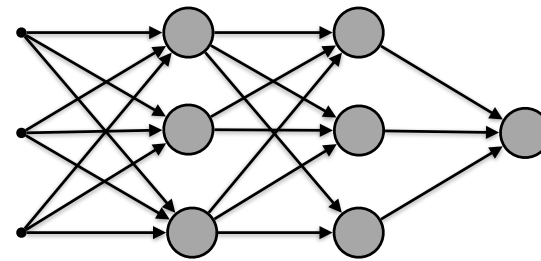
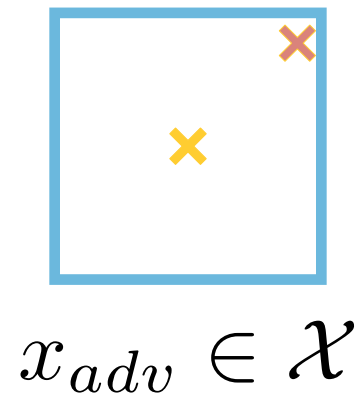
Updated parameters  $\theta'$



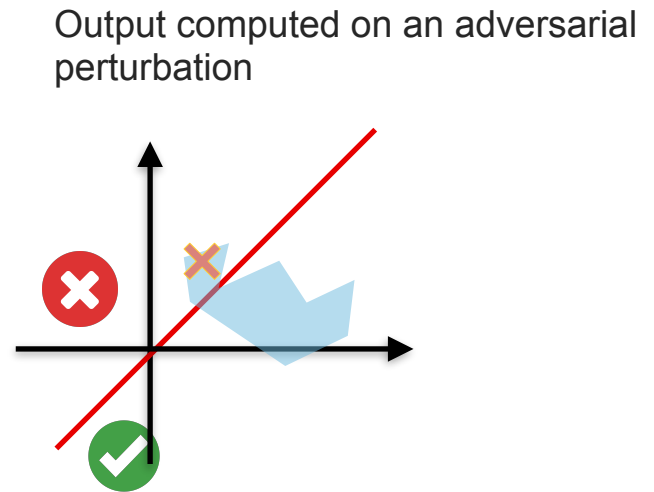
Minimizing the worst-case of the approximation: leads to tighter approximation

# Robust Training

## Adversarial Training (AT)

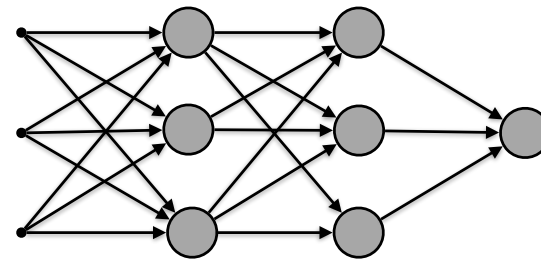
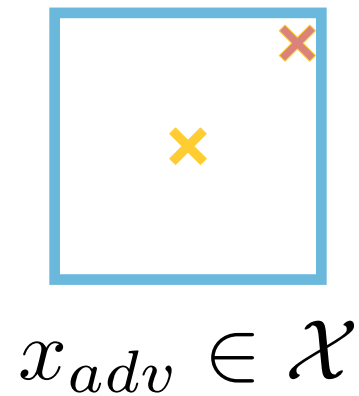


$$f_{\theta}(x_{adv})$$



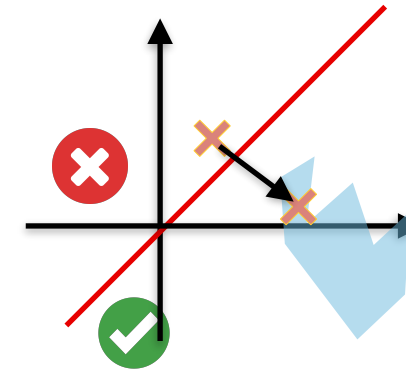
# Robust Training

## Adversarial Training (AT)



$f_{\theta'}(x_{adv})$

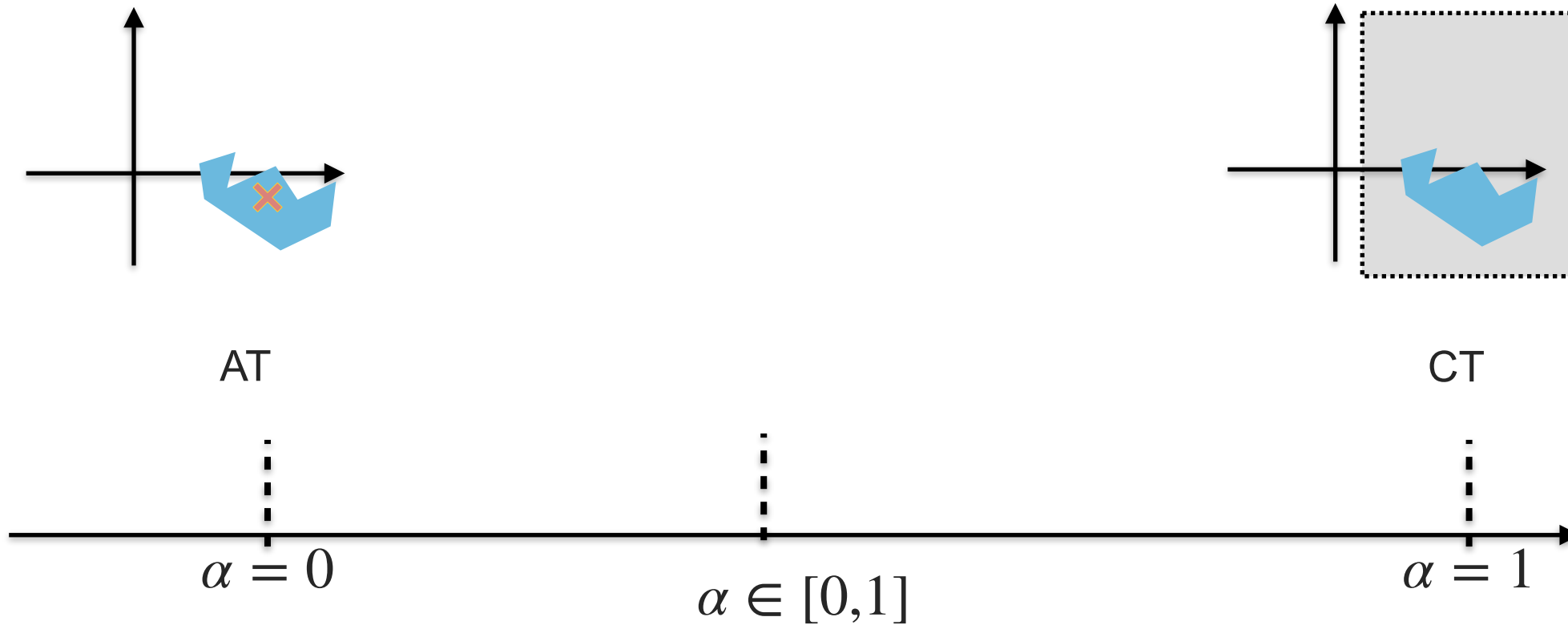
Updated parameters  $\theta'$



Minimizing the error on adversarial perturbations: leads to better **empirical** robustness



# Combining Certified and Adversarial Training (CT + AT)



# Trade-offs



CIFAR-10, CNN-7,  $\epsilon = 8 / 255$ .

| Training Method | Standard (%) | Empirical (%) | Verified (%) |
|-----------------|--------------|---------------|--------------|
| Standard        | 91.27        | 0             | 0            |

# Trade-offs



CIFAR-10, CNN-7,  $\epsilon = 8 / 255$ .

| Training Method | Standard (%) | Empirical (%) | Verified (%) |
|-----------------|--------------|---------------|--------------|
| Standard        | <b>91.27</b> | 0             | 0            |
| AT              | 77.07        | <b>40.14</b>  | 0            |

# Trade-offs



CIFAR-10, CNN-7,  $\epsilon = 8 / 255$ .

| Training Method | Standard (%) | Empirical (%) | Verified (%) |
|-----------------|--------------|---------------|--------------|
| Standard        | <b>91.27</b> | 0             | 0            |
| AT              | 77.07        | <b>40.14</b>  | 0            |
| CT              | 48.51        | 35.48         | 34.97        |

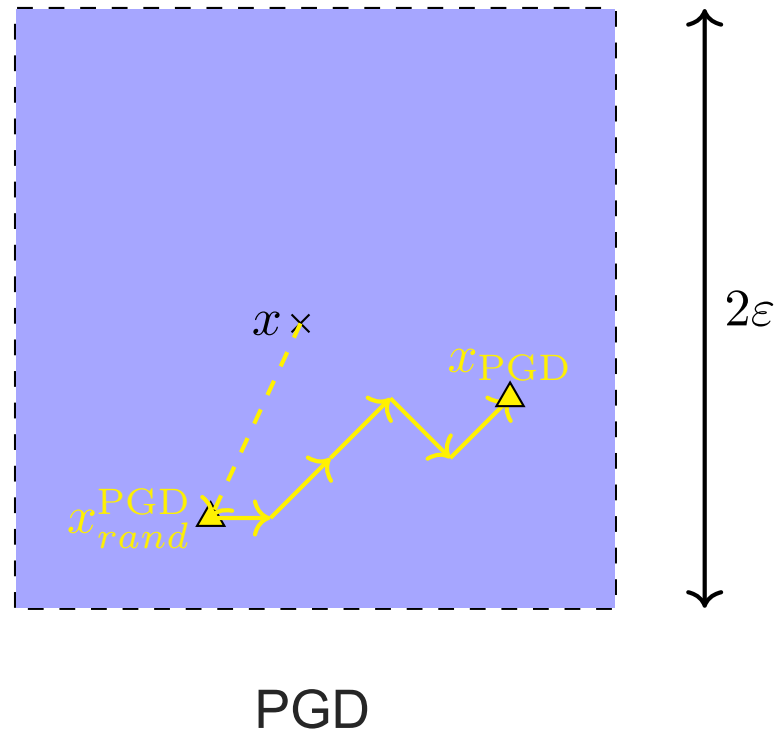
# Trade-offs



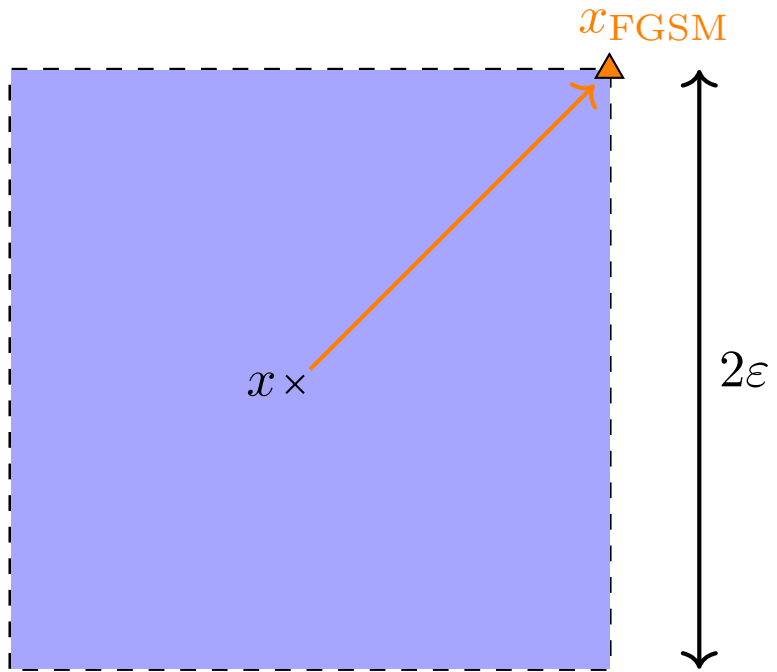
CIFAR-10, CNN-7,  $\epsilon = 8 / 255$ .

| Training Method | Standard (%) | Empirical (%) | Verified (%) |
|-----------------|--------------|---------------|--------------|
| Standard        | <b>91.27</b> | 0             | 0            |
| AT              | 77.07        | <b>40.14</b>  | 0            |
| CT              | 48.51        | 35.48         | 34.97        |
| CT + AT         | 53.35        | 36.02         | <b>35.44</b> |

# Cost of adversarial training:

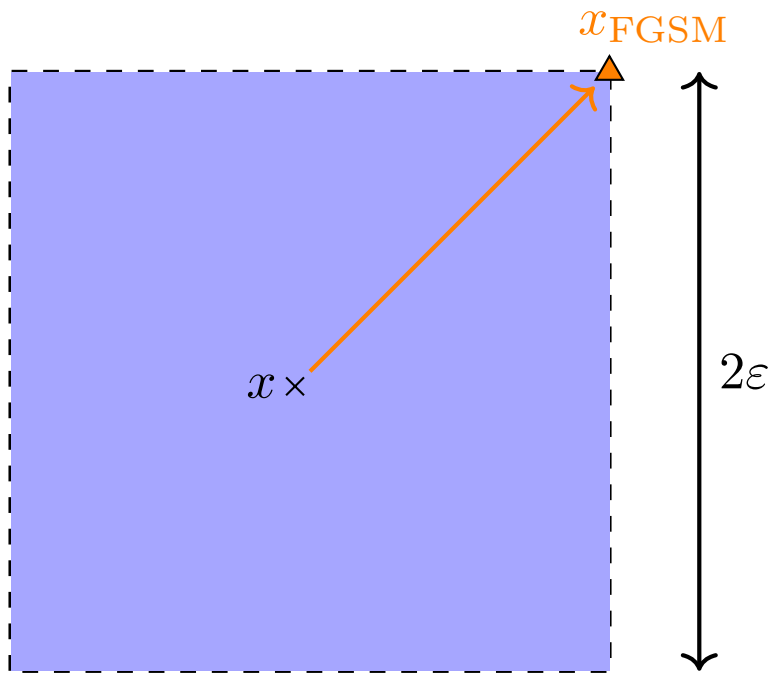


# Single Step Training?



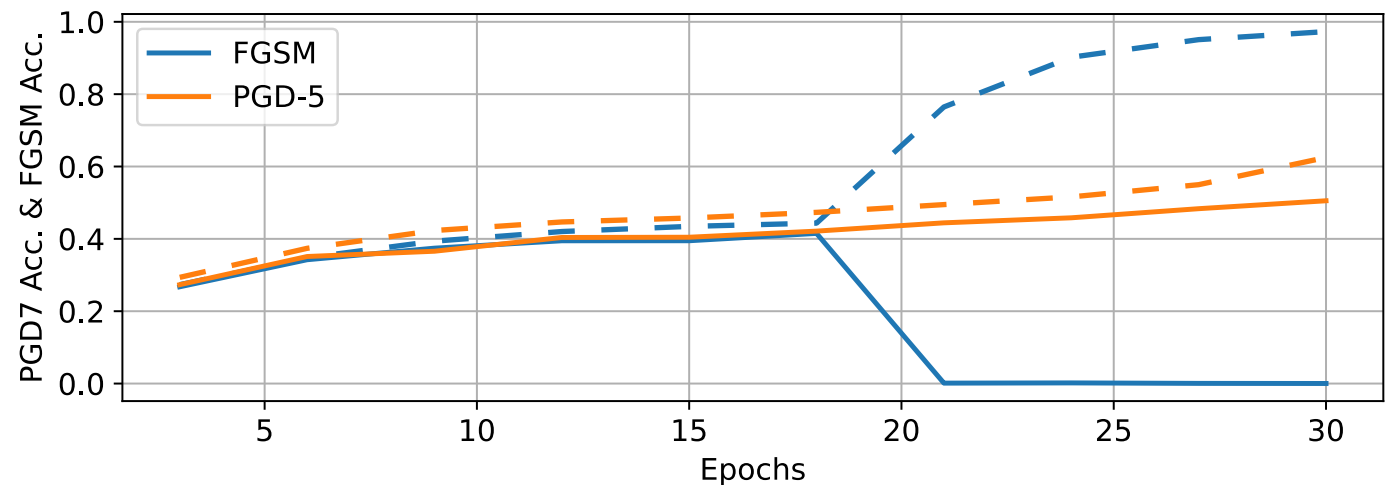
FGSM

# Catastrophic Overfitting



FGSM

CIFAR10, PreactResnet18,  $\epsilon = 8/255$



Robust Accuracy against **multi-step attacks**

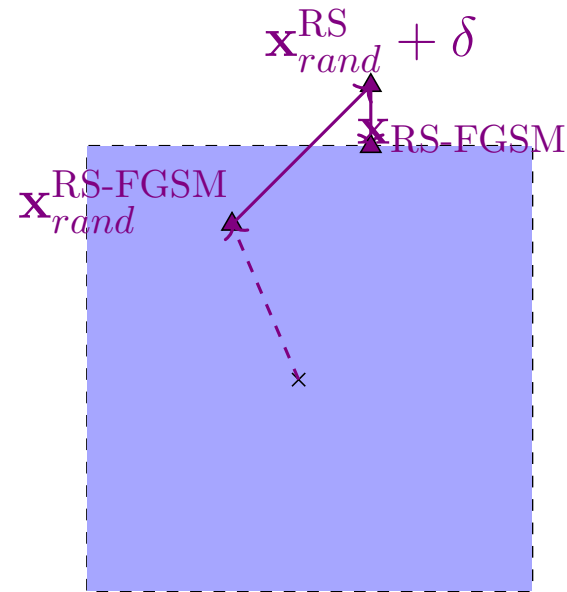


Robust Accuracy against **single-step attacks**



# Mitigations

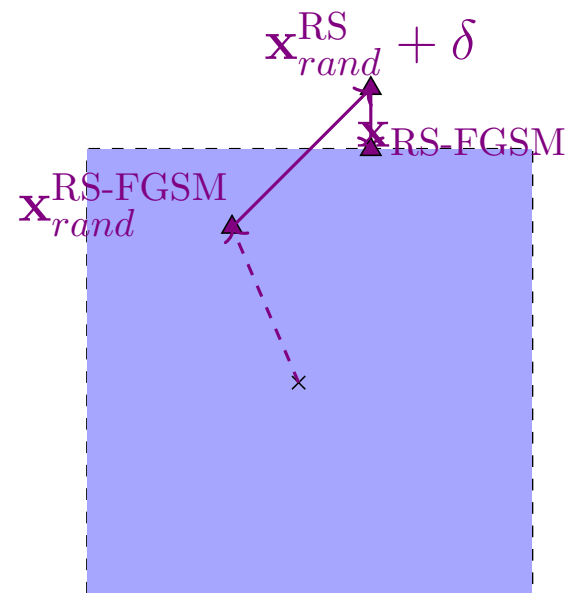
- Noisy single steps attacks



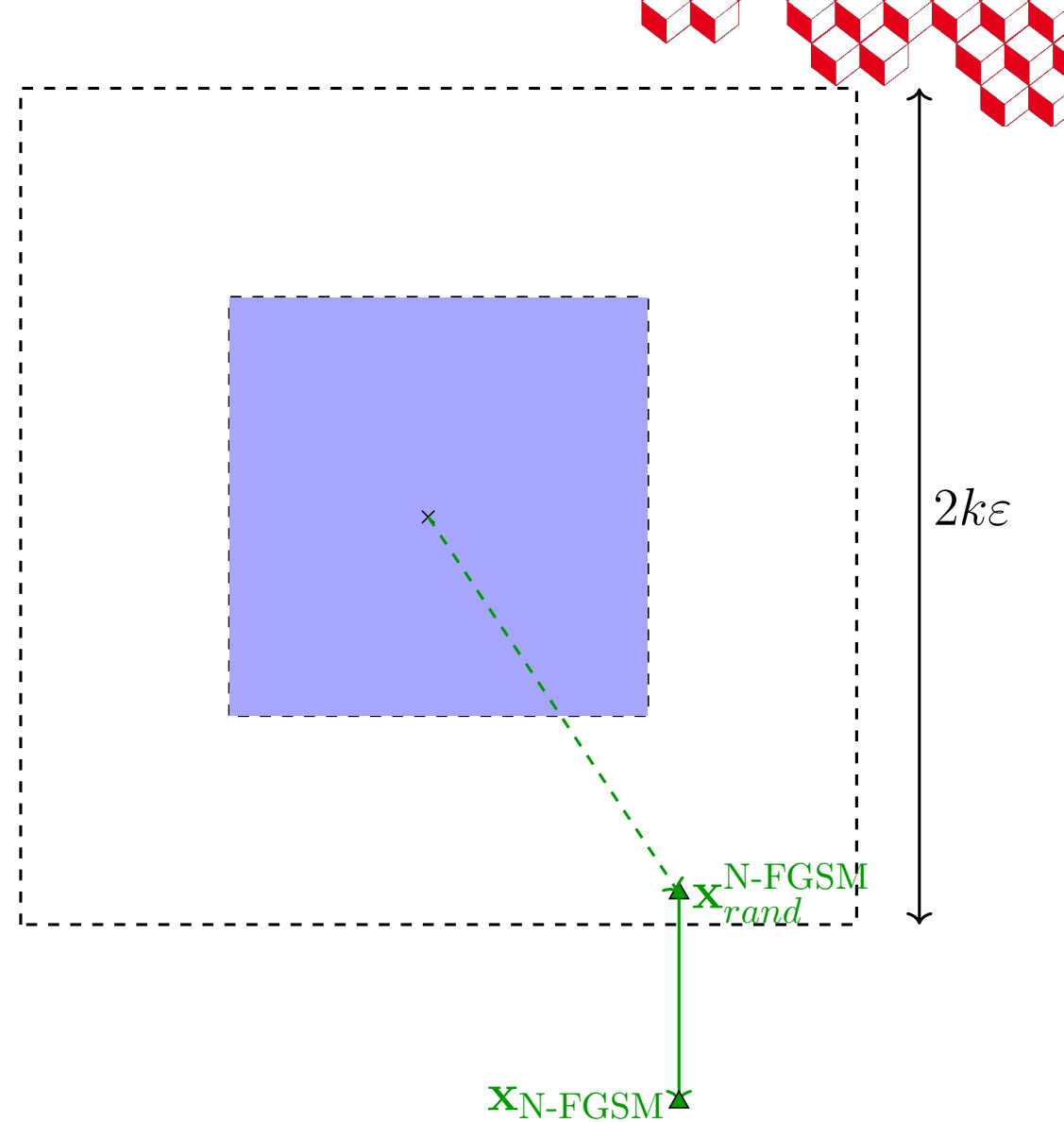
RS-FGSM

# Mitigations

- Noisy single steps attacks



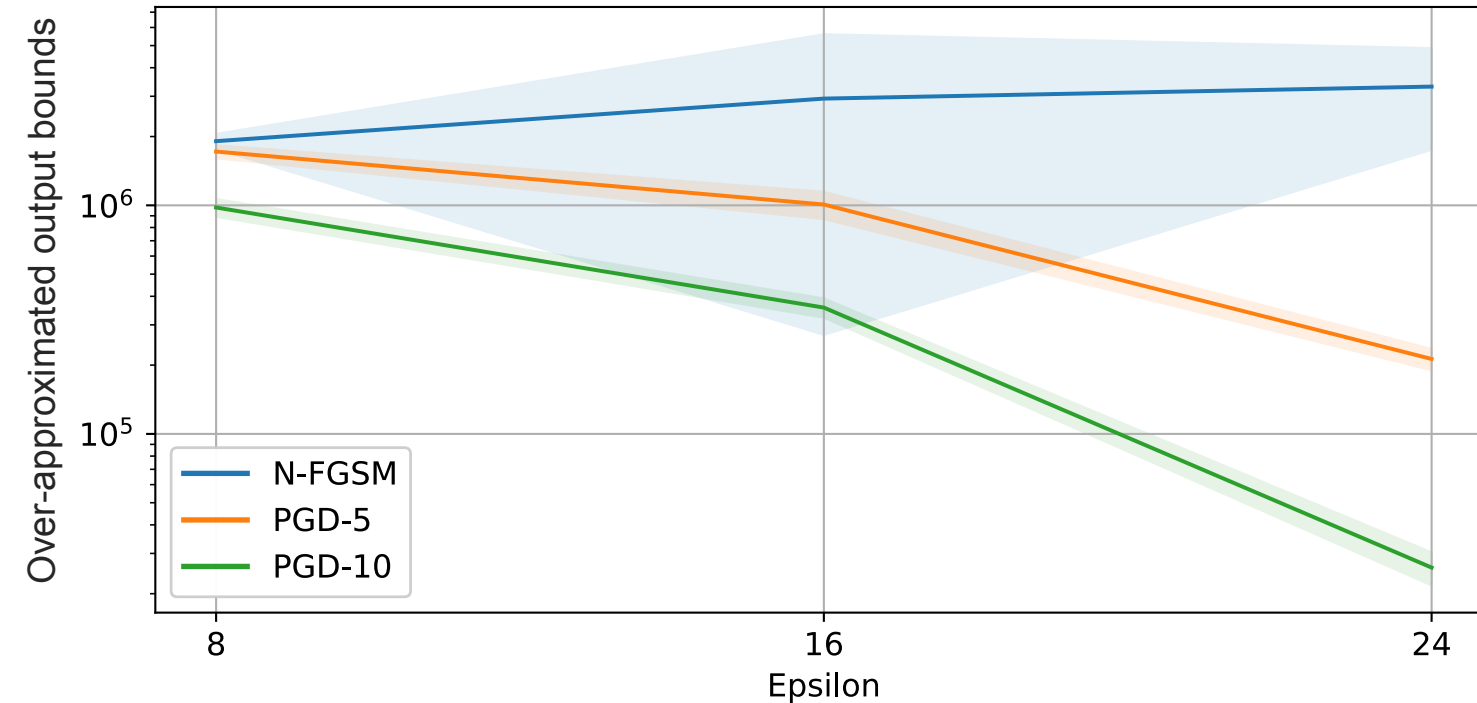
RS-FGSM



N-FGSM

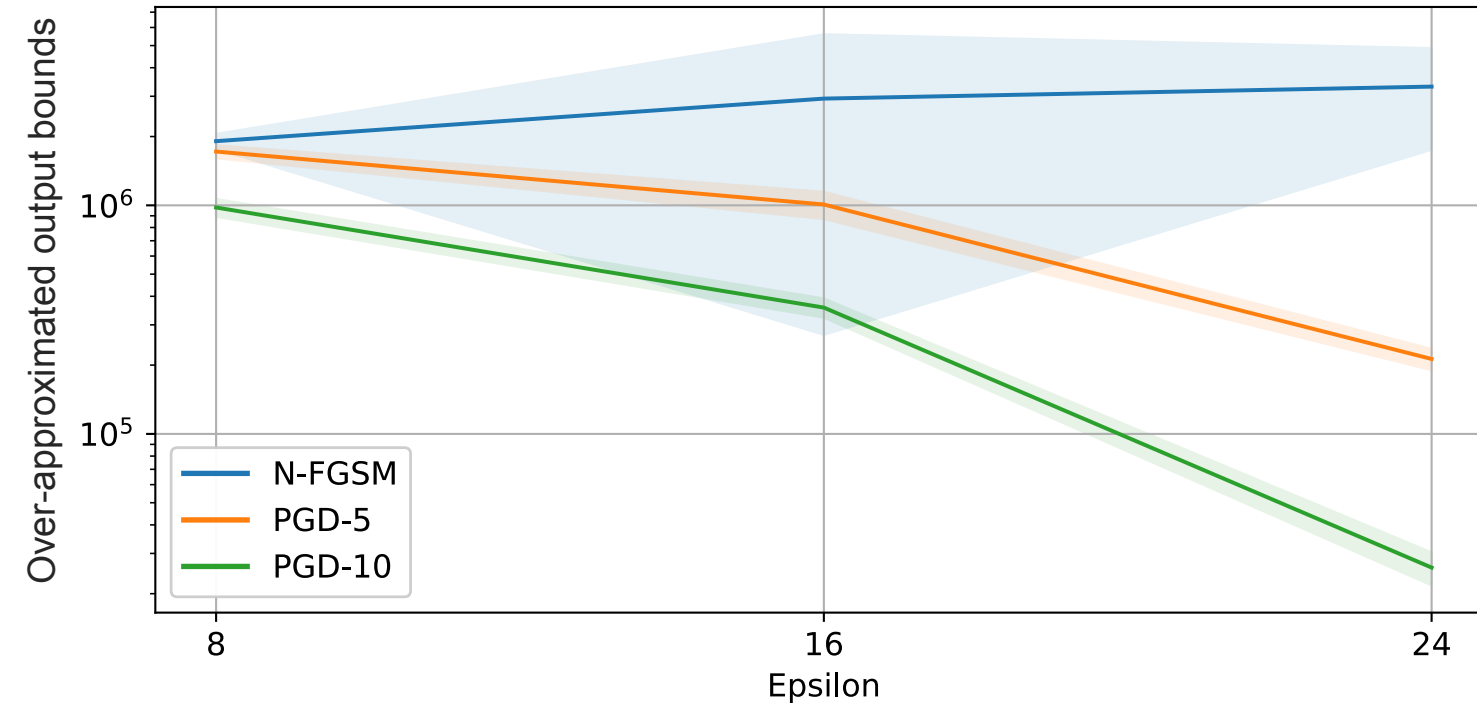
# Over-approximations and catastrophic overfitting

CIFAR10, PreactResnet18. Mean and 95% CI over 5 runs



# Over-approximations and catastrophic overfitting

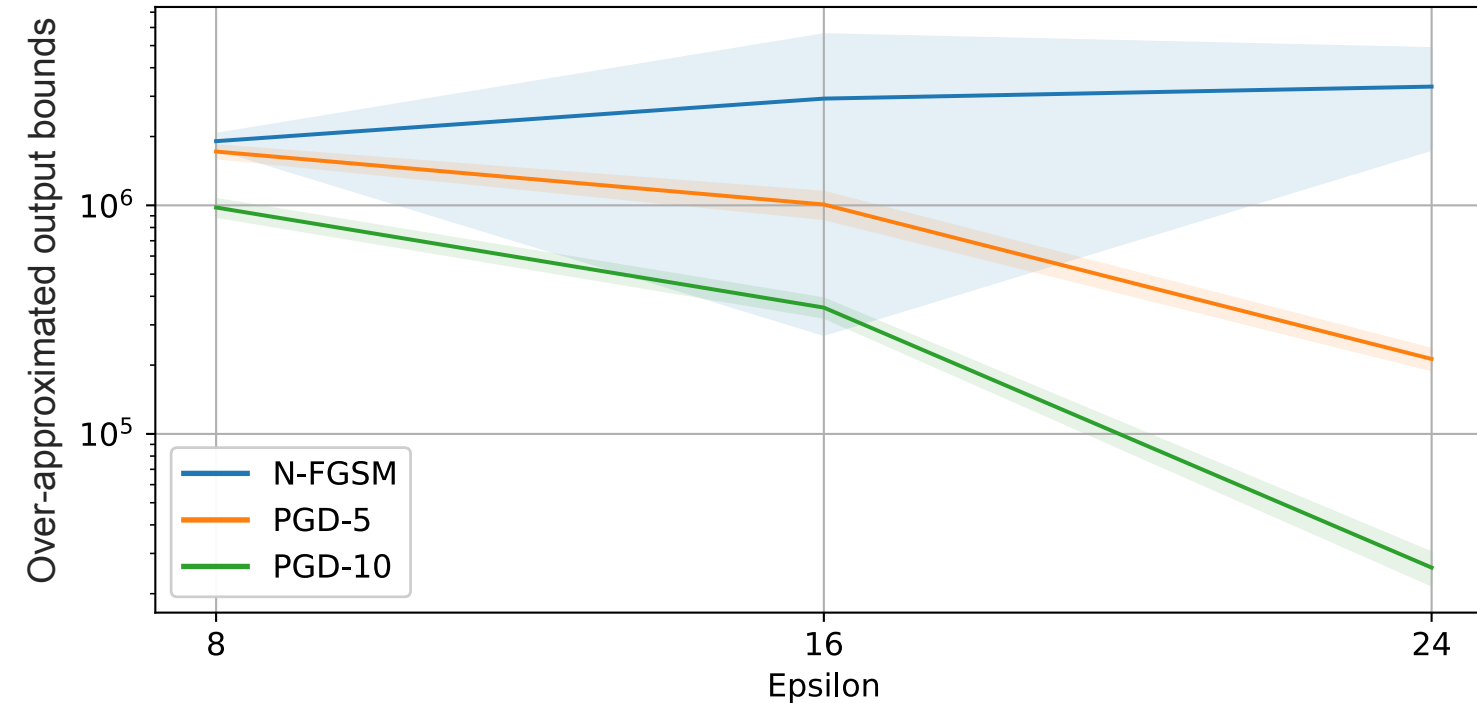
CIFAR10, PreactResnet18. Mean and 95% CI over 5 runs



- Bounds decrease with stronger attacks

# Over-approximations and catastrophic overfitting

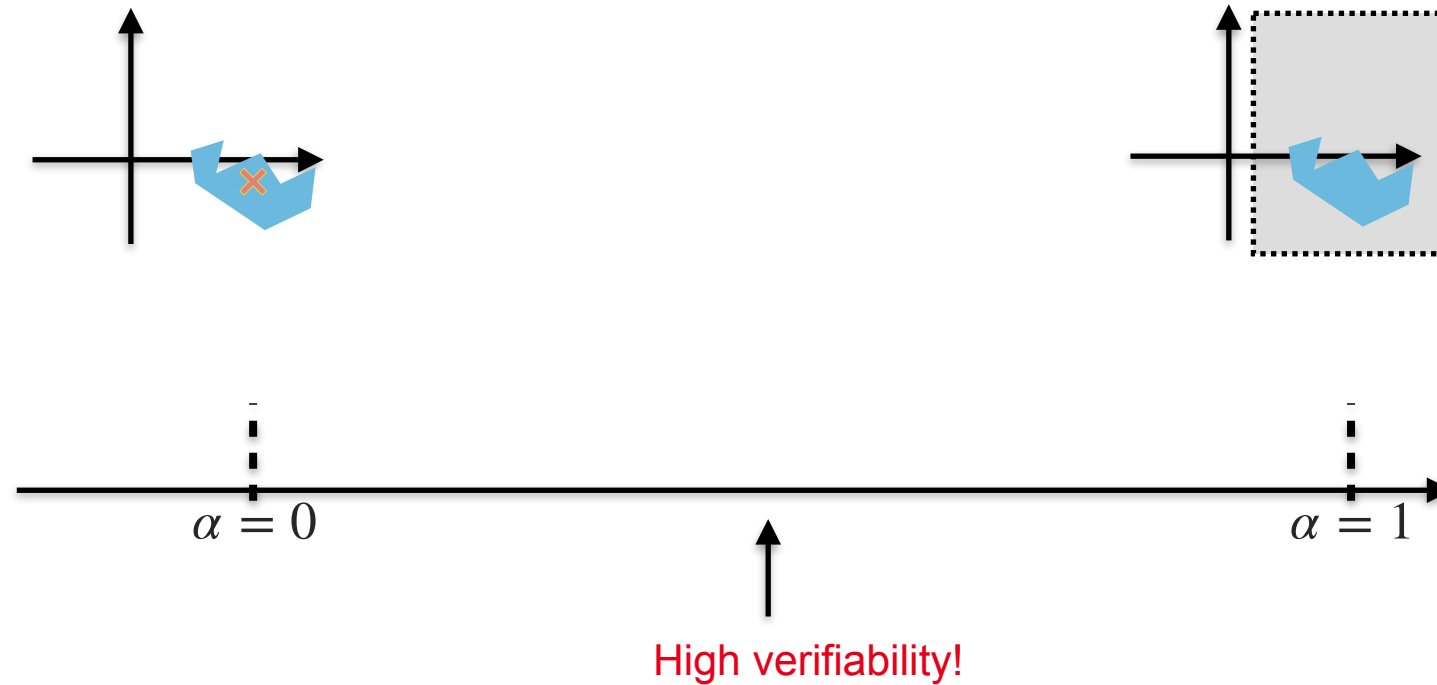
CIFAR10, PreactResnet18. Mean and 95% CI over 5 runs



- **Bounds decrease with stronger attacks**
- Adding noise is not enough to **consistently mitigate catastrophic overfitting?**

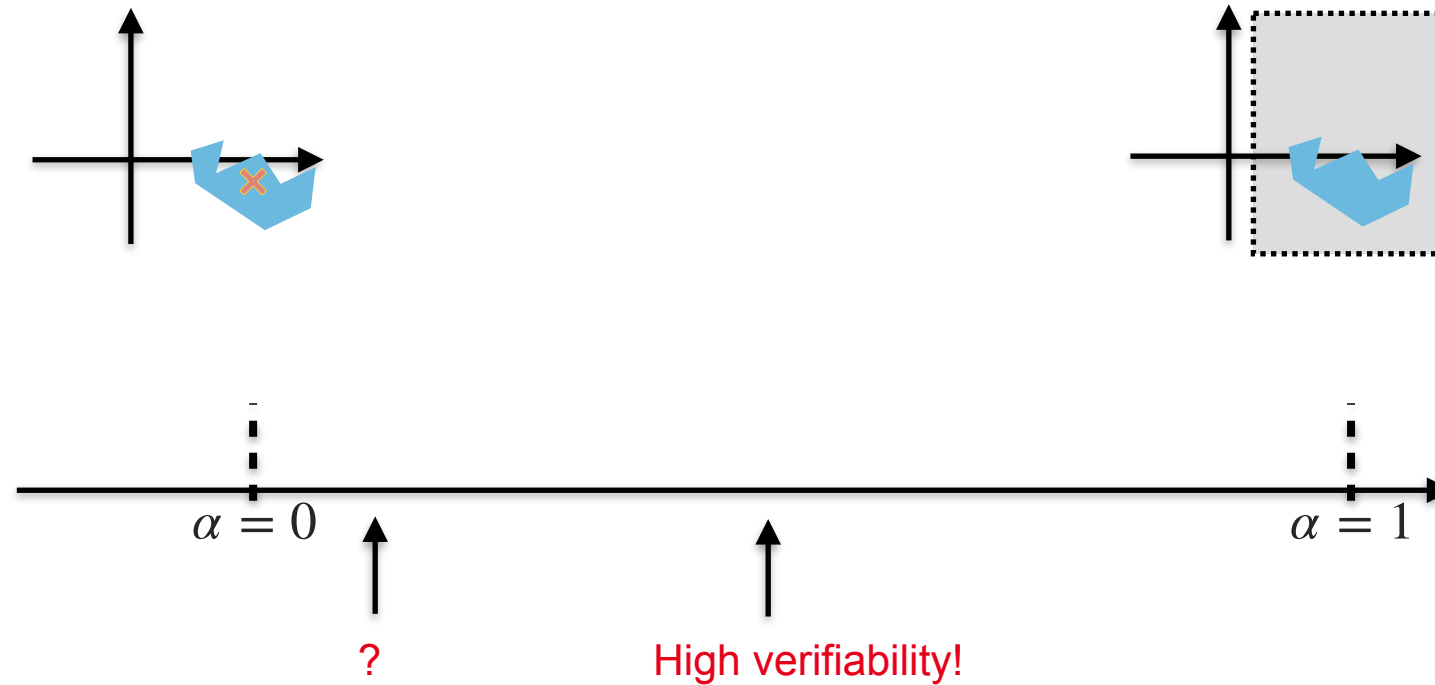
# Over-approximations and catastrophic overfitting

Our proposal: use over-approximations at training time for empirical robustness



# Over-approximations and catastrophic overfitting

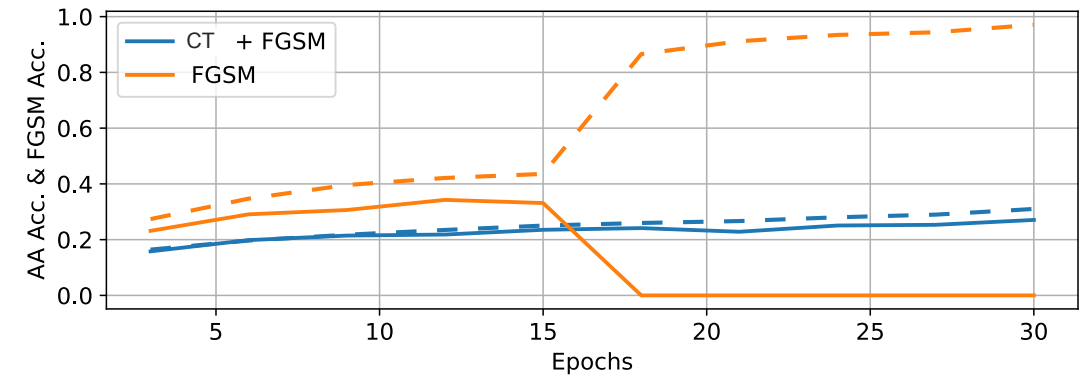
Our proposal: use over-approximations at training time for empirical robustness



# CT for Empirical Robustness

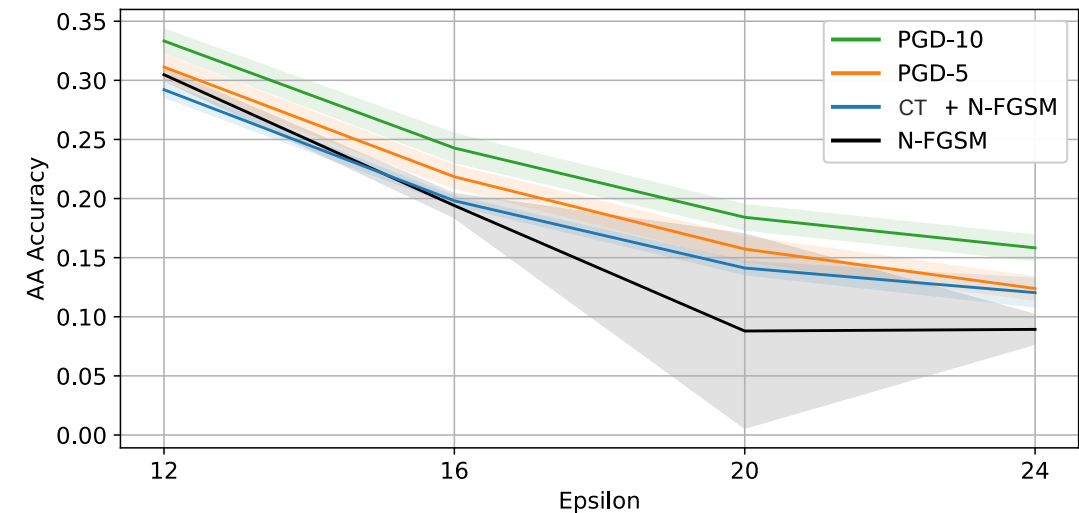
- CT + FGSM can prevent catastrophic overfitting on FGSM

CIFAR10, PreactResnet18,  $\epsilon = 8/255$



- Also prevent CO and improves robustness for N-FGSM

CIFAR10, PreactResnet18



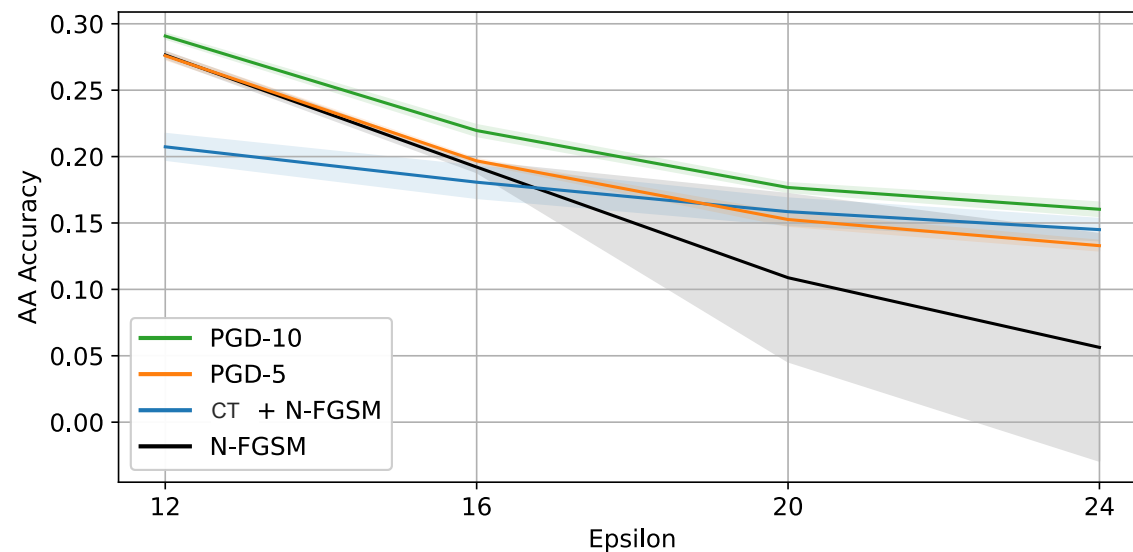


# CT for Empirical Robustness

Performance improvement (w.r.t underlying attack):

- On longer training schedules
- On easier datasets
- On shallower networks

CIFAR10, CNN-7, 30 Epochs

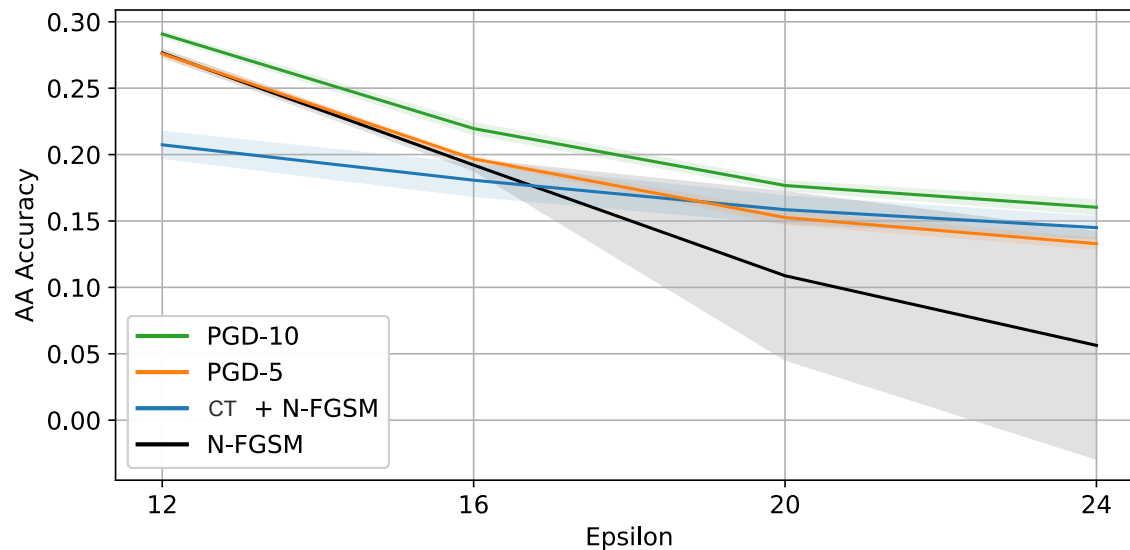


# CT for Empirical Robustness

Performance improvement (w.r.t underlying attack):

- On longer training schedules
- On easier datasets
- On shallower networks

CIFAR10, CNN-7, 30 Epochs



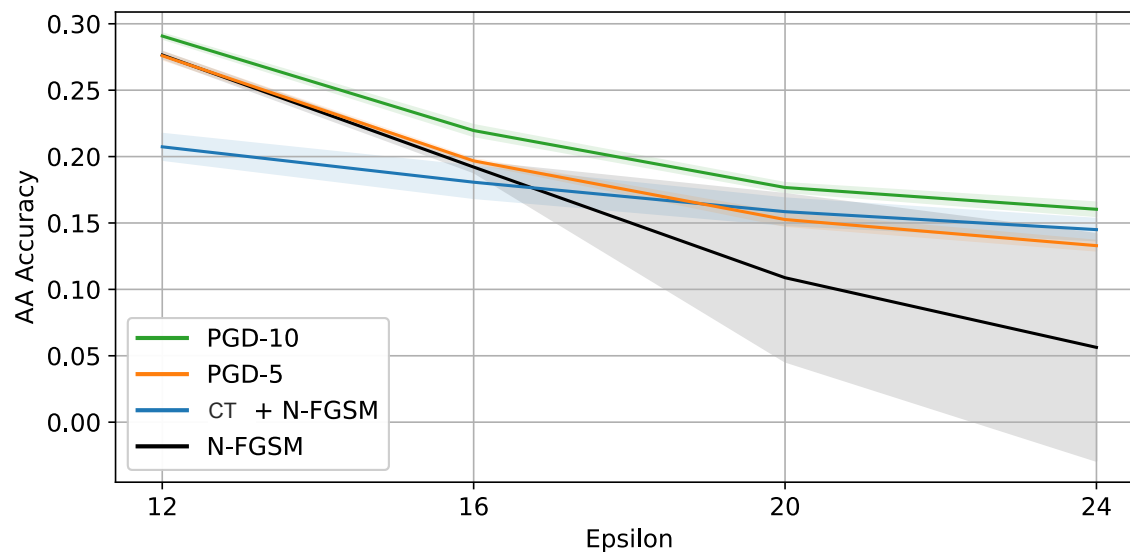
CIFAR10, CNN-7, 160 Epochs

# CT for Empirical Robustness

Performance improvement (w.r.t underlying attack):

- On longer training schedules
- On easier datasets
- On shallower networks

CIFAR10, CNN-7, 30 Epochs



CIFAR10, CNN-7, 160 Epochs

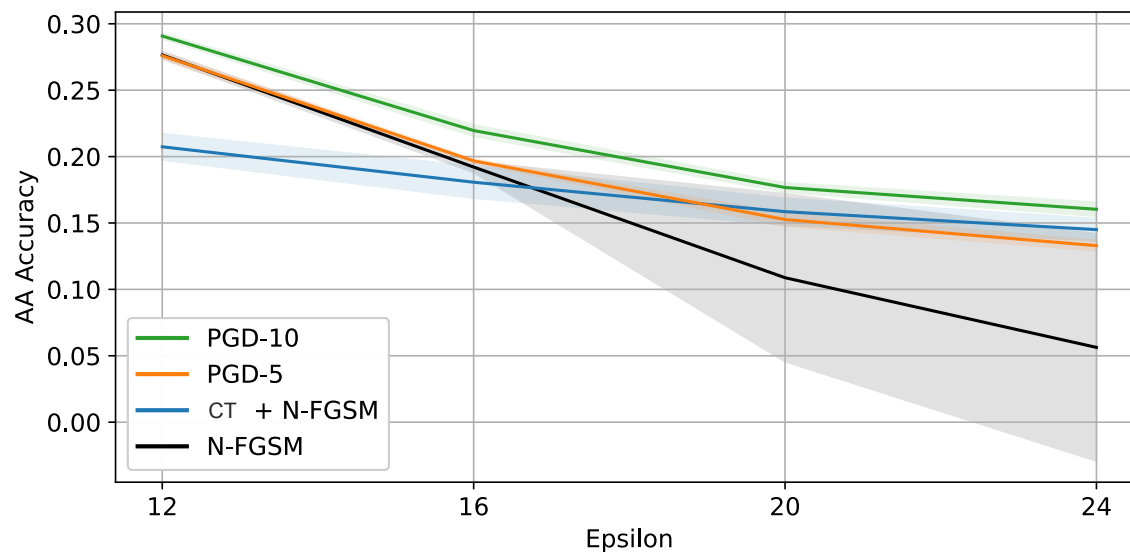
| Training Method | Empirical (%) |
|-----------------|---------------|
| CT + N-FGSM     | 14.79         |
| CT + PGD-5      | <b>16.45</b>  |
| PGD-10          | 15.14         |
| PGD-5           | 11.47         |
| N-FGSM          | 0.23          |

# CT for Empirical Robustness

Performance improvement (w.r.t underlying attack):

- On longer training schedules
- On easier datasets
- On shallower networks

CIFAR10, CNN-7, 30 Epochs



CIFAR10, CNN-7, 160 Epochs

| Training Method | Empirical (%) |
|-----------------|---------------|
| CT + N-FGSM     | 14.79         |
| CT + PGD-5      | <b>16.45</b>  |
| PGD-10          | 15.14         |
| PGD-5           | 11.47         |
| N-FGSM          | 0.23          |

Can over-approximations be used beyond robustness?

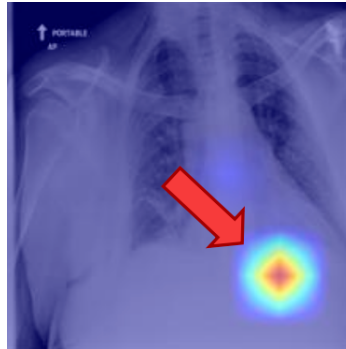
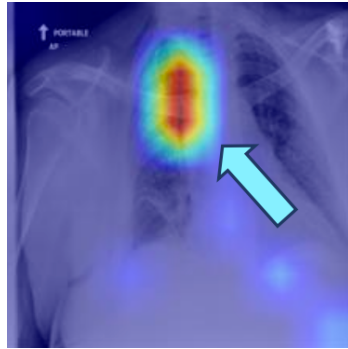
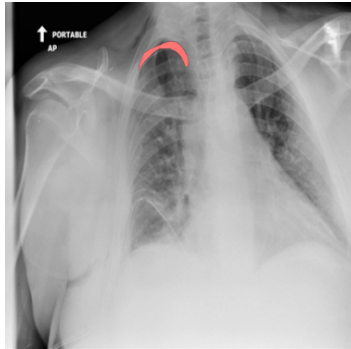


# **3 ■ Over-approximations for Explainability**

# Feature Attributions

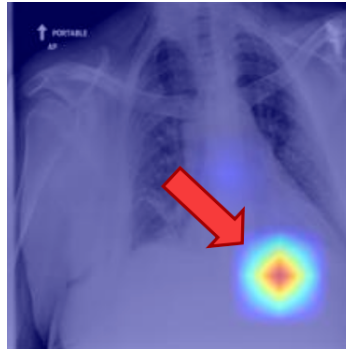
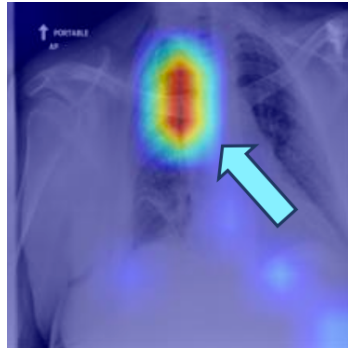
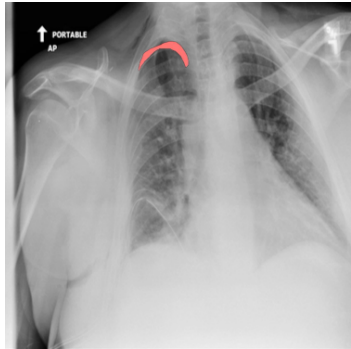
Right for the wrong reasons, the need for Explainability

- Identify inputs contributing to the network's decision: improve trust, help diagnose bad models ...



# Feature Attributions

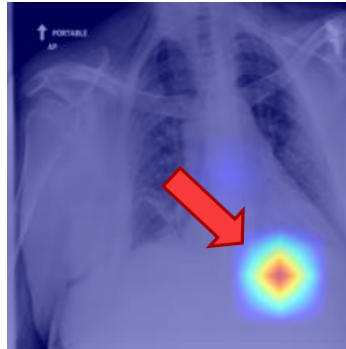
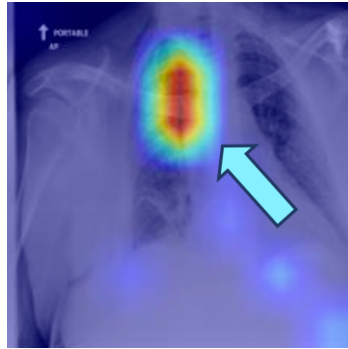
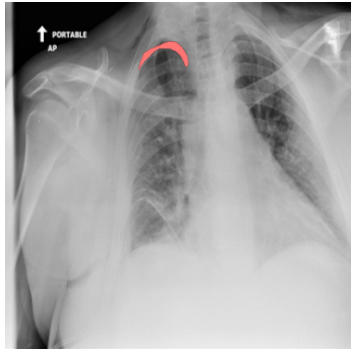
## Right for the wrong reasons, the need for Explainability



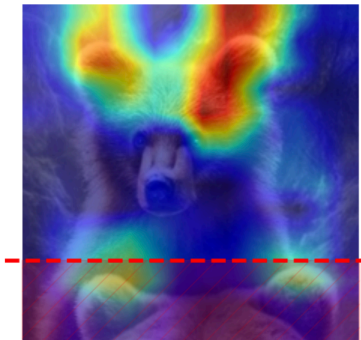
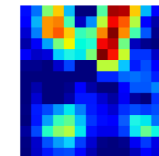
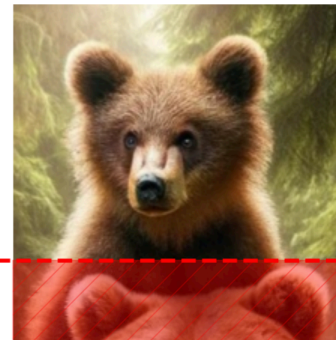
- Identify inputs contributing to the network's decision: improve trust, help diagnose bad models ...
- But what if the method does not highlight truly important inputs?

# Feature Attributions

## Right for the wrong reasons, the need for Explainability

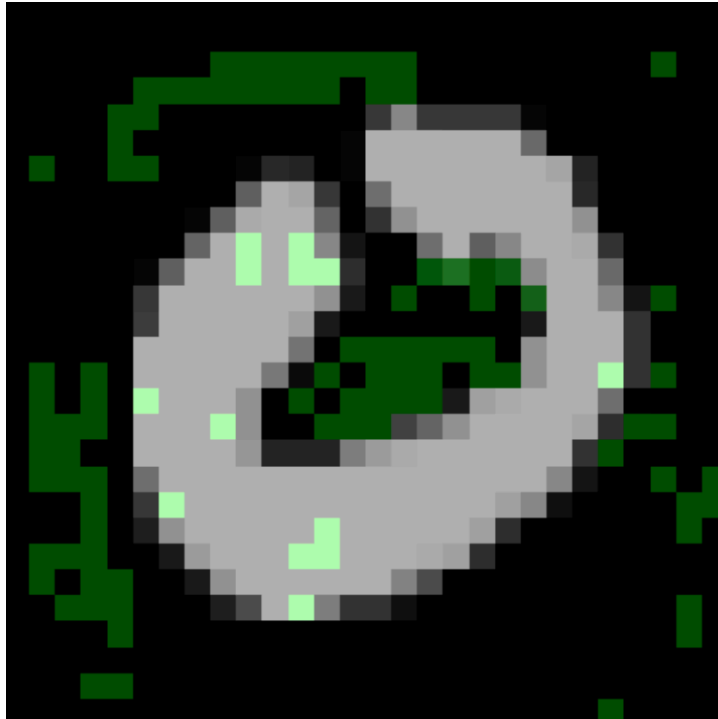


- Identify inputs contributing to the network's decision: improve trust, help diagnose bad models ...
- But what if the method does not highlight truly important inputs?



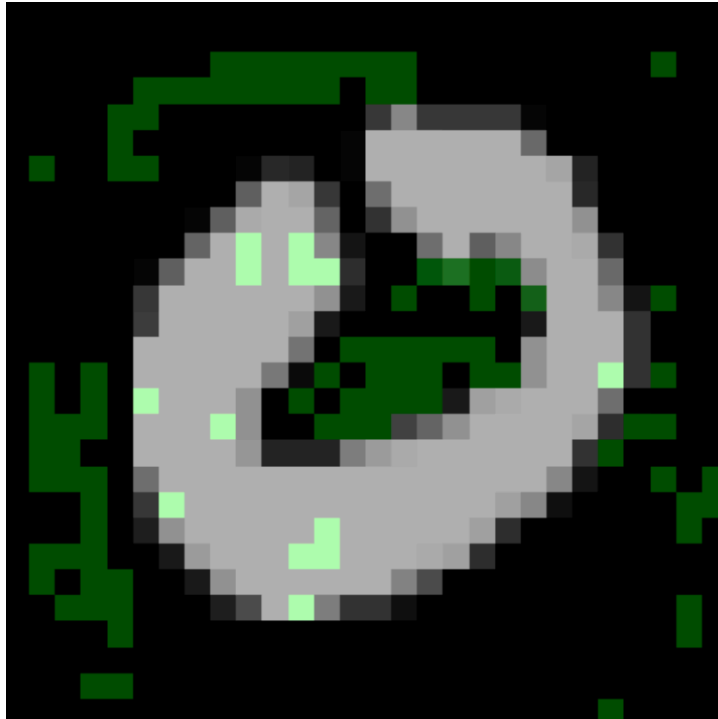


# Formally Robust Explanations



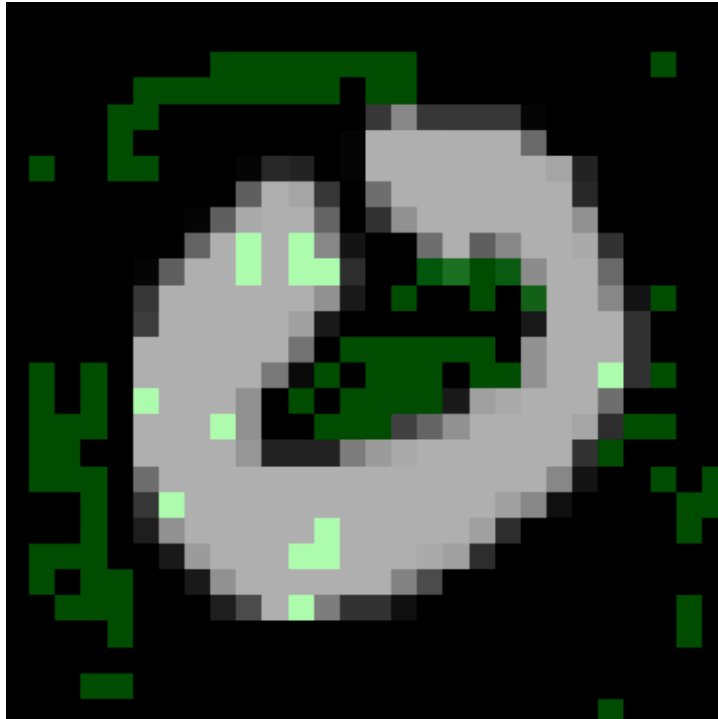
- Binary attribution: irrelevant or relevant

# Formally Robust Explanations



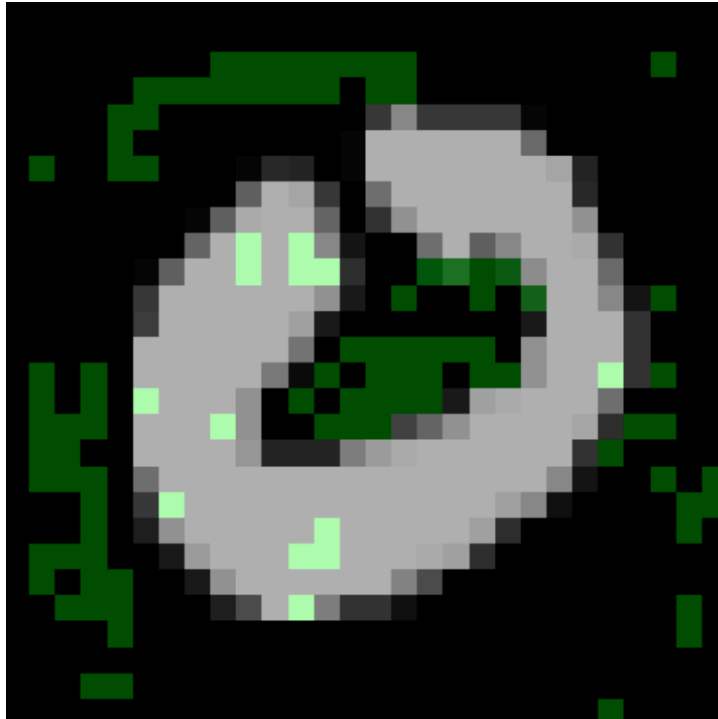
- Binary attribution: irrelevant or relevant
- **Irrelevant pixel:** perturbing it (*locally*) does not change the prediction (= provable robustness!)

# Formally Robust Explanations



- Binary attribution: irrelevant or relevant
- **Irrelevant pixel:** perturbing it (*locally*) does not change the prediction (= provable robustness!)
- **Subset-Optimal:** if perturbing any new pixel together with the irrelevant features changes the prediction

# Formally Robust Explanations



- Binary attribution: irrelevant or relevant
- **Irrelevant pixel:** perturbing it (*locally*) does not change the prediction (= provable robustness!)
- **Subset-Optimal:** if perturbing any new pixel together with the irrelevant features changes the prediction
- Key metric: size of the irrelevant feature set (larger = simpler explanation)

# Formally Robust Explanations



# Formally Robust Explanations

$x_1$

$x_2$

$x_3$

$x_4$

$x_5$

$x_6$

# Formally Robust Explanations

|       |  |
|-------|--|
| $x_1$ |  |
| $x_2$ |  |
| $x_3$ |  |
| $x_4$ |  |
| $x_5$ |  |
| $x_6$ |  |

# Formally Robust Explanations

|       |  |
|-------|--|
| $x_1$ |  |
| $x_2$ |  |
| $x_3$ |  |
| $x_4$ |  |
| $x_5$ |  |
| $x_6$ |  |

Candidate

Irrelevant

Relevant



# Formally Robust Explanations

|       |  |  |
|-------|--|--|
| $x_1$ |  |  |
| $x_2$ |  |  |
| $x_3$ |  |  |
| $x_4$ |  |  |
| $x_5$ |  |  |
| $x_6$ |  |  |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |  |  |  |
|-------|--|--|--|
| $x_1$ |  |  |  |
| $x_2$ |  |  |  |
| $x_3$ |  |  |  |
| $x_4$ |  |  |  |
| $x_5$ |  |  |  |
| $x_6$ |  |  |  |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |  |  |  |  |
|-------|--|--|--|--|
| $x_1$ |  |  |  |  |
| $x_2$ |  |  |  |  |
| $x_3$ |  |  |  |  |
| $x_4$ |  |  |  |  |
| $x_5$ |  |  |  |  |
| $x_6$ |  |  |  |  |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|
| $x_1$ | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_2$ | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant |
| $x_3$ | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   |
| $x_4$ | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant |
| $x_5$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  |
| $x_6$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |            |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|------------|
| $x_1$ | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_2$ | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_3$ | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   | Relevant   |
| $x_4$ | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant | Irrelevant |
| $x_5$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Relevant   |
| $x_6$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |            |            |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|------------|------------|
| $x_1$ | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_2$ | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_3$ | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   | Relevant   | Relevant   |
| $x_4$ | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant |
| $x_5$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   |
| $x_6$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant |

Candidate

Irrelevant

Relevant

# Formally Robust Explanations

|       |            |            |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|------------|------------|
| $x_1$ | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_2$ | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_3$ | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   | Relevant   | Relevant   |
| $x_4$ | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant |
| $x_5$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   |
| $x_6$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant |

Candidate

Irrelevant

Relevant

- Each step: solving a verification problem!

# Formally Robust Explanations

|       |            |            |            |            |            |            |            |
|-------|------------|------------|------------|------------|------------|------------|------------|
| $x_1$ | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_2$ | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant |
| $x_3$ | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   | Relevant   | Relevant   |
| $x_4$ | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant | Irrelevant | Irrelevant |
| $x_5$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Relevant   | Relevant   |
| $x_6$ | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Irrelevant | Candidate  | Irrelevant |

Candidate

Irrelevant

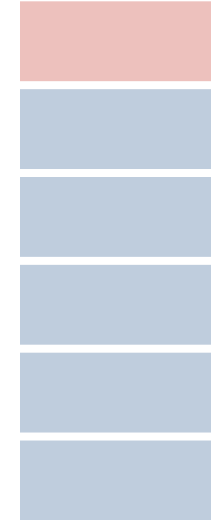
Relevant

- Each step: solving a verification problem!
- Active area of research to increase scalability



# Over-approximations For Explainability

- Use over-approximations for scalability?



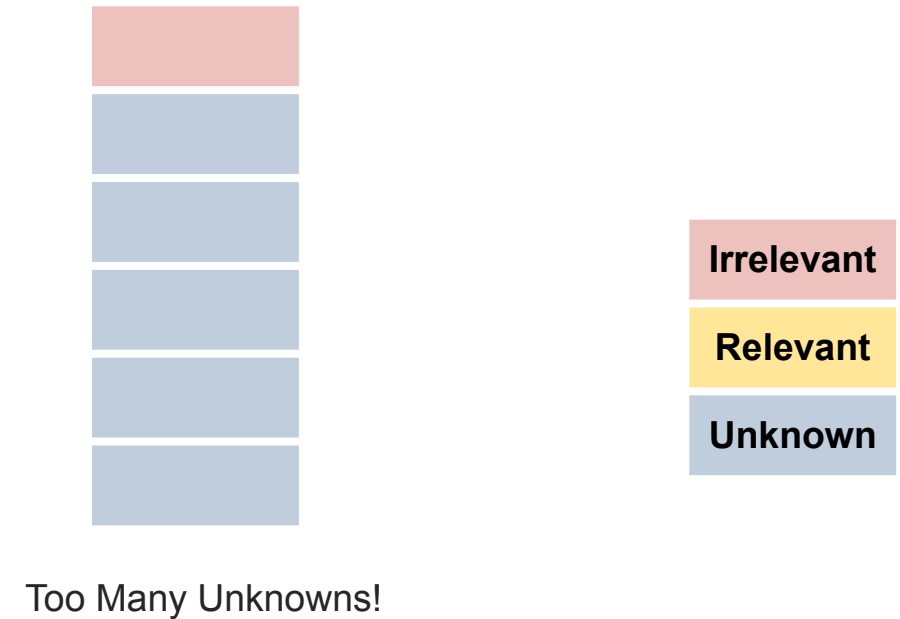
**Irrelevant**

**Relevant**

**Unknown**

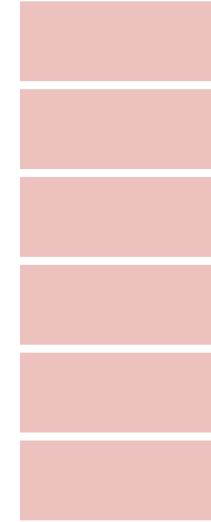
# Over-approximations For Explainability

- Use over-approximations for scalability?



# Over-approximations For Explainability

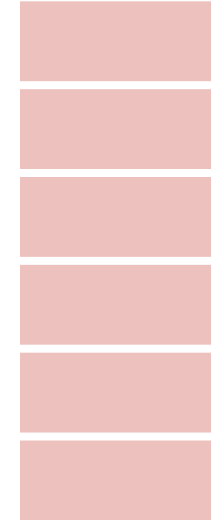
- Use over-approximations for scalability?
- Use robust models?



Everything Irrelevant!  
Explanation is empty

# Over-approximations For Explainability

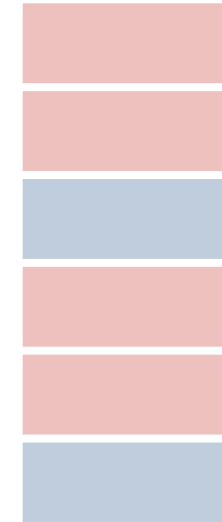
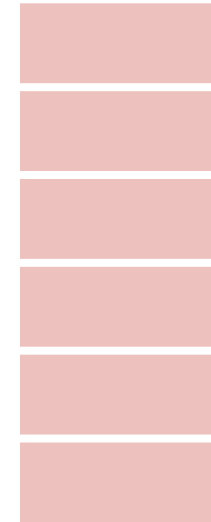
- Use over-approximations for scalability?
- Use robust models?



Everything Irrelevant!  
Explanation is empty

# Over-approximations For Explainability

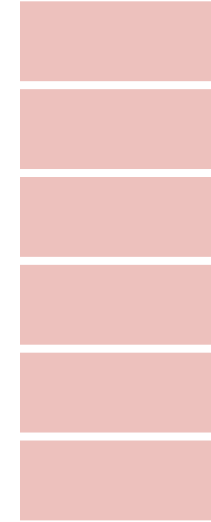
- Use over-approximations for scalability?
- Use robust models?



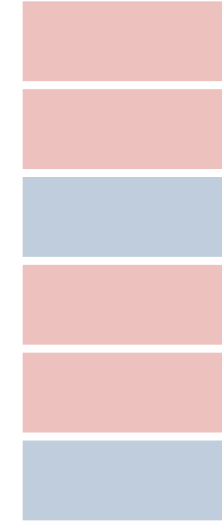
Everything Irrelevant!  
Explanation is empty

# Over-approximations For Explainability

- Use over-approximations for scalability?
- Use robust models?



Everything Irrelevant!  
Explanation is empty



Unknowns might hide  
this possibility!

# Training For Explainability

## Our propositions:

- **Make sure the entire input is Empirically Formally Explainable (EFX)**
  1. there exist a successful perturbation (empirically estimated)
  2. there is at least one irrelevant feature (provably assessed with over-approximations)
- **Use robust training with lowered robustness goal to prevent full irrelevancy**

# Lowering the robustness goal

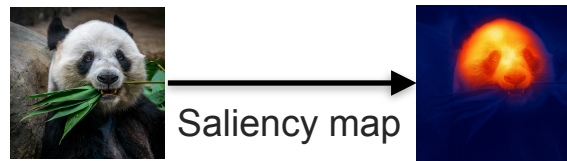




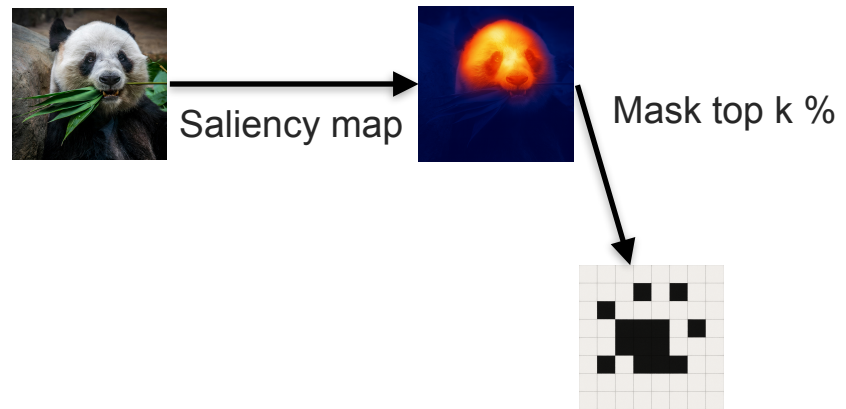
# Lowering the robustness goal



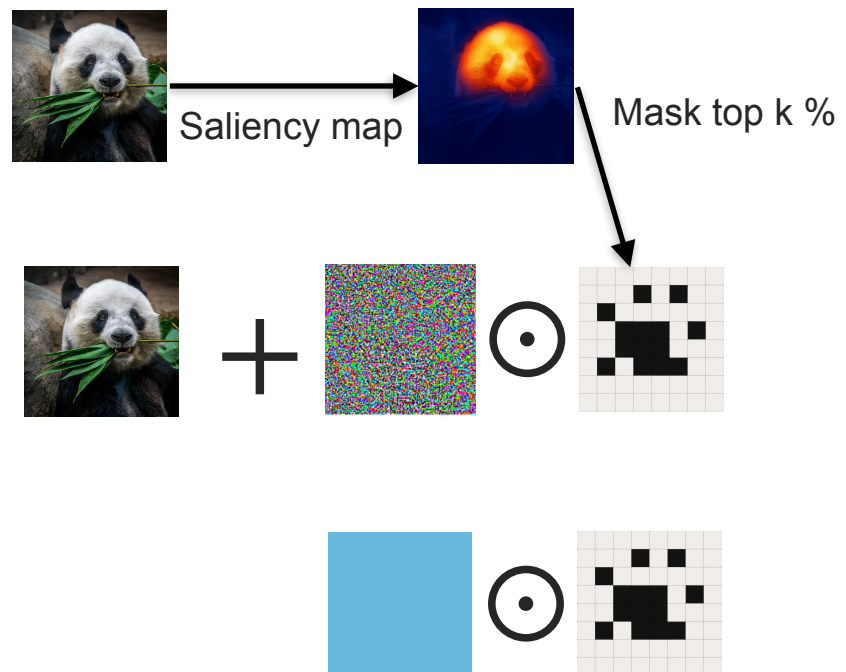
# Lowering the robustness goal



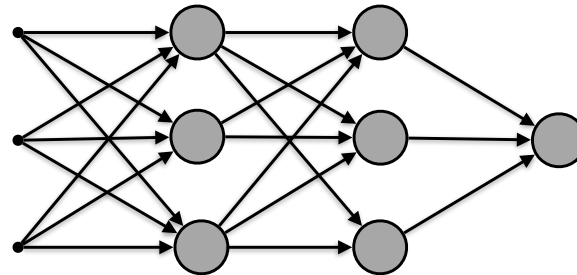
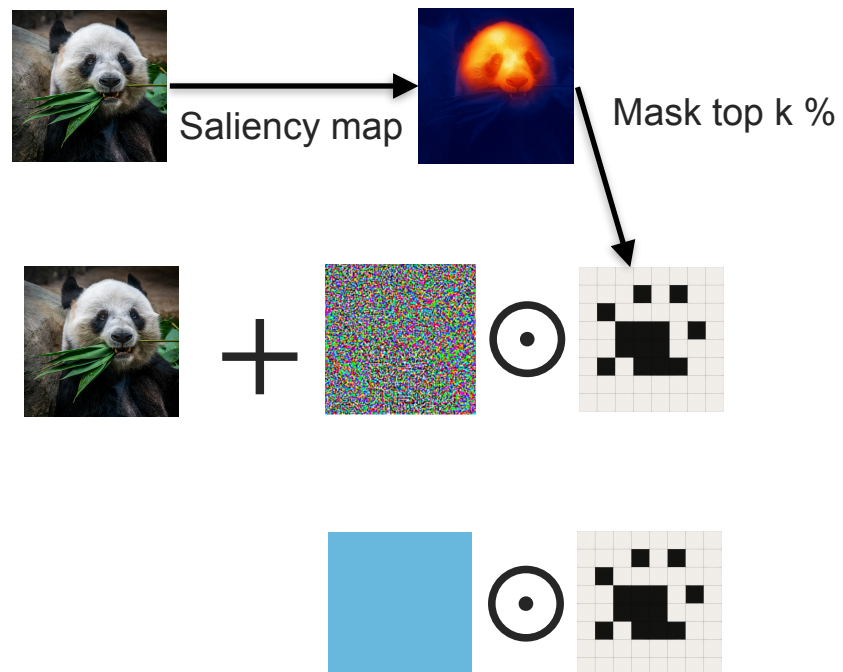
# Lowering the robustness goal



# Lowering the robustness goal



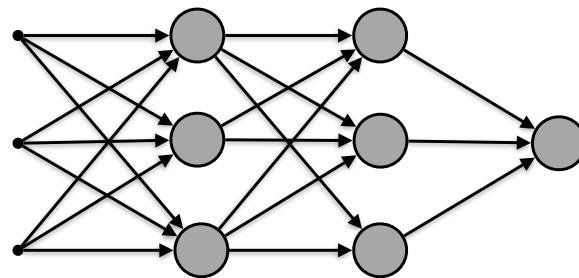
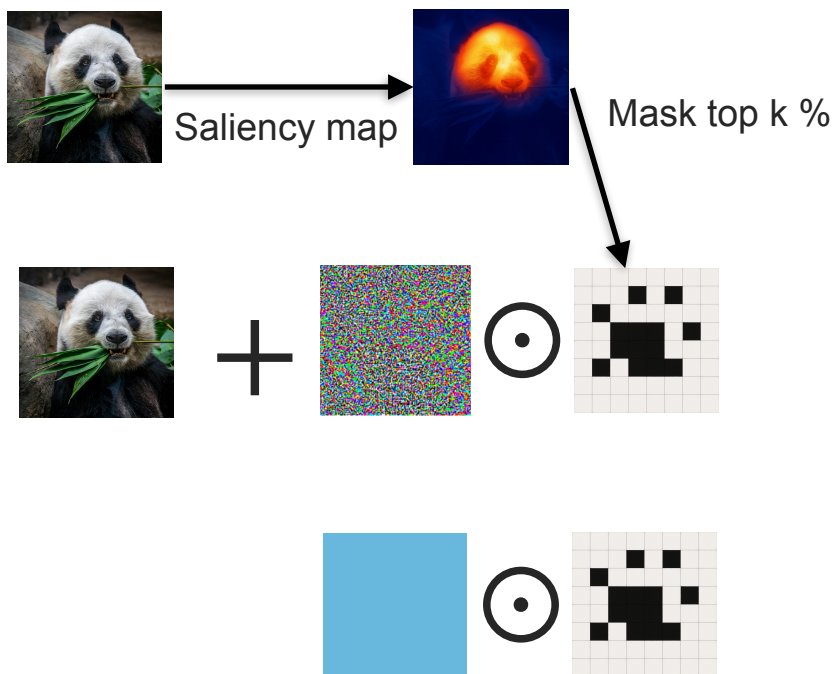
# Lowering the robustness goal



$$\alpha \cdot \text{[Diagram of a point in a region]} + (1 - \alpha) \cdot \text{[Diagram of a point in a region]}$$

# Lowering the robustness goal

- Adversarial training: lower the perturbation radius
- Expressive Losses: lower the coefficient associated with the IBP loss
- Train to be robust on a subset of inputs (Feature Subset Certified Training)



$$\alpha \cdot \text{[Diagram of a point in a region]} + (1 - \alpha) \cdot \text{[Diagram of a point in a region]}$$

The equation shows a weighted sum of two diagrams. The first diagram, labeled  $\alpha$ , shows a point (yellow star) in a region (blue shape) within a square. The second diagram, labeled  $(1 - \alpha)$ , shows a point (red cross) in a region (blue shape) within a square. A red line is drawn through both diagrams, indicating a decision boundary.

# Results

TinyImageNet, CNN-7

| Perturbation | Training Method       | Accuracy (%) ↑ | Avg Irrelevant set ↑<br>(# features: 4096) | # EFX(AA) ↑<br>(samples: 1000) |
|--------------|-----------------------|----------------|--------------------------------------------|--------------------------------|
| 4 / 255      | Standard              |                |                                            |                                |
|              | Adversarial (1 / 255) |                |                                            |                                |
|              | CT + Adv (0.5 / 255)  |                |                                            |                                |
|              | FS CT + Adv (2 / 255) |                |                                            |                                |
| 8 / 255      | Standard              |                |                                            |                                |
|              | Adversarial (4 / 255) |                |                                            |                                |
|              | CT + Adv (1 / 255)    |                |                                            |                                |
|              | FS CT + Adv (4 / 255) |                |                                            |                                |

# Results

TinyImageNet, CNN-7

| Perturbation | Training Method       | Accuracy (%) ↑ | Avg Irrelevant set ↑<br>(# features: 4096) | # EFX(AA) ↑<br>(samples: 1000) |
|--------------|-----------------------|----------------|--------------------------------------------|--------------------------------|
| 4 / 255      | Standard              | 54.80          | 23                                         | 990                            |
|              | Adversarial (1 / 255) | <b>52.61</b>   | 340                                        | 911                            |
|              | CT + AT (0.5 / 255)   | 44.79          | <b>804</b>                                 | 947                            |
|              | FS CT + AT (2 / 255)  | 48.60          | 506                                        | <b>996</b>                     |
| 8 / 255      | Standard              | 54.80          | 6                                          | 887                            |
|              | Adversarial (4 / 255) | 41.41          | 263                                        | 925                            |
|              | CT + AT (1 / 255)     | 42.36          | 465                                        | 988                            |
|              | FS CT + AT (4 / 255)  | <b>44.49</b>   | <b>684</b>                                 | <b>998</b>                     |





# 3. Conclusion

# Contributions

# Contributions

- 1. ReCIPH: a heuristic to improve verification relying on partitioning and over-approximations (Workshop Poster WFVML 2022)**

# Contributions

- 1. ReCIPH: a heuristic to improve verification relying on partitioning and over-approximations (Workshop Poster WFVML 2022)**
- 2. Using Certified Training for Empirical Robustness (Journal TMLR 2025)**

# Contributions

- 1. ReCIPH: a heuristic to improve verification relying on partitioning and over-approximations (Workshop Poster WFVML 2022)**
- 2. Using Certified Training for Empirical Robustness (Journal TMLR 2025)**
- 3. Using Certified Training to scale Formal Explainability (submission in preparation)**

# Future Work and important challenges



**Verification**

**Robust training**

**Explainability**

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency

## Robust training

- Ever-going problem of efficiency

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning  
(beyond bisection)**

## Robust training

- Ever-going problem of efficiency

## Explainability

- Ever-going problem of efficiency



# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures

## Robust training

- Ever-going problem of efficiency

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency
- **True end to end learning of which input are relevant jointly with the robust training for explainability**

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency
- **True end to end learning of which input are relevant jointly with the robust training for explainability**
- **Explore other notions than input level explanations -> in the latent space**



# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

- Ever-going problem of efficiency
- **True end to end learning of which input are relevant jointly with the robust training for explainability**
- **Explore other notions than input level explanations -> in the latent space**

# Future Work and important challenges

## Verification

- Ever-going problem of efficiency
- **Automated input partitioning (beyond bisection)**
- Verification of more complex architectures
- Verification of more complex properties

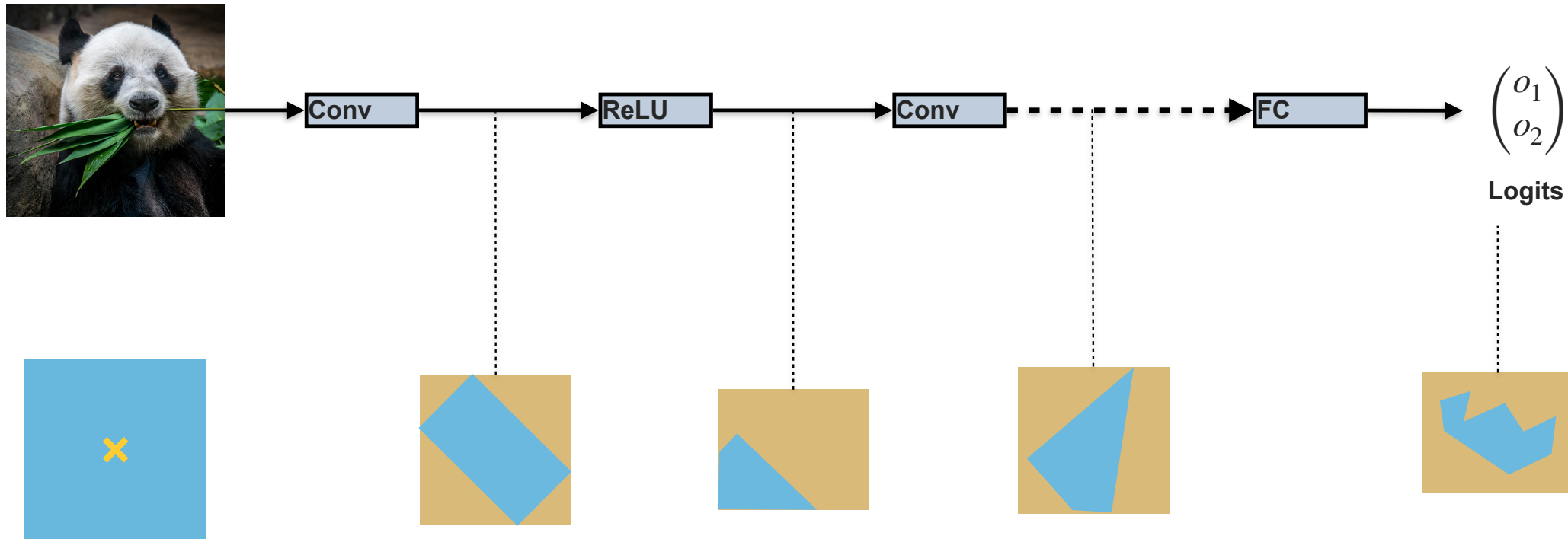
## Robust training

- Ever-going problem of efficiency
- **Re-visit tighter approximation for training?**
- **Applications beyond robustness and explainability: contrastive learning, robust segmentation...**

## Explainability

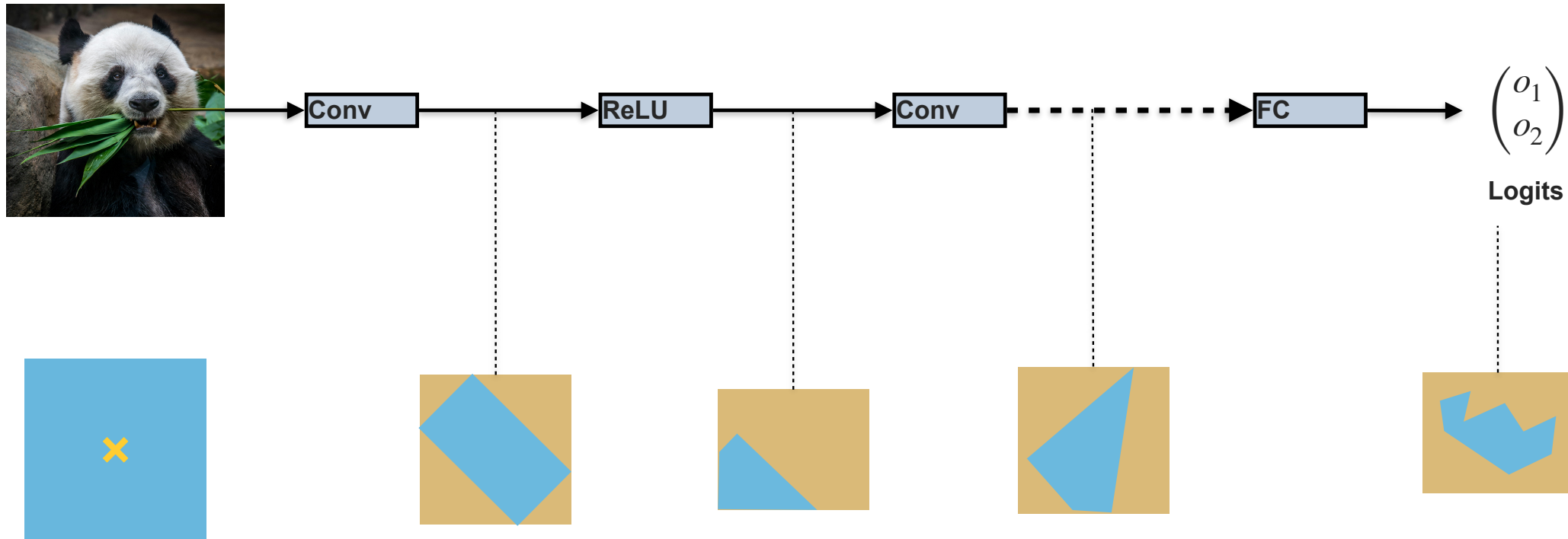
- Ever-going problem of efficiency
- **True end to end learning of which input are relevant jointly with the robust training for explainability**
- **Explore other notions than input level explanations -> in the latent space**

# If you must only remember one thing



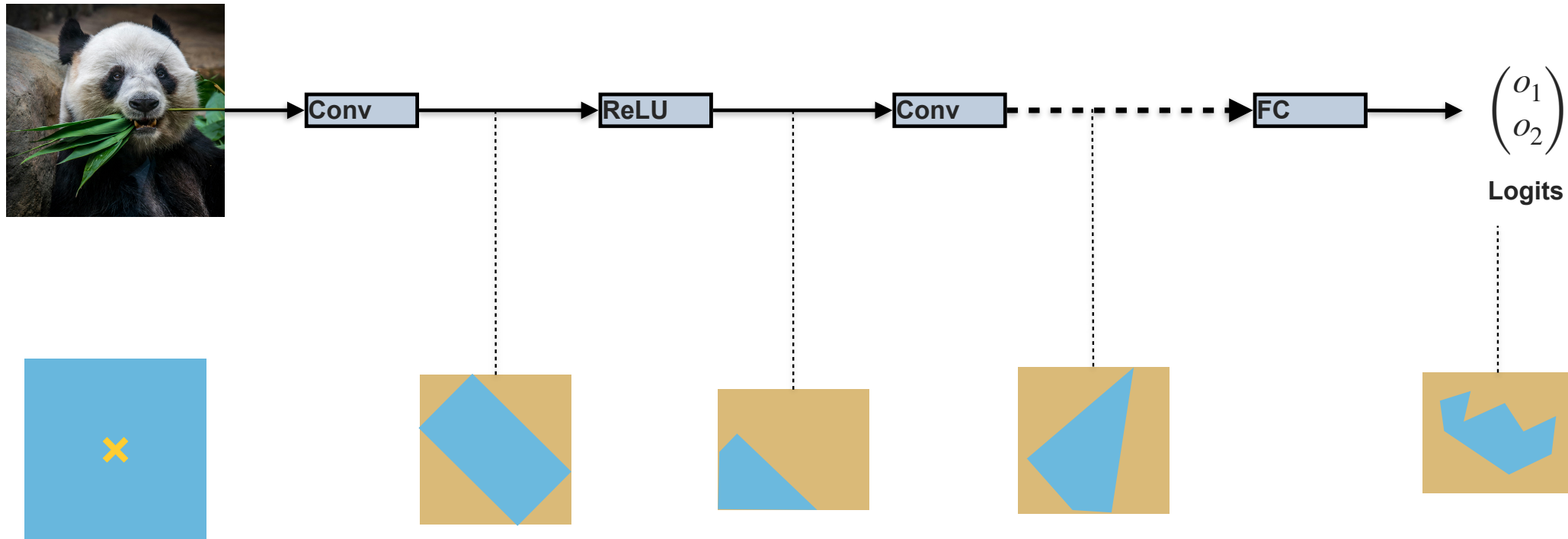
- Over-approximations: versatile tool that can be leveraged in multiple ways to bring trust to AI systems

# If you must only remember one thing



- Over-approximations: versatile tool that can be leveraged in multiple ways to bring trust to AI systems
- Trust should be an integral part of the design / training choices!

# If you must only remember one thing



- Over-approximations: versatile tool that can be leveraged in multiple ways to bring trust to AI systems
- Trust should be an integral part of the design / training choices!

**Questions ?**

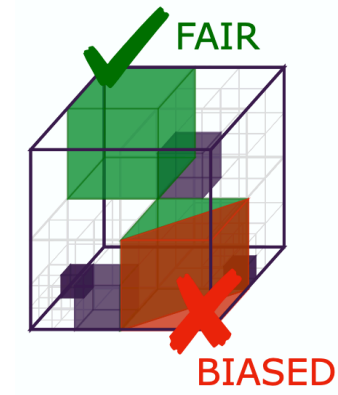
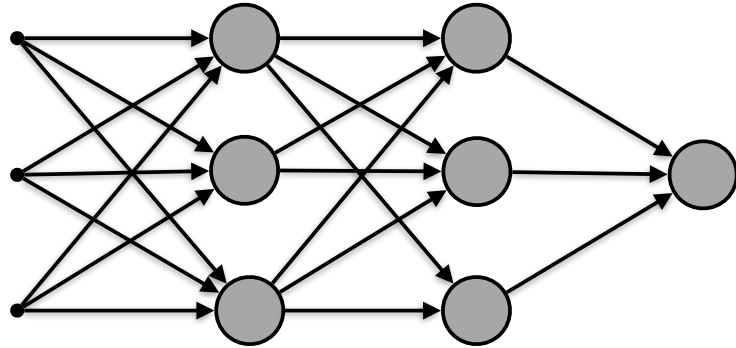


# 4 ■ Additional content

# LIBRA (Fairness)

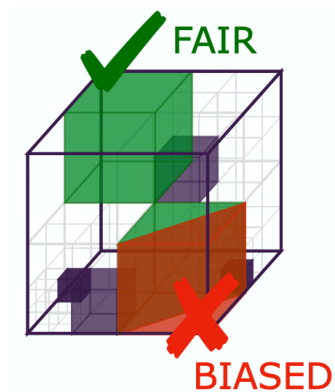
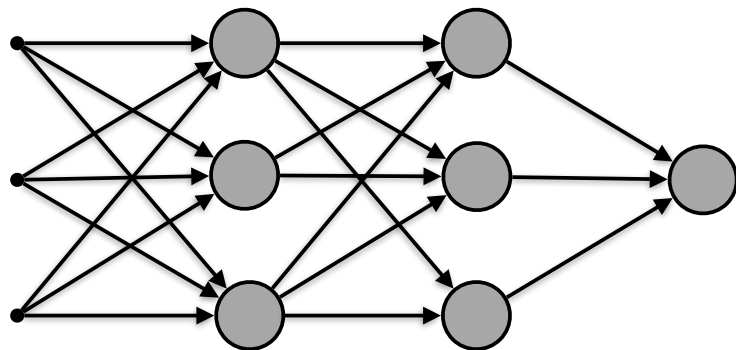


# LIBRA (Fairness)





# LIBRA (Fairness)



|   |   | Coverage (%) |        | Execution Time (s) |        |
|---|---|--------------|--------|--------------------|--------|
| L | U | Width        | ReCIPH | Width              | ReCIPH |
| 1 | 2 | 68.8%        | 87.5 % | 0.26               | 0.11   |
|   | 6 | 68.8%        | 87.5 % | 0.51               | 0.20   |
| 4 | 2 | 100 %        | 100 %  | 2.60               | 2.10   |
|   | 6 | 100 %        | 100 %  | 2.65               | 2.10   |

# VNN Comp 2024



PyRAT with ReCIPH input partitioning is used for 6 out of 12 benchmarks

| Benchmark        | PyRAT Rank | Notes                                                                                                         |
|------------------|------------|---------------------------------------------------------------------------------------------------------------|
| cGAN             | 1 / 6      | First rank position are shared with at least one other tool.<br>Global ranking was 2 out of 8 submitted tools |
| LinearizeNN      | 1 / 4      |                                                                                                               |
| Collins RUL CNN  | 5 / 7      |                                                                                                               |
| TLL Verify Bench | 1 / 8      |                                                                                                               |
| ACAS Xu          | 3 / 8      |                                                                                                               |
| Dist Shift       | 1 / 5      |                                                                                                               |

# VNN Comp 2024



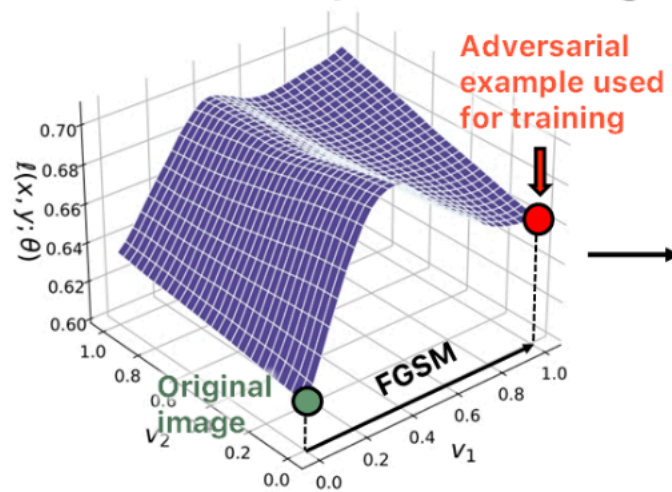
PyRAT with ReCIPH input partitioning is used for 6 out of 12 benchmarks

| Benchmark        | PyRAT Rank | Notes                                                                                                         |
|------------------|------------|---------------------------------------------------------------------------------------------------------------|
| cGAN             | 1 / 6      | First rank position are shared with at least one other tool.<br>Global ranking was 2 out of 8 submitted tools |
| LinearizeNN      | 1 / 4      |                                                                                                               |
| Collins RUL CNN  | 5 / 7      |                                                                                                               |
| TLL Verify Bench | 1 / 8      |                                                                                                               |
| ACAS Xu          | 3 / 8      |                                                                                                               |
| Dist Shift       | 1 / 5      |                                                                                                               |

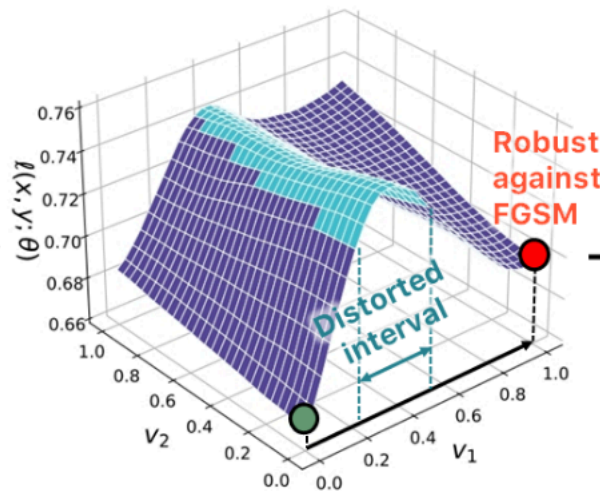
Verification is still a hard challenge!

# Local Linearity and Catastrophic Overfitting

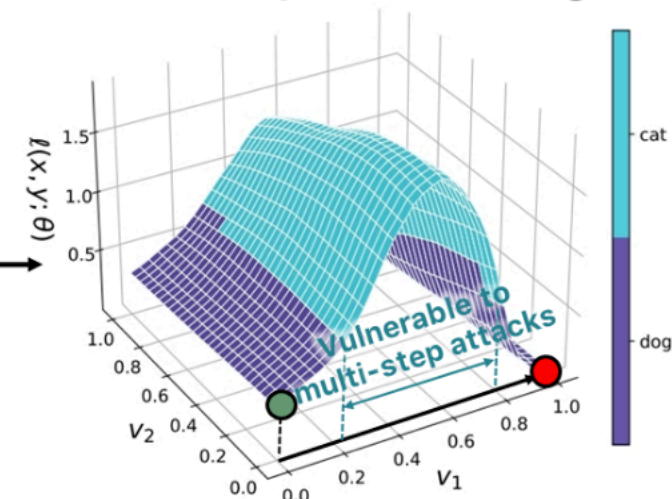
Before Catastrophic Overfitting



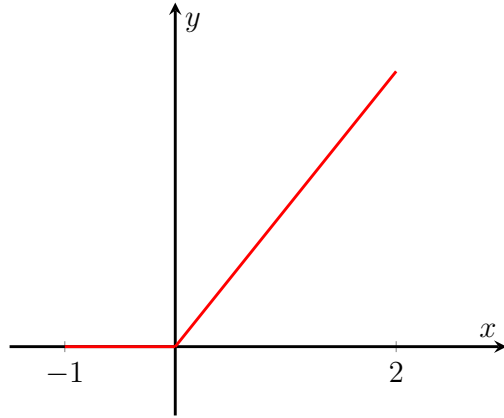
Decision Boundary Distortion



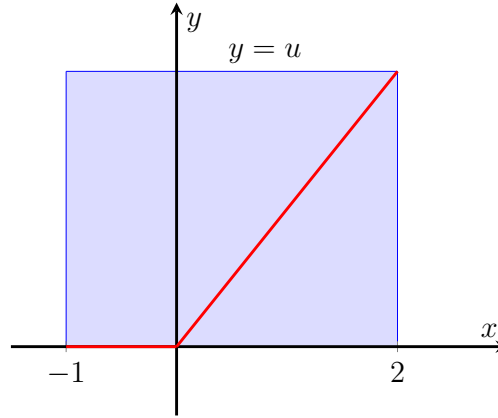
Catastrophic Overfitting



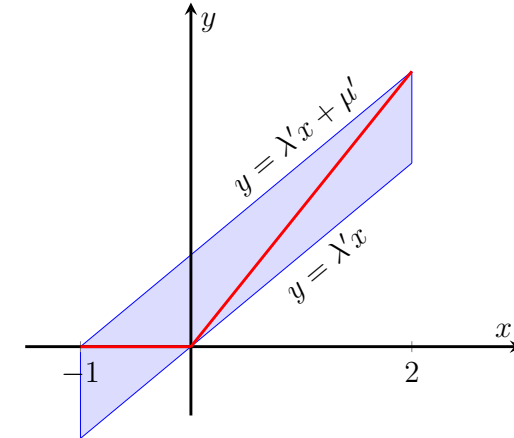
## Source of approximations: handling of the non-linearities (activation functions)



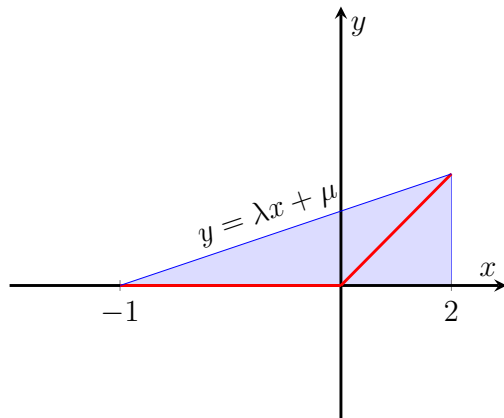
(a) Exact ReLU



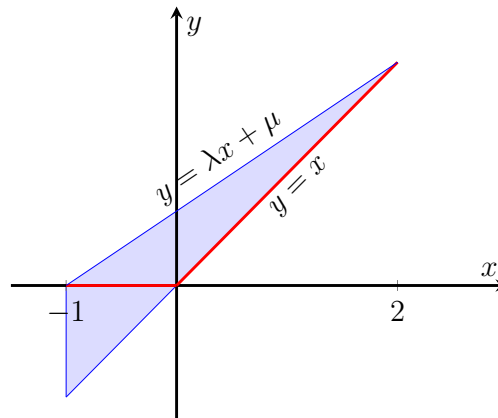
(b) Box approximation



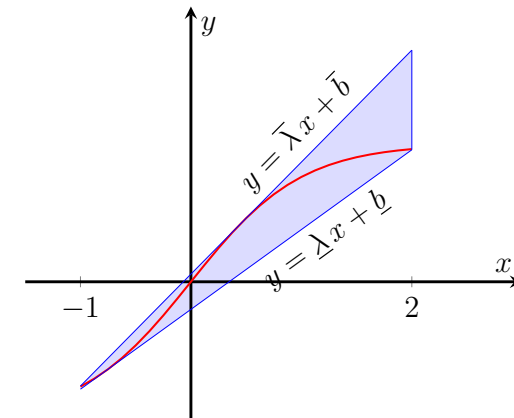
(c) Parallel Linear Approximation



(d) DeepPoly/CROWN (case  $u \leq -l$ )

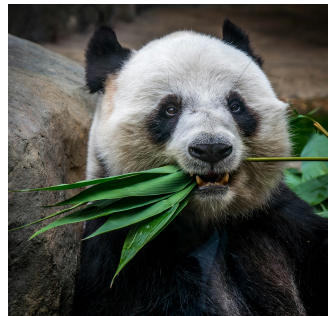


(e) DeepPoly/CROWN (case  $u > -l$ )

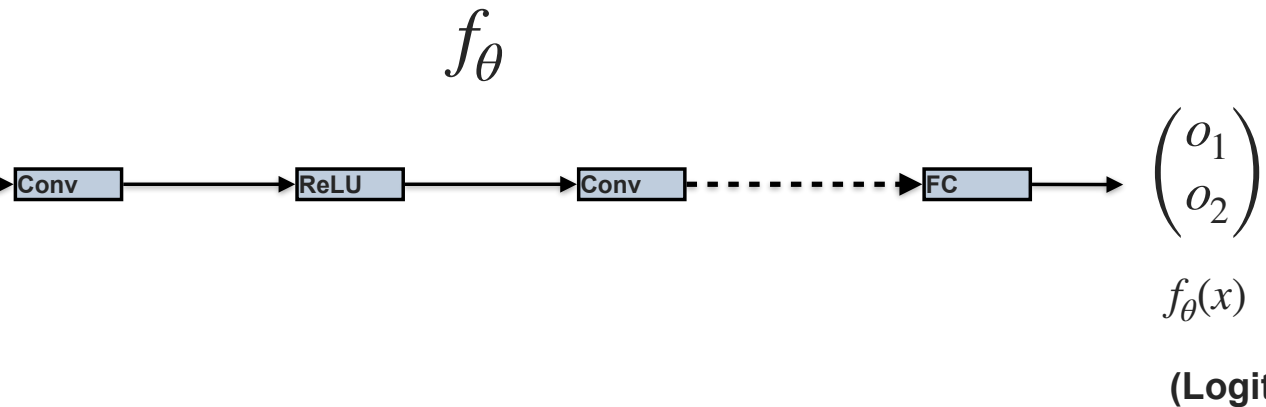


(f) Parallel relaxation of tanh

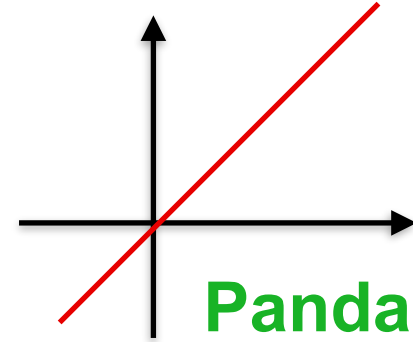
# Supervised Training (Classification)



$x$



Not Panda



$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_\theta(x_i), y_i)$$

$$\theta' \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(f_\theta(x), y)$$

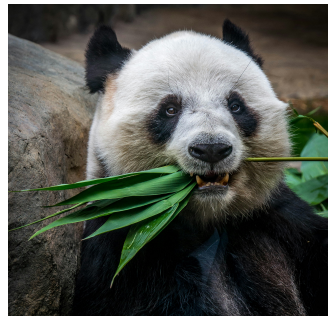
- Goal: empirical risk minimization. Expectation of the loss function over the dataset

- Gradient Descent: Update of the parameters in the opposite direction of the gradient to minimize the error.

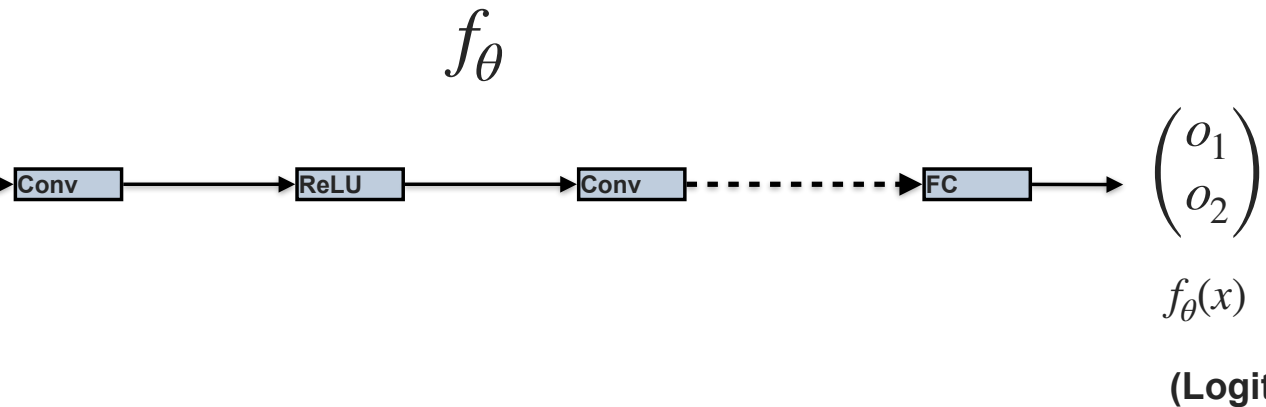
Stochastic Gradient Descent (SGD): compute the gradients and do the update over small batch of inputs randomly drawn from the dataset.

One epoch = one full pass over the dataset

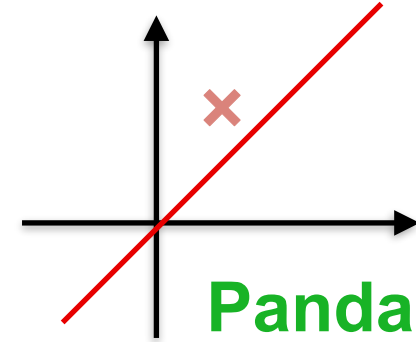
# Supervised Training (Classification)



$x$



Not Panda

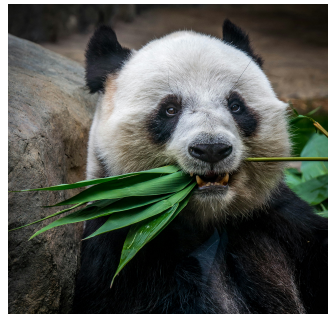


$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_\theta(x_i), y_i)$$

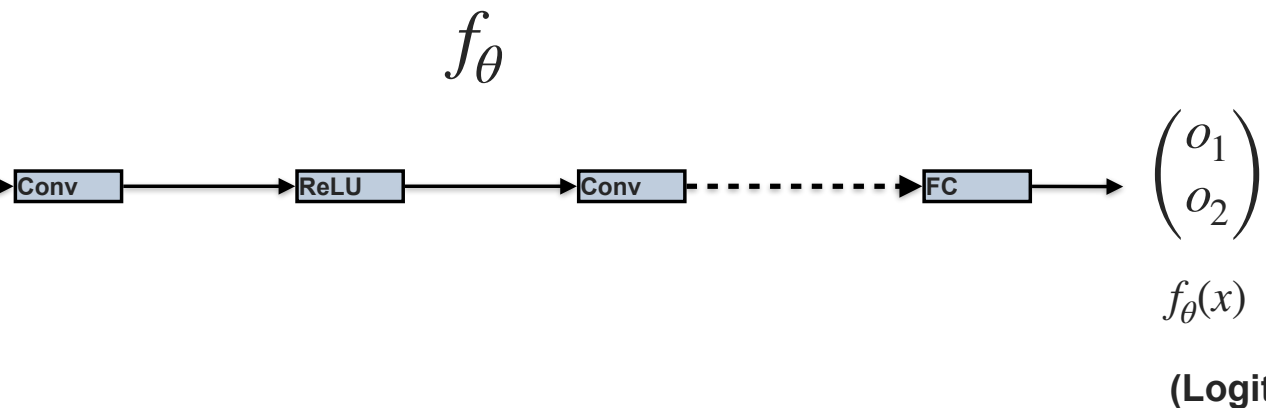
$$\theta' \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(f_\theta(x), y)$$

- Goal: empirical risk minimization. Expectation of the loss function over the dataset
- Gradient Descent: Update of the parameters in the opposite direction of the gradient to minimize the error.  
Stochastic Gradient Descent (SGD): compute the gradients and do the update over small batch of inputs randomly drawn from the dataset.  
One epoch = one full pass over the dataset

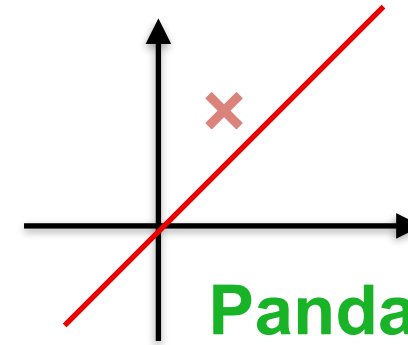
# Supervised Training (Classification)



$x$



Not Panda



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

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f_\theta(x_i), y_i)$$

$$\theta' \leftarrow \theta - \eta \nabla_{\theta} \mathcal{L}(f_\theta(x), y)$$

- Estimation of the error of the model (using a surrogate loss function)
- Goal: empirical risk minimization. Expectation of the loss function over the dataset
- Gradient Descent: Update of the parameters in the opposite direction of the gradient to minimize the error.  
Stochastic Gradient Descent (SGD): compute the gradients and do the update over small batch of inputs randomly drawn from the dataset.  
One epoch = one full pass over the dataset



# Robust Learning, more formally

- [Madry 2018] Goal of robust training: 
$$\min_{\theta} \sum_{i=1}^n \max_{x' \in B(x^{(i)}, \varepsilon)} \mathcal{L} \left( f_{\theta}(x'), y^{(i)} \right)$$

$\mathcal{L}$  = typically the cross-entropy

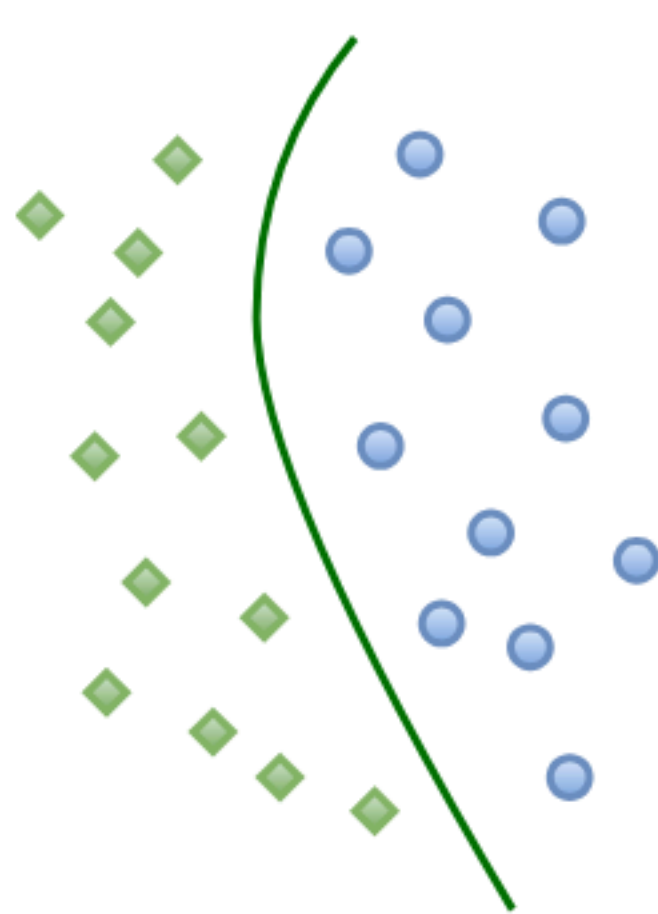
$$\mathcal{L}_{\text{CE}}(y, \hat{y}) = - \sum_{k=1}^K y_k \log(\hat{y}_k)$$

- Adversarial training: under-approximate solution of inner-maximization using some attack algorithm

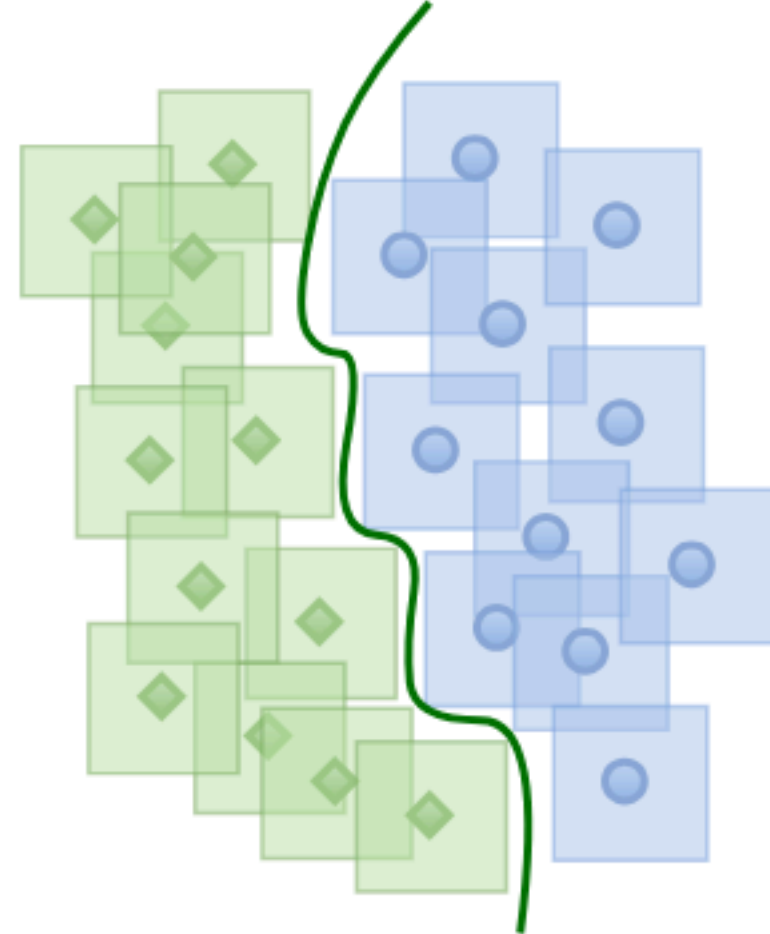
$$\mathcal{L}_{adv} := \mathcal{L}(f_{\theta}(x_{adv}), y) \quad \text{with } x_{adv} \text{ an adversarial perturbation}$$

- Certified training: over-approximate solution using **differentiable** bounding algorithm (in practice IBP)

$$\mathcal{L}_{IBP} := \mathcal{L}(\underline{f_{\theta}(x)}[y] \mathbf{1} - \overline{f_{\theta}(x)}) \quad \text{with } \overline{f_{\theta}(x)} \text{ and } \underline{f_{\theta}(x)} \text{ the output bounds computed with IBP}$$



Non-robust classification

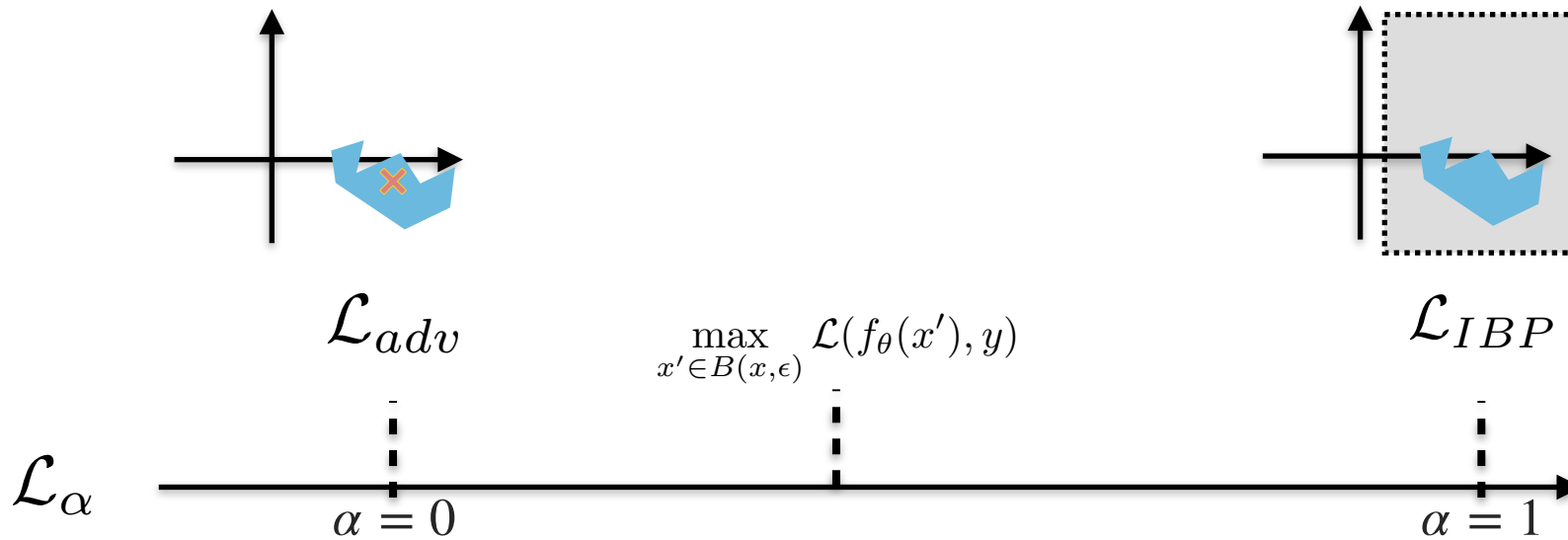


Robust classification

# Interval Bound Propagation



# Expressive Losses



Different ways of combining have been proposed:

- MTL-IBP
- CC-IBP
- Exp-IBP

# Expressive Losses

$$\mathcal{L}_{\alpha}^{MTL-IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y) := (1 - \alpha) \mathcal{L}_{adv}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y) + \alpha \mathcal{L}_{IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y),$$

$$\mathcal{L}_{\alpha}^{Exp-IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y) := \mathcal{L}_{adv}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y)^{(1-\alpha)} \mathcal{L}_{IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y)^{\alpha}.$$

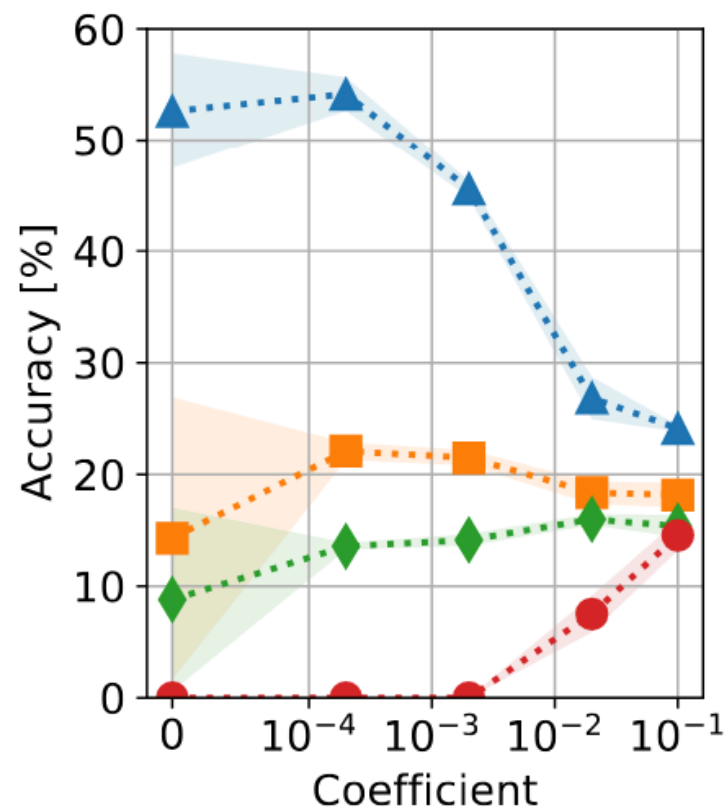
$$\mathcal{L}_{\alpha}^{SABR}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y) := \mathcal{L}_{IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\alpha\epsilon}(\mathbf{x}_{\alpha}); y),$$

$$\mathcal{L}_{\alpha}^{CC-IBP}(\mathbf{f}_{\theta}, \mathcal{B}_{\epsilon}(\mathbf{x}); y) := \mathcal{L} \left( - \left[ (1 - \alpha) \mathbf{z}_{\mathbf{f}_{\theta}}(\mathbf{x}_{adv, \mathbb{A}}, y) + \alpha \mathbf{l}_{\mathbf{f}_{\theta}}^{\mathcal{B}_{\epsilon}(\mathbf{x}), y} \right], y \right),$$

# Trade Offs of CT + AT in CO setting

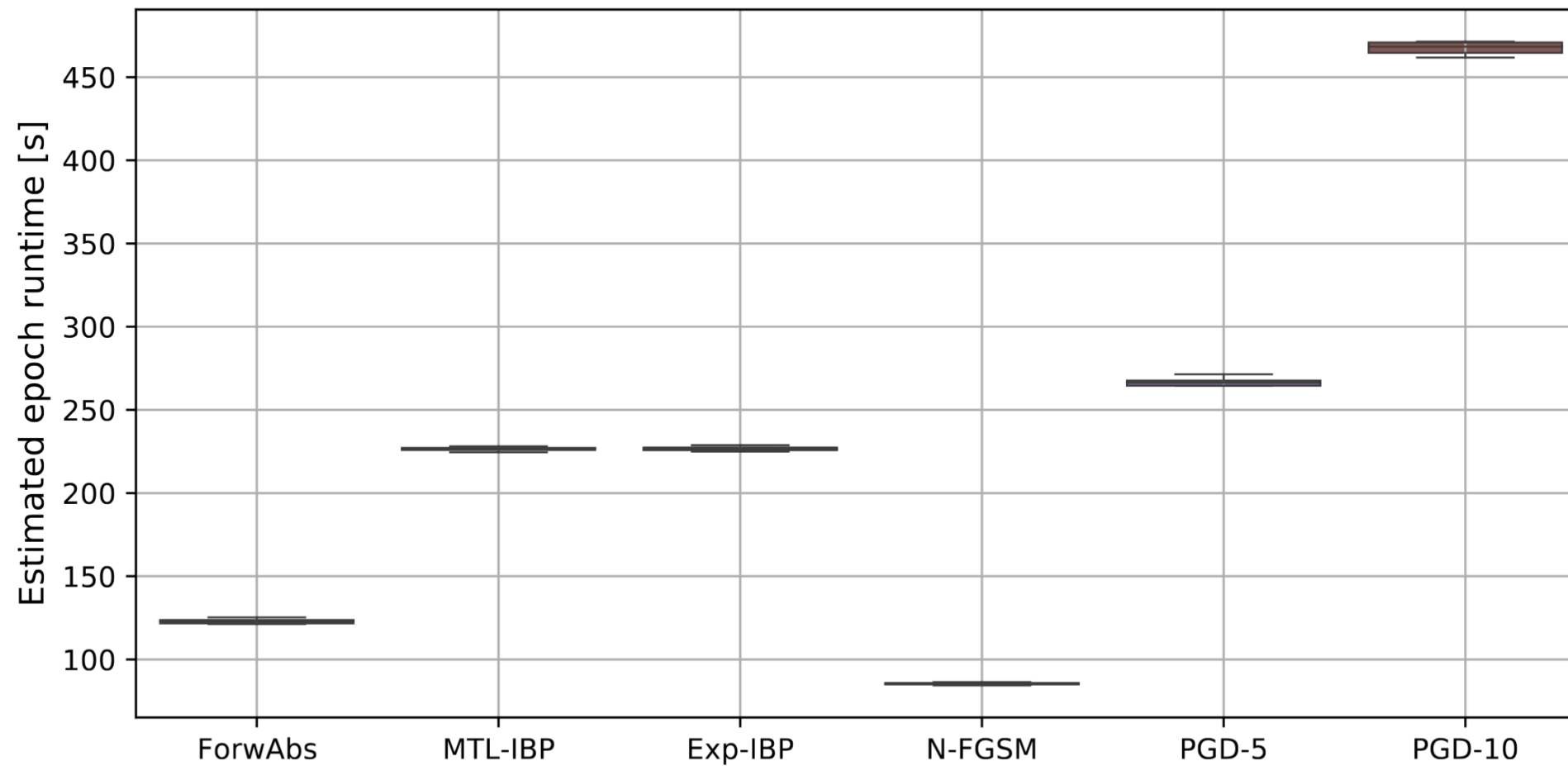


# Sensitivity of CT + AT to $\alpha$



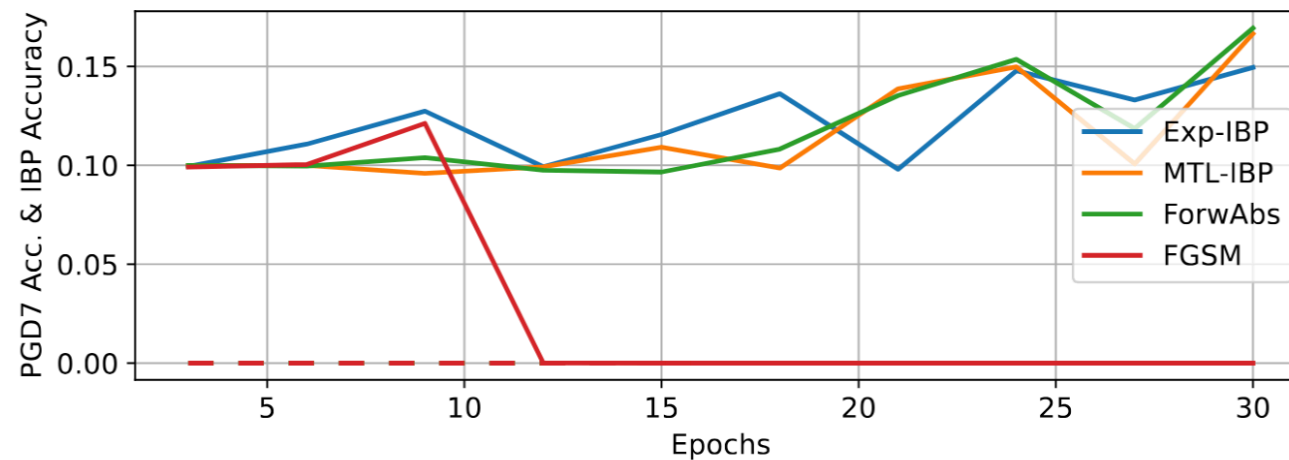
(a) Exp-IBP,  
CIFAR-10.

# Time trade offs

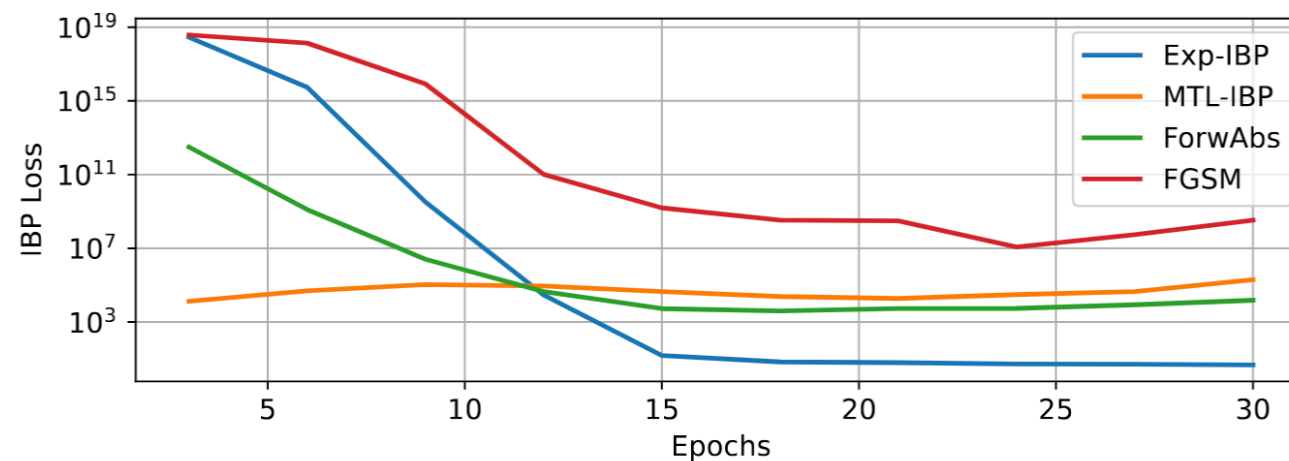




# Prevent CO in Softplus networks

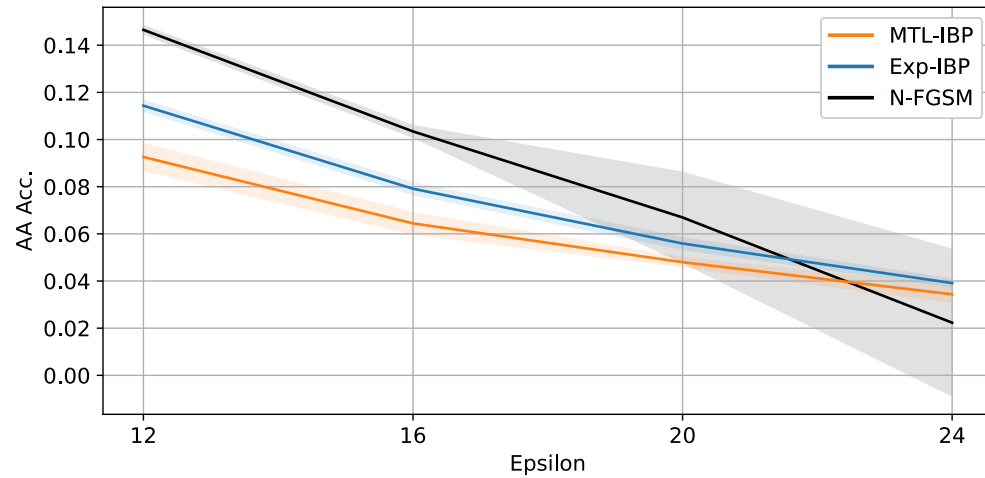


(a) PGD-7 (solid) and IBP (dashed) acc., CIFAR-10.

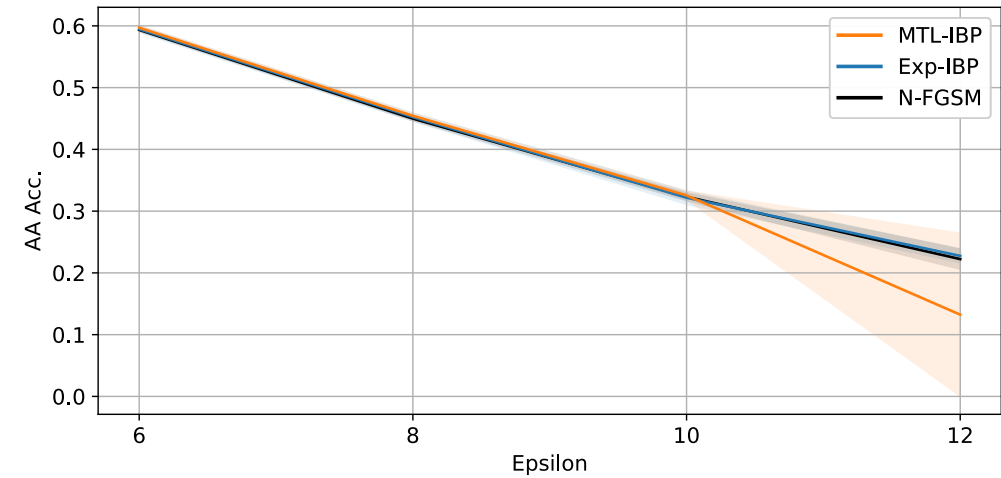


(c) IBP loss for CIFAR-10.

# Limits



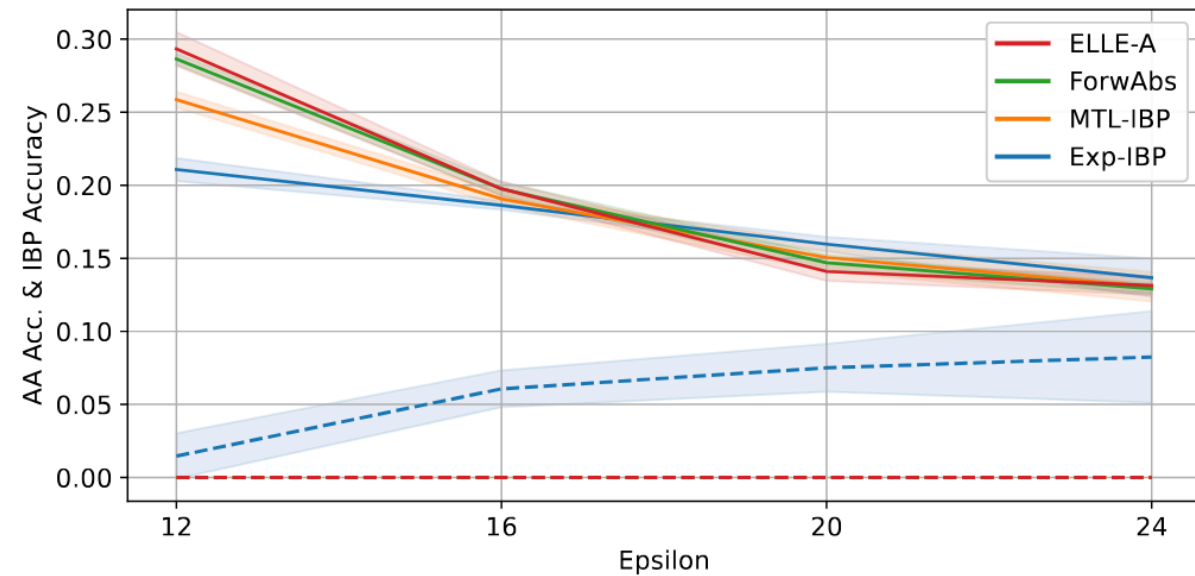
CIFAR-100



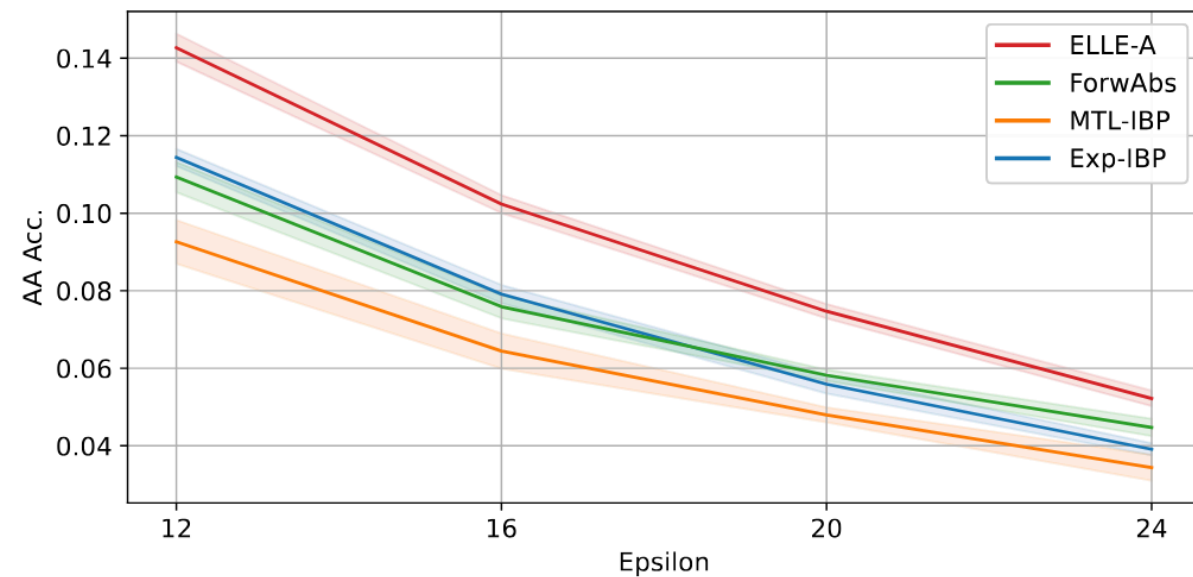
CIFAR SVHN

- On CIFAR-100: CO is mitigated, but high cost in empirical robustness (tuning adapted to the epsilon could help)
- On SVHN: MTL-IBP fails. MTL-IBP tuning is very sensitive compared to Exp-IBP, especially for deep networks.

# Comparisons with ELLE

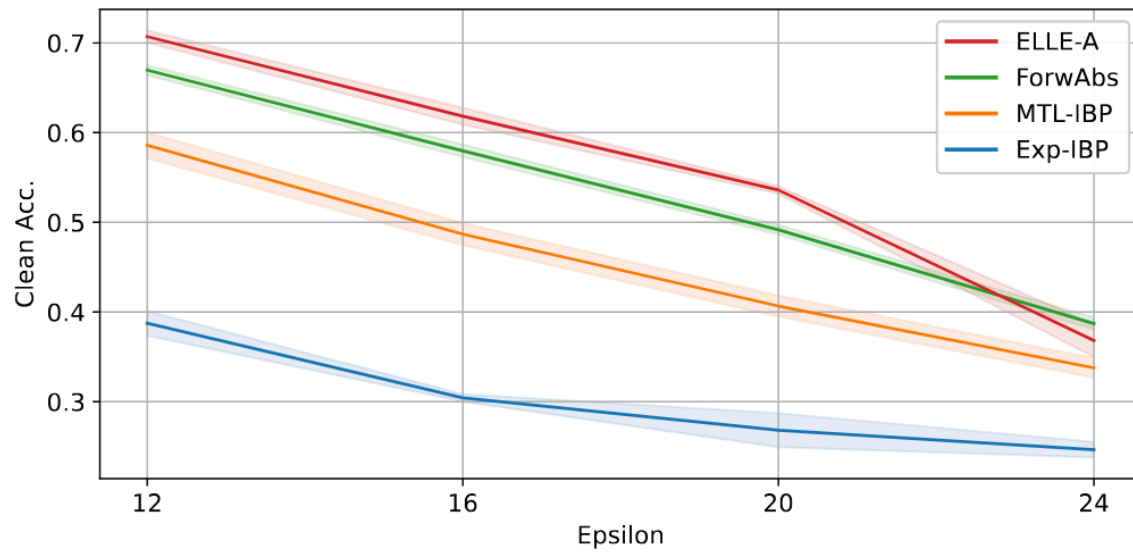


(a) CIFAR-10, AA (solid lines) and IBP accuracies (dashed).

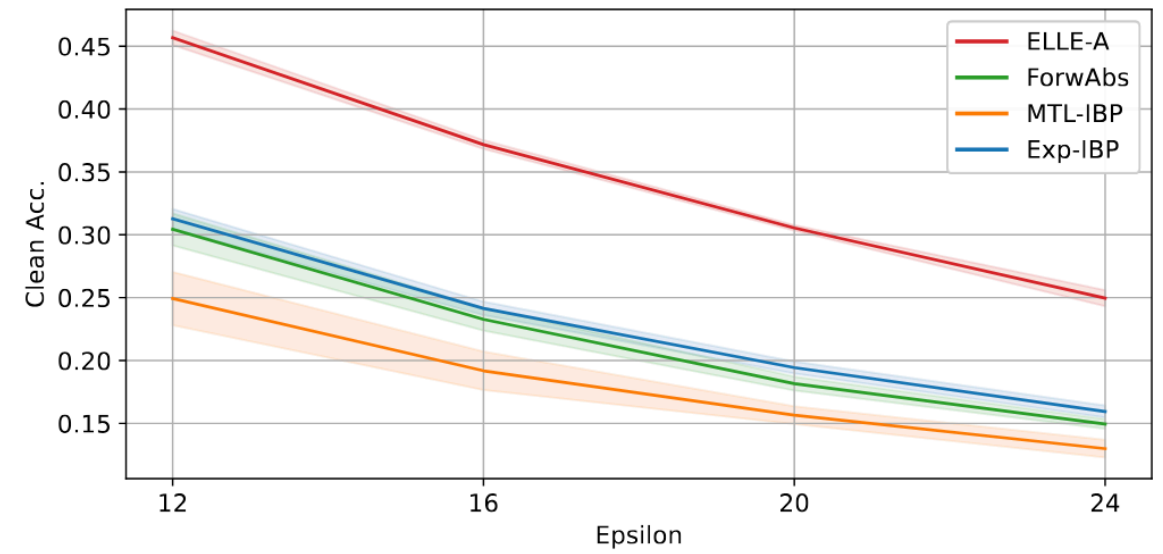


(b) CIFAR-100.

# Comparisons with ELLE



(a) CIFAR10.



(b) CIFAR-100.

# Some clean accuracies

CIFAR-10, CNN-7, 160 epochs

| $\epsilon$       | Method  | Clean acc. [%]   | IBP loss                      |
|------------------|---------|------------------|-------------------------------|
| $\frac{8}{255}$  | EXP-IBP | $76.30 \pm 0.43$ | $(4.90 \pm 0.55) \times 10^4$ |
|                  | IBP     | $39.37 \pm 1.51$ | $1.92 \pm 0.01$               |
|                  | FORWABS | $74.26 \pm 0.28$ | $(2.05 \pm 0.11) \times 10^4$ |
|                  | PGD-10  | $77.07 \pm 0.31$ | $(9.79 \pm 0.85) \times 10^5$ |
|                  | PGD-5   | $79.81 \pm 0.38$ | $(1.72 \pm 0.11) \times 10^6$ |
|                  | N-FGSM  | $77.28 \pm 0.26$ | $(1.91 \pm 0.14) \times 10^6$ |
| $\frac{16}{255}$ | EXP-IBP | $31.30 \pm 2.13$ | $2.15 \pm 0.02$               |
|                  | IBP     | $31.65 \pm 0.55$ | $2.13$                        |
|                  | FORWABS | $46.54 \pm 0.31$ | $(7.72 \pm 0.37) \times 10^1$ |
|                  | PGD-10  | $58.44 \pm 0.24$ | $(3.57 \pm 0.33) \times 10^5$ |
|                  | PGD-5   | $64.07 \pm 0.14$ | $(1.01 \pm 0.13) \times 10^6$ |
|                  | N-FGSM  | $62.38 \pm 9.16$ | $(2.93 \pm 2.66) \times 10^6$ |

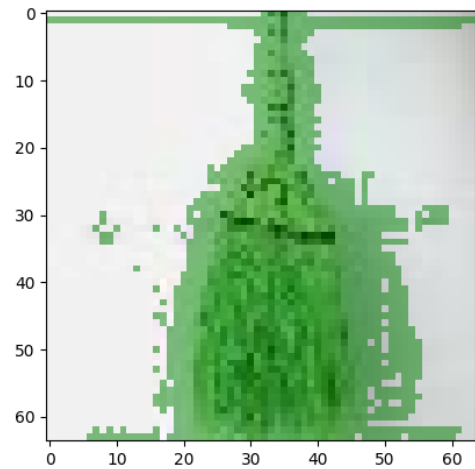
# Low Epsilon on standard network?

| $\epsilon$ -test | # $\text{EFX}_{AA}^\epsilon$ | $ \mathcal{I} $ |
|------------------|------------------------------|-----------------|
| 0.25/255         | 40/300                       | $747 \pm 234$   |
| 0.5/255          | 82/300                       | $418 \pm 160$   |
| 1/255            | 174/300                      | $210 \pm 81$    |

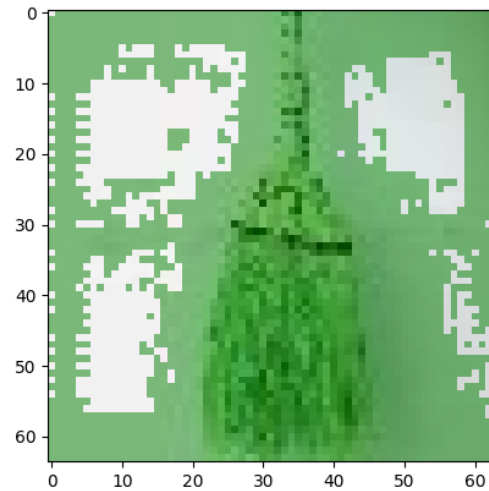
# ReCIPH as traversal order

| Dataset      | Method    | $ \mathcal{I} $ -Heuristic | $ \mathcal{I} $ -IBP Bounds   | $ \mathcal{I} $ -CROWN Coefs    | $ \mathcal{I} $ -Grad |
|--------------|-----------|----------------------------|-------------------------------|---------------------------------|-----------------------|
| CIFAR-10     | Adv       | $89 \pm 55$                | $110 \pm 72$                  | <b><math>112 \pm 74</math></b>  | $99 \pm 69$           |
|              | CC-IBP    | $178 \pm 99$               | $305 \pm 154$                 | <b><math>326 \pm 154</math></b> | $302 \pm 151$         |
|              | Standard  | $12 \pm 10$                | <b><math>37 \pm 22</math></b> | $20 \pm 11$                     | $25 \pm 15$           |
|              | FS-CC-IBP | $67 \pm 51$                | $124 \pm 96$                  | <b><math>140 \pm 93</math></b>  | $129 \pm 92$          |
| TinyImageNet | Adv       | $228 \pm 147$              | $318 \pm 218$                 | <b><math>344 \pm 228</math></b> | $260 \pm 200$         |
|              | CC-IBP    | $416 \pm 299$              | $671 \pm 475$                 | <b><math>782 \pm 498</math></b> | $723 \pm 489$         |
|              | Standard  | $21 \pm 19$                | <b><math>68 \pm 46</math></b> | $23 \pm 23$                     | $39 \pm 29$           |
|              | FS-CC-IBP | $185 \pm 144$              | $364 \pm 394$                 | <b><math>503 \pm 429</math></b> | $421 \pm 396$         |

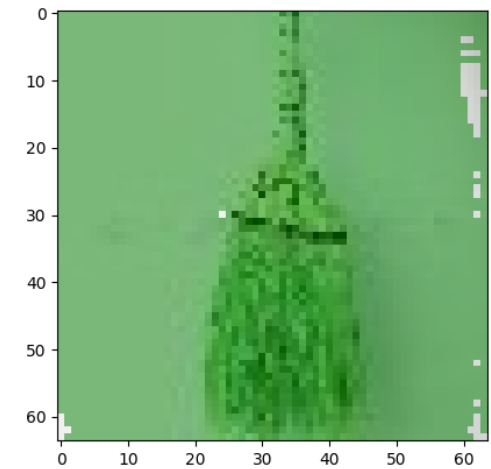
# Qualitative Results



FS+CT+AT



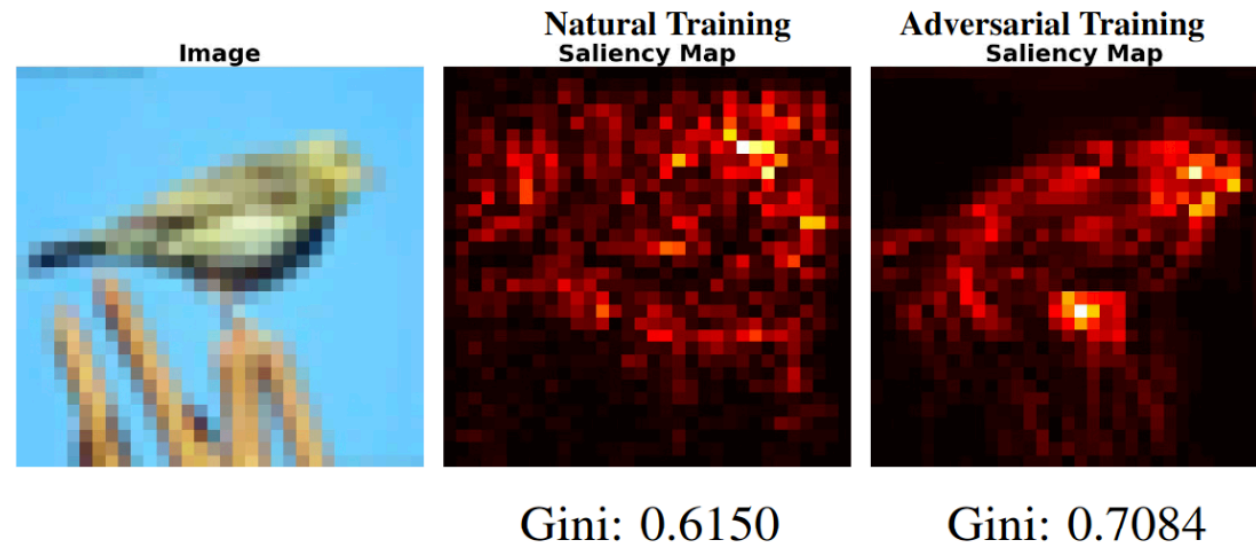
CT + AT



AT

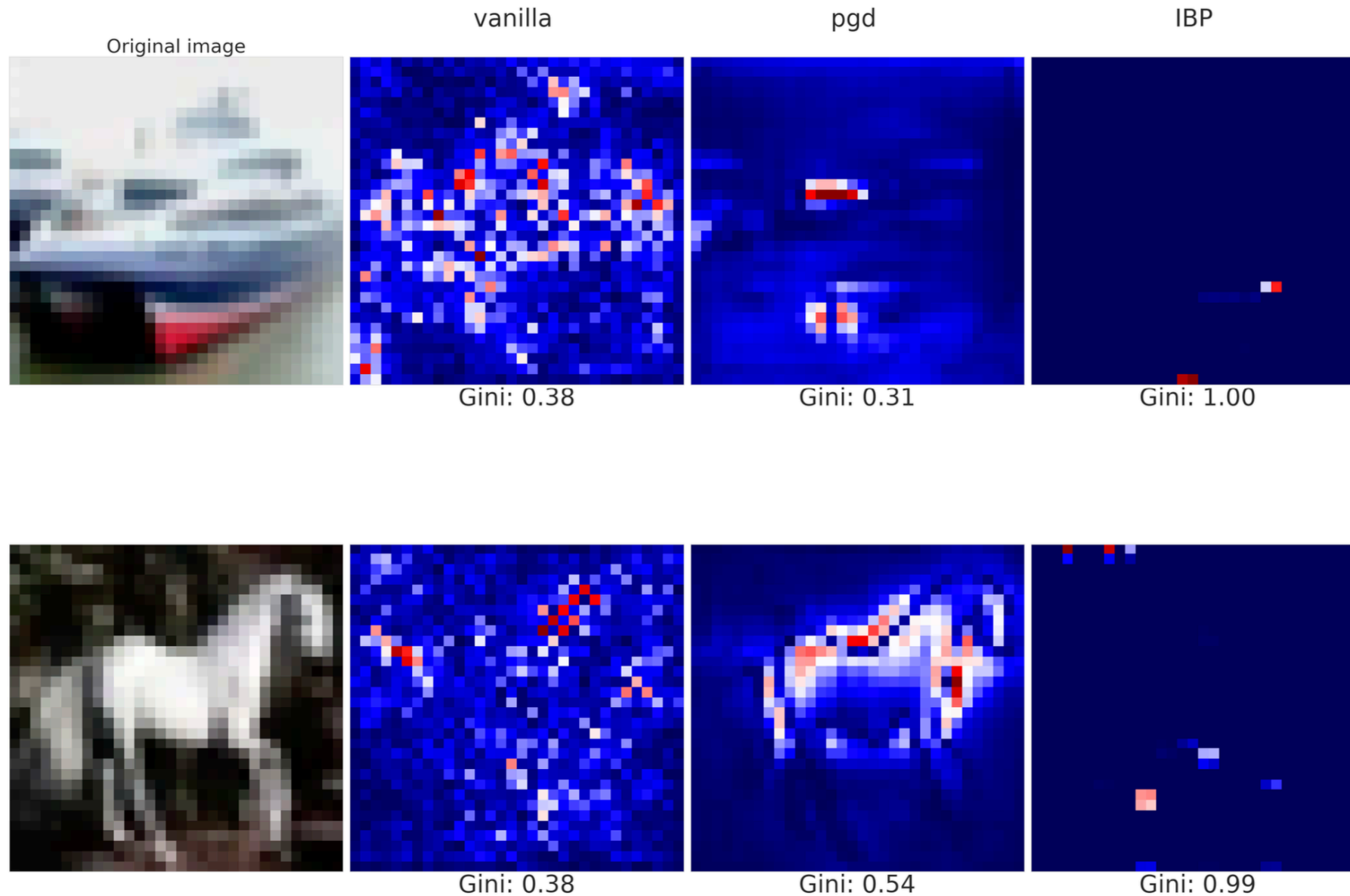


# Robustness and XAI 1

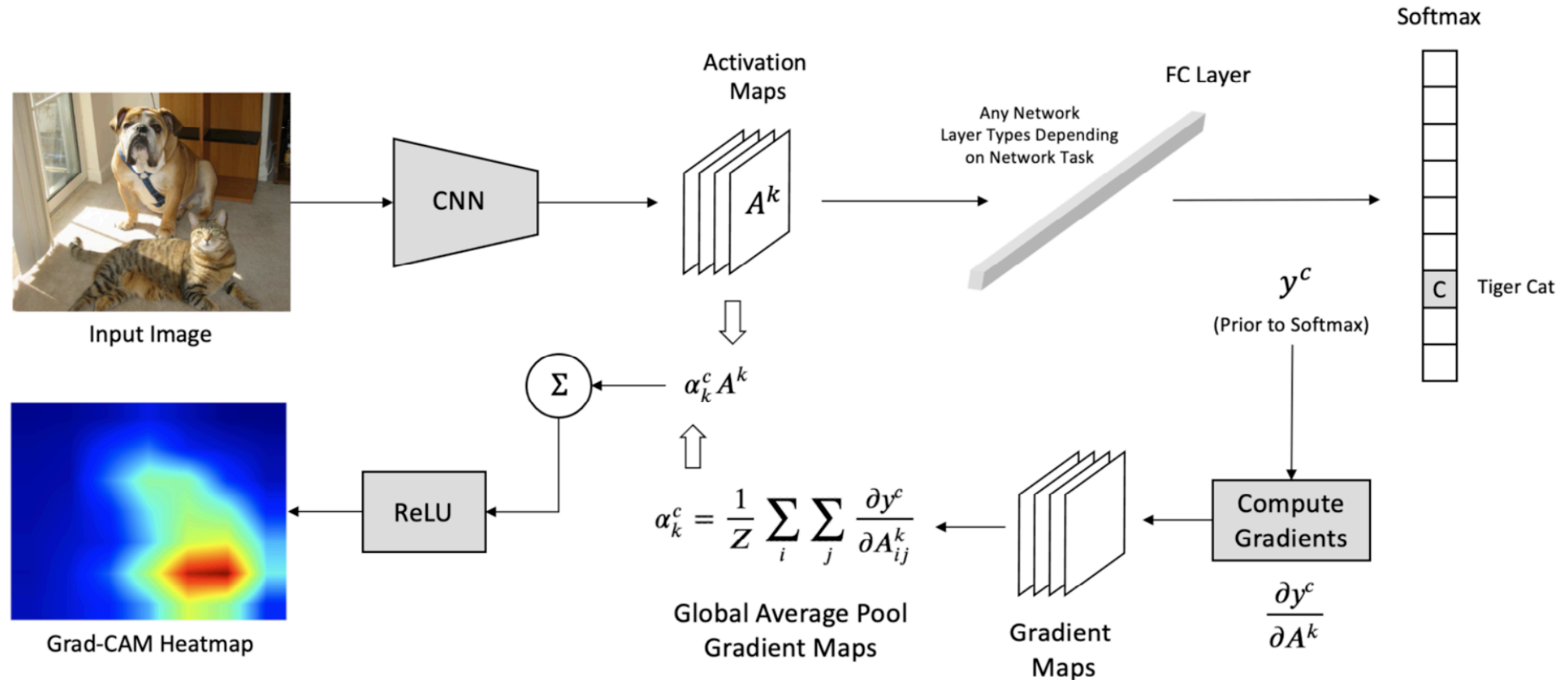


Saliency maps and their coefficient of two models predicting correctly "Bird", Figure from (Chalasani et al. 2020)

# Robustness and XAI 2



# Robustness and XAI 2



GradCam method, figure from learnopencv.com

# Robustness and XAI 3

