Optimization for Robust Deep Learning

M. Pawan Kumar

University of Oxford

Joint work with **Rudy Bunel, Alban Desmaison**,
Krishnamurthy Dvijotham, Pushmeet Kohli and Philip Torr

ImageNet Challenge

~1 M images, 1000 classes

Flute

Strawberry

Traffic light

Backpack

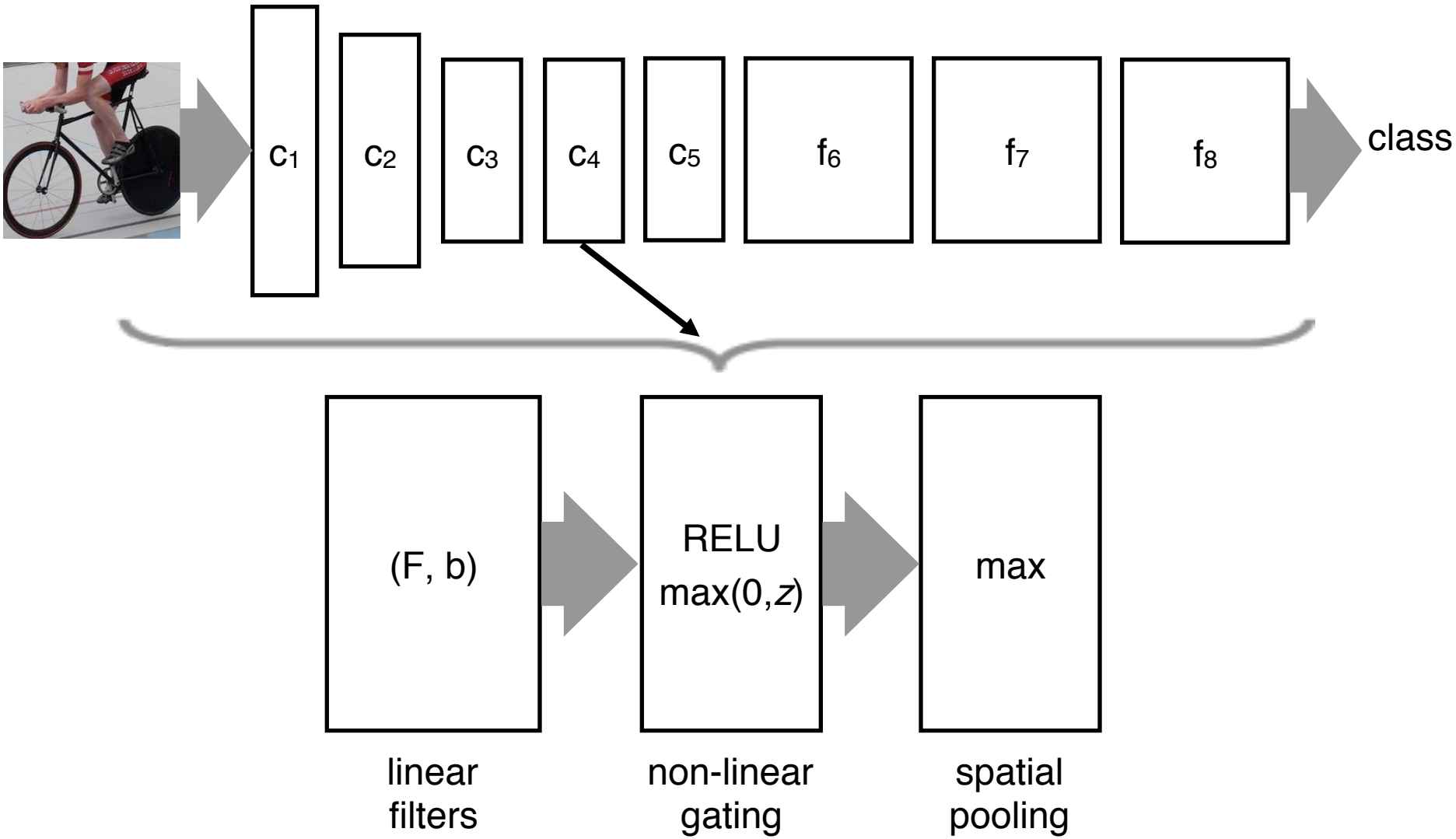
Matchstick

Sea lion

Bathing cap

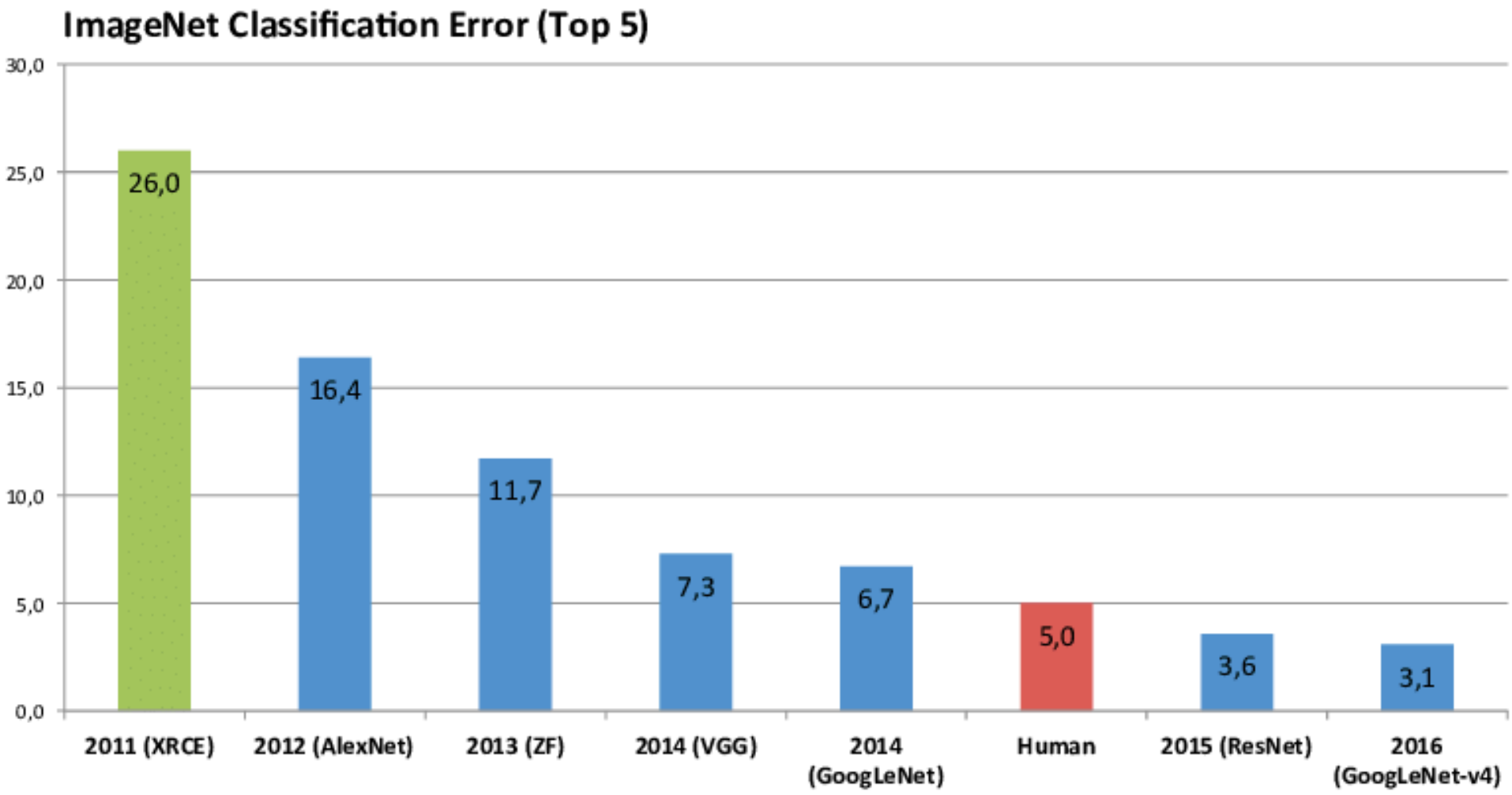
Racket

Deep Learning

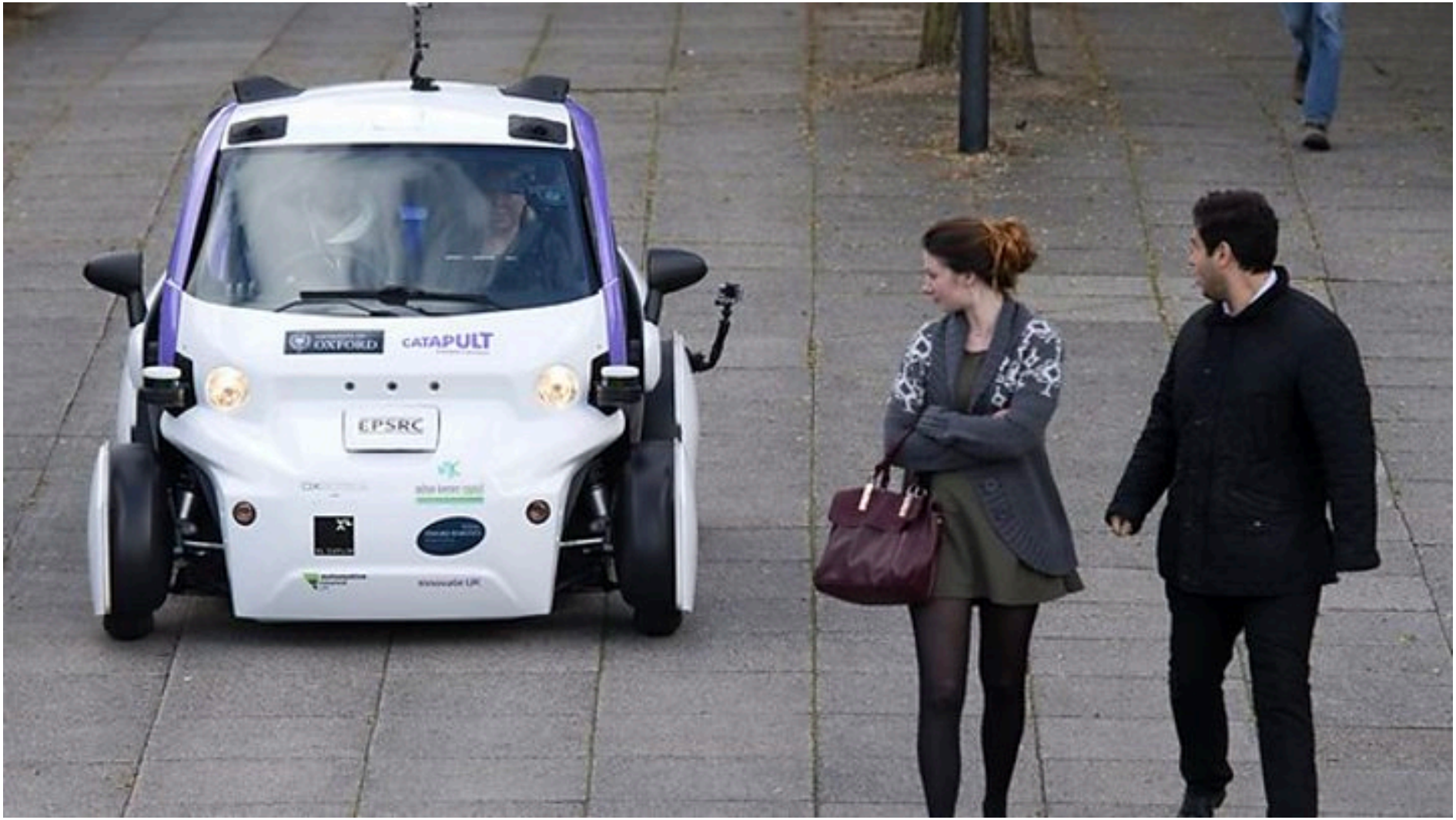


Linear functions followed by piecewise linear non-linearities

ImageNet Challenge



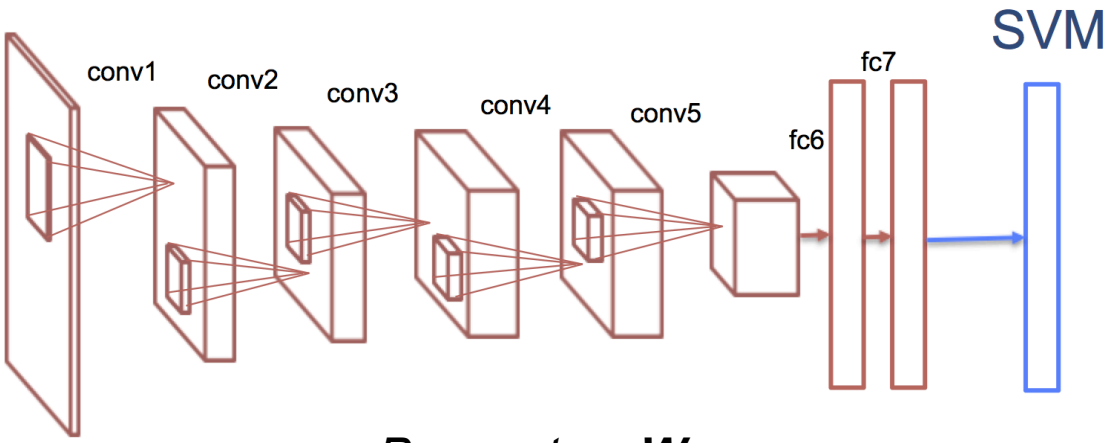
Self-Driving Car



Human drivers replaced by deep neural networks

Deep Learning

Training Data



*Parameters **W***

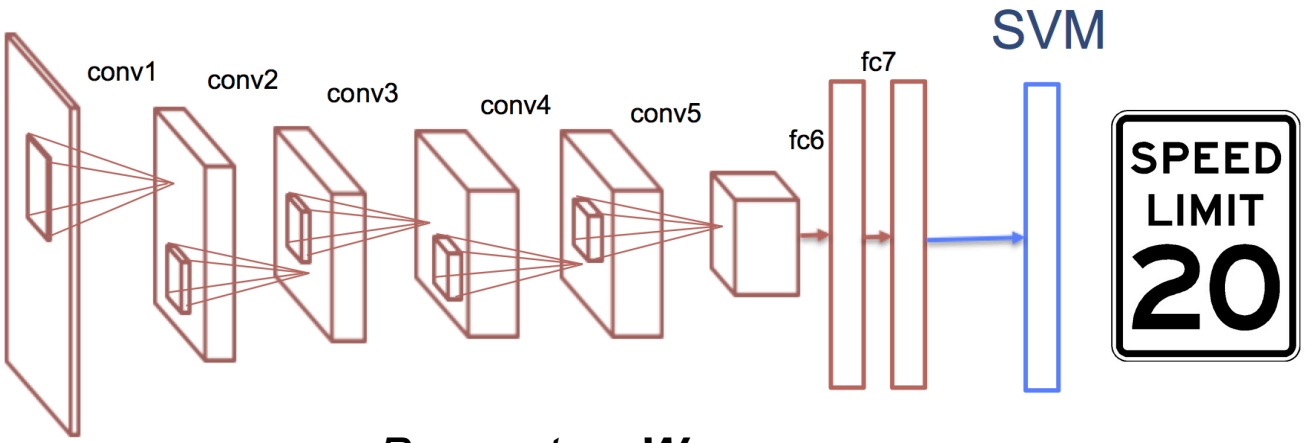
Road sign recognition

Parameters **W** perform multiclass classification

Estimate **W** using training data

Deep Learning

Test Image

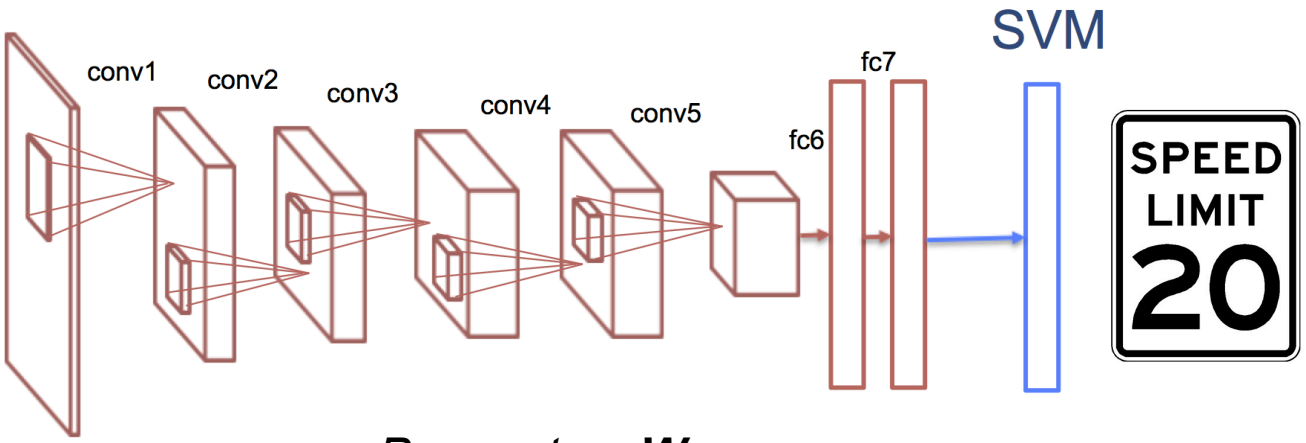


*Parameters **W***

Parameters **W** select the class of a new input image

Deep Learning

Test Image



*Parameters **W***

Small deformations can cause fatal errors

Input Image

Blurring, Saturation

Prediction

Aung et al., 2017

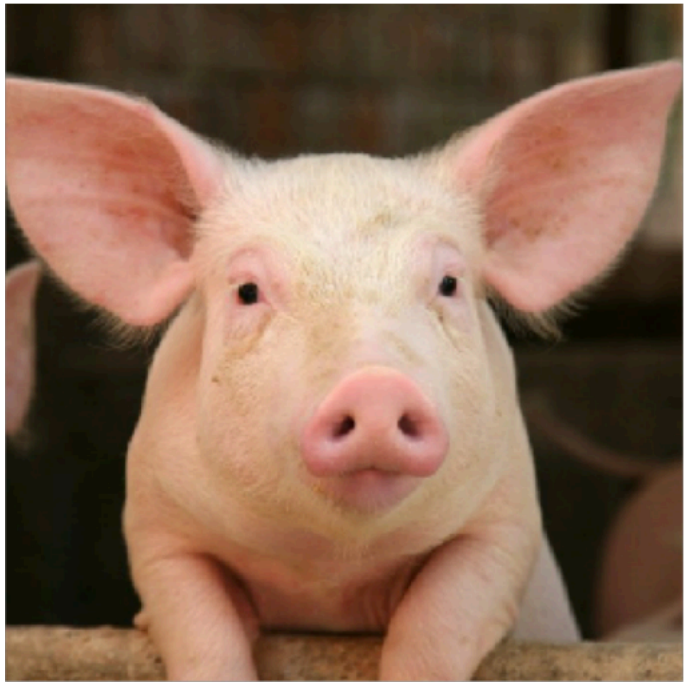
Input Image

Pixel Errors

Prediction

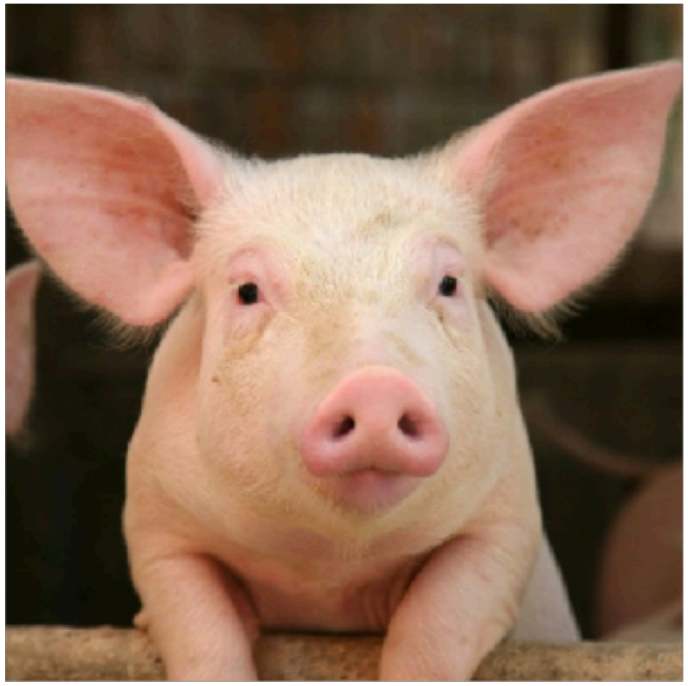
Evtimov et al., 2017

Spot the Difference

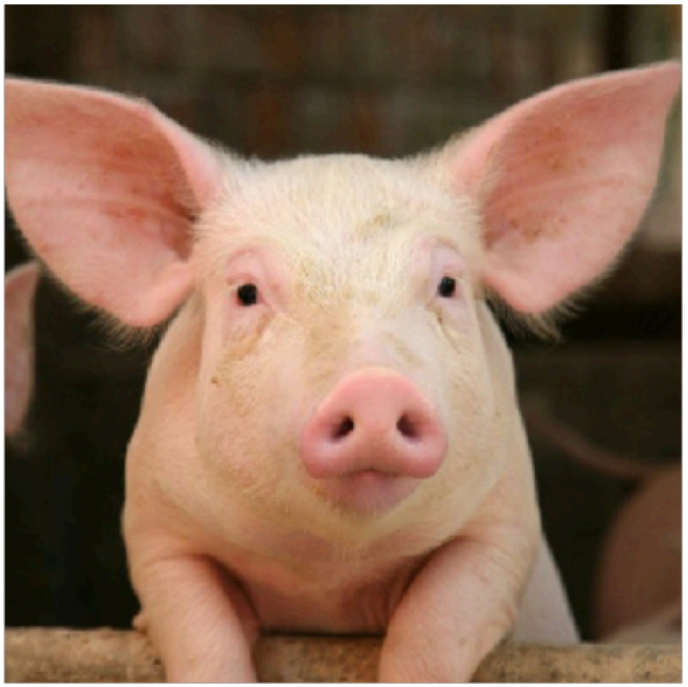


Spot the Difference

“pig” (91%)

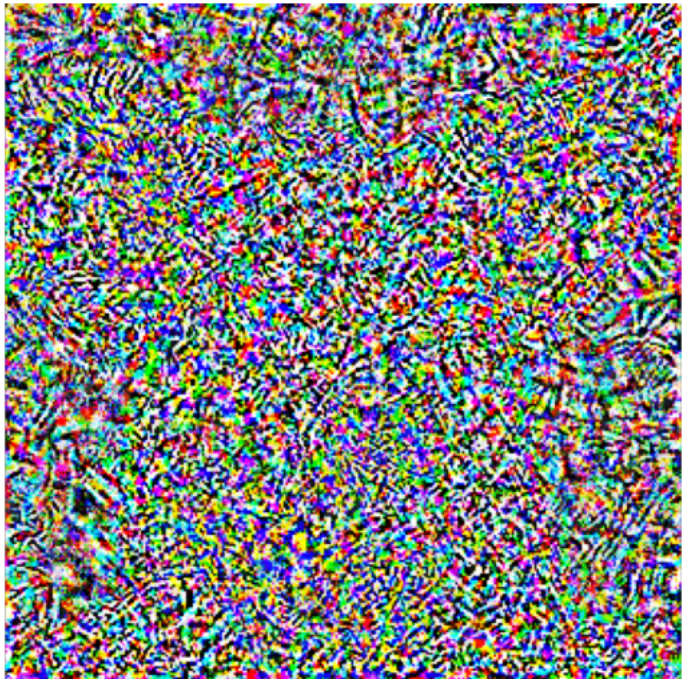


“airliner” (99%)

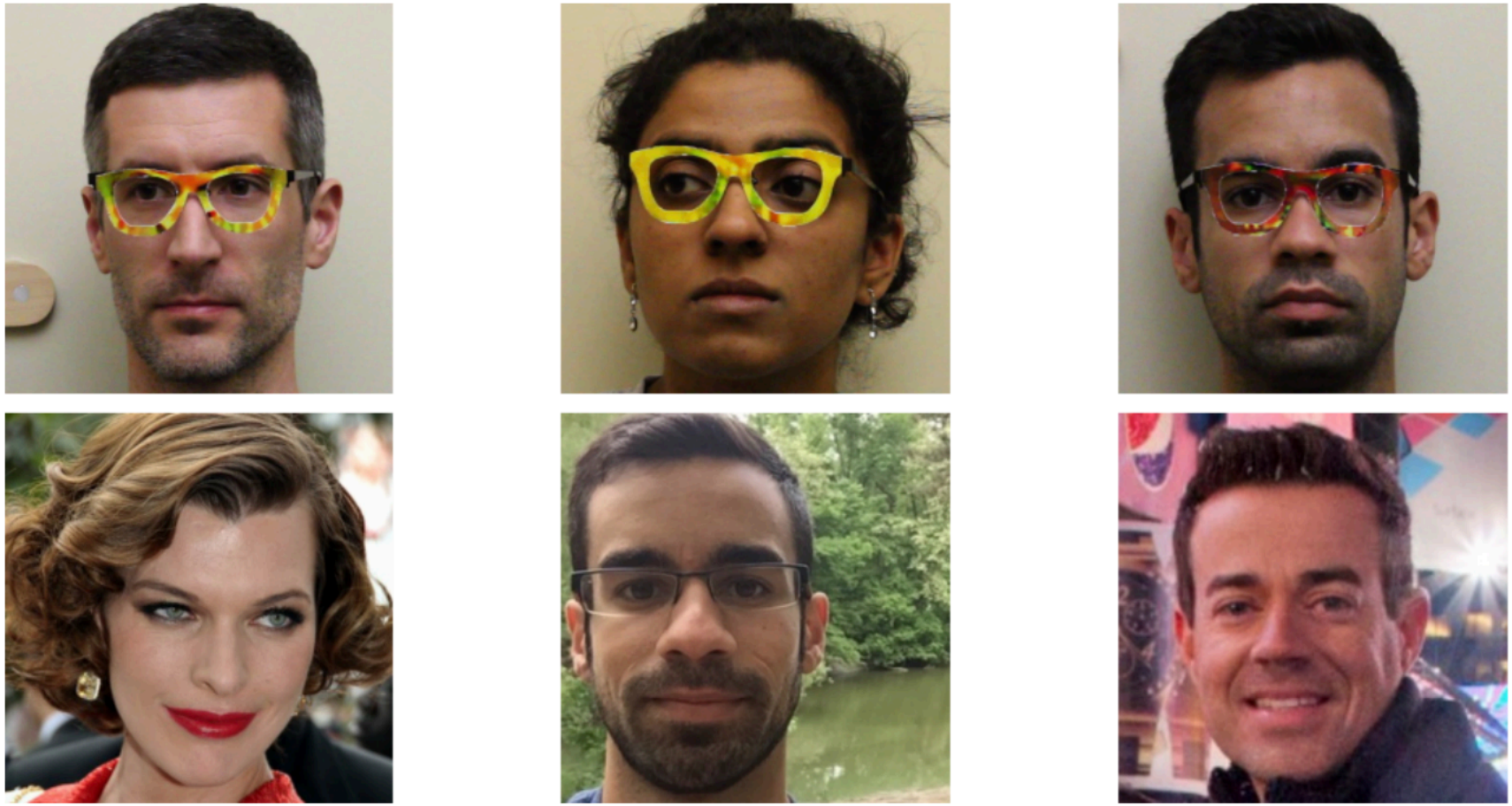


Difference

0.005 x



Glasses



Sharif, Bhagavatula, Bauer and Reiter, 2016

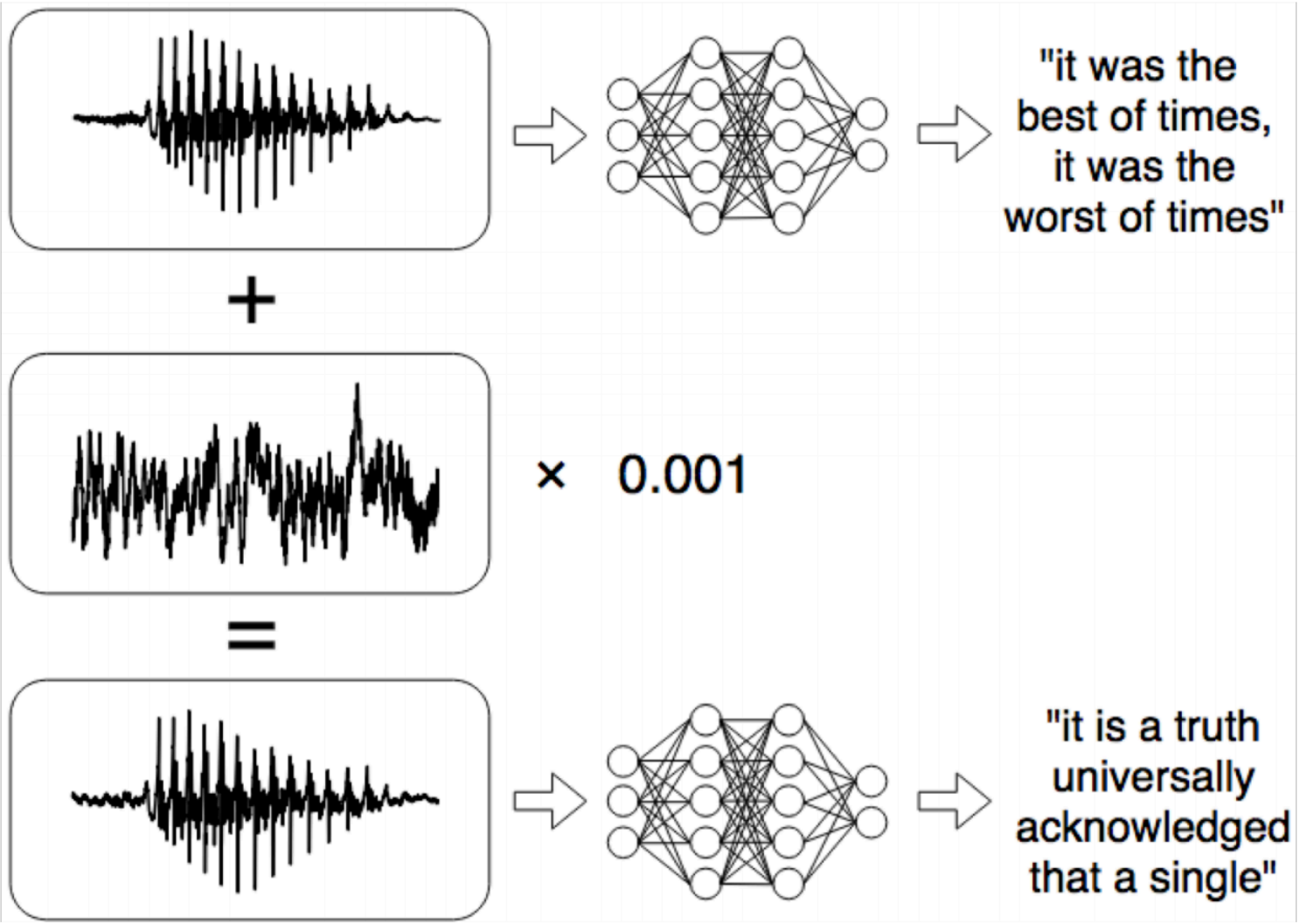
3D Object

classified as turtle classified as rifle

classified as other

Athalye, Engstrom, Ilyas and Kwok, 2017

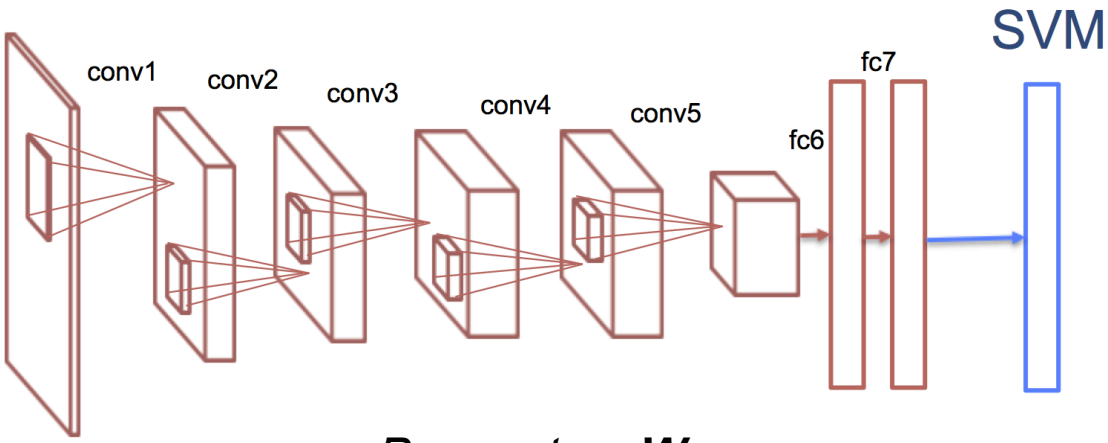
Audio



Carlini and Wagner, 2018

Robust Deep Learning

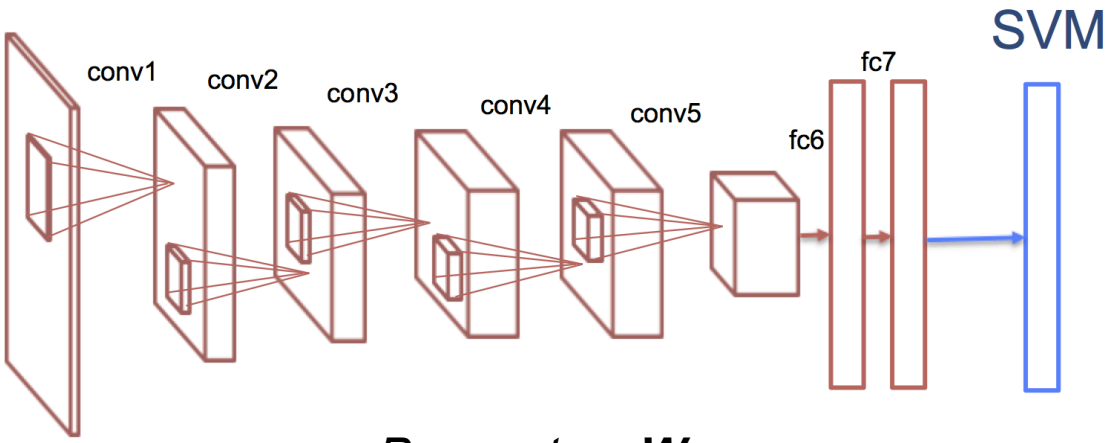
Training Data



*Parameters **W***

Robust Deep Learning

Training Data



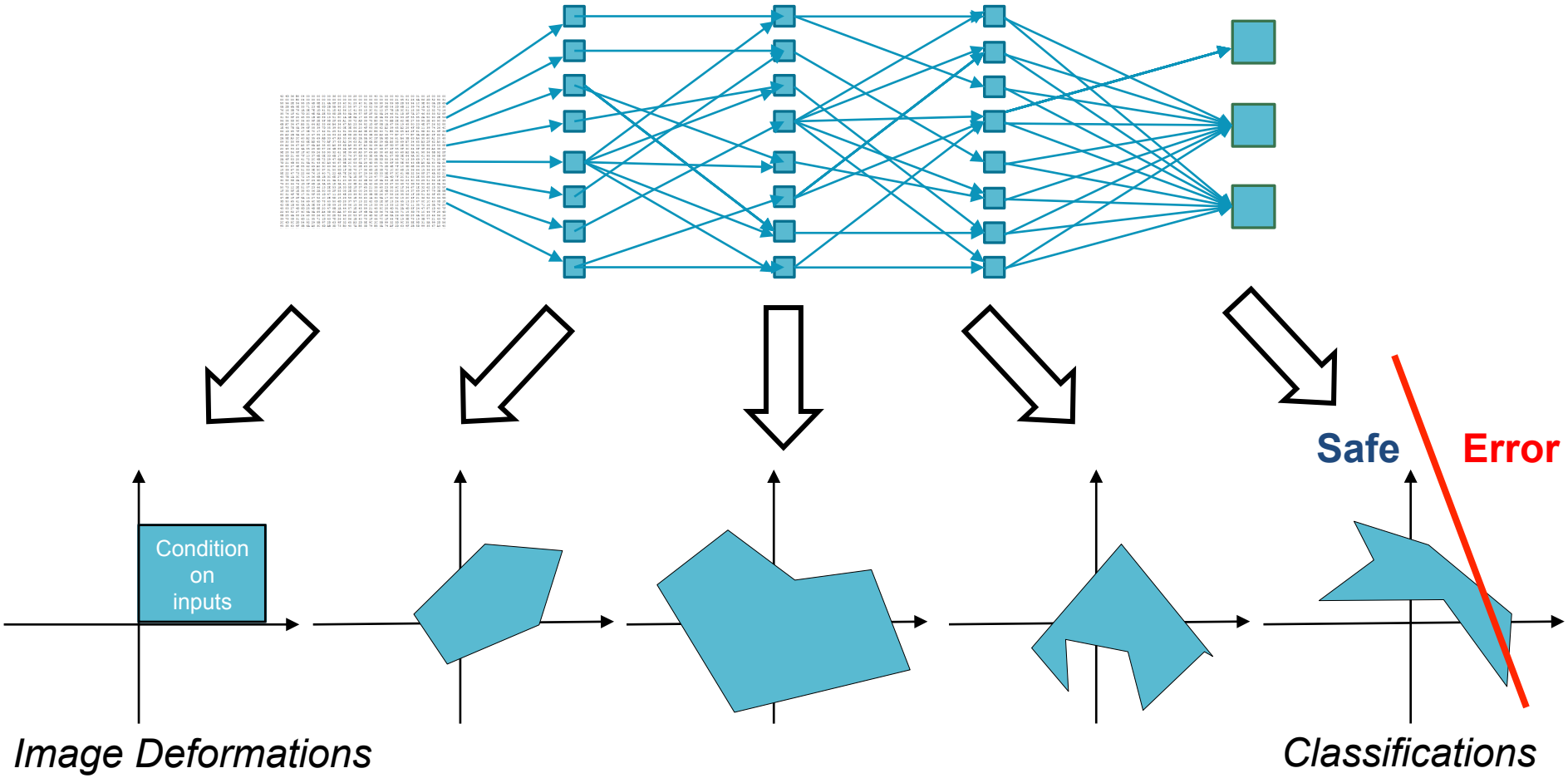
*Parameters **W***

Augment Training Data Set

Identify Deformation With Error

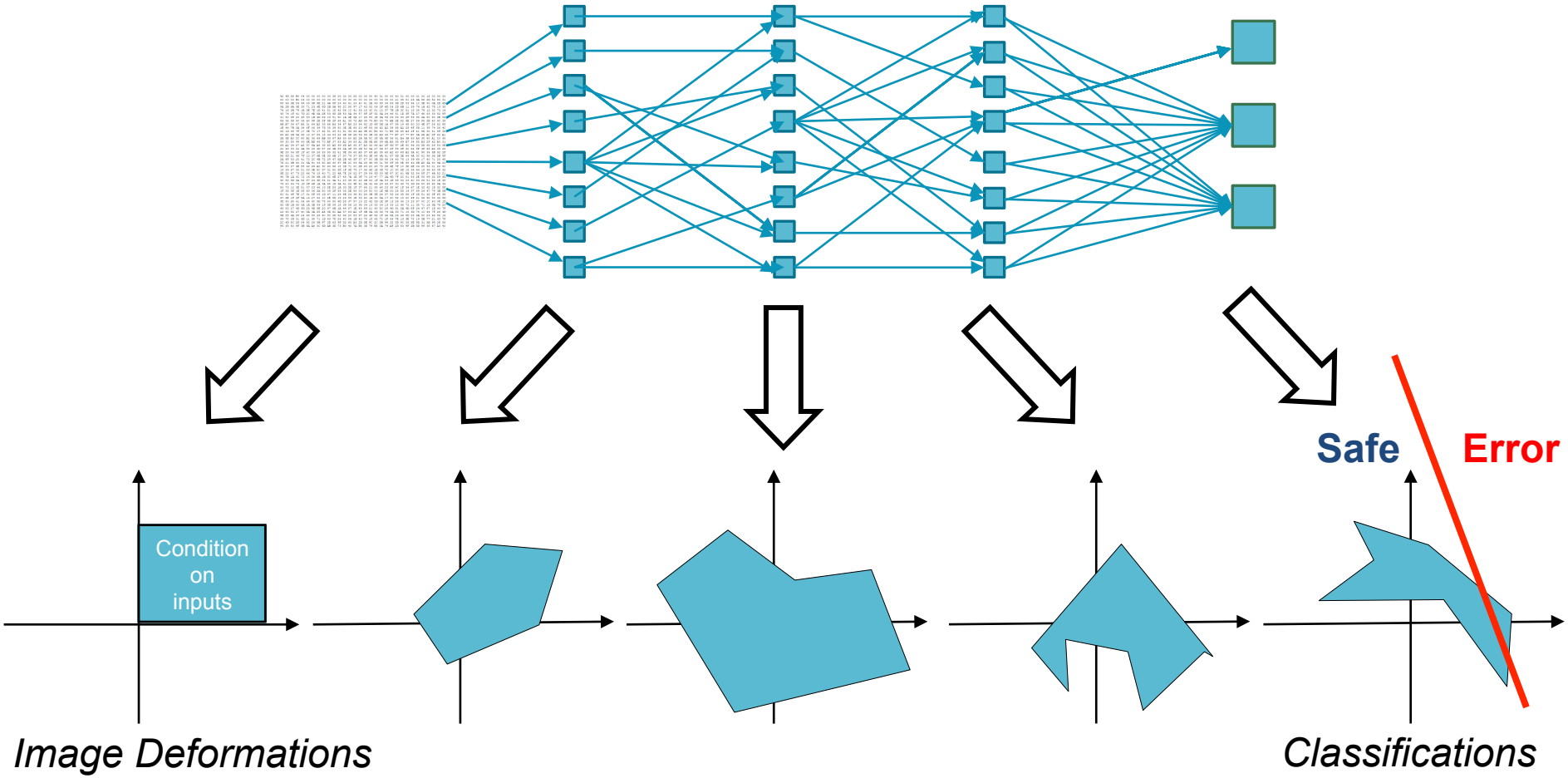
Robust Deep Learning

Is there an erroneous output?



Robust Deep Learning

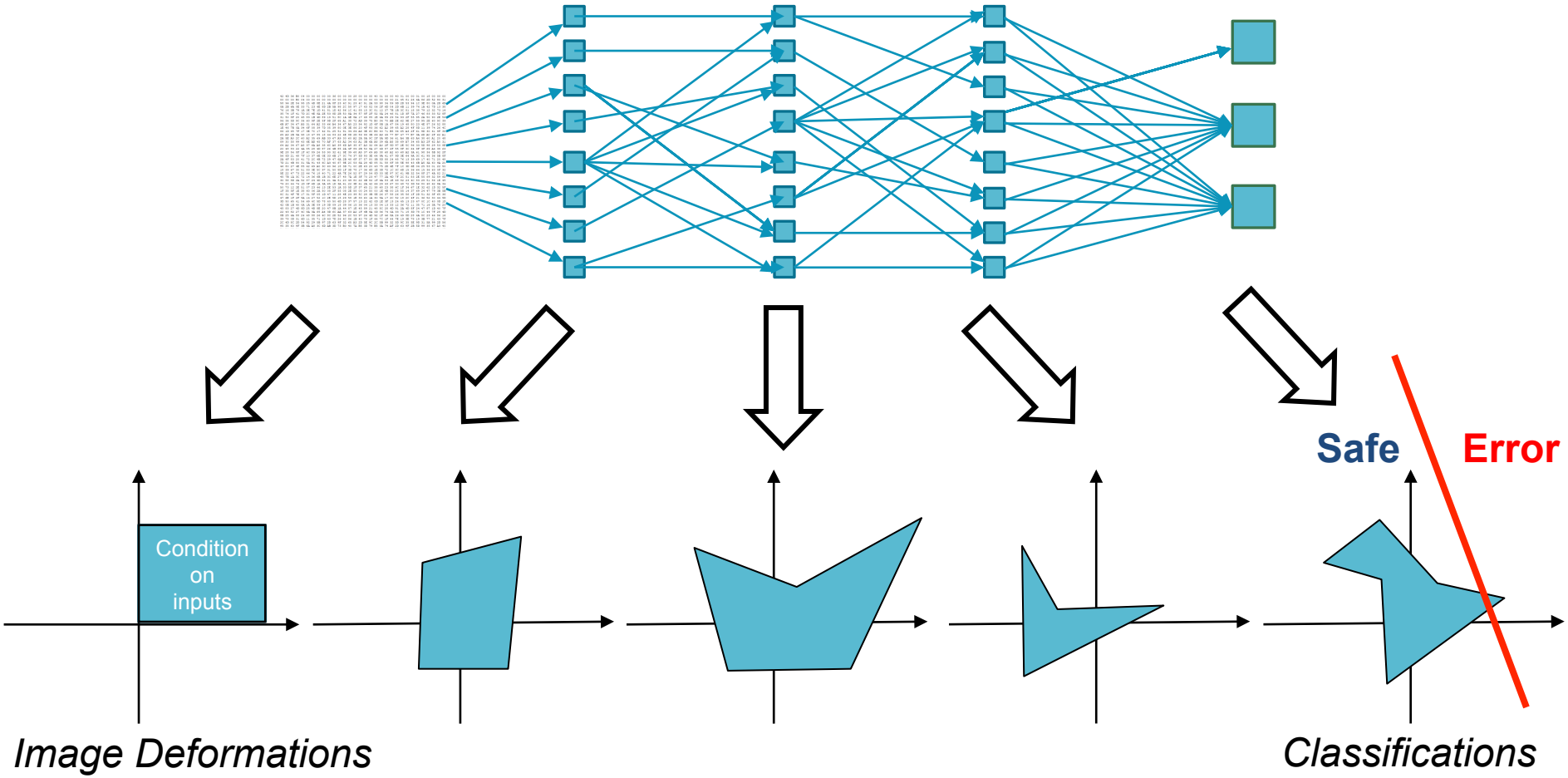
Is there an erroneous output?



Re-estimate parameters **W**

Robust Deep Learning

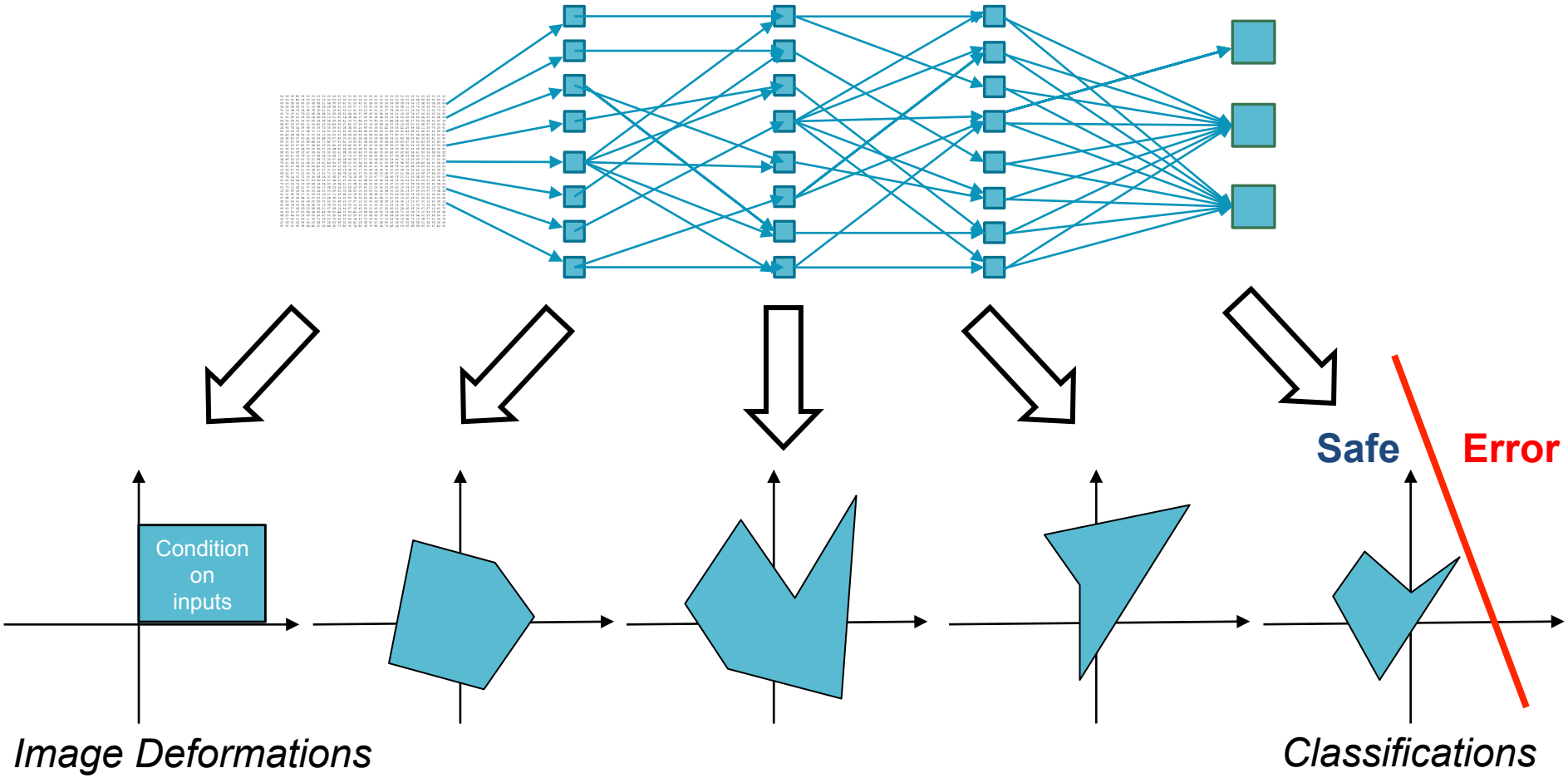
Is there an erroneous output?



Re-estimate parameters **W**

Robust Deep Learning

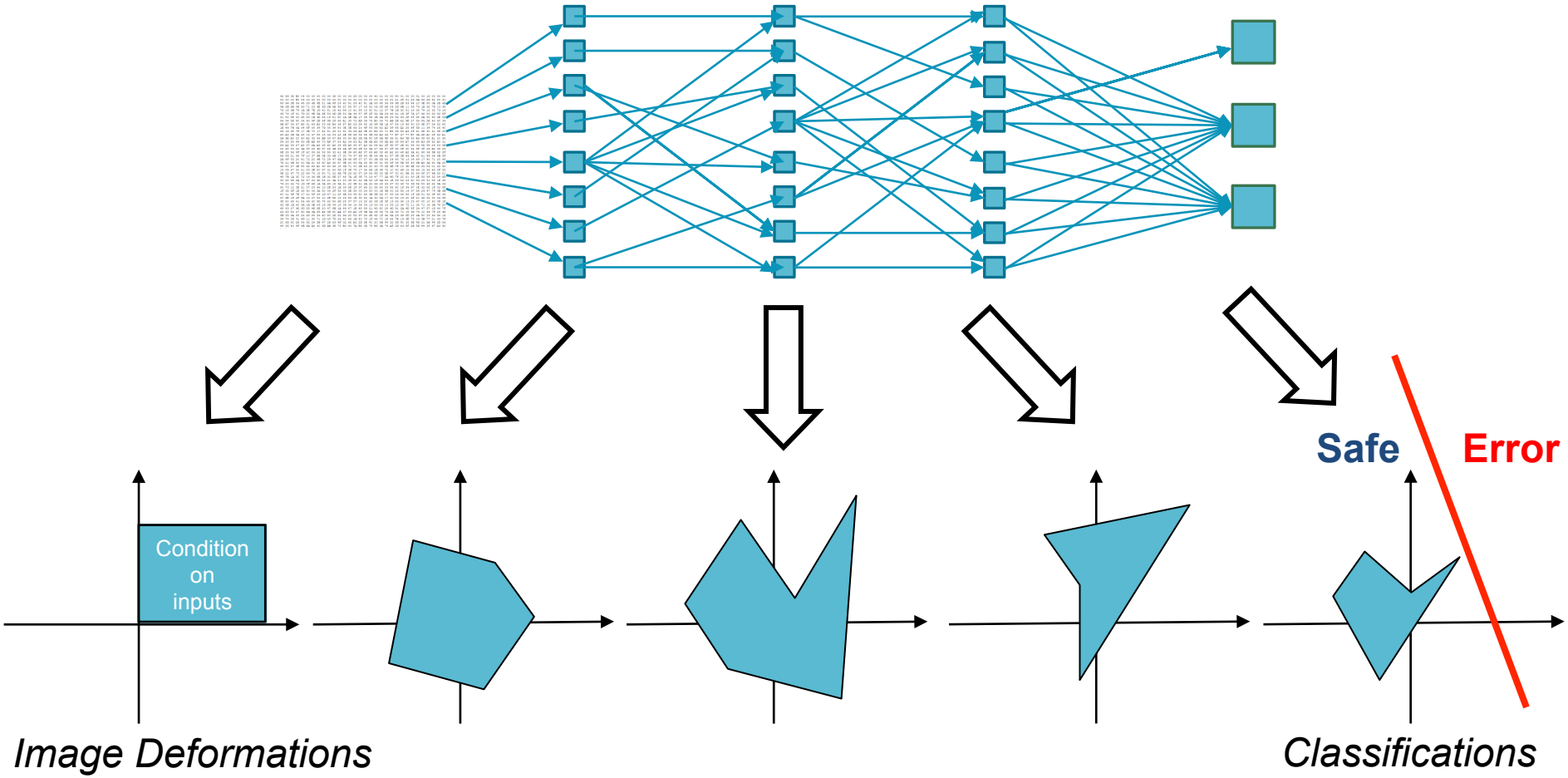
Is there an erroneous output?



Re-estimate parameters **W**

Robust Deep Learning

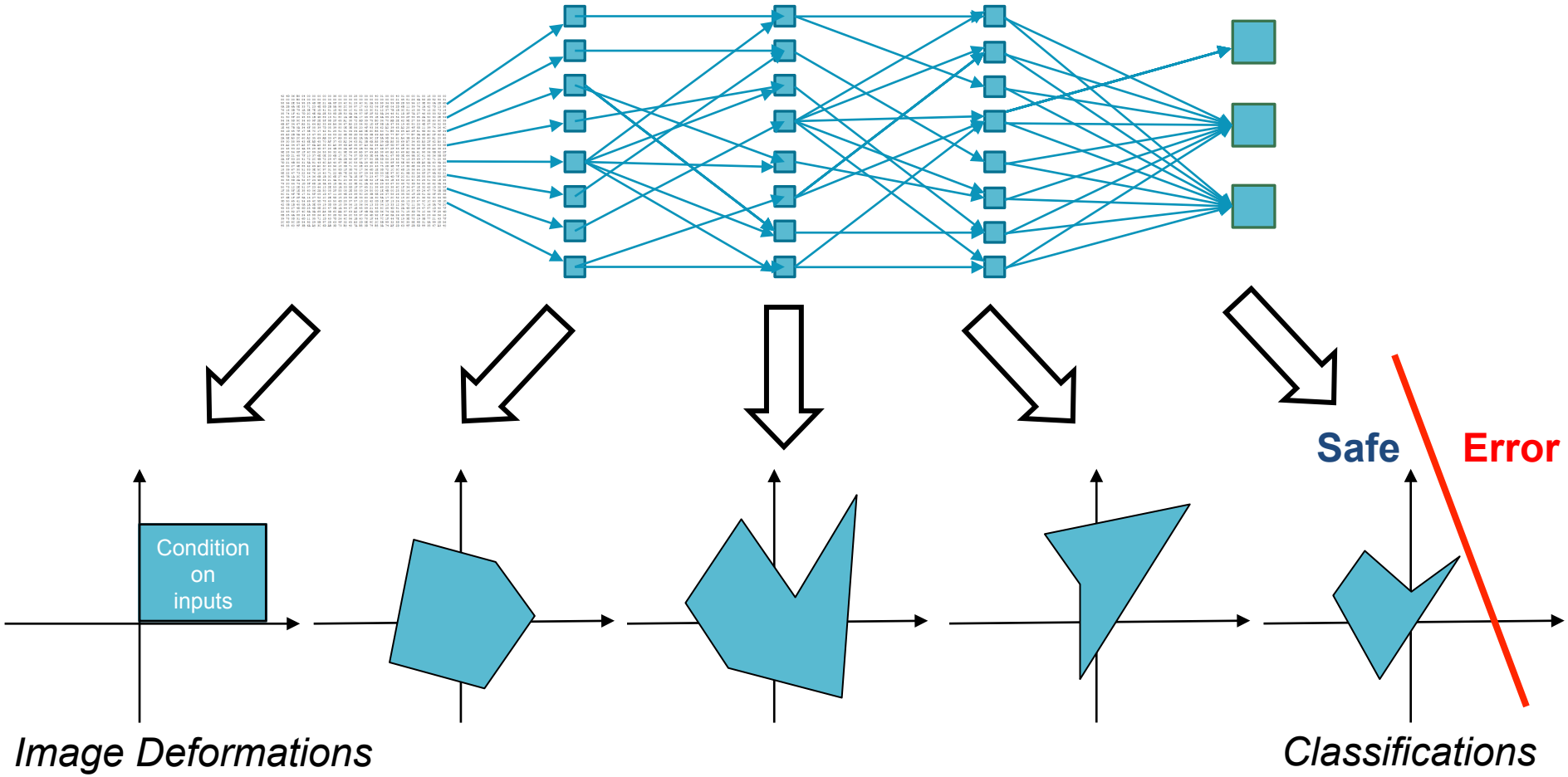
Is there an erroneous output?



Terminate when all putative outputs are safe

Robust Deep Learning

Is there an erroneous output?



Terminate when all putative outputs are safe

Outline

- Neural Network Bounds

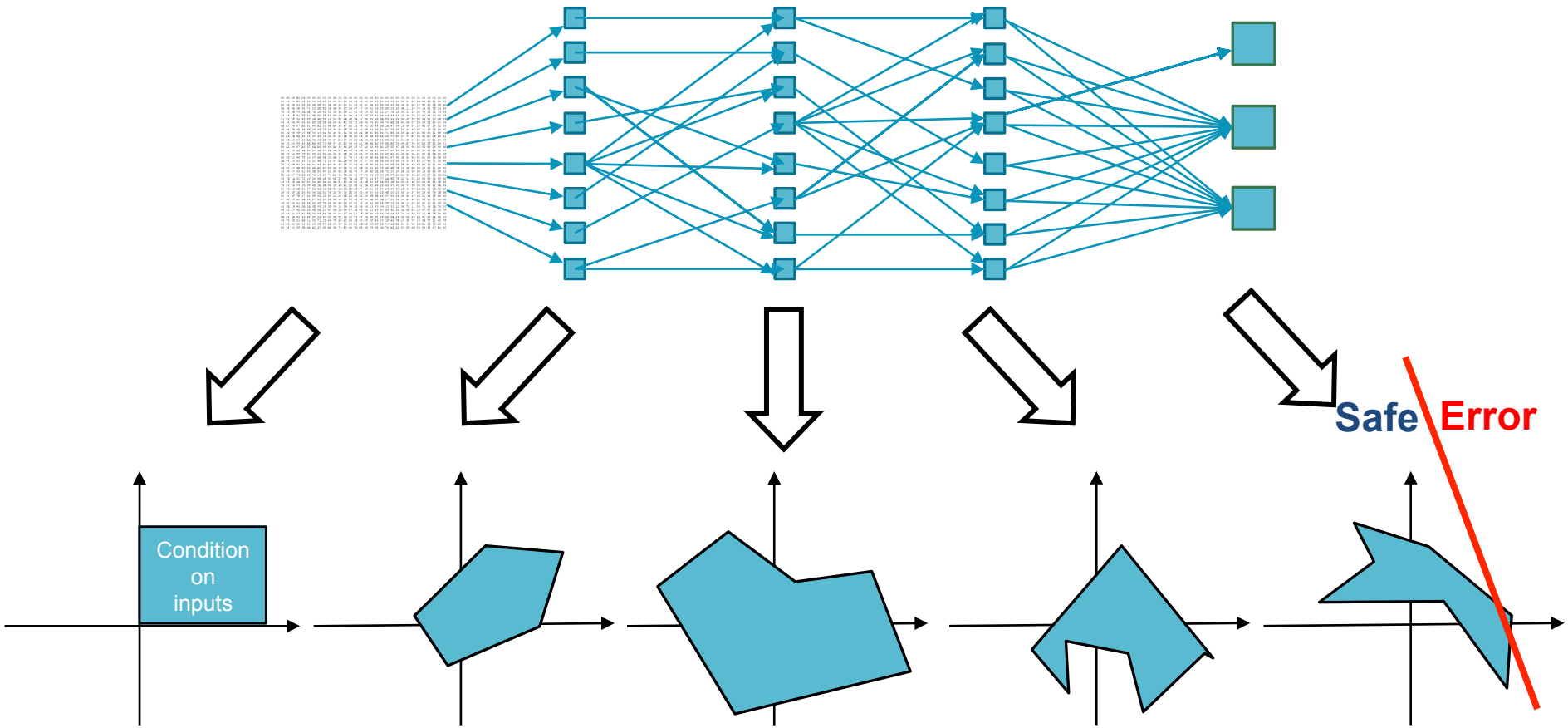
- Lagrangian Decomposition
- Proximal Minimization

Method of Multipliers

- Results

Robust Deep Learning

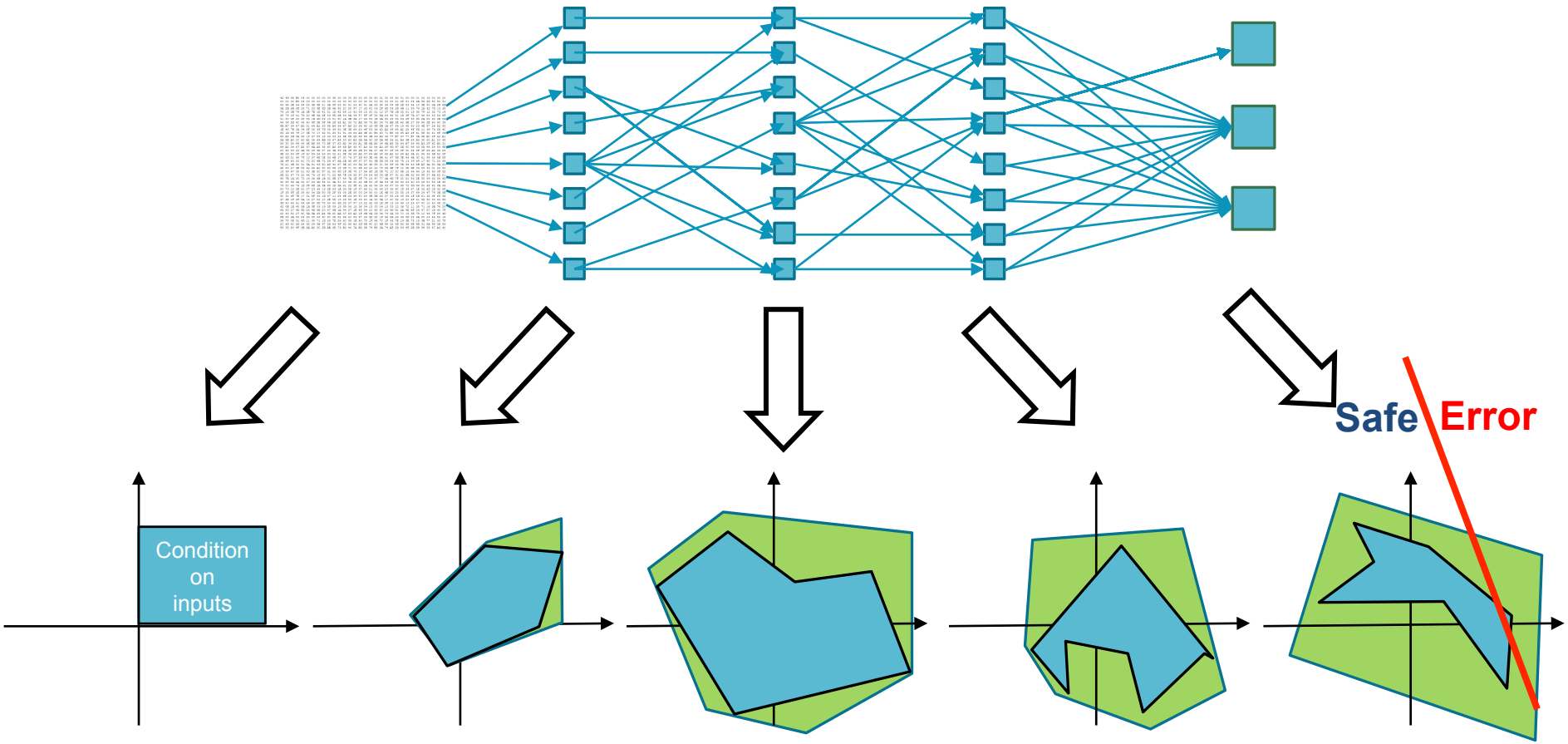
Is there an erroneous output?



Non-convexity makes the problem NP-hard

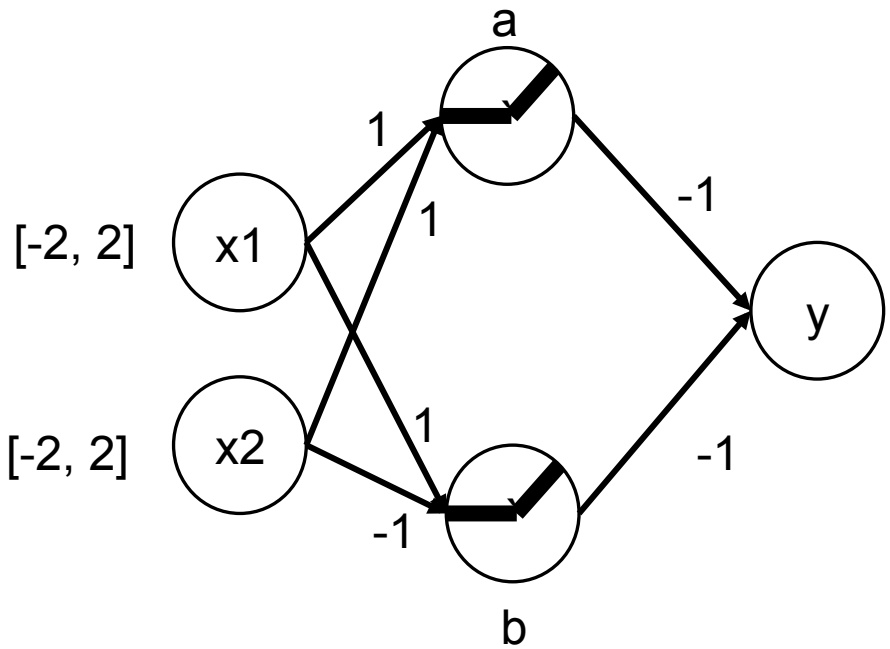
Robust Deep Learning

Is there an erroneous output?



Replace by a convex superset

Example



Find the minimum valid output

min $\quad y$

s.t. $\quad -2 \le x_1 \le 2$

$-2 \le x_2 \le 2$

$a_{in} = x_1 + x_2$

$b_{in} = x_1 - x_2$

$a_{out} = \max\{a_{in}, 0\}$

$b_{out} = \max\{b_{in}, 0\}$

$y = - a_{out} - b_{out}$

Example

Linear constraints

Easy to handle

$$\min \quad y$$

$$\text{s.t.} \quad -2 \le x_1 \le 2$$

$$-2 \le x_2 \le 2$$

$$a_{in} = x_1 + x_2$$

$$b_{in} = x_1 - x_2$$

$$a_{out} = \max\{a_{in}, 0\}$$

$$b_{out} = \max\{b_{in}, 0\}$$

$$y = -a_{out} - b_{out}$$

Example

$$\min \quad y$$

$$\text{s.t.} \quad -2 \le x_1 \le 2$$

$$-2 \le x_2 \le 2$$

$$a_{in} = x_1 + x_2$$

$$b_{in} = x_1 - x_2$$

$$\mathbf{a_{out} = \max\{a_{in}, 0\}}$$

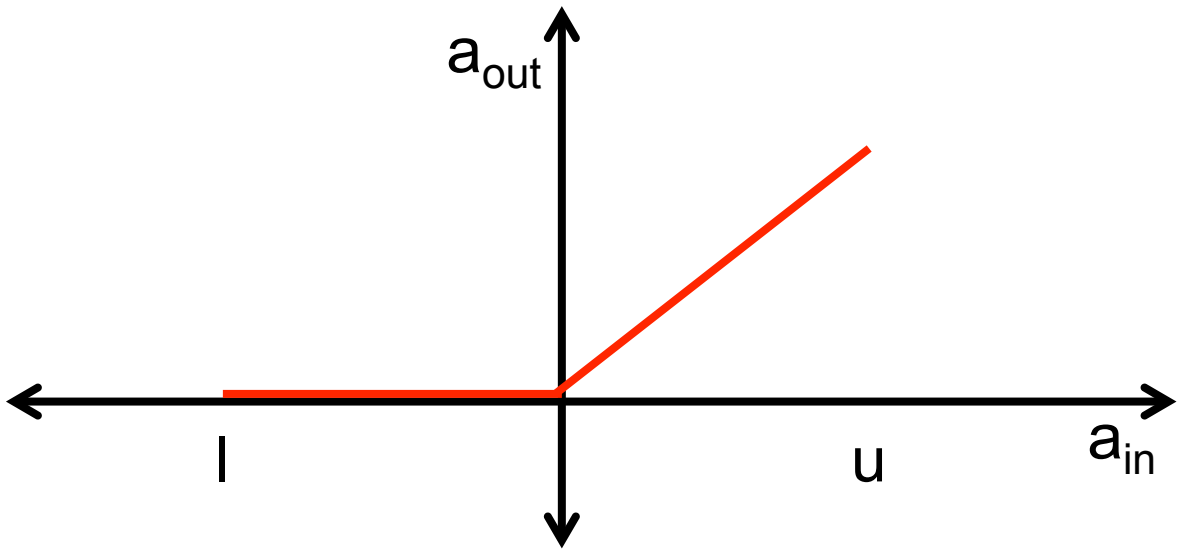
$$\mathbf{b_{out} = \max\{b_{in}, 0\}}$$

$$y = -a_{out} - b_{out}$$

Non-linear constraints

ReLU

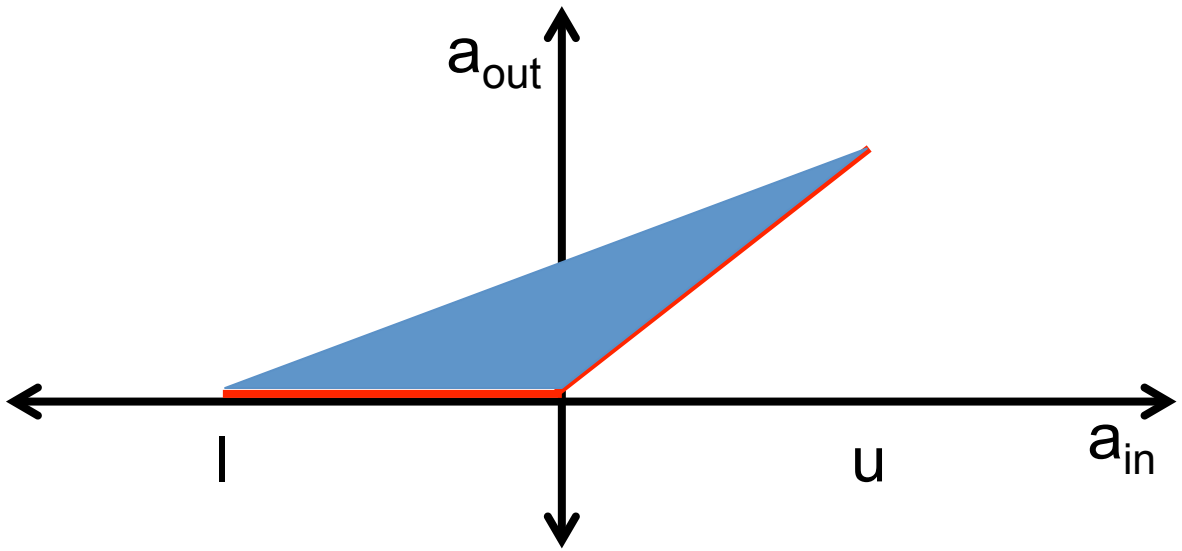
$a_{out} = \max\{a_{in}, 0\}$ $\qquad$ $a_{in} \in [l,u]$



Set of feasible (a_{in}, a_{out}) is not convex

Convex Relaxation

$a_{out} = \max\{a_{in}, 0\}$ $\qquad$ $a_{in} \in [l,u]$



Ehlers 2017

Replace with smallest convex superset

Original Problem

$$\min \quad y$$

$$\text{s.t.} \quad -2 \le x_1 \le 2$$

$$-2 \le x_2 \le 2$$

$$a_{in} = x_1 + x_2$$

$$b_{in} = x_1 - x_2$$

$$\mathbf{a_{out} = \max\{a_{in}, 0\}}$$

$$\mathbf{b_{out} = \max\{b_{in}, 0\}}$$

$$y = -a_{out} - b_{out}$$

Convex Relaxation

Linear Program

$$\min \quad y$$

$$\text{s.t.} \quad -2 \le x_1 \le 2$$

$$-2 \le x_2 \le 2$$

$$a_{in} = x_1 + x_2$$

$$b_{in} = x_1 - x_2$$

$$a_{out} \ge 0,\ a_{out} \ge a_{in},\ a_{out} \le 0.5a_{in} + 2$$

$$b_{out} \ge 0,\ b_{out} \ge b_{in},\ b_{out} \le 0.5b_{in} + 2$$

$$y = -a_{out} - b_{out}$$

General Formulation

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x}$$

$$\text{s.t.} \quad A_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

Linear mapping of the k-th layer (“easy” constraints)

General Formulation

$$\min_{\mathbf{x}} \quad \mathbf{c}^T\mathbf{x}$$

$$\text{s.t.} \quad A_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

Linear mapping of the k-th layer (“easy” constraints)

Convex hull of activations (relaxations)

Outline

- Neural Network Bounds
- **Lagrangian Decomposition**
- Proximal Minimization
- Results

Additional Variables

$$\min_{\mathbf{x},\, \mathbf{x}_k,\, \mathbf{z}_k} \quad \mathbf{c}^T\mathbf{x}$$

$$\text{s.t.} \quad A_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{x}) \qquad k = 1, \ldots, L$$

Additional Variables

$$\min_{\mathbf{x},\, \mathbf{x}_k,\, \mathbf{z}_k} \quad \mathbf{c}^T\mathbf{x}$$

$$\begin{aligned}
\text{s.t.} \quad & A_k(\mathbf{x}_k) && k = 1, \ldots, L \\
& B_k(\mathbf{z}_k) && k = 1, \ldots, L \\
& \mathbf{x}_k = \mathbf{x} && k = 1, \ldots, L \\
& \mathbf{z}_k = \mathbf{x} && k = 1, \ldots, L
\end{aligned}$$

Partial Lagrangian

$$\max_{\boldsymbol{\lambda},\, \boldsymbol{\mu}} \ \min_{\mathbf{x},\, \mathbf{x}_k,\, \mathbf{z}_k} \quad \mathbf{c}^T\mathbf{x} + \Sigma_k\, \boldsymbol{\lambda}_k^T(\mathbf{x}_k - \mathbf{x}) + \Sigma_k\, \boldsymbol{\mu}_k^T(\mathbf{z}_k - \mathbf{x})$$

$$\text{s.t.} \quad A_k(\mathbf{x}_k) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{z}_k) \qquad k = 1, \ldots, L$$

Eliminating **x** using KKT conditions

Lagrangian Decomposition

$$\max_{\boldsymbol{\lambda},\, \boldsymbol{\mu}} \ \min_{\mathbf{x}_k,\, \mathbf{z}_k} \quad \Sigma_k\, \boldsymbol{\lambda}_k^T \mathbf{x}_k + \Sigma_k\, \boldsymbol{\mu}_k^T \mathbf{z}_k$$

$$\text{s.t.} \quad A_k(\mathbf{x}_k) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{z}_k) \qquad k = 1, \ldots, L$$

$$\Sigma_k\, \boldsymbol{\lambda}_k + \Sigma_k\, \boldsymbol{\mu}_k = \mathbf{c}$$

Projected subgradient descent (Dvijotham et al., 2018)

Tuning step-size schedule is hard

Outline

- Neural Network Bounds
- Lagrangian Decomposition
- **Proximal Minimization**
- Results

Proximal Minimization

$$\max_{\boldsymbol{\lambda},\, \boldsymbol{\mu}} \; \min_{\mathbf{x}_k,\, \mathbf{z}_k} \quad \Sigma_k \, \boldsymbol{\lambda}_k^T \mathbf{x}_k + \Sigma_k \, \boldsymbol{\mu}_k^T \mathbf{z}_k$$

$$+ \eta \, ||\boldsymbol{\lambda} - \boldsymbol{\lambda}'||^2 + \eta \, ||\boldsymbol{\mu} - \boldsymbol{\mu}'||^2$$

$$\text{s.t.} \quad A_k(\mathbf{x}_k) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{z}_k) \qquad k = 1, \ldots, L$$

$$\Sigma_k \, \boldsymbol{\lambda}_k + \Sigma_k \, \boldsymbol{\mu}_k = \mathbf{c}$$

Initialize $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$ and $\boldsymbol{\mu} = \boldsymbol{\mu}'$ Solve proximal problem

Proximal Minimization

$$\max_{\boldsymbol{\lambda},\, \boldsymbol{\mu}} \; \min_{\mathbf{x}_k,\, \mathbf{z}_k} \quad \Sigma_k \, \boldsymbol{\lambda}_k^T \mathbf{x}_k + \Sigma_k \, \boldsymbol{\mu}_k^T \mathbf{z}_k$$

$$+ \eta \, ||\boldsymbol{\lambda} - \boldsymbol{\lambda}'||^2 + \eta \, ||\boldsymbol{\mu} - \boldsymbol{\mu}'||^2$$

$$\text{s.t.} \quad A_k(\mathbf{x}_k) \qquad k = 1, \ldots, L$$

$$B_k(\mathbf{z}_k) \qquad k = 1, \ldots, L$$

$$\Sigma_k \, \boldsymbol{\lambda}_k + \Sigma_k \, \boldsymbol{\mu}_k = \mathbf{c}$$

Smooth quadratic dual Conditional gradient

Repeatedly solve proximal problems until convergence

Advantages

Conditional gradient of dual is the subgradient

Bach, 2015

Analytically computable optimal step-size

A single easy-to-tune parameter η

Highly parallelizable over GPUs

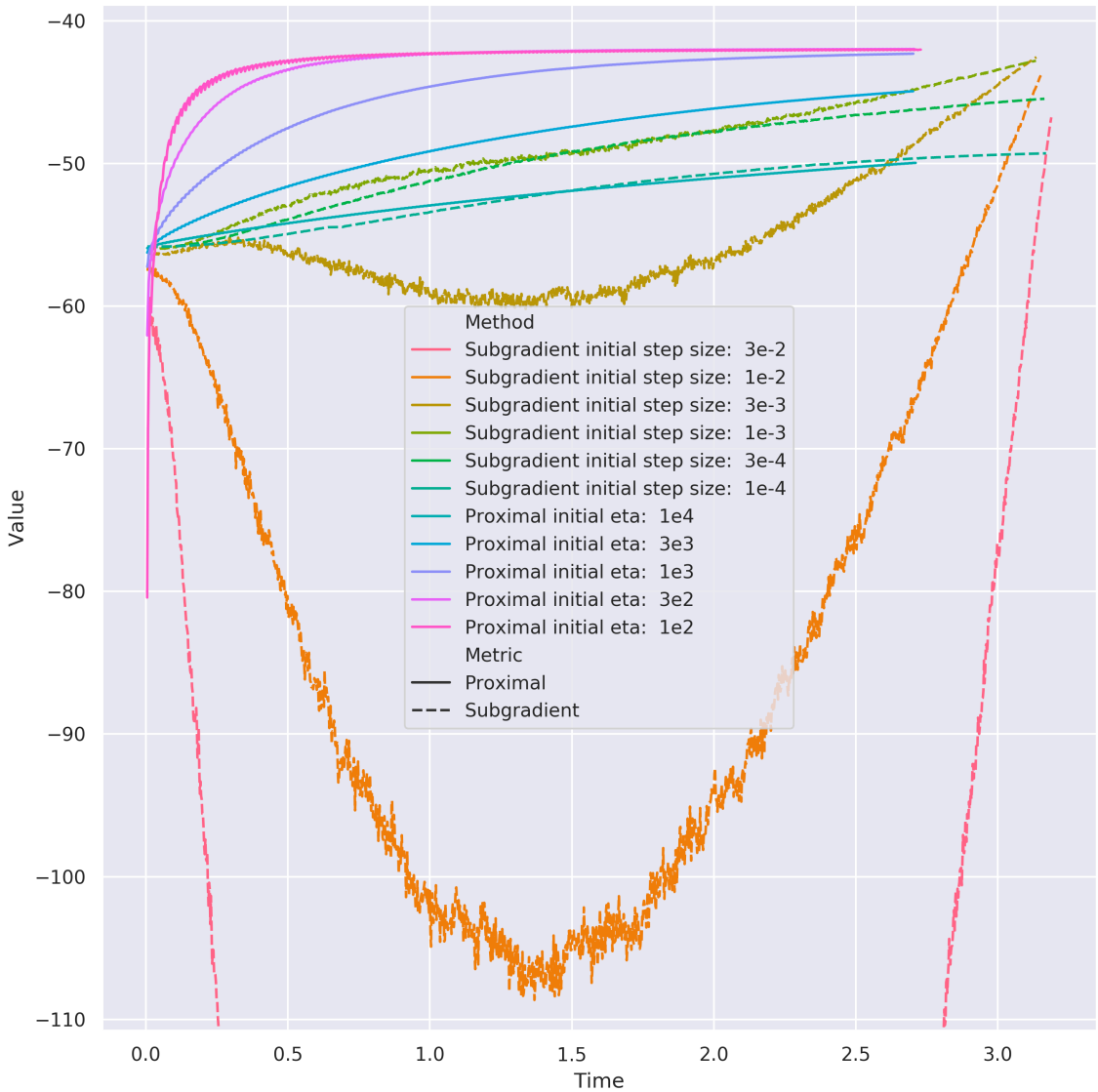
Outline

- Neural Network Bounds
- Lagrangian Decomposition
- Proximal Minimization
- **Results**

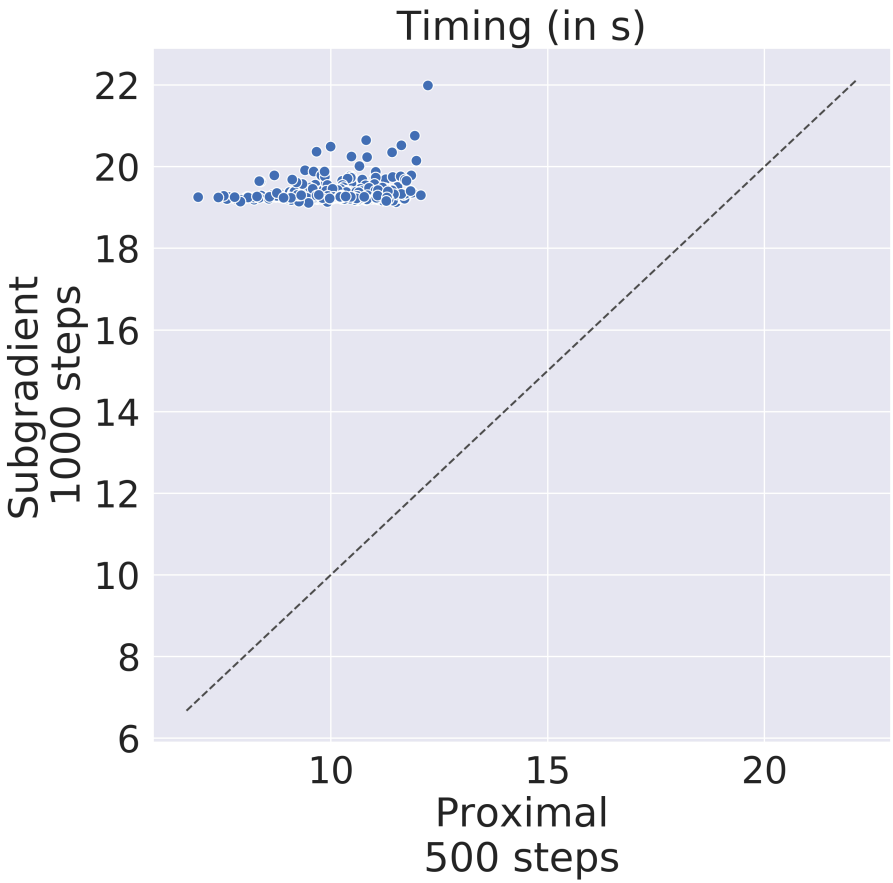
Network Architecture

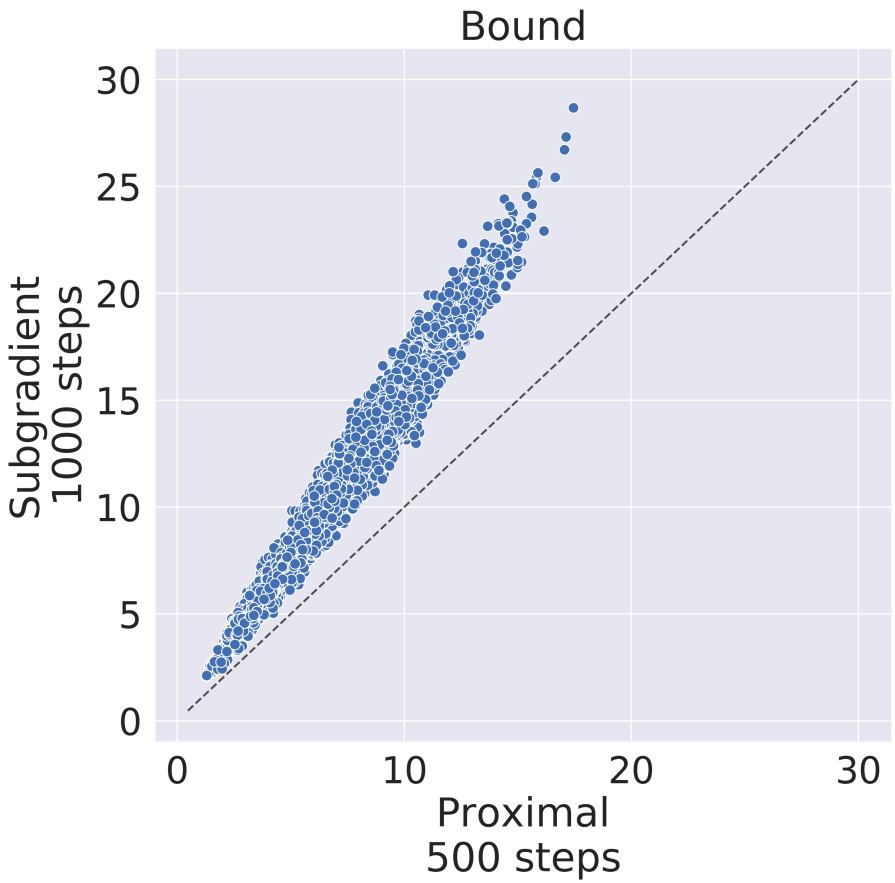
Conv2D(channels=16, kernel_size=4, stride=2) ReLU
Conv2D(channels=32, kernel_size=4, stride=2) ReLU
Linear(channels=100) ReLU
Linear(channels=10)

Results

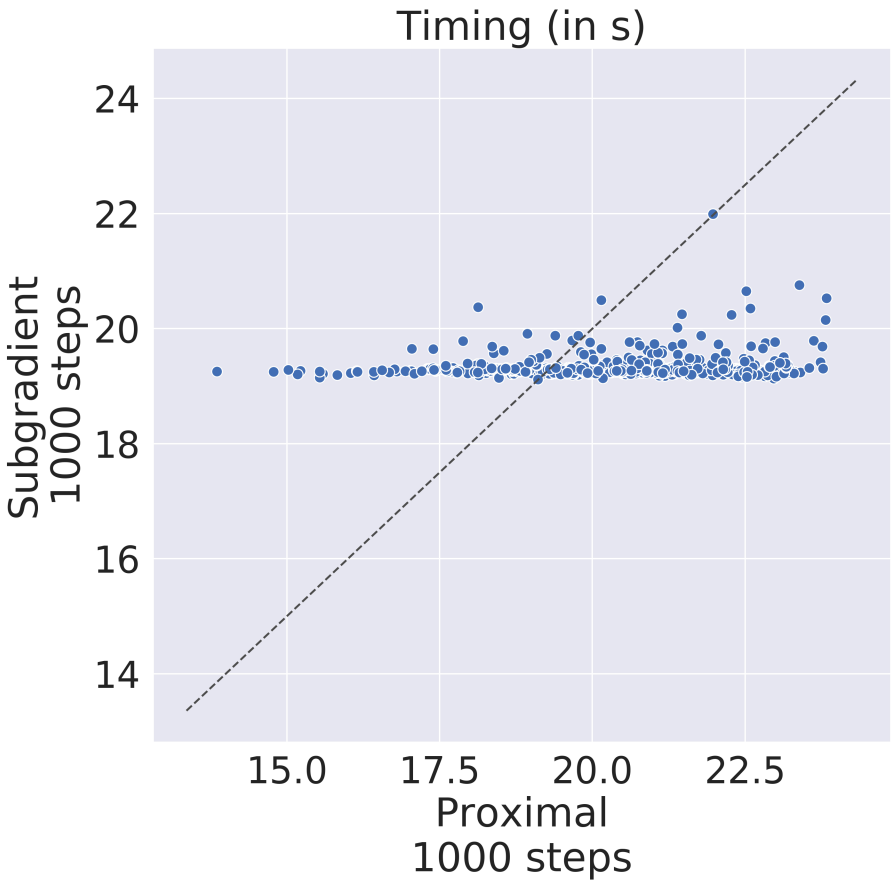

−40
−50
−60
−70
−80
−90
−100
−110
Value
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Time
Method
Subgradient initial step size: 3e-2
Subgradient initial step size: 1e-2
Subgradient initial step size: 3e-3
Subgradient initial step size: 1e-3
Subgradient initial step size: 3e-4
Subgradient initial step size: 1e-4
Proximal initial eta: 1e4
Proximal initial eta: 3e3
Proximal initial eta: 1e3
Proximal initial eta: 3e2
Proximal initial eta: 1e2
Metric
Proximal
Subgradient

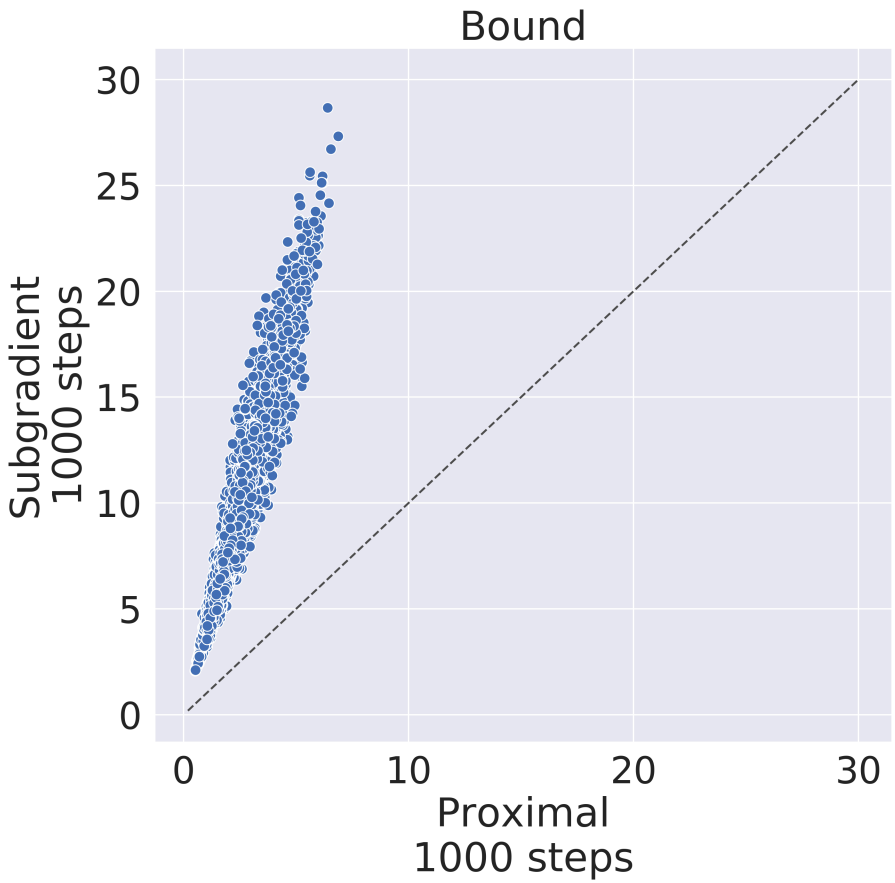
Results


Timing (in s)
Subgradient
1000 steps
Proximal
500 steps
22
20
18
16
14
12
10
8
6
10
15
20

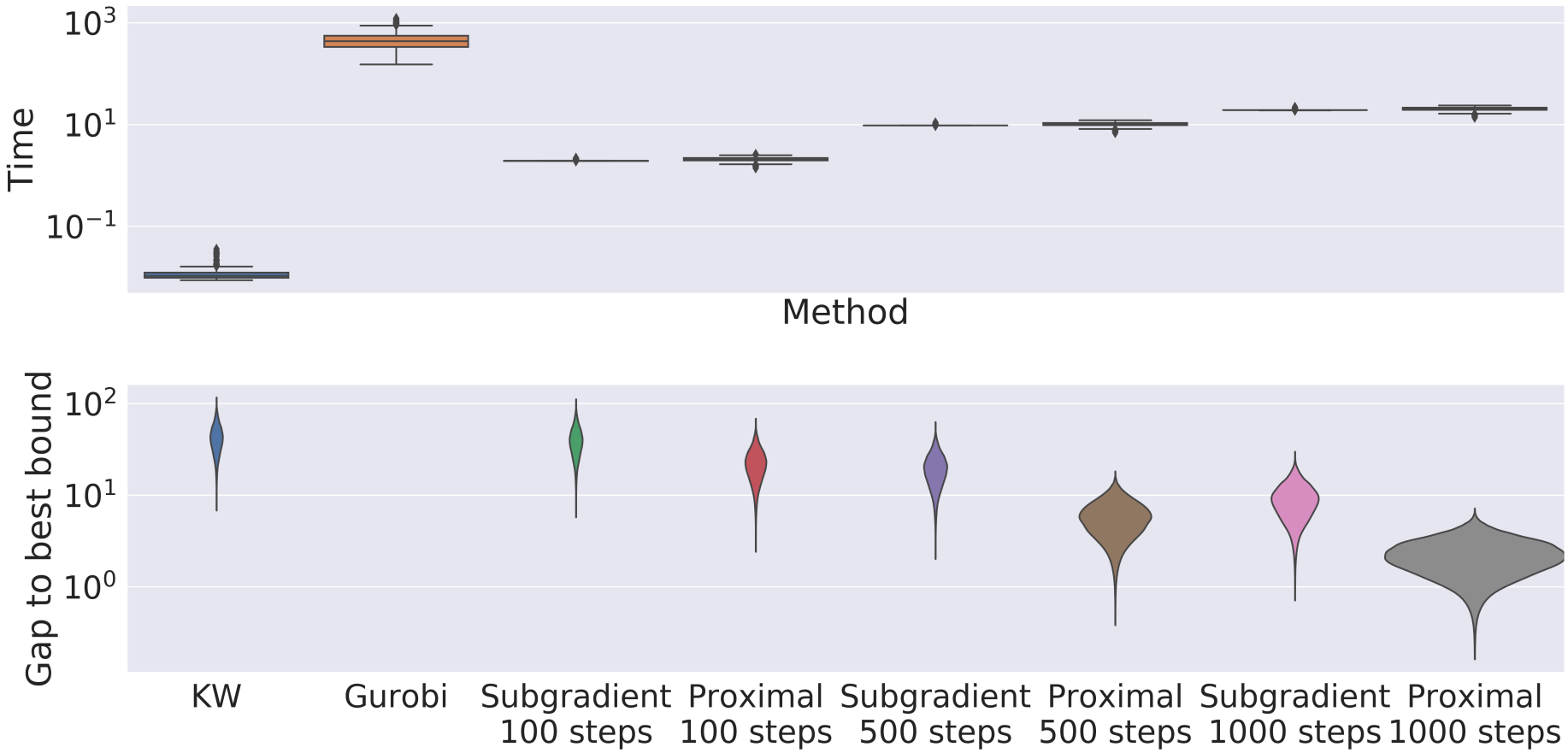

Bound
Subgradient
1000 steps
Proximal
500 steps
30
25
20
15
10
5
0
0
10
20
30

Results


Timing (in s)
Subgradient
1000 steps
24
22
20
18
16
14
15.0
17.5
20.0
22.5
Proximal
1000 steps


Bound
Subgradient
1000 steps
30
25
20
15
10
5
0
0
10
20
30
Proximal
1000 steps

Results


Time
10^3
10^1
10^{-1}
Method
Gap to best bound
10^2
10^1
10^0
KW
Gurobi
Subgradient 100 steps
Proximal 100 steps
Subgradient 500 steps
Proximal 500 steps
Subgradient 1000 steps
Proximal 1000 steps

Questions?

Paper + Code + Data Available*

http://www.robots.ox.ac.uk/~oval