

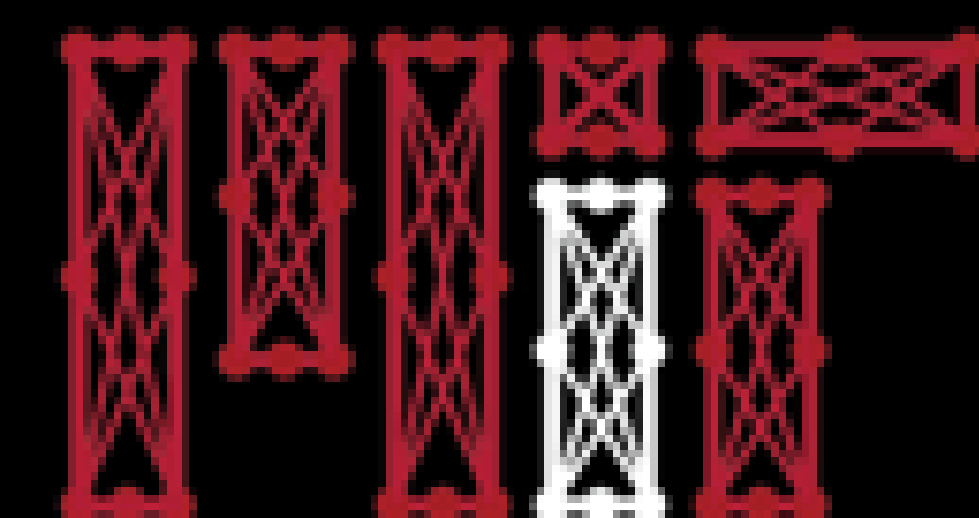


# Limitations and New Frontiers

Ava Soleimany

MIT 6.S191

January 29, 2020



6.S191 Introduction to Deep Learning

[introtodeeplearning.com](http://introtodeeplearning.com) [@MITDeepLearning](https://twitter.com/MITDeepLearning)



# T-shirts! Today!



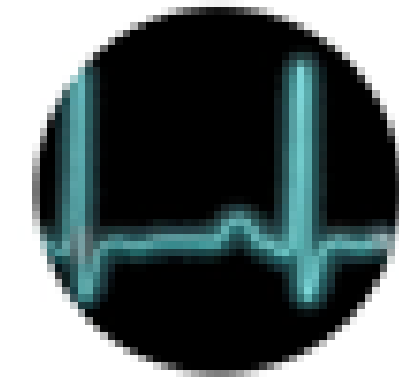
# Lecture Schedule



## Intro to Deep Learning

### Lecture 1

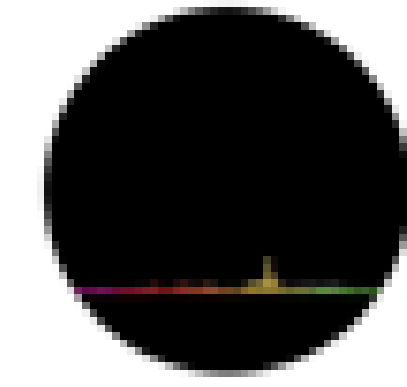
[Slides] [Video] *coming soon!*



## Deep Sequence Modeling

### Lecture 2

[Slides] [Video] *coming soon!*



## Intro to Tensorflow; Music Generation

### Lab Session 1

[Code] *coming soon!*



## Deep Computer Vision

### Lecture 3

[Slides] [Video] *coming soon!*



## Deep Generative Modeling

### Lecture 4

[Slides] [Video] *coming soon!*



## De-biasing Facial Recognition Systems

### Lab Session 2

[Code] [[Paper](#)] *coming soon!*



## Deep Reinforcement Learning

### Lecture 5

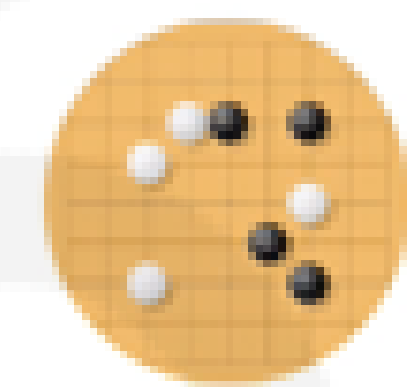
[Slides] [Video] *coming soon!*



## Limitations and New Frontiers

### Lecture 6

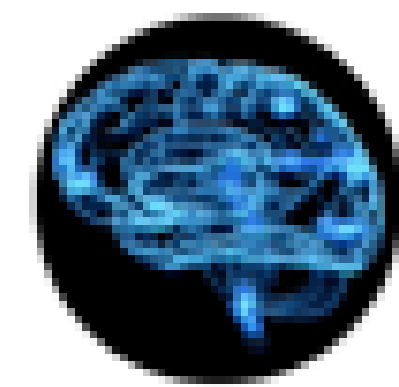
[Slides] [Video] *coming soon!*



## Pixels-to-Control Learning

### Lab Session 3

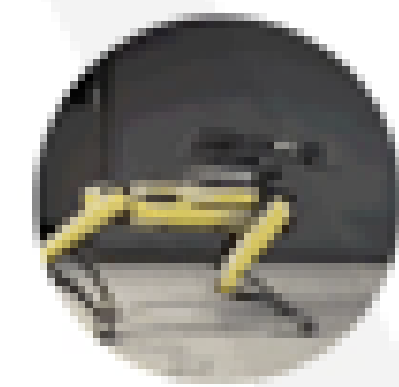
[Code] *coming soon!*



## Guest Lecture

### Lecture 7

[[Info](#)] [Slides] [Video] *coming soon!*



## Robot Learning

### Lecture 8

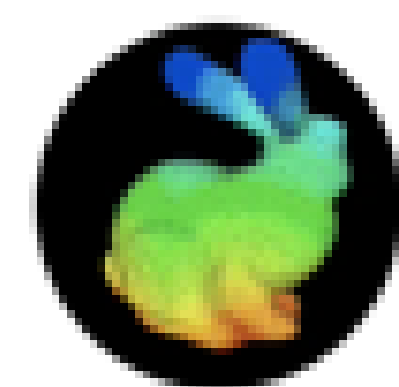
[[Info](#)] [Slides] [Video] *coming soon!*



## Final Projects

### Lab Session 4

[Video] *coming soon!*



## Neural Rendering

### Lecture 9

[[Info](#)] [Slides] [Video] *coming soon!*



## ML for Scent

### Lecture 10

[[Info](#)] [Slides] [Video] *coming soon!*



## Final Projects and Awards Ceremony

### Lab Session 5

[Video] *coming soon!*

- Mon Jan 27 – Fri Jan 31
- 1:00 pm – 4:00pm, 32-123
- Lecture + Lab Breakdown
- Graded P/D/F; 3 Units
- 1 Final Assignment
- Lab submissions: Thursday 1/30, 5pm

# Final Class Project

## Option 1: Proposal Presentation

- At least 1 registered student to be prize eligible
- Present a novel deep learning research idea or application
- 3 minutes (strict)
- Presentations on **Friday, Jan 31**
- Submit groups by **Wednesday 11:59pm** to be eligible
- Submit slide by **Thursday 11:59pm** to be eligible
- Instructions: [shorturl.at/wxBK7](https://shorturl.at/wxBK7)

- Judged by a panel of judges
- Top winners are awarded:



3x NVIDIA 2080 Ti (\$4000)



4x Google Home (\$400)



3x Display Monitors (\$300)



3x SSD 1TB (\$200)



# Final Class Project

## Option 1: Proposal Presentation

- At least 1+ registered student to be prize eligible
- Present a novel deep learning research idea or application
- 3 minutes (strict)
- Presentations on **Friday, Jan 31**
- Submit groups by **Wednesday 11:59pm** to be eligible
- Submit slide by **Thursday 11:59pm** to be eligible
- Instructions: [shorturl.at/wxBK7](https://shorturl.at/wxBK7)

## Proposal Logistics

- Prepare slides on Google Slides
- **Group submit by today 11:59pm:**  
[shorturl.at/mxBWZ](https://shorturl.at/mxBWZ)
- In class project work: **Thu, Jan 30**
- **Slide submit by Thu 11:59 pm:**  
[shorturl.at/pqCL9](https://shorturl.at/pqCL9)
- Presentations on **Friday, Jan 31**

# Final Class Project

## Option 1: Proposal Presentation

- At least 1+ registered student to be prize eligible
- Present a novel deep learning research idea or application
- 3 minutes (strict)
- Presentations on **Friday, Jan 31**
- Submit groups by **Wednesday 11:59pm** to be eligible
- Submit slide by **Thursday 11:59pm** to be eligible
- Instructions: [shorturl.at/wxBK7](https://shorturl.at/wxBK7)

## Option 2: Write a 1-page review of a deep learning paper

- Grade is based on clarity of writing and technical communication of main ideas
- Due **Friday Jan 31 1:00pm** (before lecture) by email

# Thursday: AI for Human Creativity + Robot Learning



**David Cox,**  
IBM Director,  
MIT-IBM Watson AI Lab  
Towards Robust AI

IBM **Research**



**Animesh Garg,**  
U Toronto,  
**NVIDIA**  
Robot Learning



## Lab + Final Project Work

Ask us questions!

Open office hours!

Work with group members!

# Friday: Neural Rendering + Learning to Smell Project Proposals + Awards!



**Chuan Li,**  
**CSO,**  
**Lambda Labs**  
Neural Rendering



**Alex Wiltschko,**  
**Senior Research Scientist,**  
**Google Brain**  
Machine Learning for Scent



**Project Proposals!**

**Judging and Awards!**

**Pizza Celebration!**

So far in 6.S191...



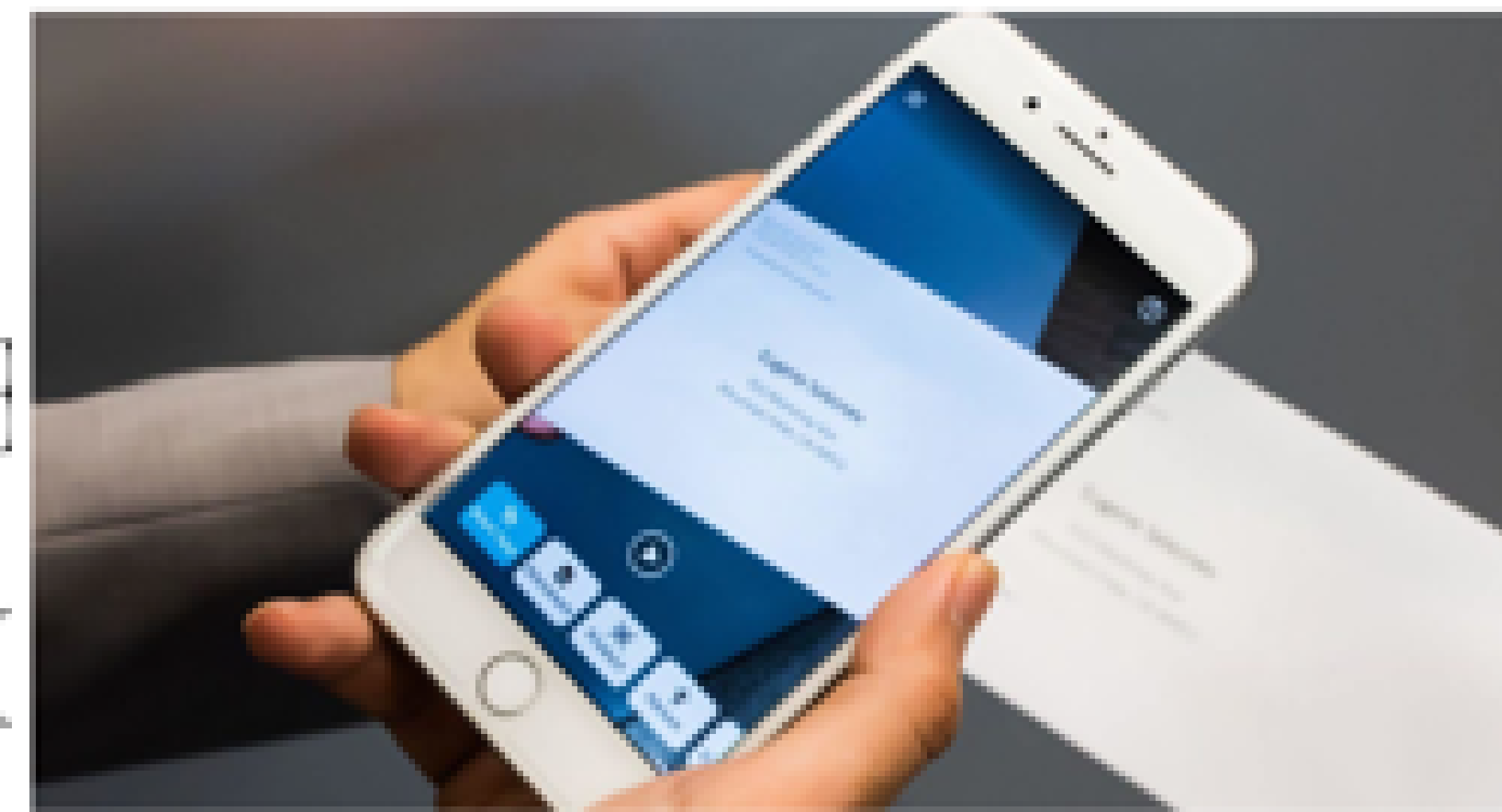
# The Rise of Deep Learning

## 'Deep Voice' Software Can Clone Anyone's Voice With Just 3.7 Seconds of Audio

Using snippets of voices, Baidu's 'Deep Voice' can generate new speech, accents, and tones.



## Let There Be Sight: How Deep Learning Is Helping the Blind 'See'



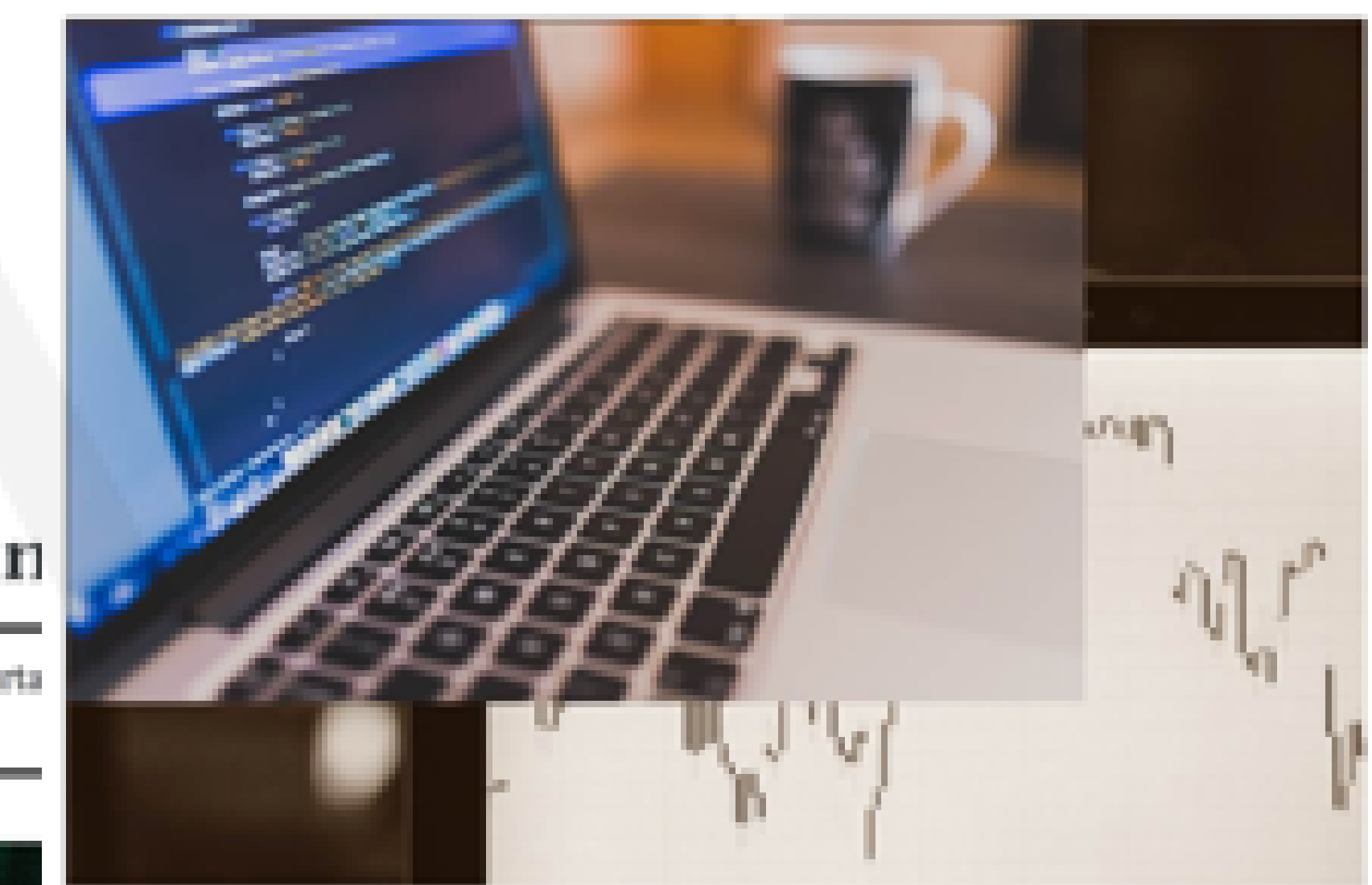
## Technology outpacing security measures

Facial Recognition | Features and Interviews

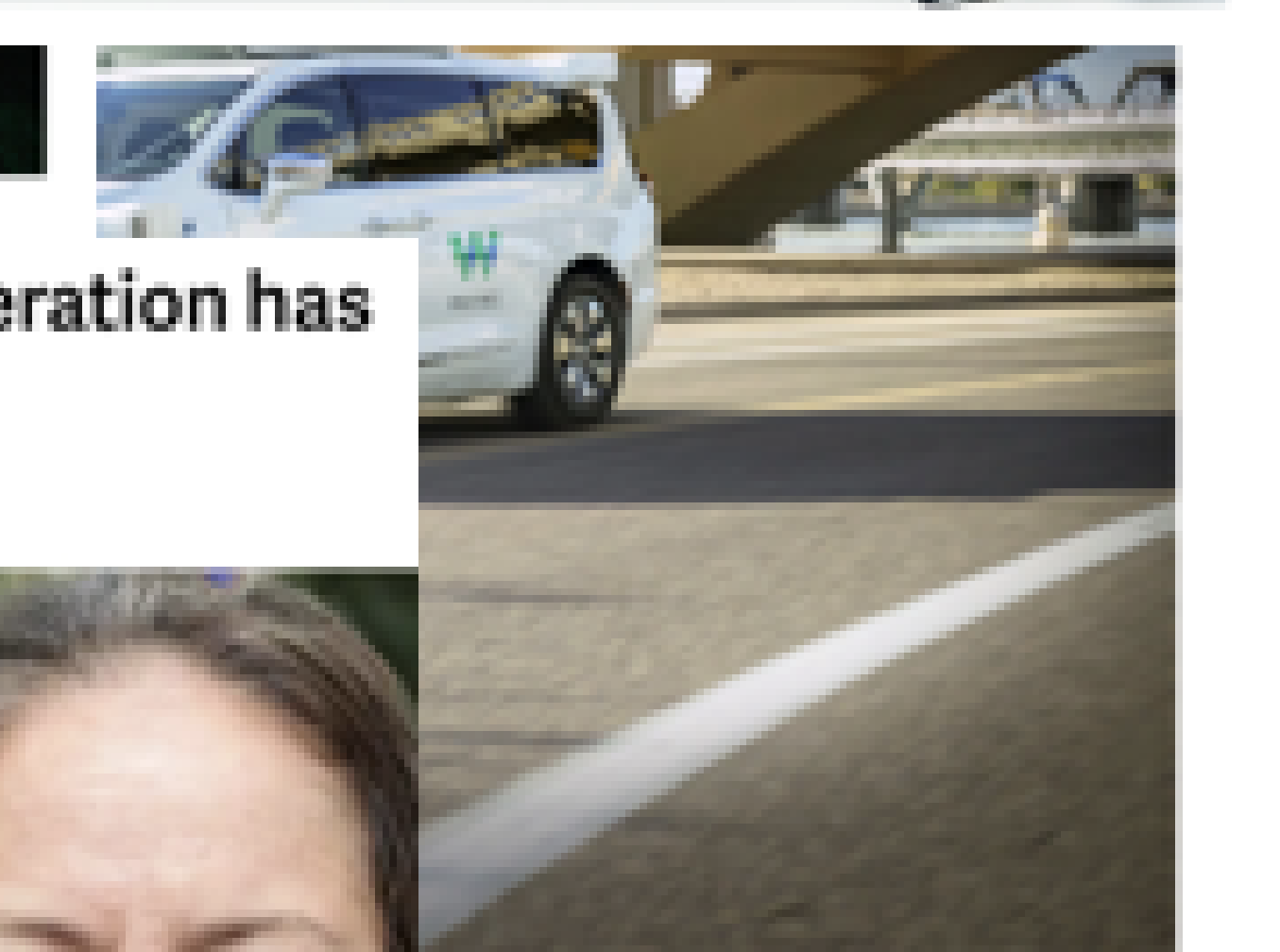
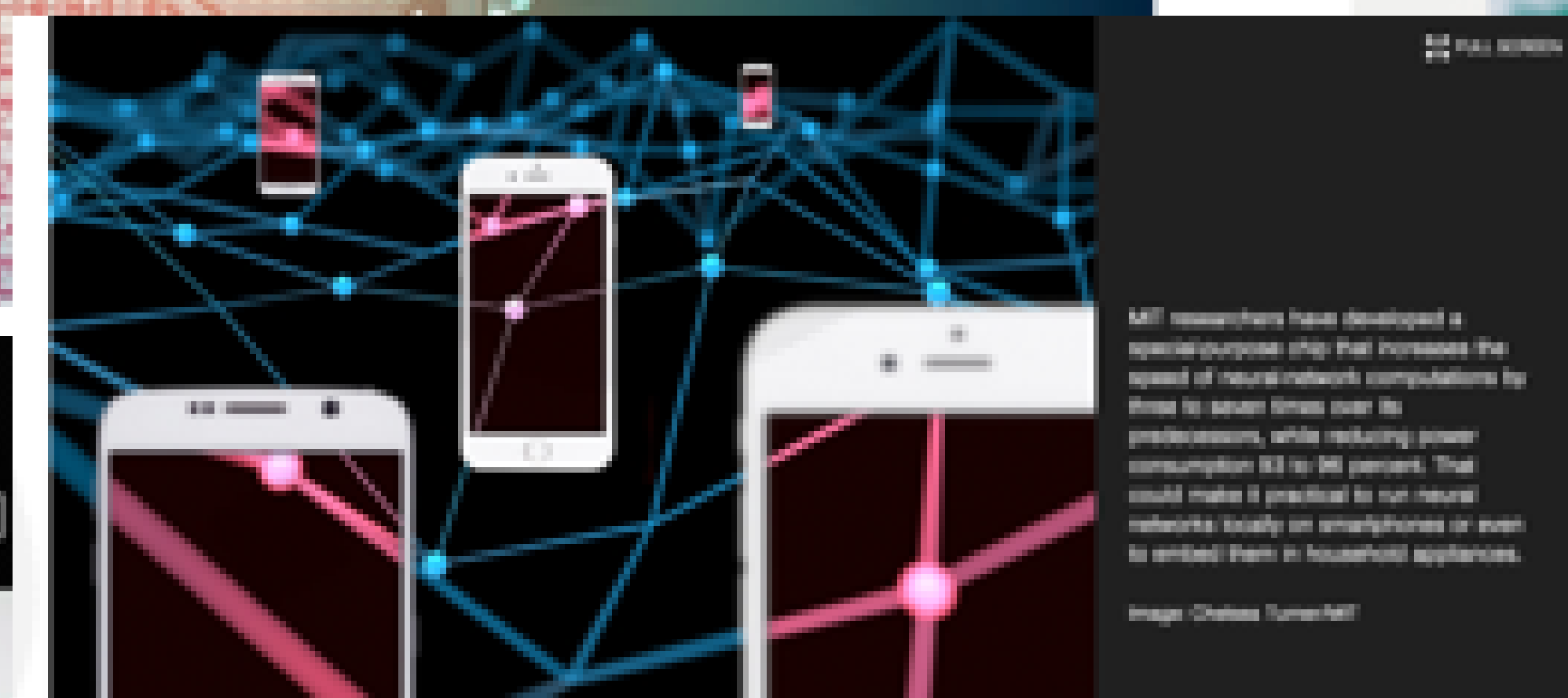
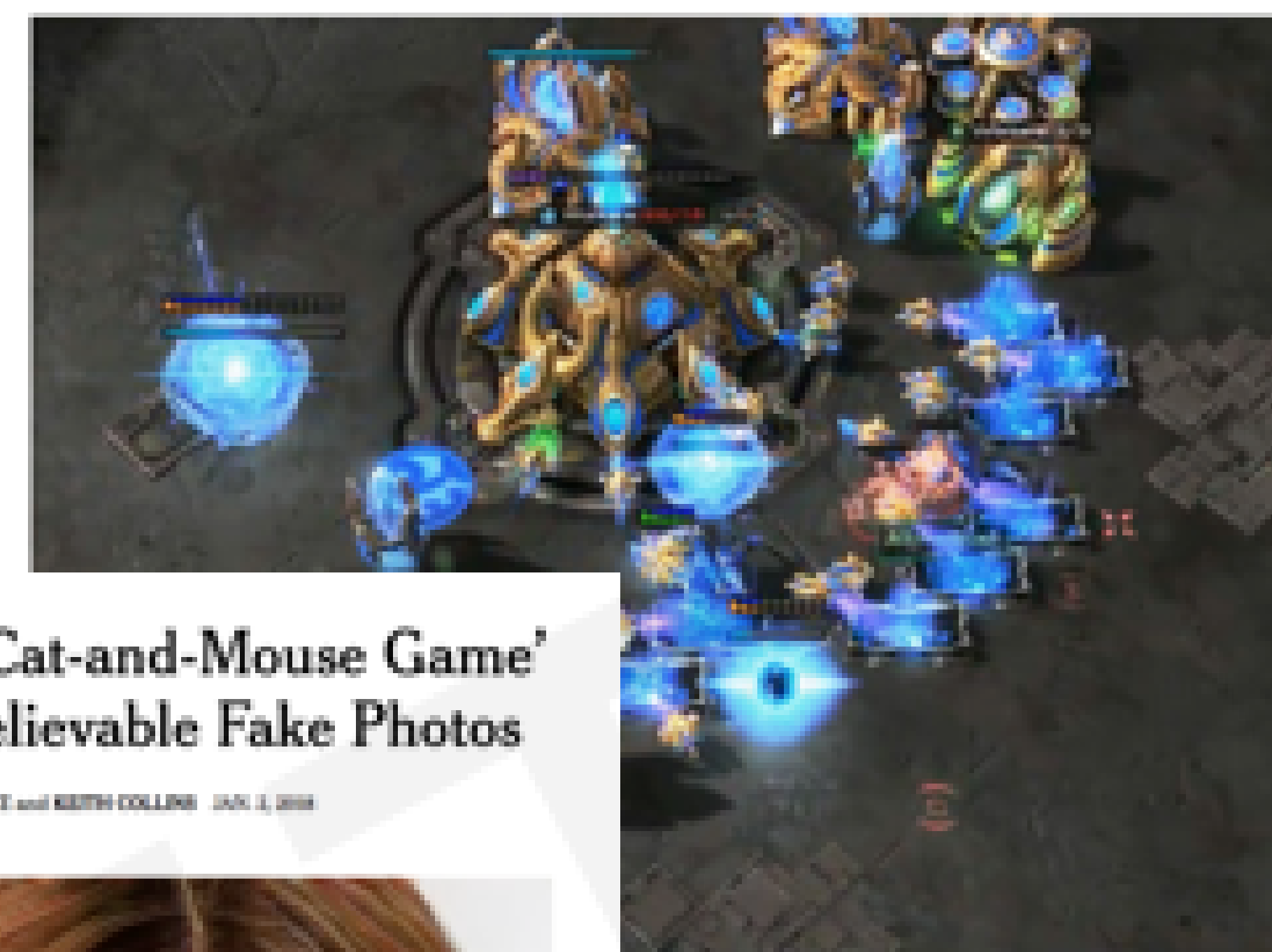
## AI beats docs in cancer spotting

A new study provides a fresh example of machine learning as an important diagnostic tool. Paul Biegler reports.

## AI Can Help In Predicting Cryptocurrency Value

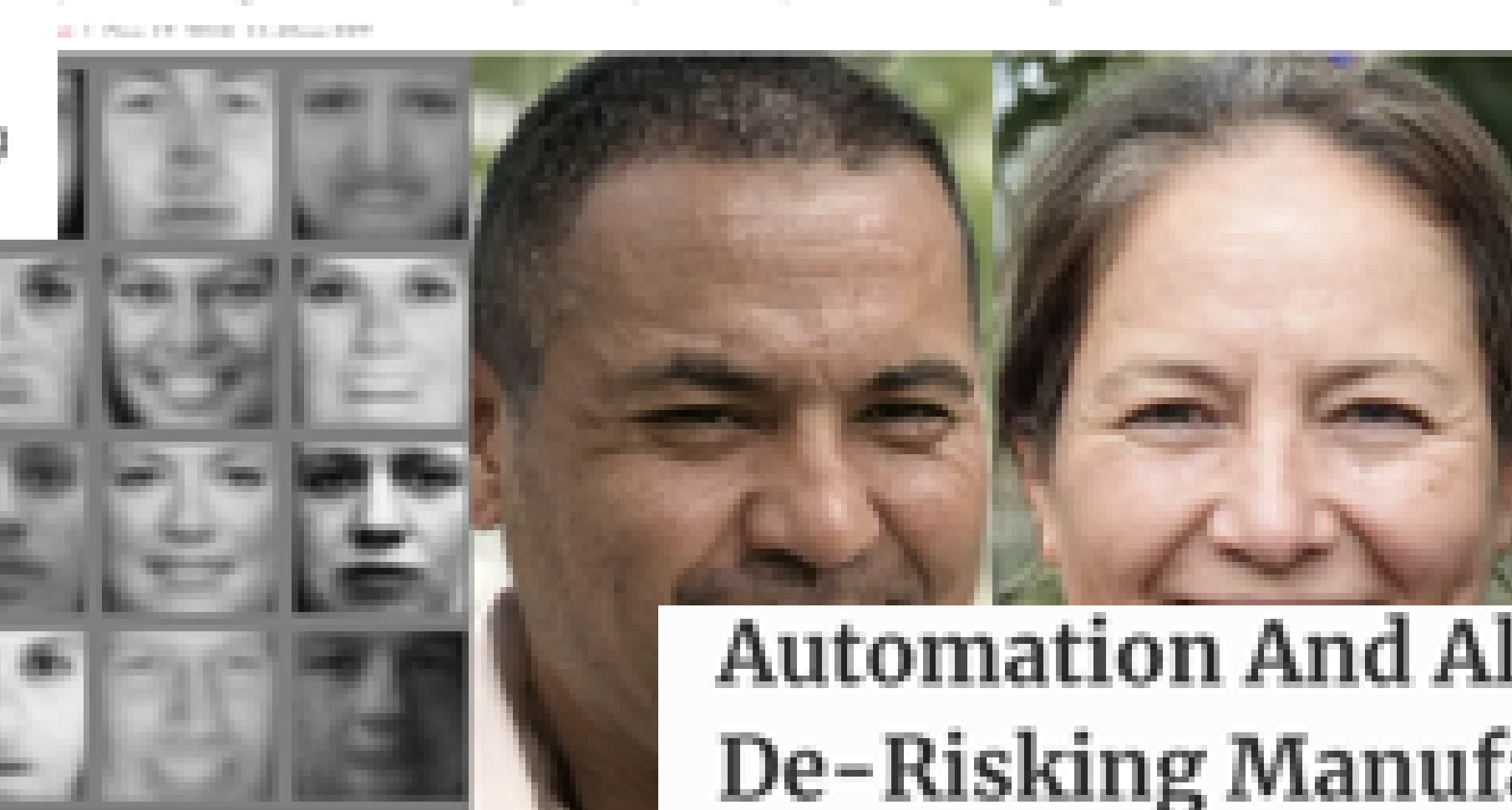


## DEEPMIND'S STARCRRAFT TRIUMPH



## AI faces show how far AI image generation has come in just four years

The faces on the right aren't real; they're the product of machine learning



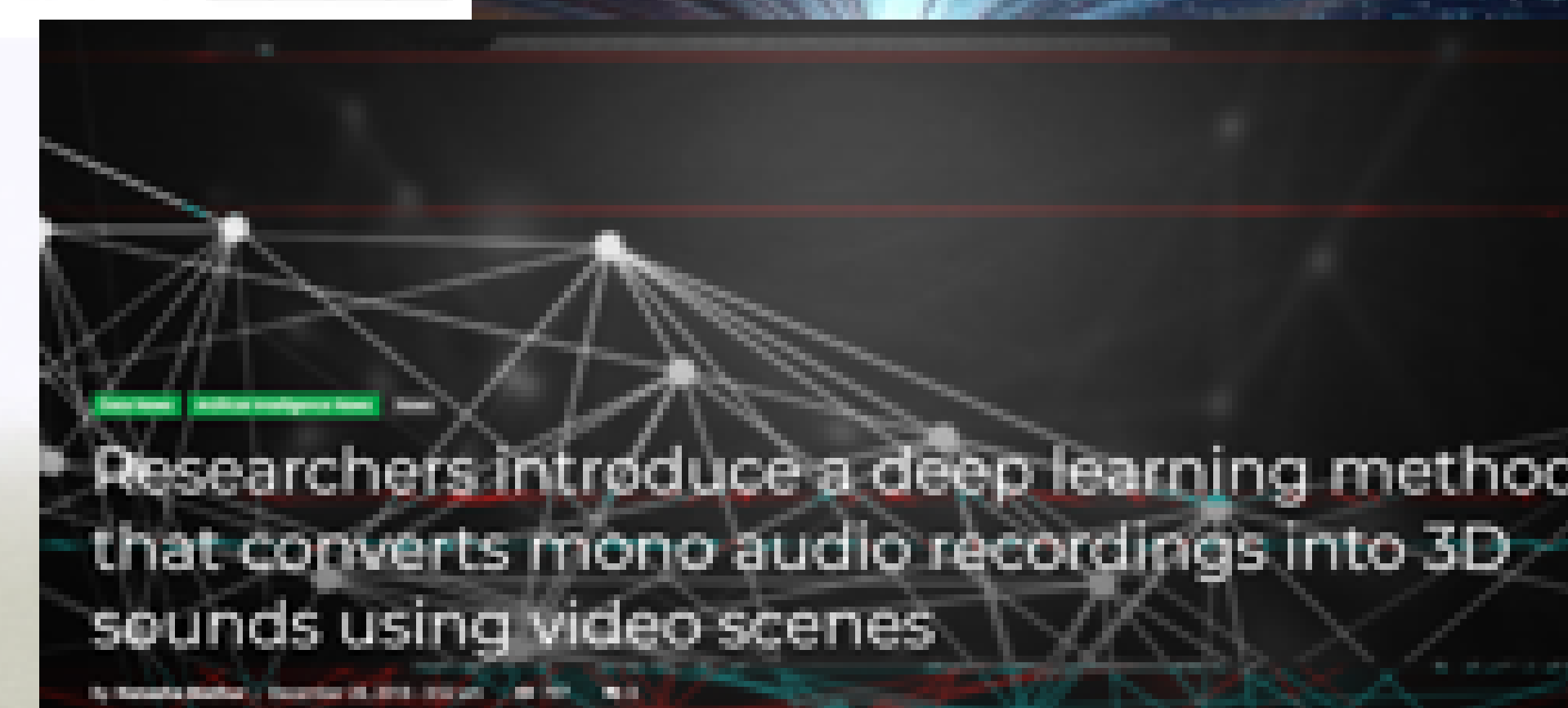
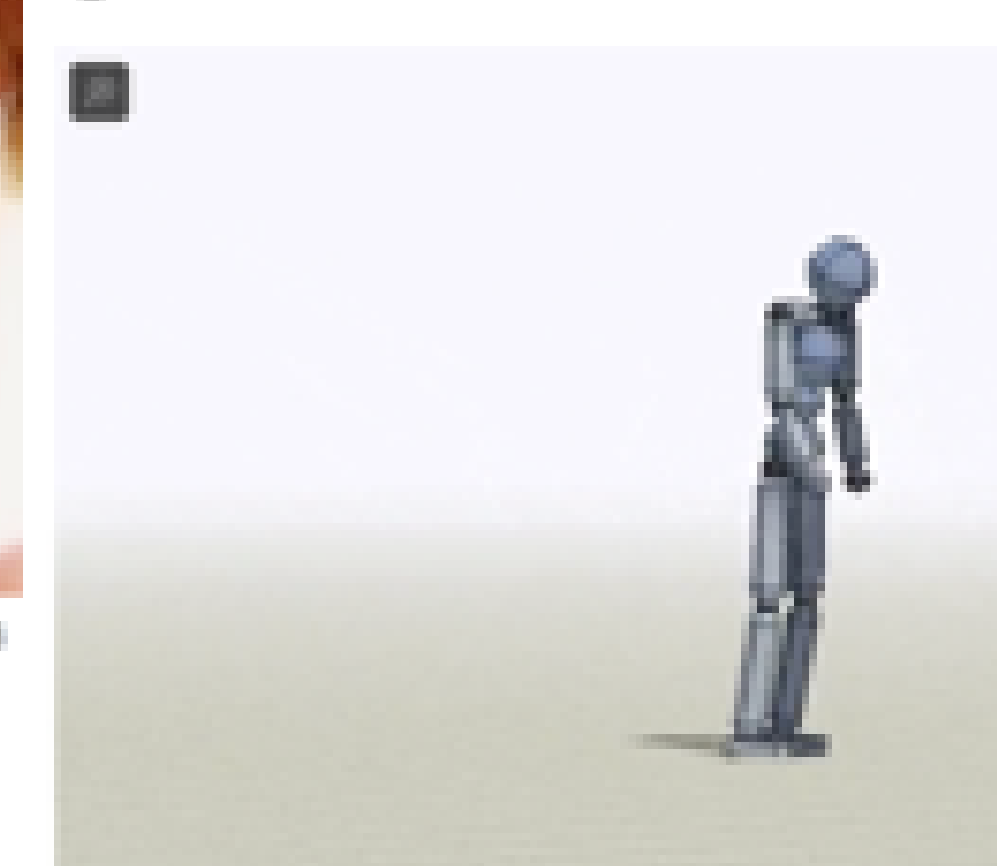
## Neural networks everywhere

New chip reduces neural networks' power consumption by up to 95 percent, making them practical for battery-powered devices.

Deep L | Wed, 01/16/2019 - 8:00am | Comment by Kenny Walter - Digital Reporter - @RandMagazine

## After Millions of Trials, These Simulated Humans Learned to Do Perfect Backflips and Cartwheels

George Siu | 4/16/18 | 10:00am - Filed to ML



Researchers introduce a deep learning method that converts mono audio recordings into 3D sounds using video scenes

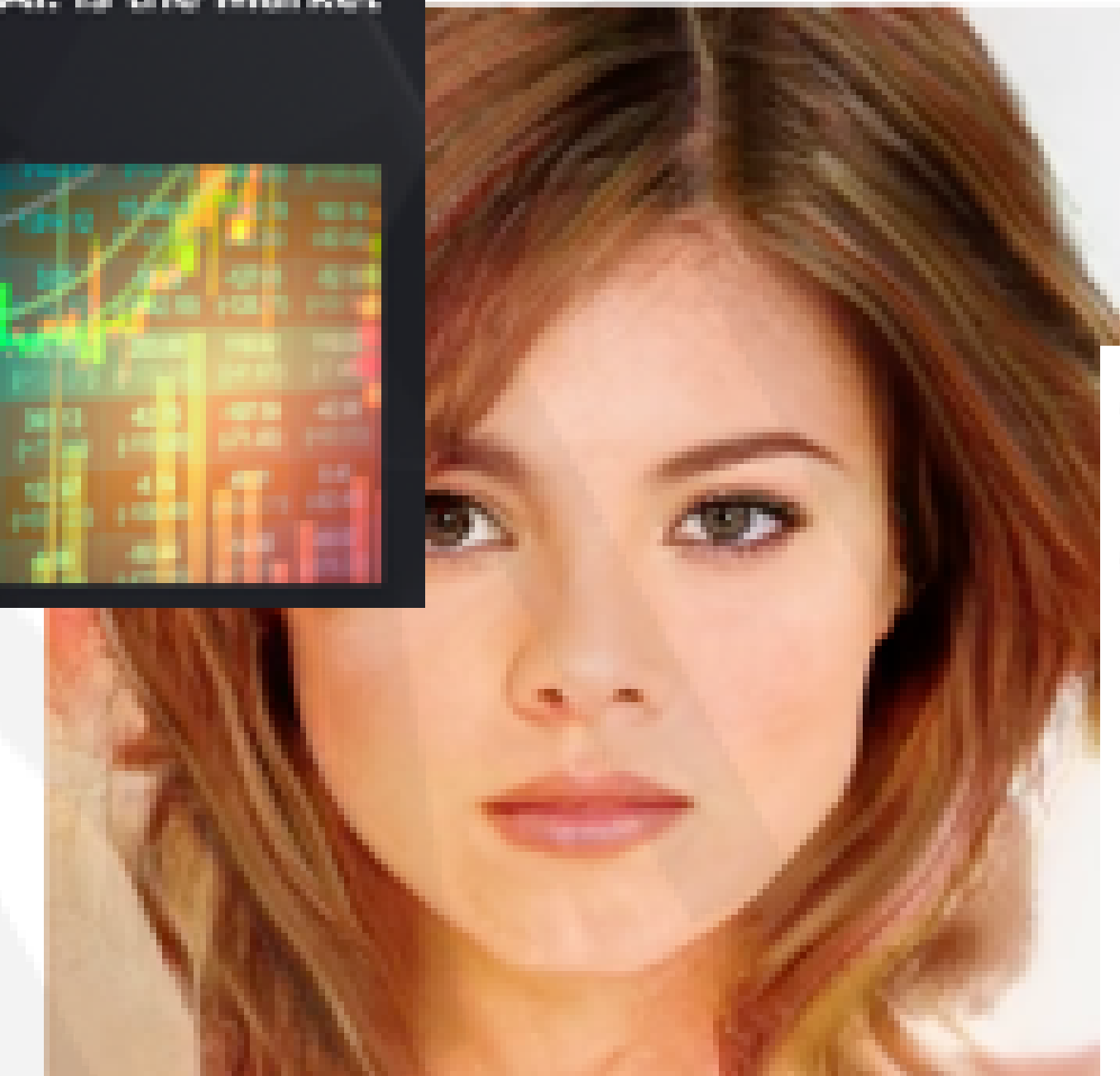
## 'Creative' AlphaZero leads way for chess computers and, maybe, science

Former chess world champion Garry Kasparov likes what he sees of computer that could be used to find cures for diseases



## How an A.I. 'Cat-and-Mouse Game' Generates Believable Fake Photos

By CADE METZ and KEITH COLLINS | JAN. 1, 2018



To create the final image in this set, the system generated 10 million revisions over 18 days.

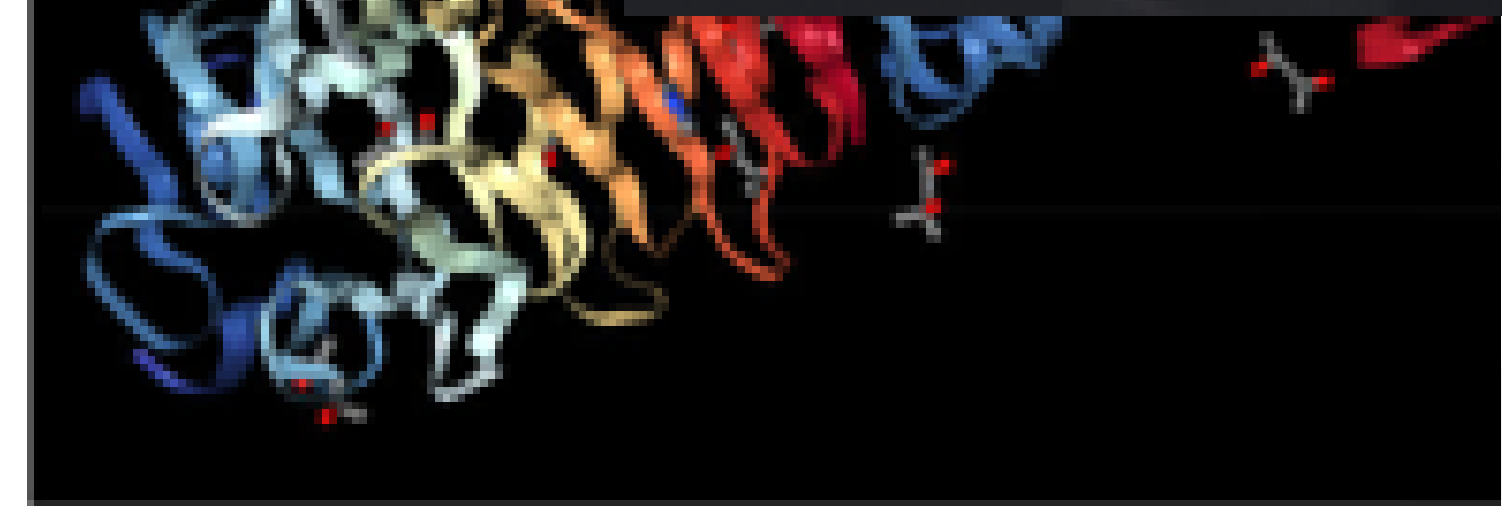
## Stock Predictions Based On AI: Is the Market Truly Predictable?

By Paul Biegler | Oct 10, 2018 | 12:00pm



## Google's DeepMind acs protein folding

By Robert F. Service | Dec. 6, 2018, 12:05 PM

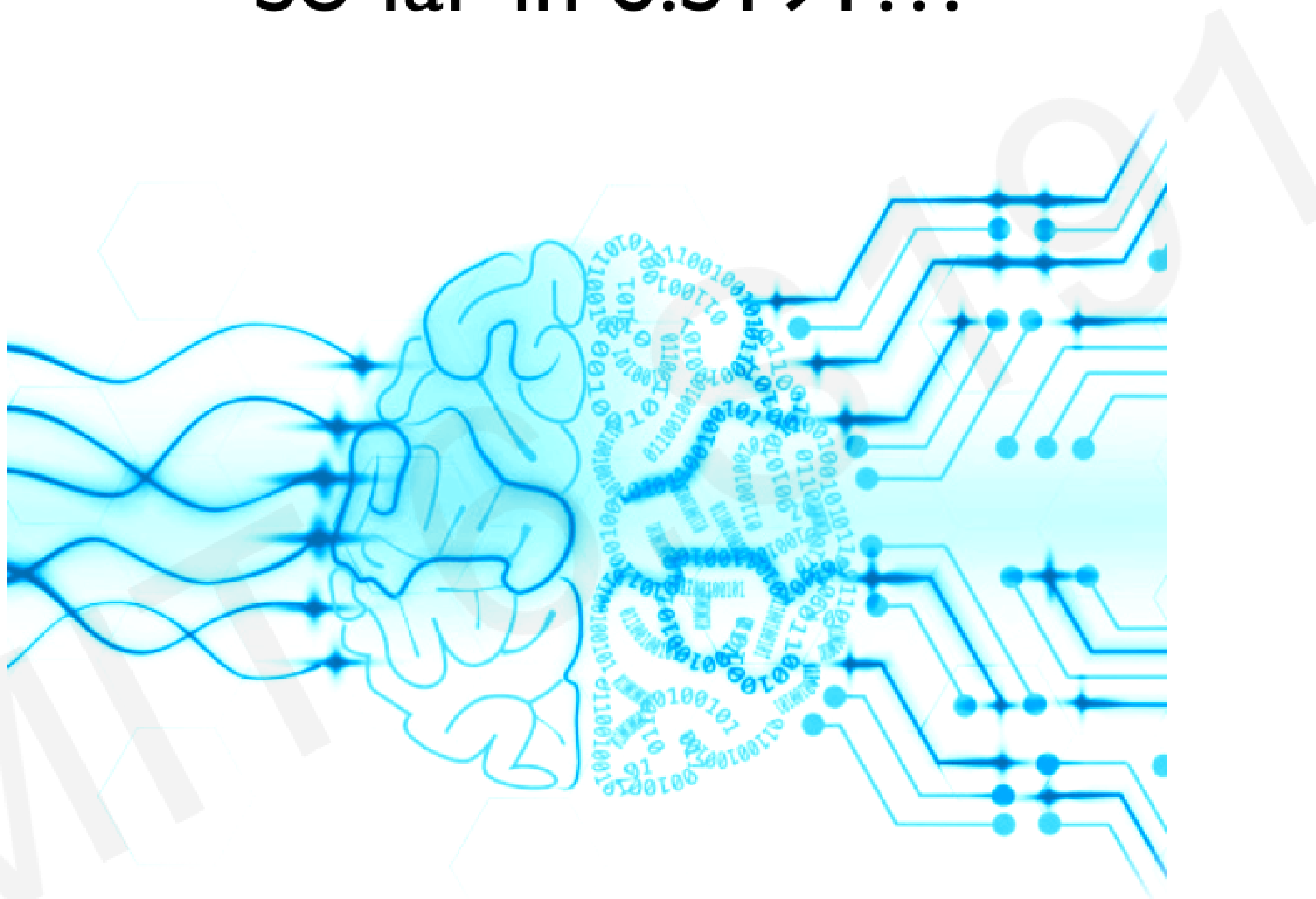


Complex of bacteria-infecting viral proteins modeled in CASP 13. The complex consists of proteins that were modeled individually. PROTEIN DATA BANK

# So far in 6.S191...

## Data

- Signals
- Images
- Sensors
- ...



# So far in 6.S191...

## Data

- Signals
- Images
- Sensors
- ...



## Decision

- Prediction
- Detection
- Action
- ...

# So far in 6.S191...

## Data

- Signals
- Images
- Sensors
- ...



## Decision

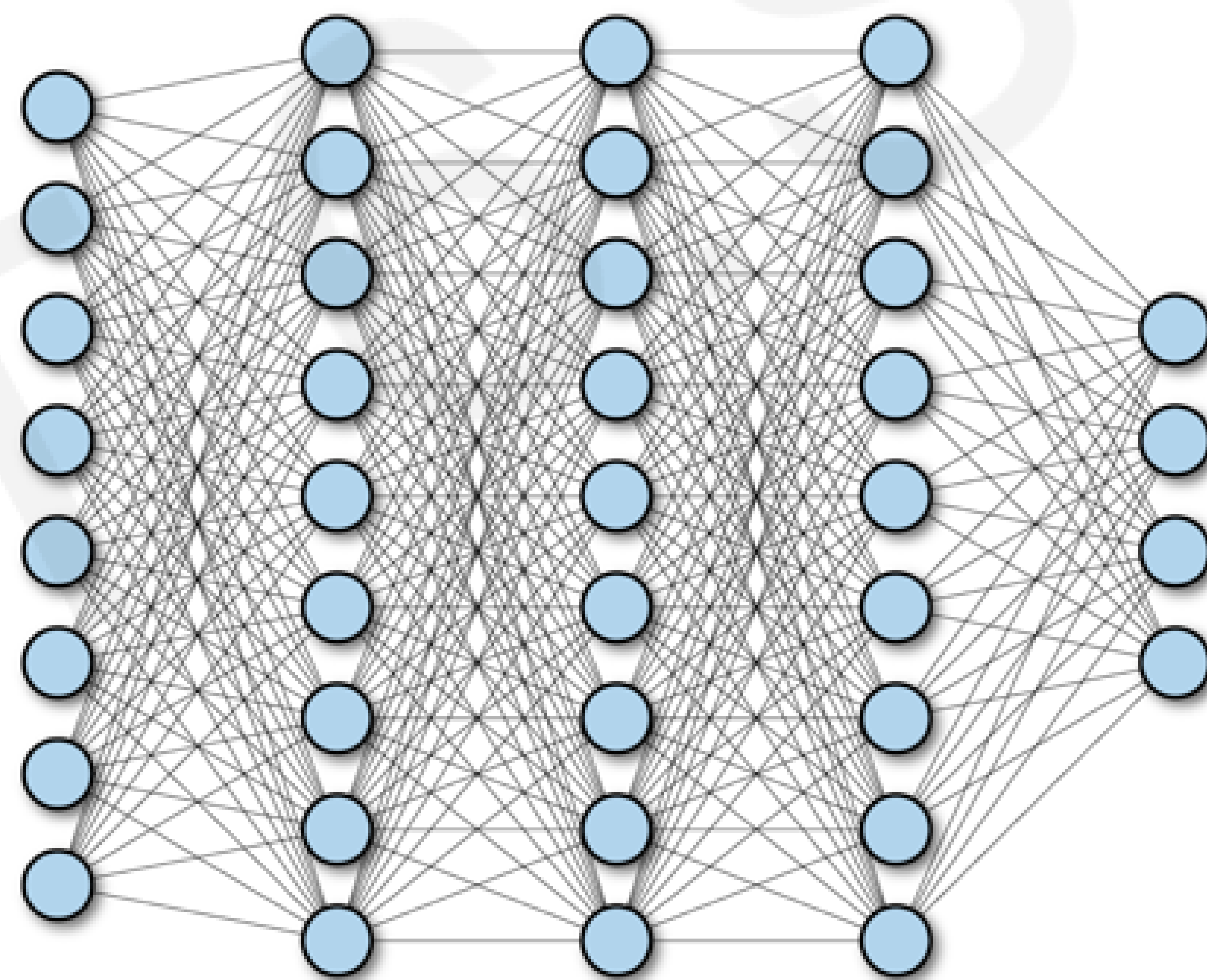
- Prediction
- Detection
- Action
- ...



# Power of Neural Nets

## Universal Approximation Theorem

*A feedforward network with a single layer is sufficient to approximate, to an arbitrary precision, any continuous function.*





# Power of Neural Nets

## Universal Approximation Theorem

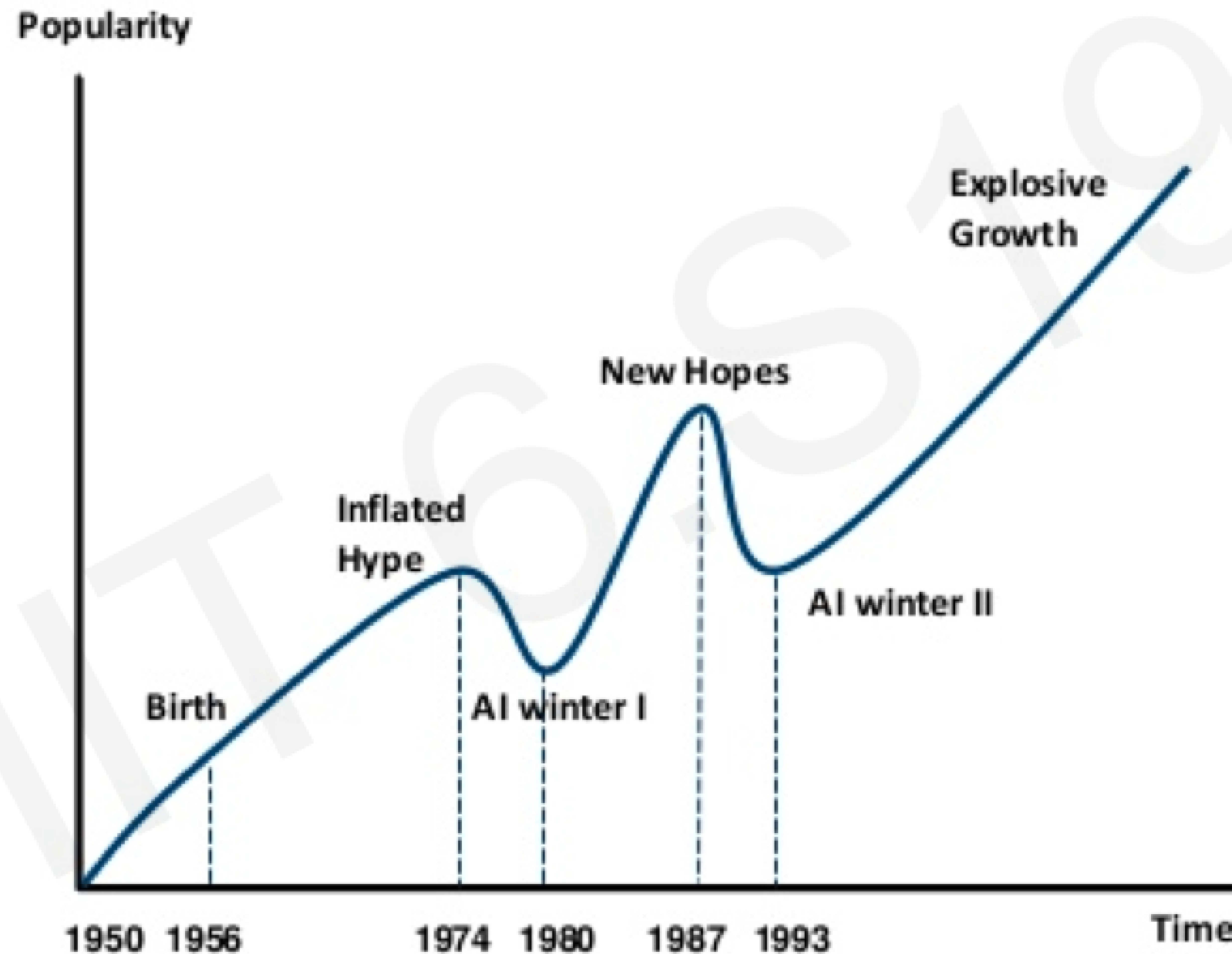
*A feedforward network with a single layer is sufficient to approximate, to an arbitrary precision, any continuous function.*

### Caveats:

*The number of hidden units may be infeasibly large*

*The resulting model may not generalize*

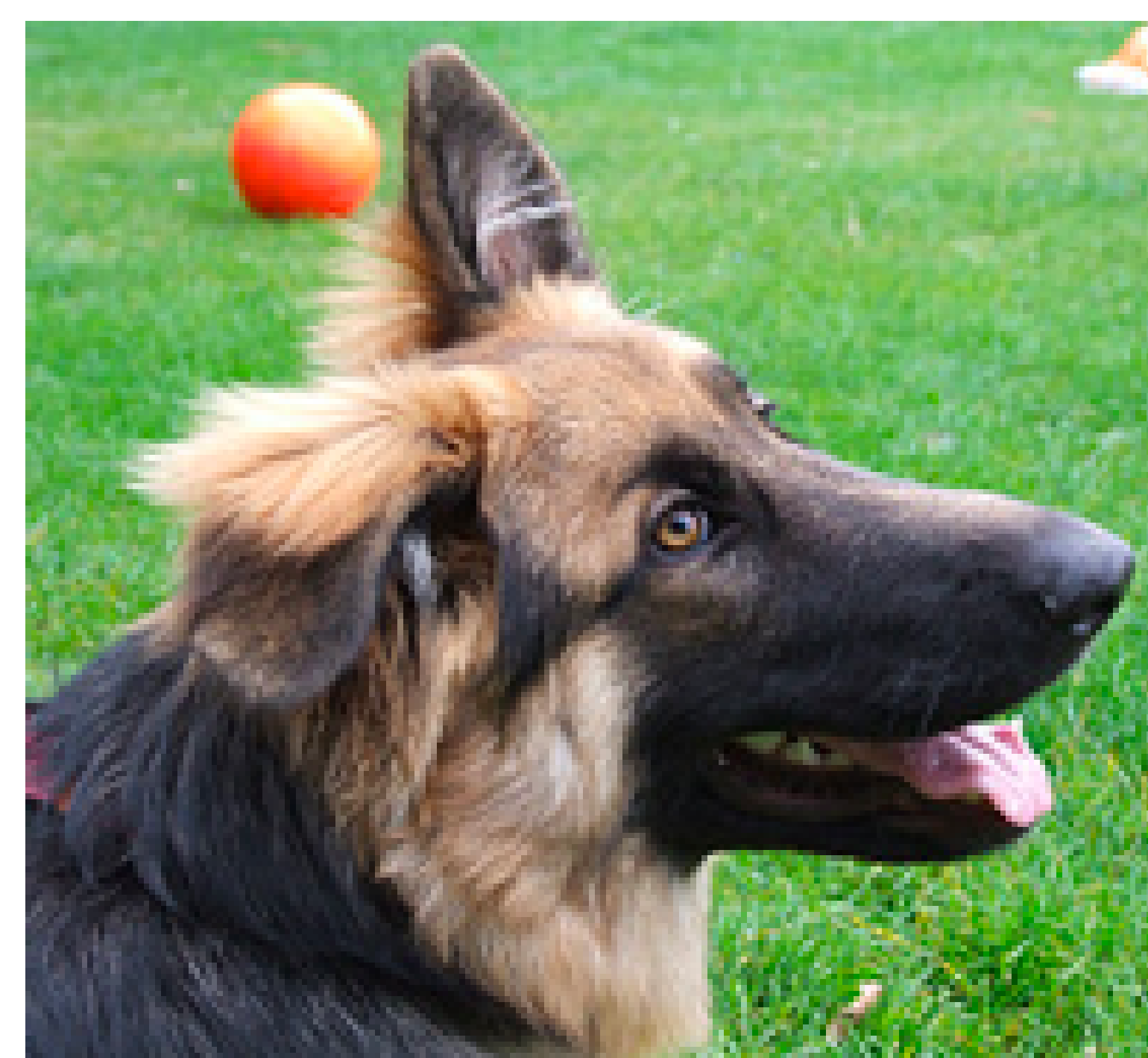
# Artificial Intelligence “Hype”: Historical Perspective



# Limitations

# Rethinking Generalization

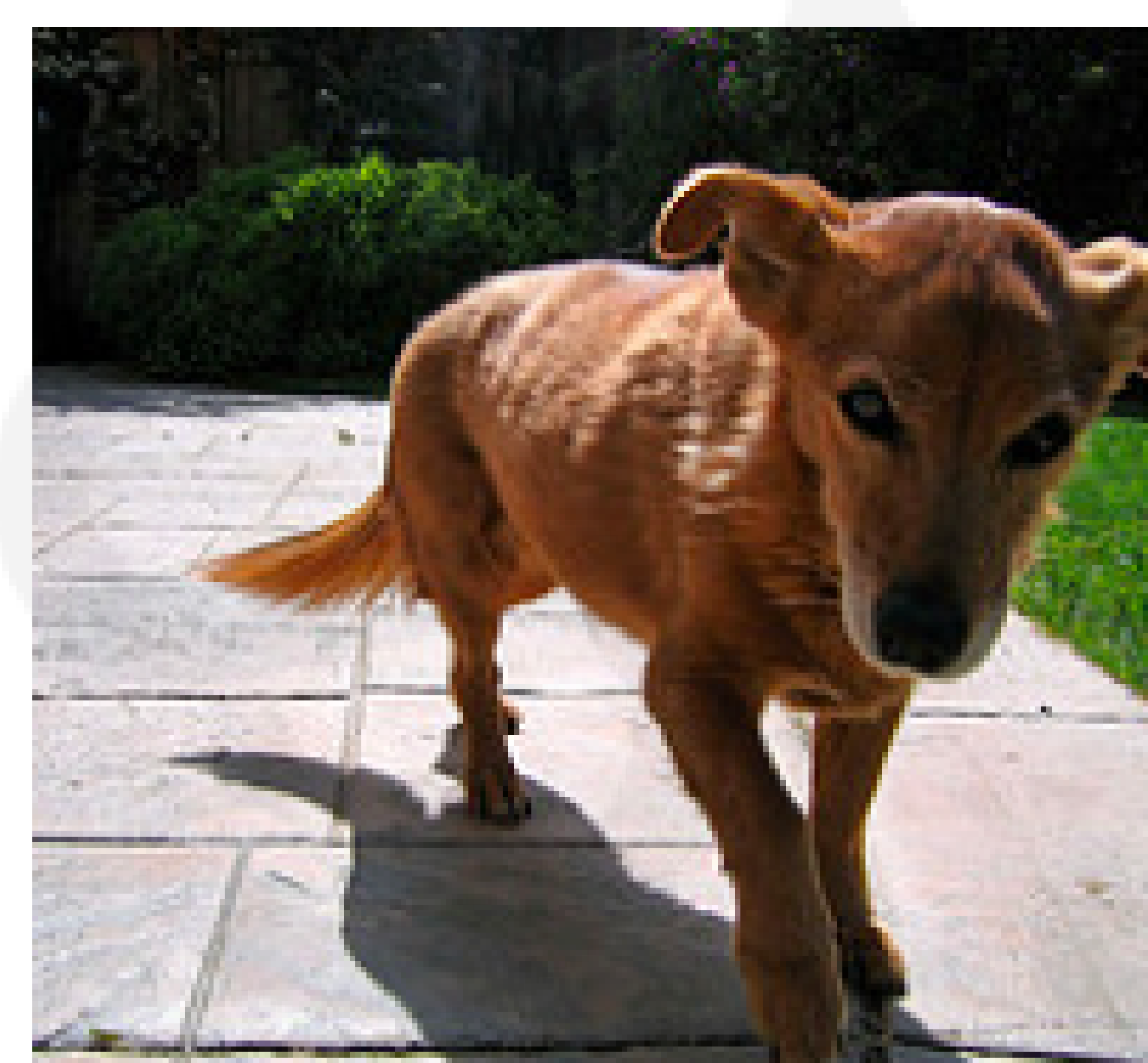
“Understanding Deep Neural Networks Requires Rethinking Generalization”



dog



banana



dog



tree

# Rethinking Generalization

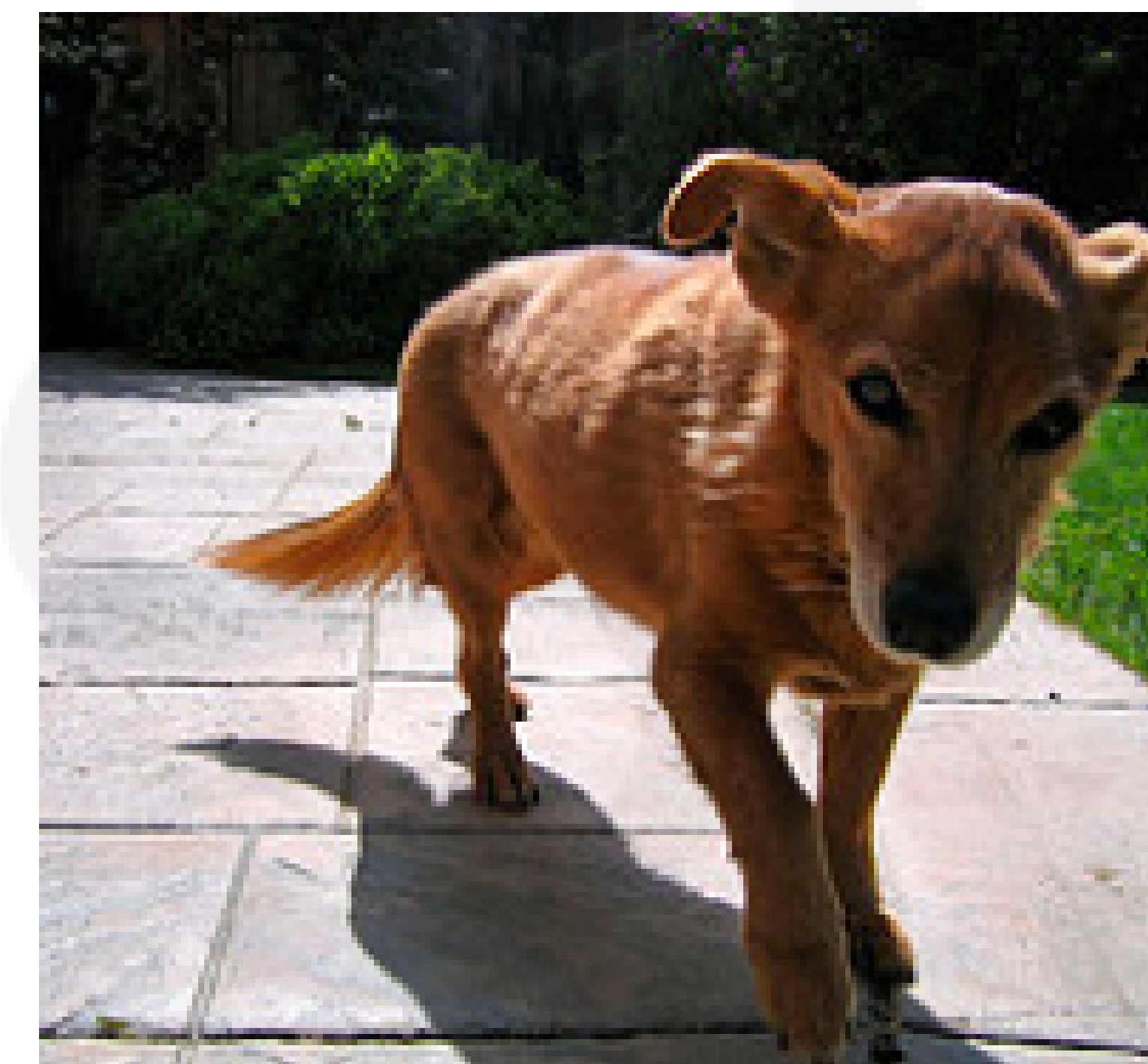
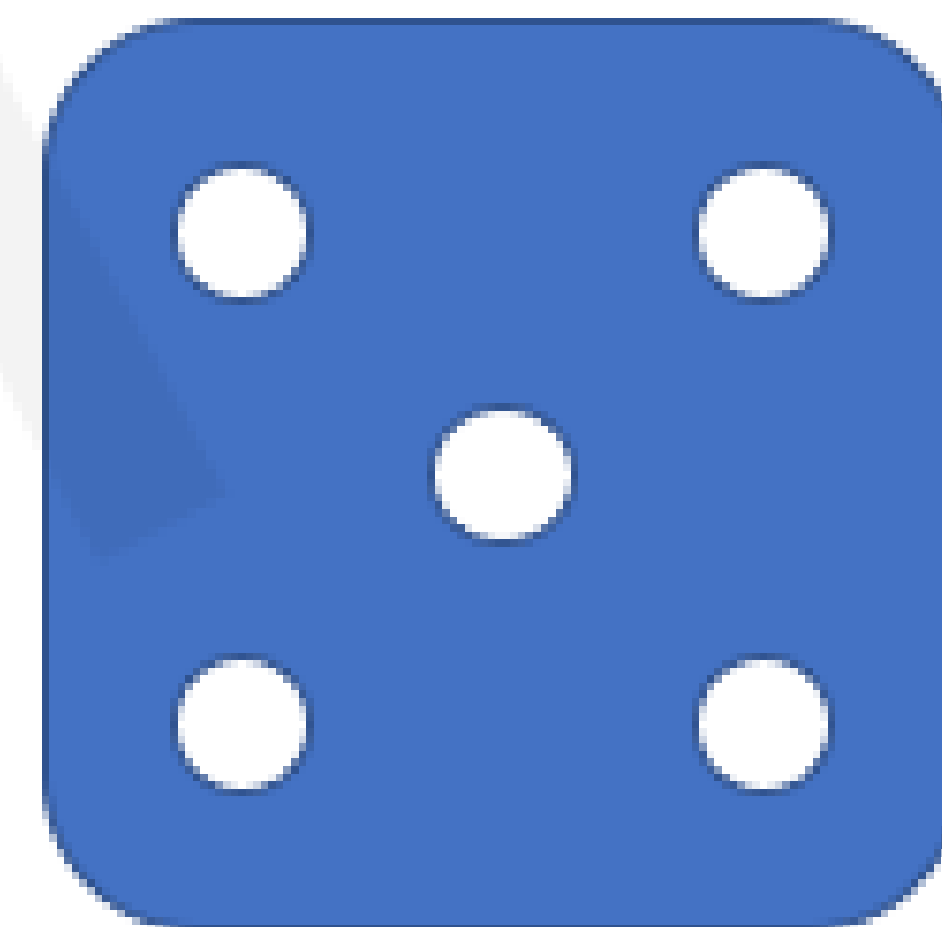
“Understanding Deep Neural Networks Requires Rethinking Generalization”



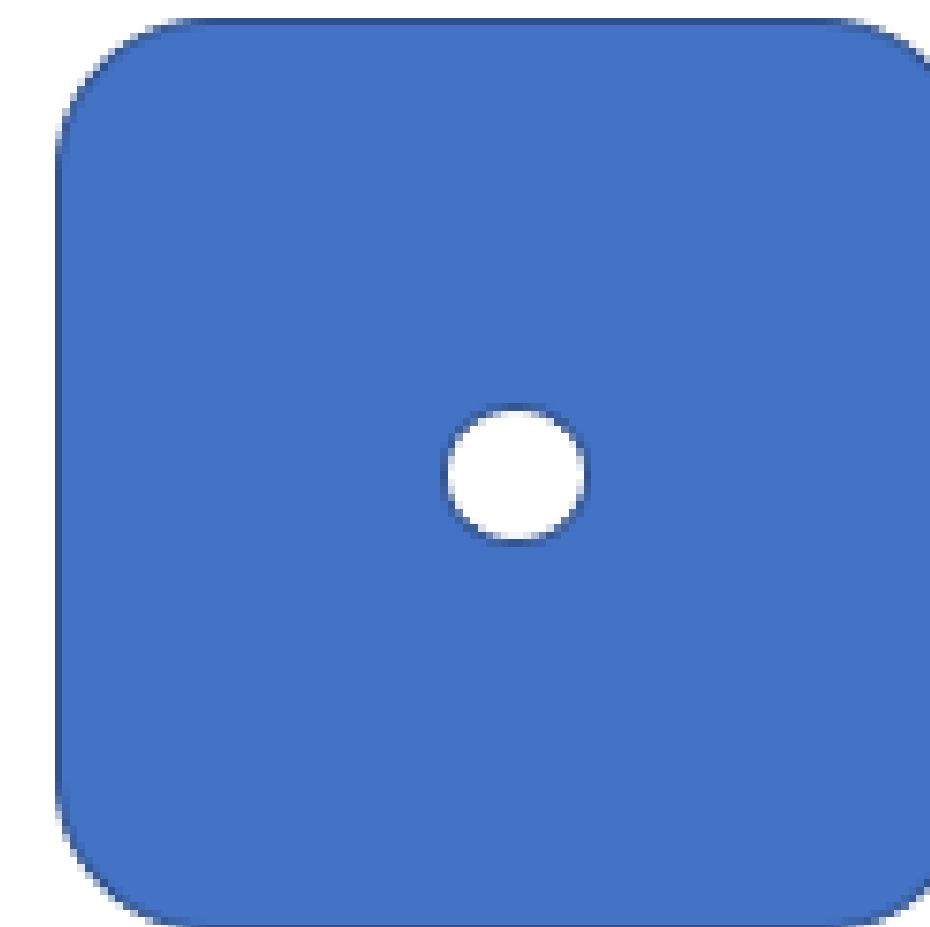
dog



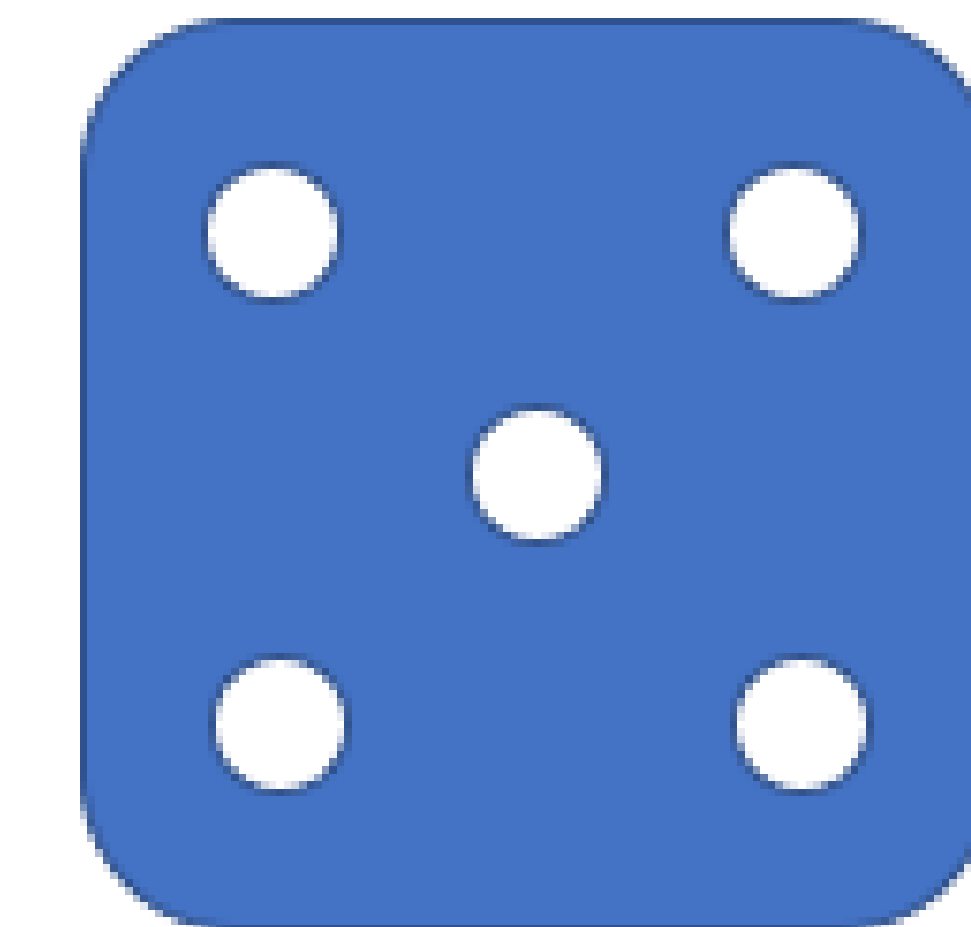
banana



dog



tree





# Rethinking Generalization

“Understanding Deep Neural Networks Requires Rethinking Generalization”



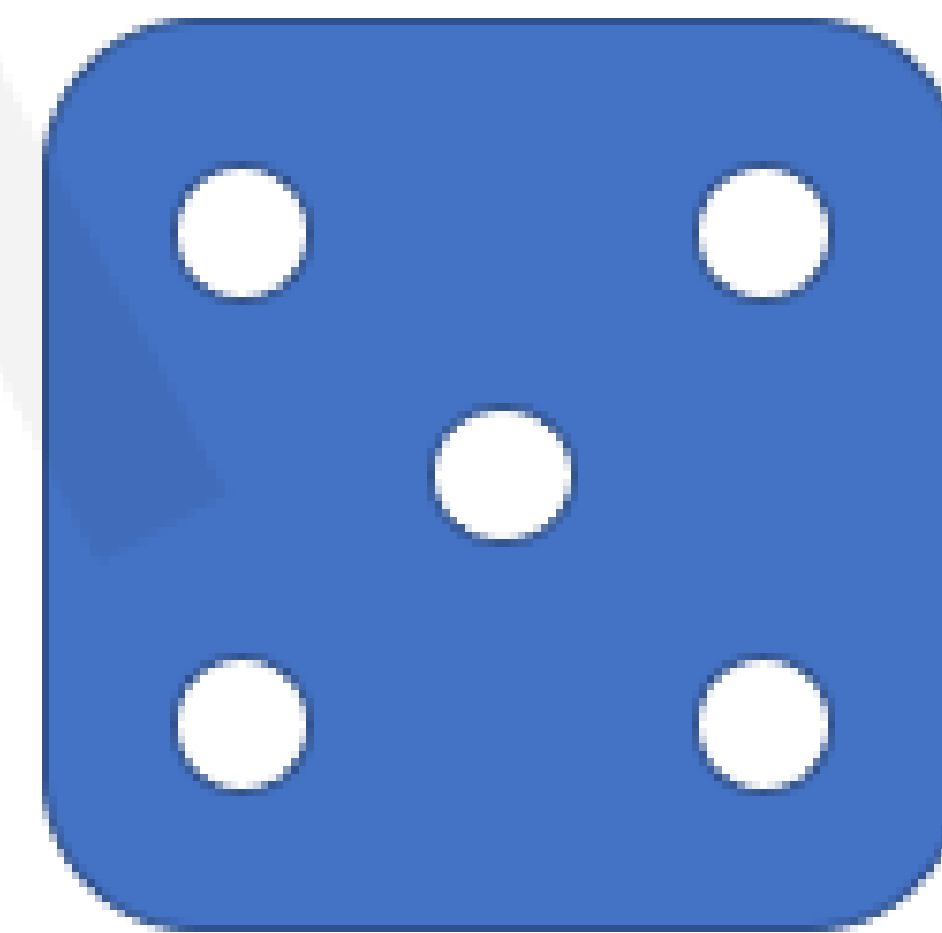
dog



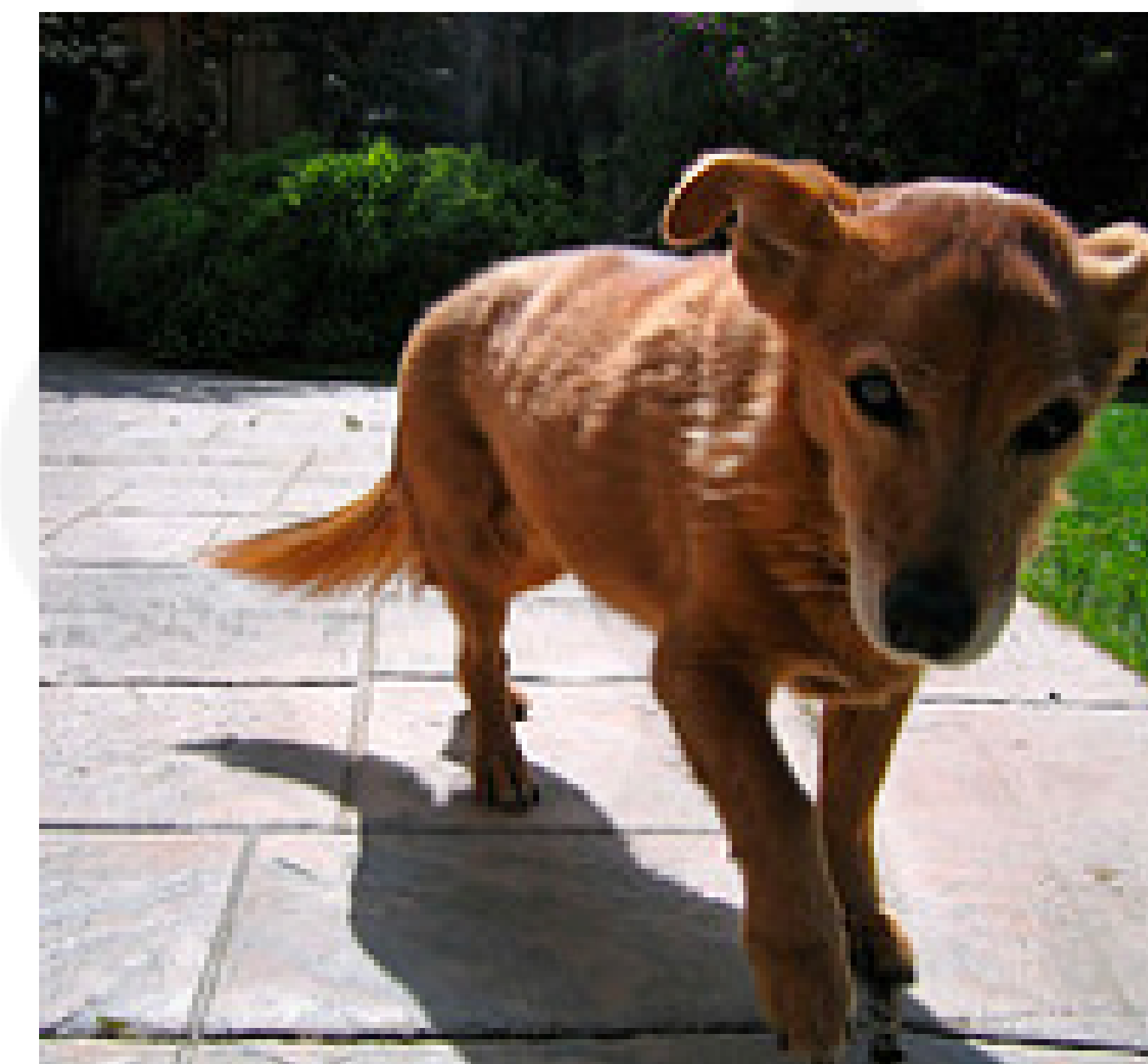
banana



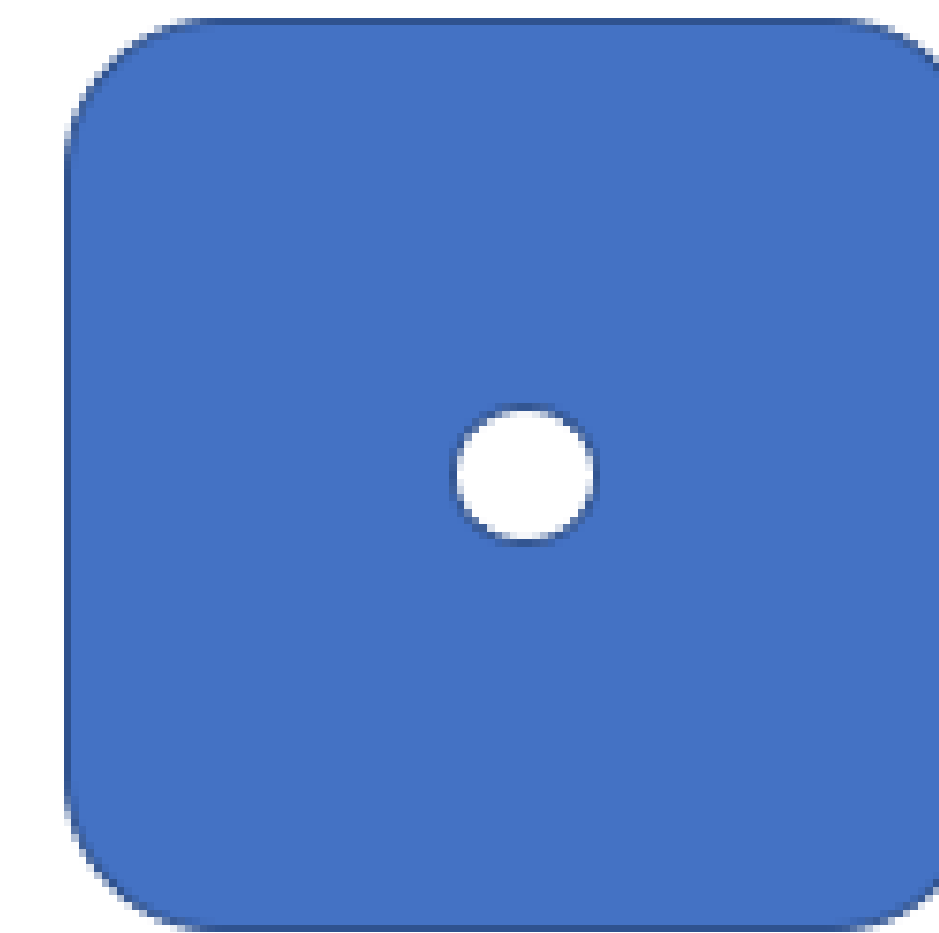
banana



dog



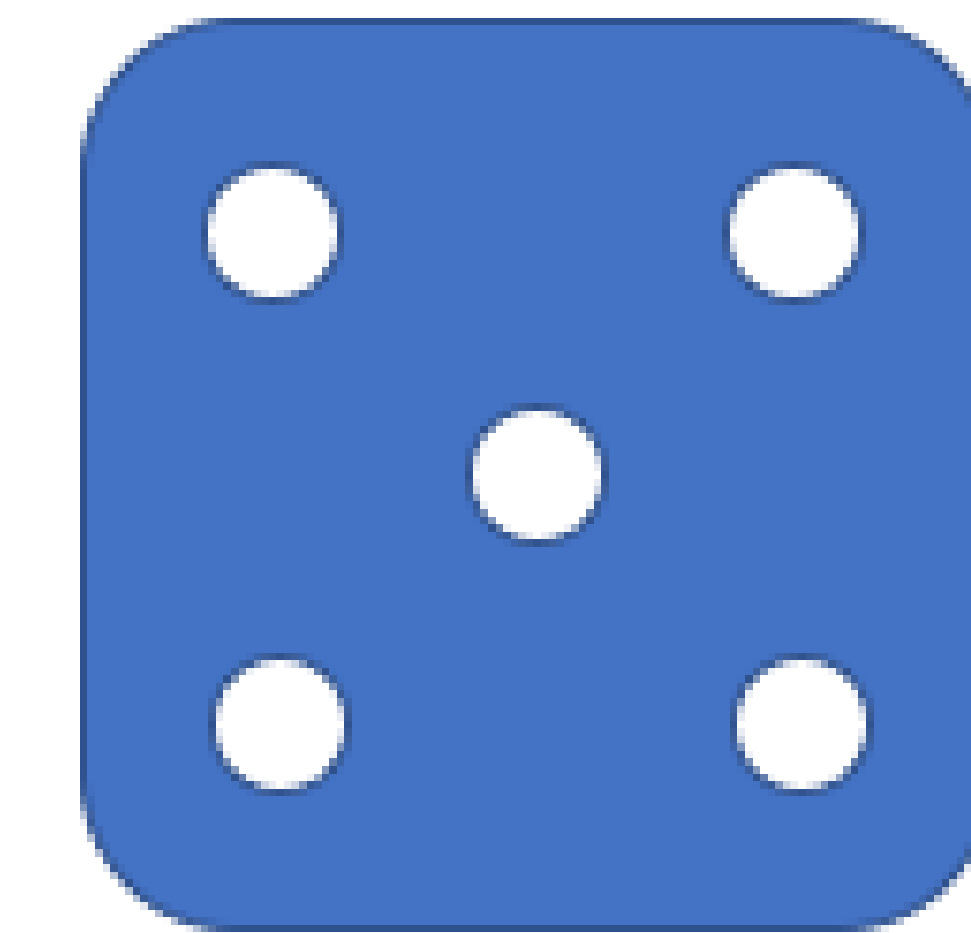
dog



tree



tree



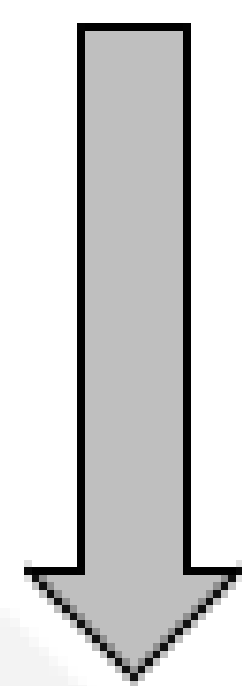
dog

# Rethinking Generalization

“Understanding Deep Neural Networks Requires Rethinking Generalization”



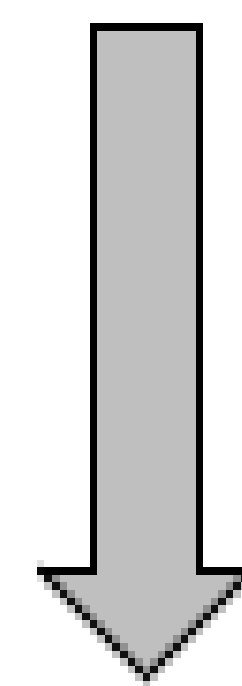
~~dog~~



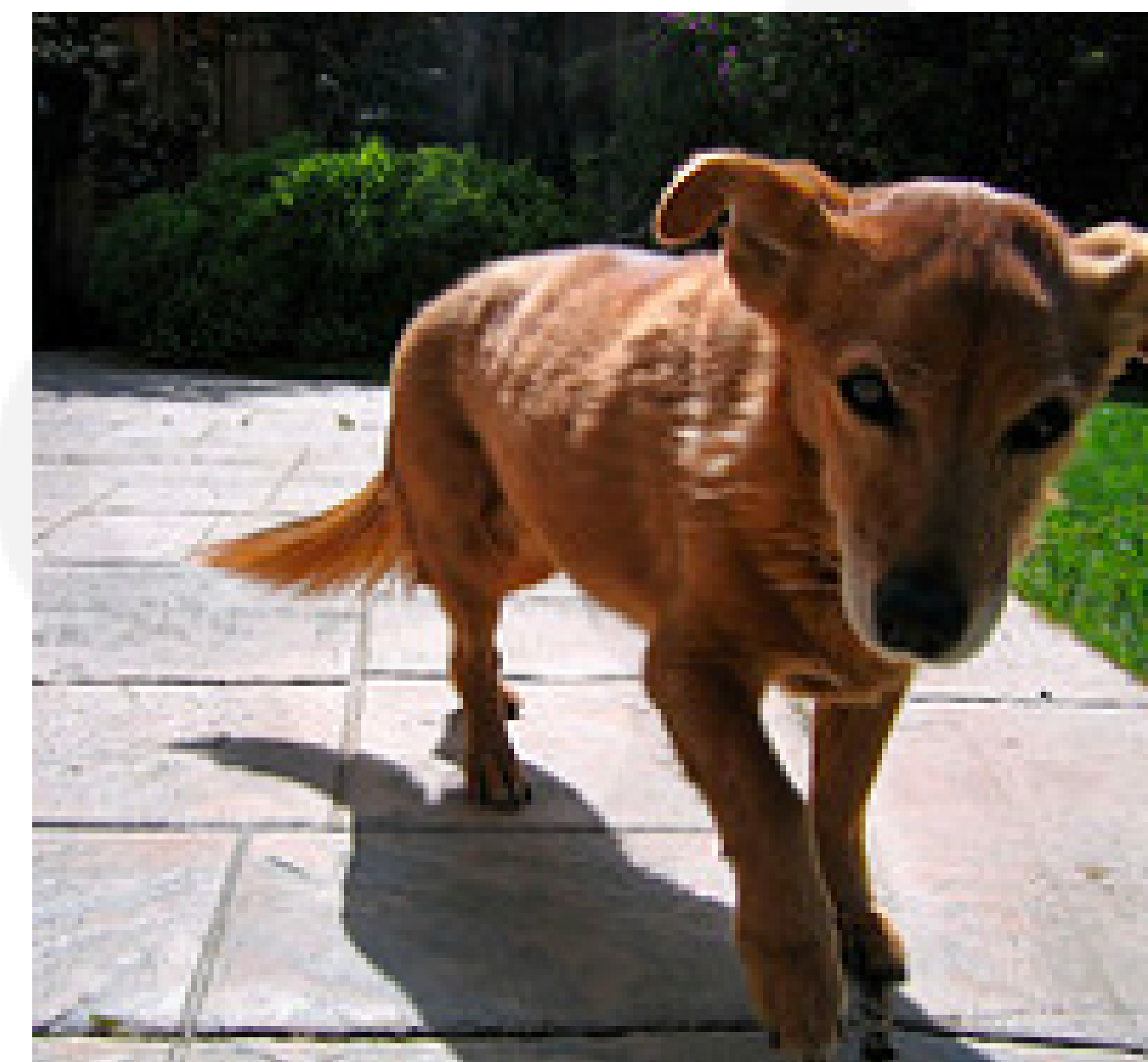
banana



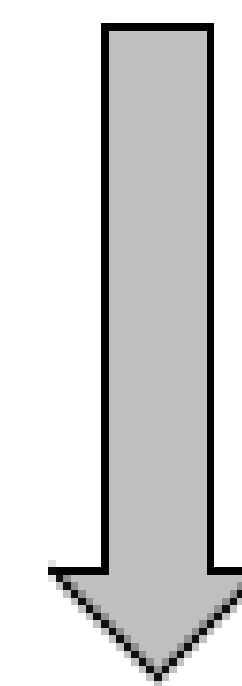
~~banana~~



dog



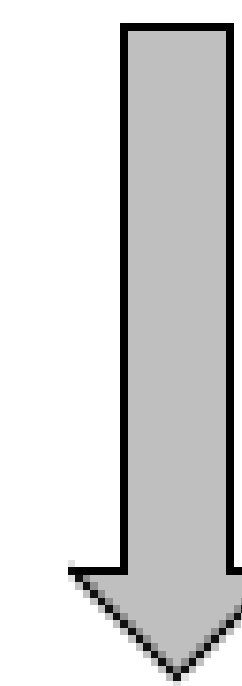
~~dog~~



tree



~~tree~~



dog

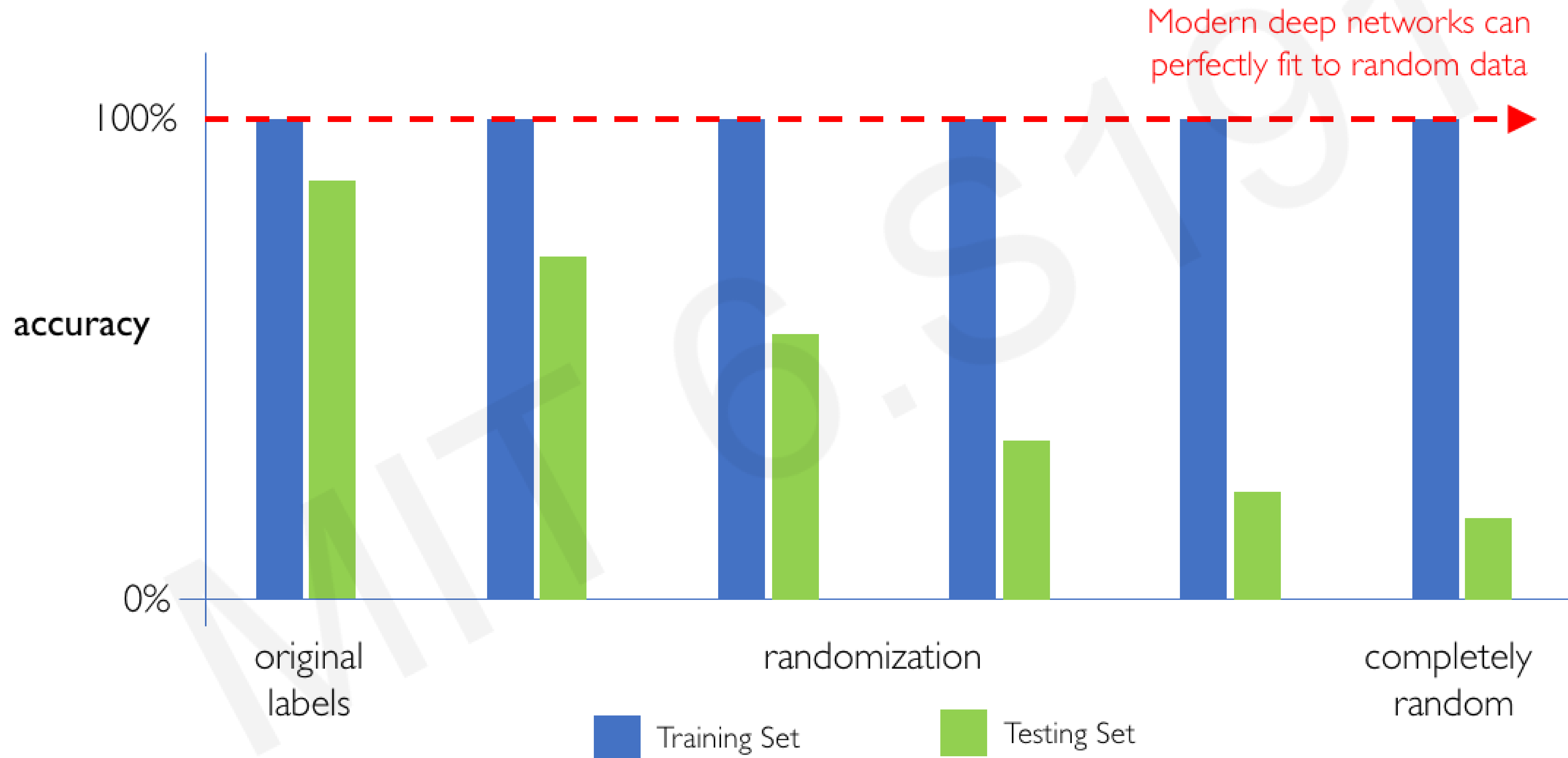
# Capacity of Deep Neural Networks



# Capacity of Deep Neural Networks



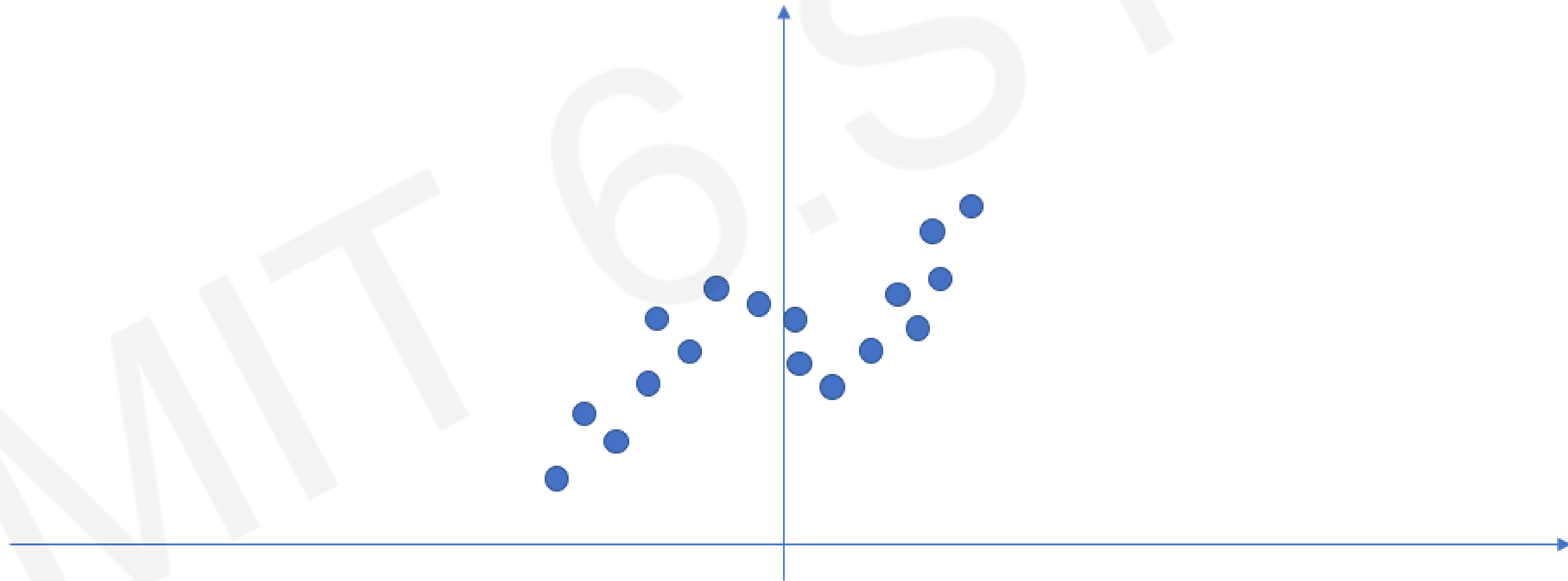
# Capacity of Deep Neural Networks





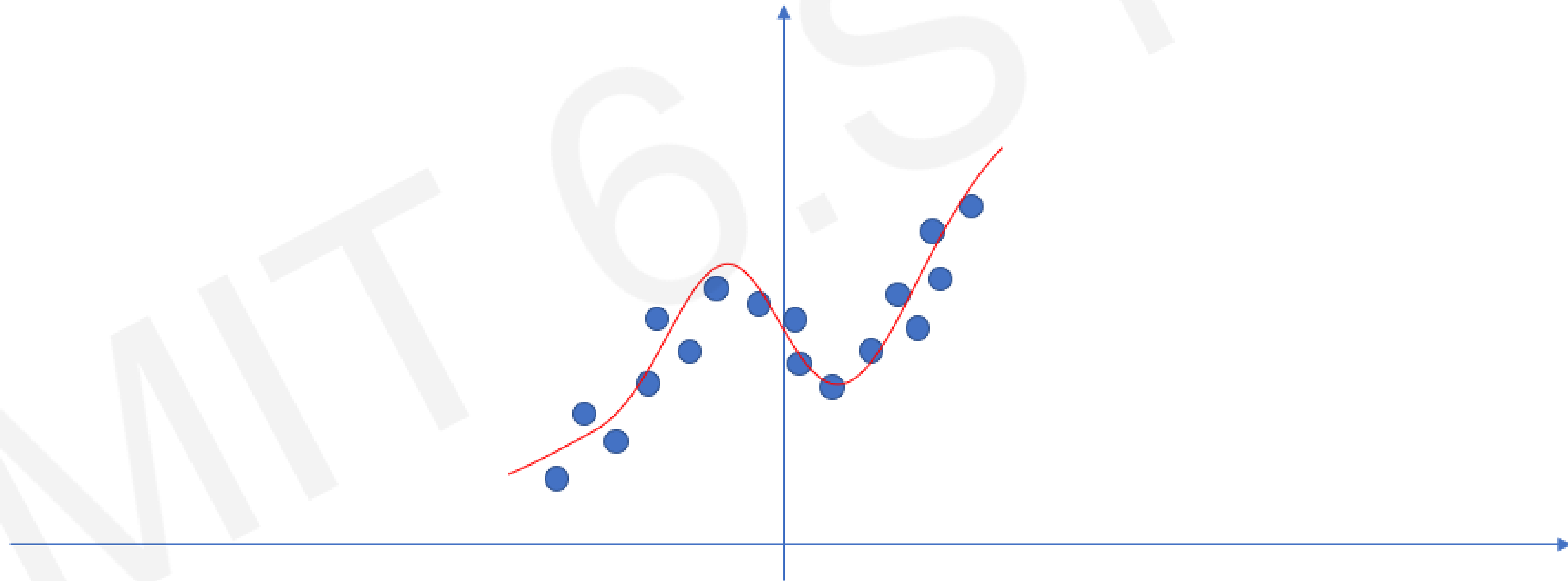
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



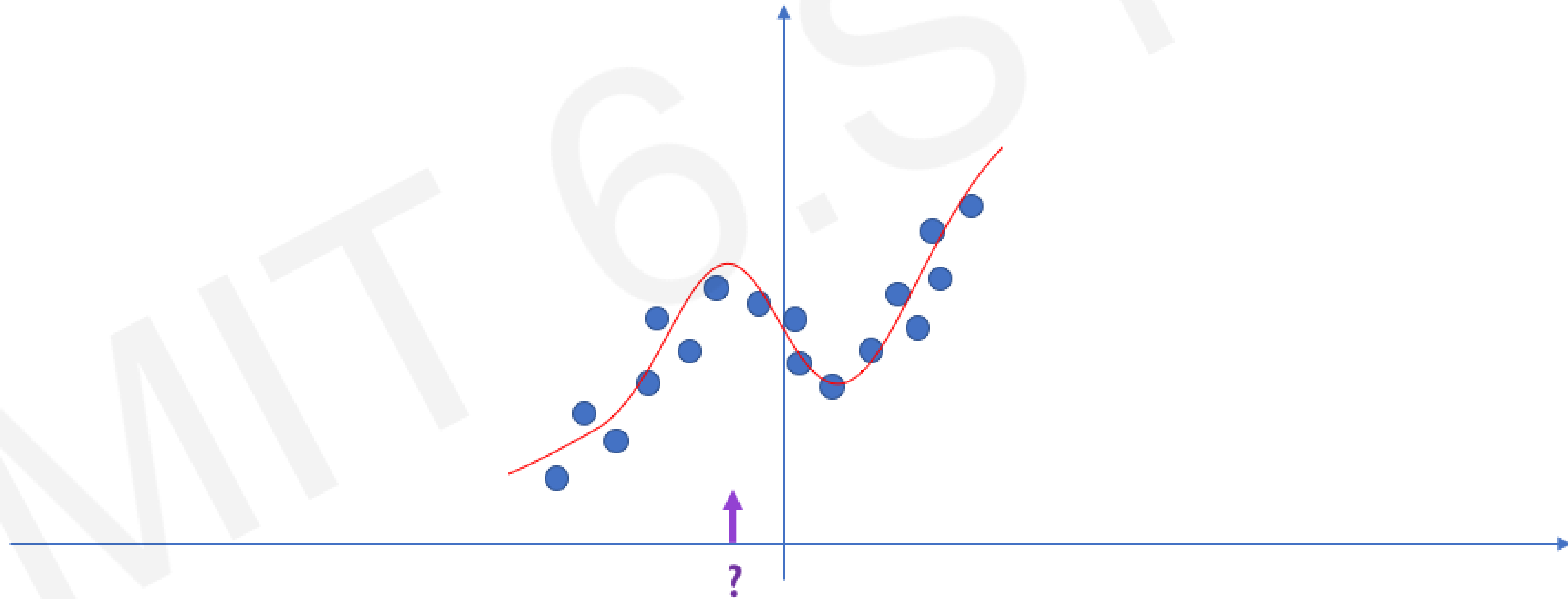
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



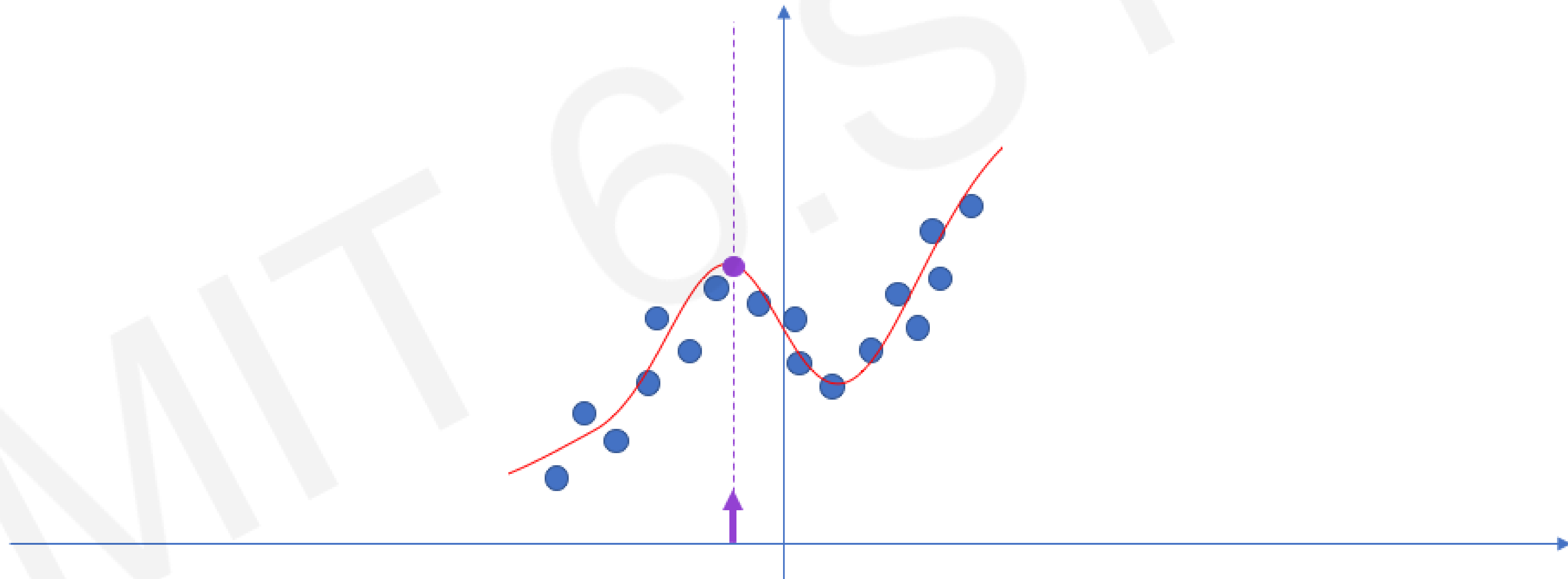
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



# Neural Networks as Function Approximators

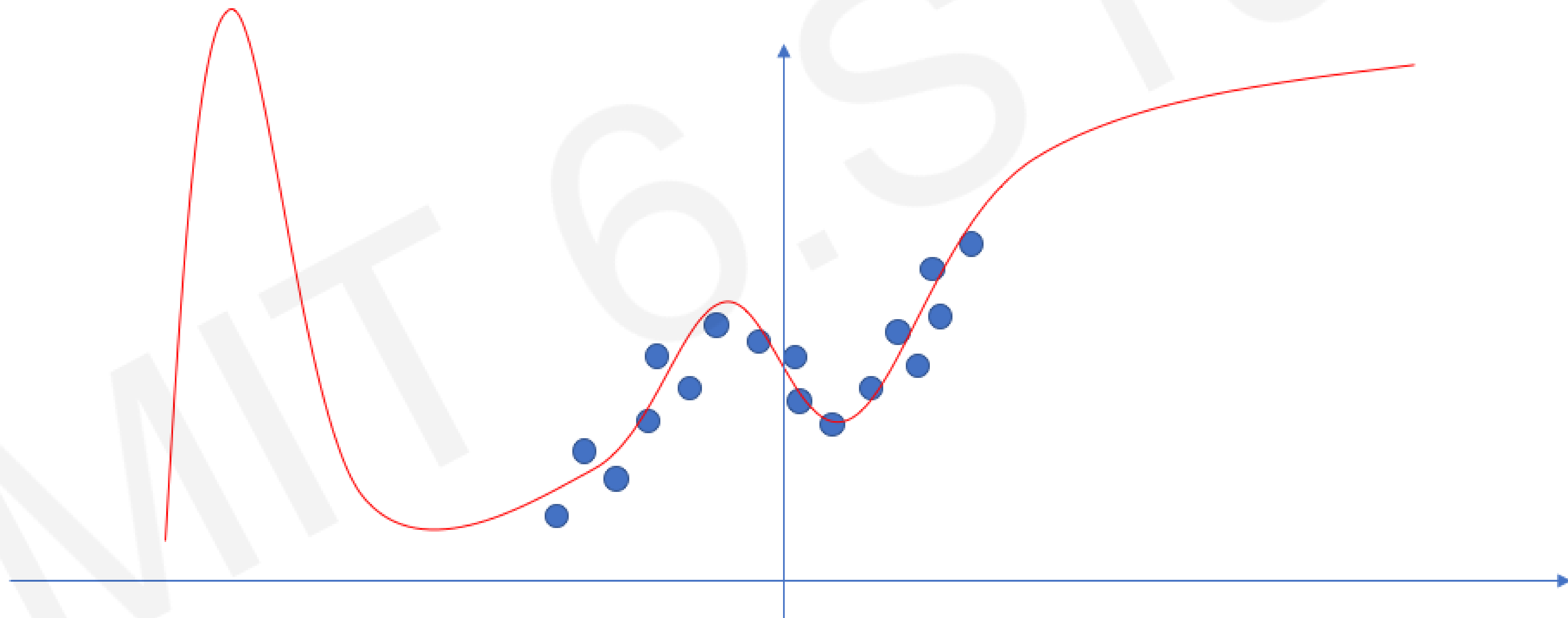
Neural networks are **excellent** function approximators





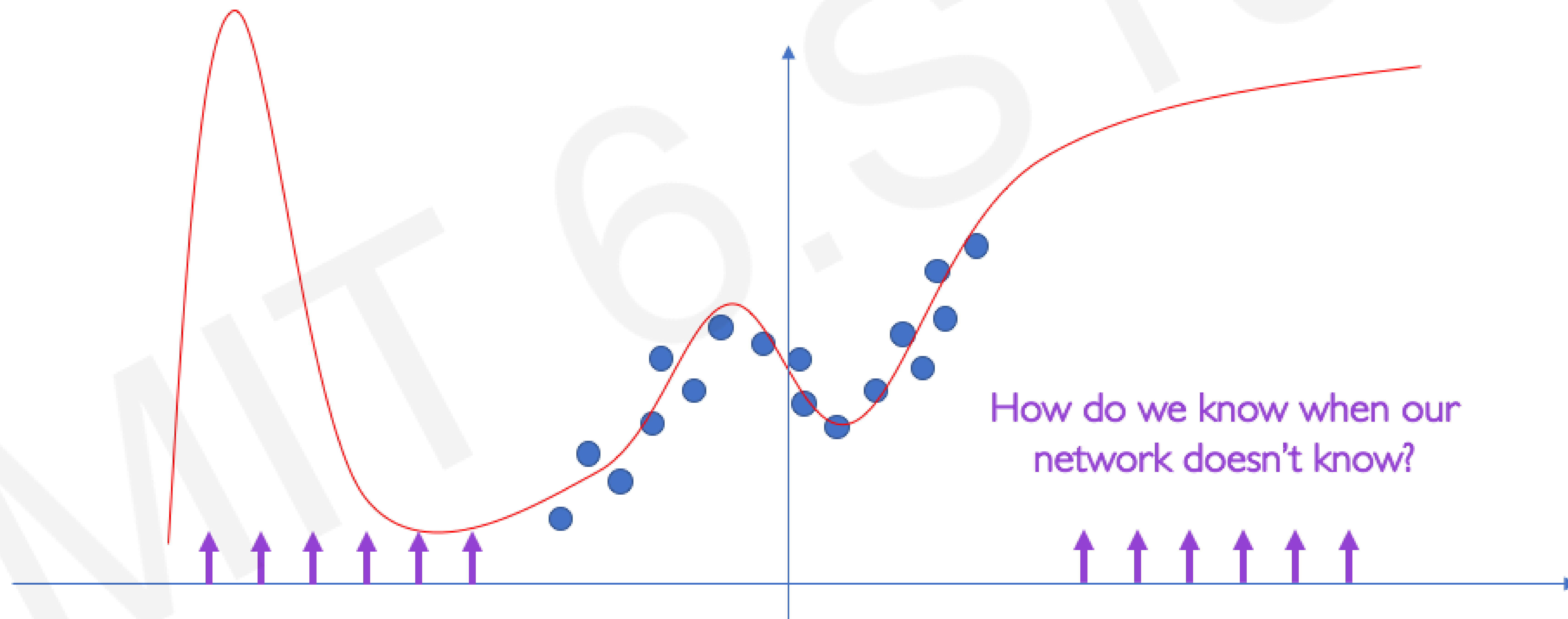
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators

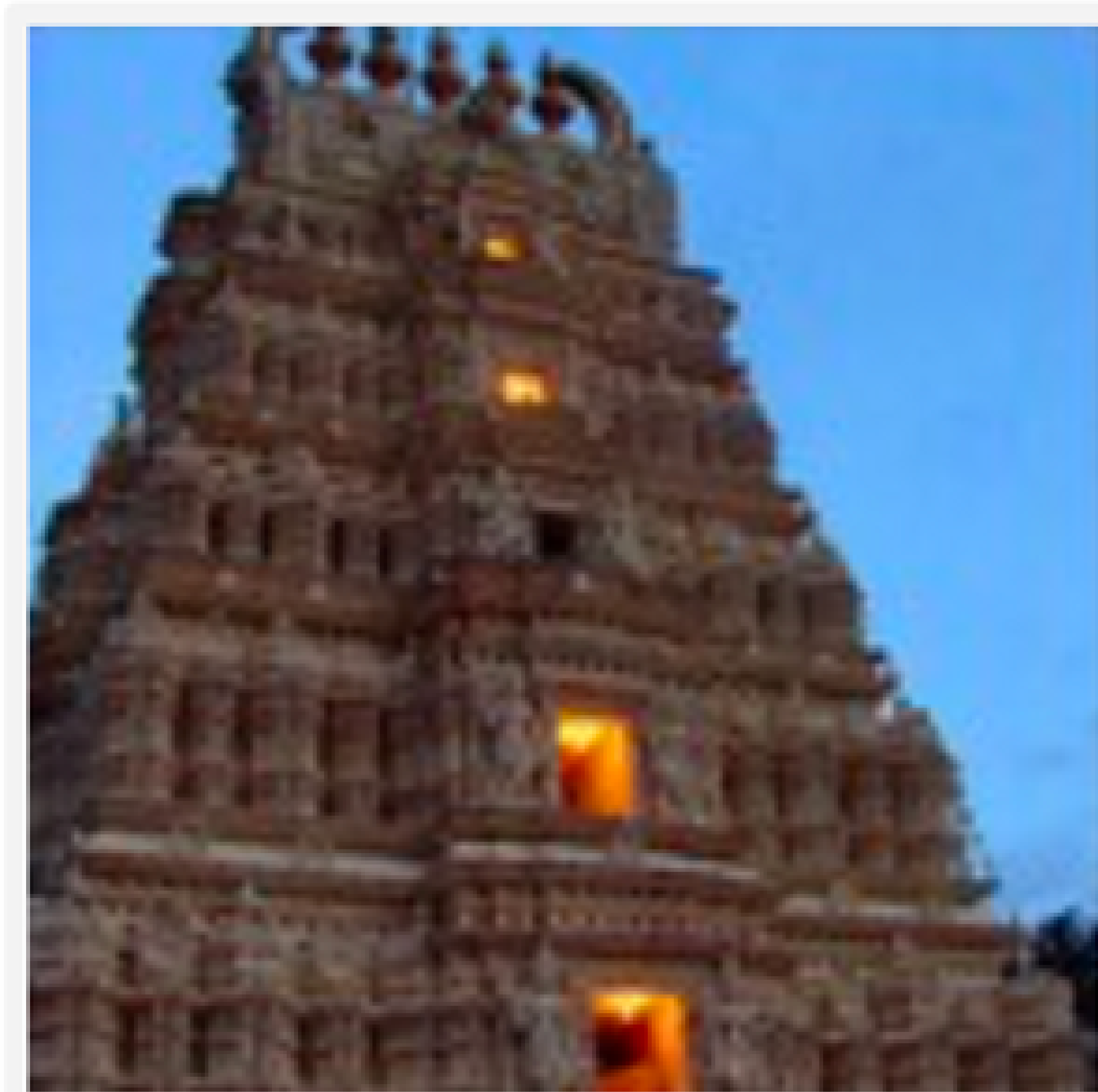


# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators  
...when they have training data

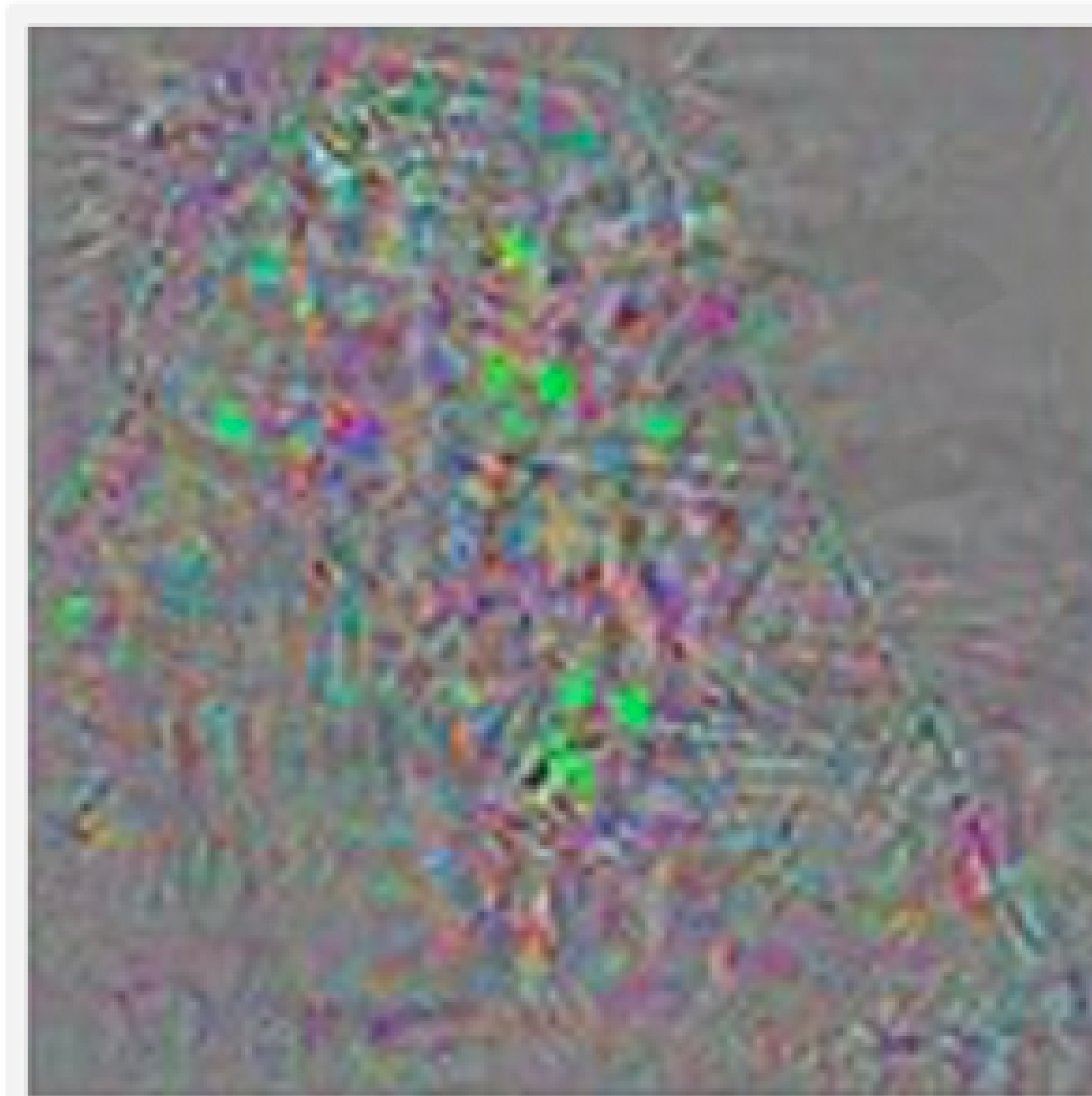
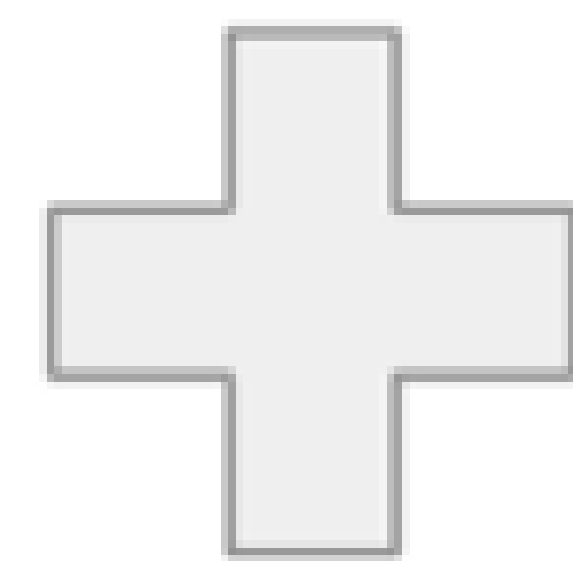


# Adversarial Attacks on Neural Networks

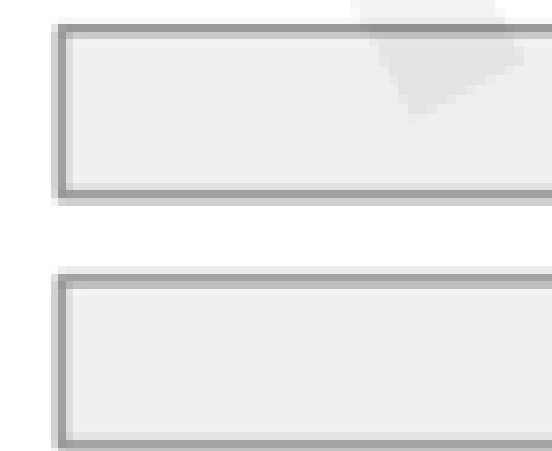


**Original image**

Temple (97%)



**Perturbations**



**Adversarial example**

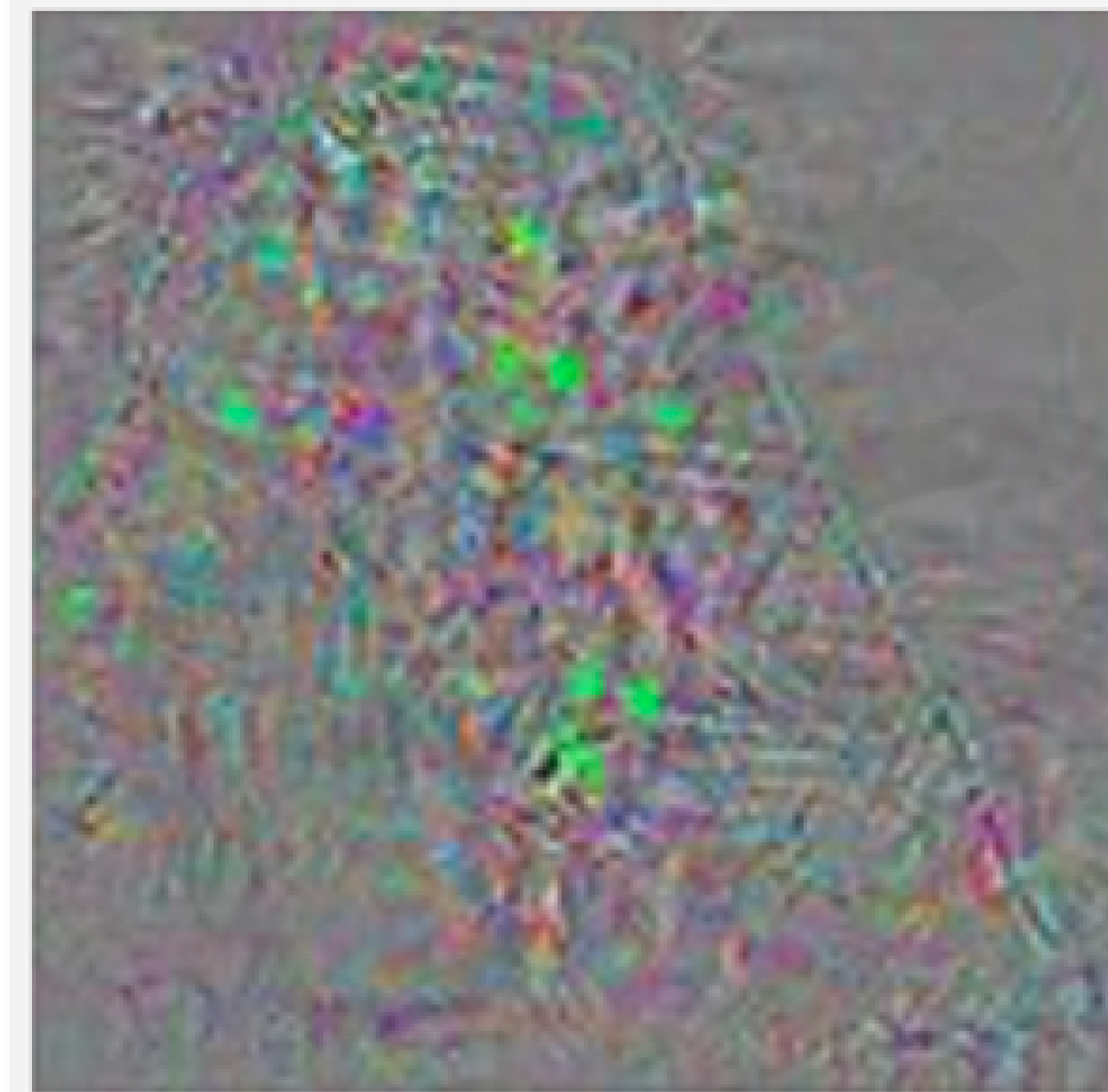
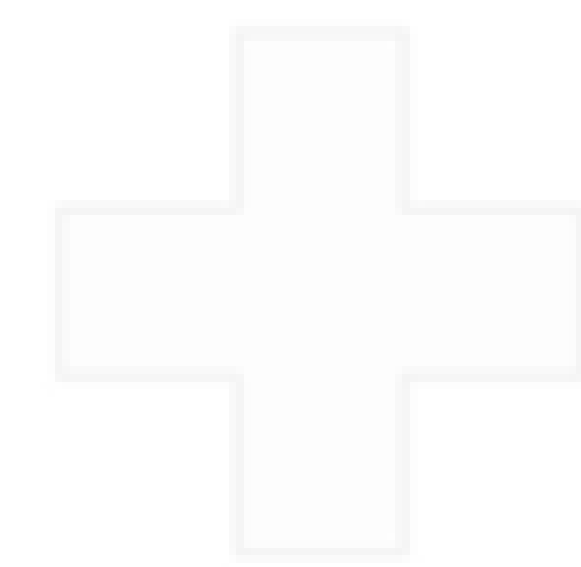
Ostrich (98%)

# Adversarial Attacks on Neural Networks

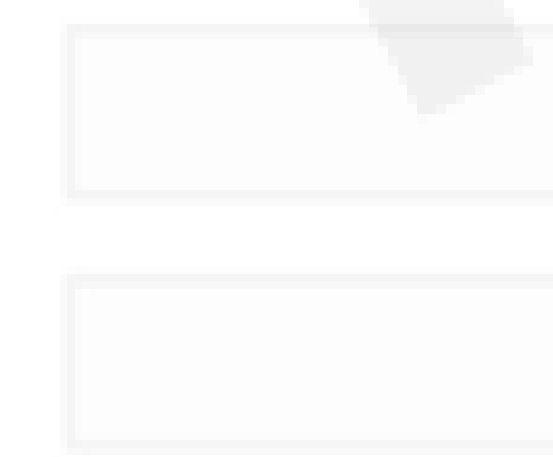


Original image

Temple (97%)



**Perturbations**



Adversarial example

Ostrich (98%)

# Adversarial Attacks on Neural Networks

## Remember:

We train our networks with gradient descent

$$W \leftarrow W - \eta \frac{\partial J(W, x, y)}{\partial W}$$

*“How does a small change in weights decrease our loss”*



# Adversarial Attacks on Neural Networks

## Remember:

We train our networks with gradient descent

$$W \leftarrow W - \eta \frac{\partial J(W, x, y)}{\partial W}$$

*“How does a small change in weights decrease our loss”*

# Adversarial Attacks on Neural Networks

## Remember:

We train our networks with gradient descent

$$W \leftarrow W - \eta \frac{\partial J(W, x, y)}{\partial W}$$

Fix your image  $x$ ,  
and true label  $y$

“How does a small change in weights decrease our loss”

# Adversarial Attacks on Neural Networks

## Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(W, x, y)}{\partial x}$$

*“How does a small change in the input increase our loss”*

# Adversarial Attacks on Neural Networks

## Adversarial Image:

Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(W, x, y)}{\partial x}$$

“How does a small change in the input increase our loss”

# Adversarial Attacks on Neural Networks

## Adversarial Image:

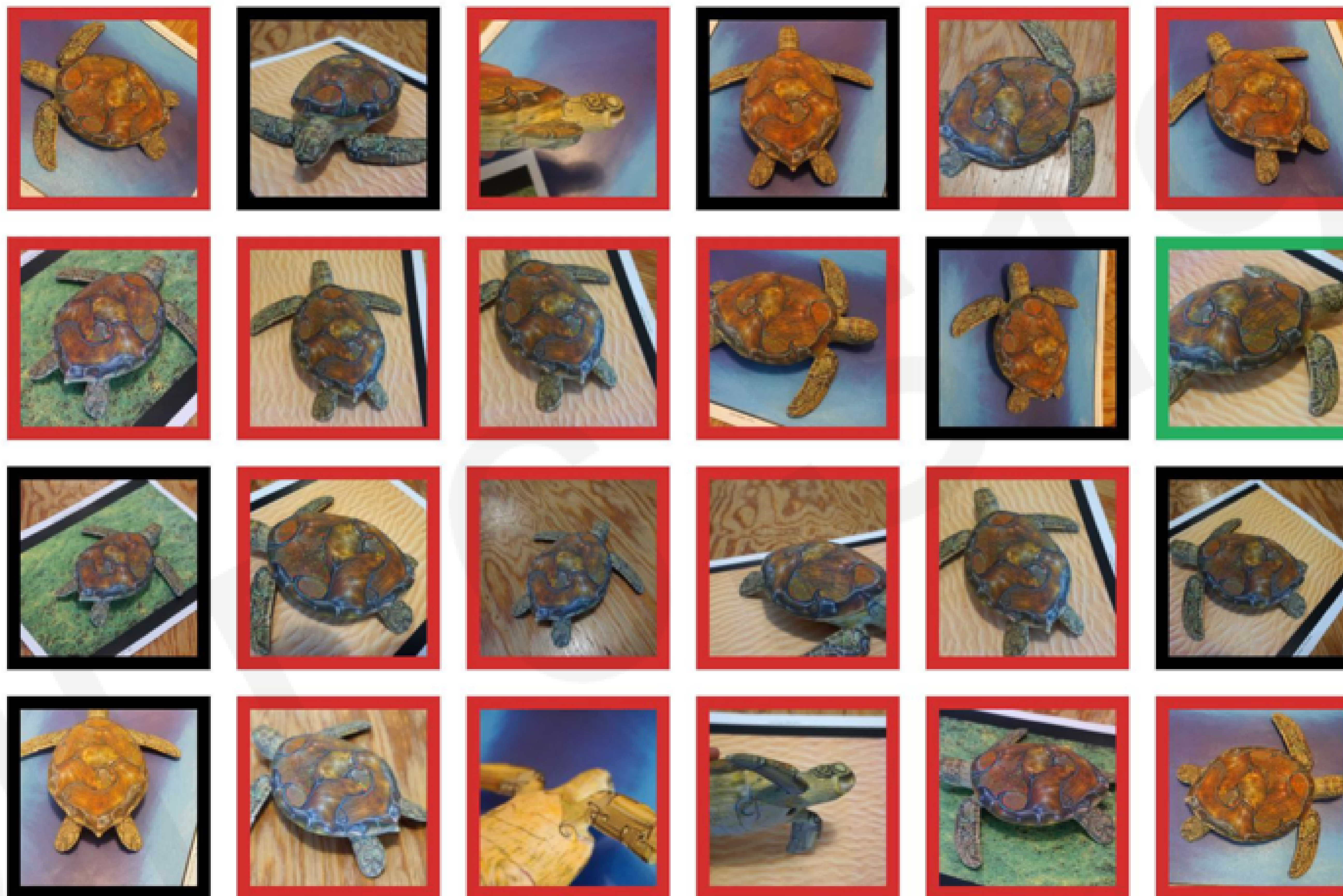
Modify image to increase error

$$x \leftarrow x + \eta \frac{\partial J(W, x, y)}{\partial x}$$

Fix your weights  $\theta$ ,  
and true label  $y$

“How does a small change in the input increase our loss”

# Synthesizing Robust Adversarial Examples



■ classified as turtle    ■ classified as rifle  
■ classified as other



# Neural Network Limitations...

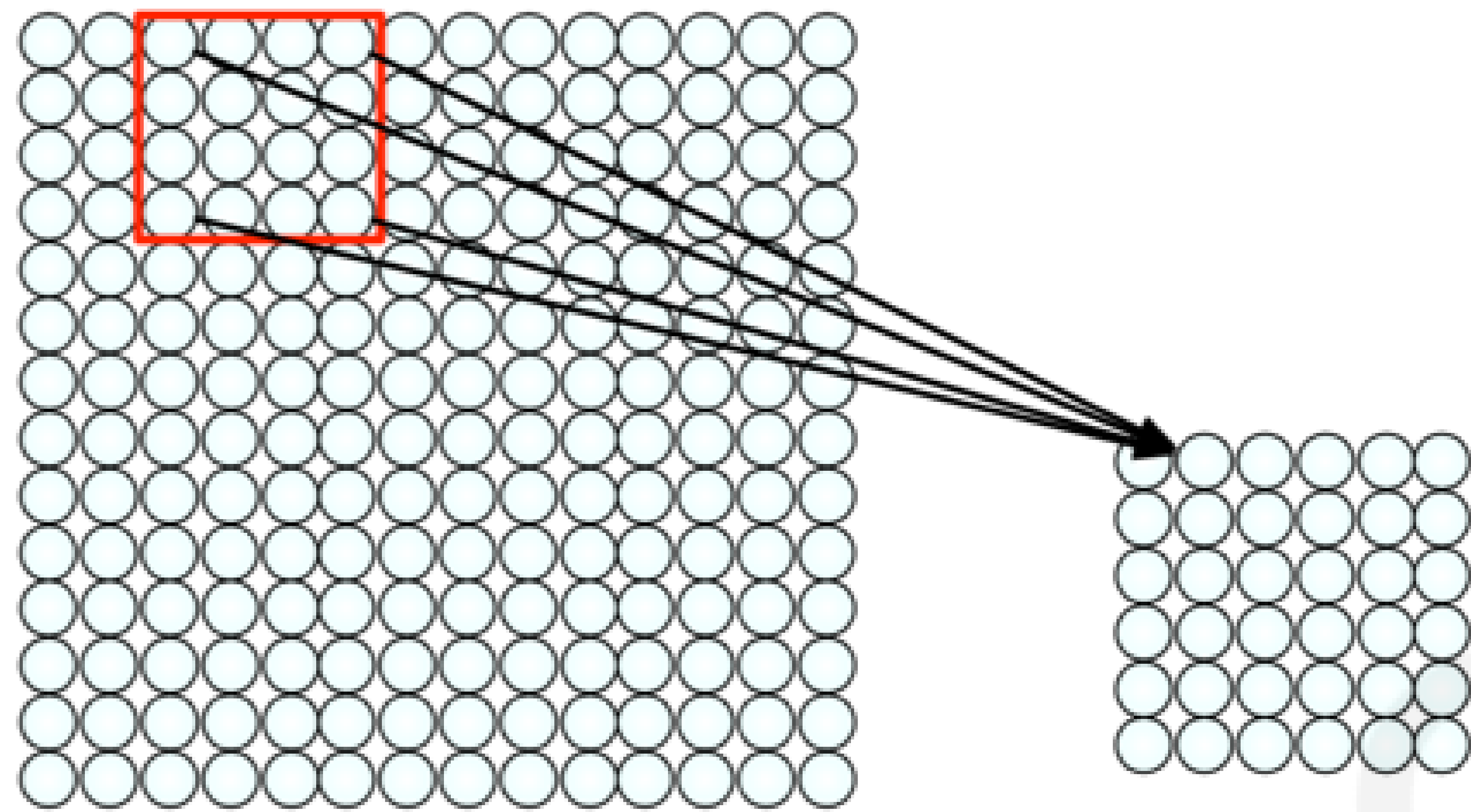
- Very **data hungry** (eg. often millions of examples)
- **Computationally intensive** to train and deploy (tractably requires GPUs)
- Easily fooled by **adversarial examples**
- Can be subject to **algorithmic bias**
- Difficult to **encode structure** and prior knowledge during learning
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- **Finicky to optimize**: non-convex, choice of architecture, learning parameters
- Often require **expert knowledge** to design, fine tune architectures

# Neural Network Limitations...

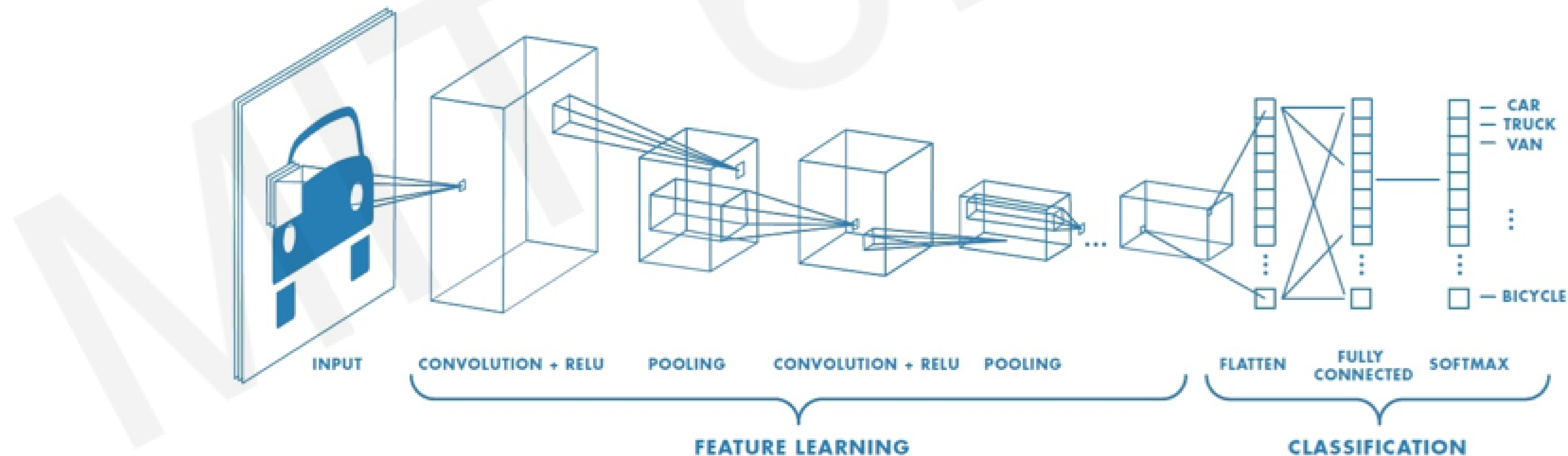
- Very **data hungry** (eg. often millions of examples)
- **Computationally intensive** to train and deploy (tractably requires GPUs)
- Easily fooled by **adversarial examples**
- Can be subject to **algorithmic bias**
- Difficult to **encode structure** and prior knowledge during learning
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- **Finicky to optimize**: non-convex, choice of architecture, learning parameters
- Often require **expert knowledge** to design, fine tune architectures

# New Frontiers I: Encoding Structure into Deep Learning

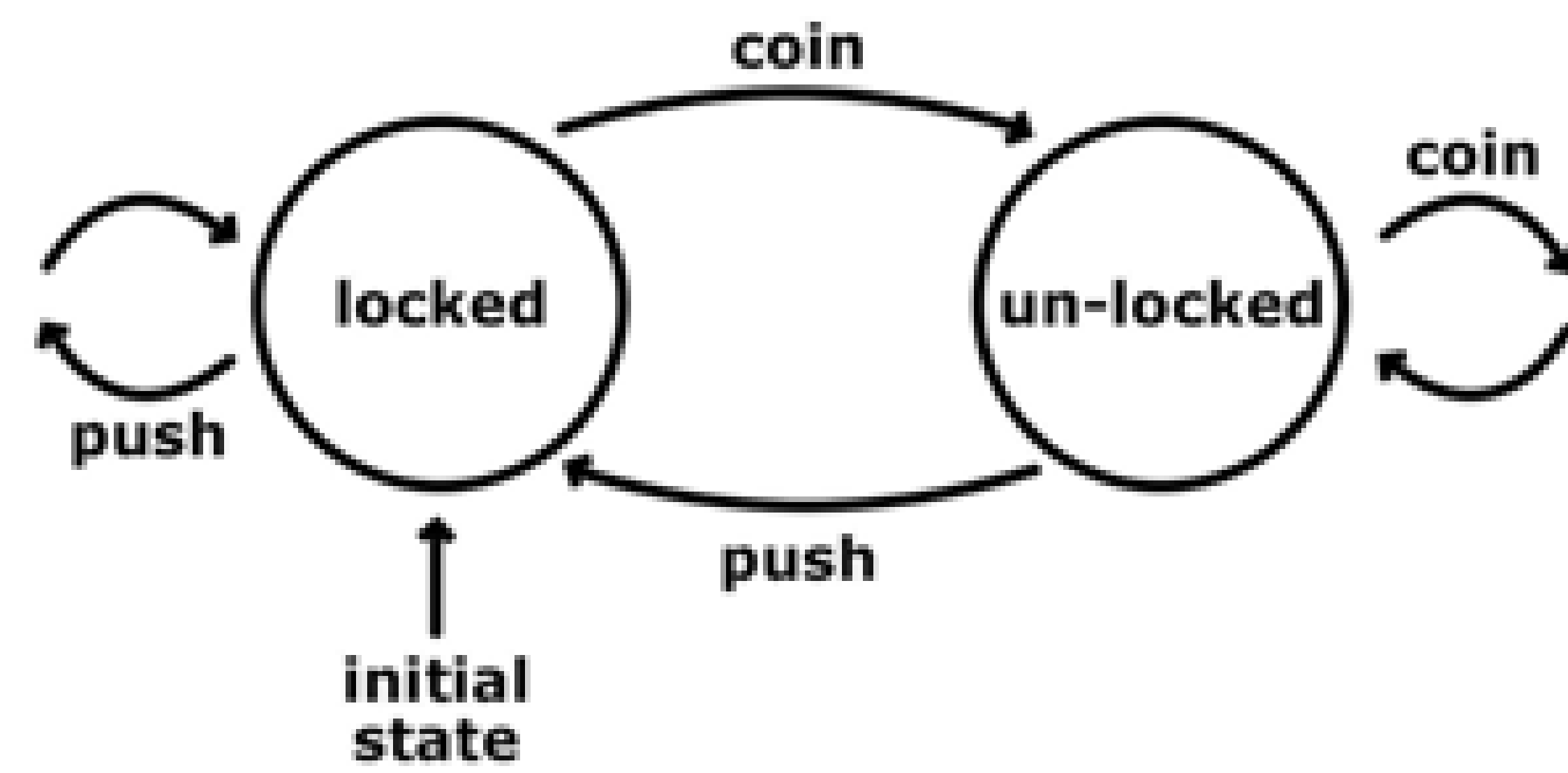
# CNNs: Using Spatial Structure



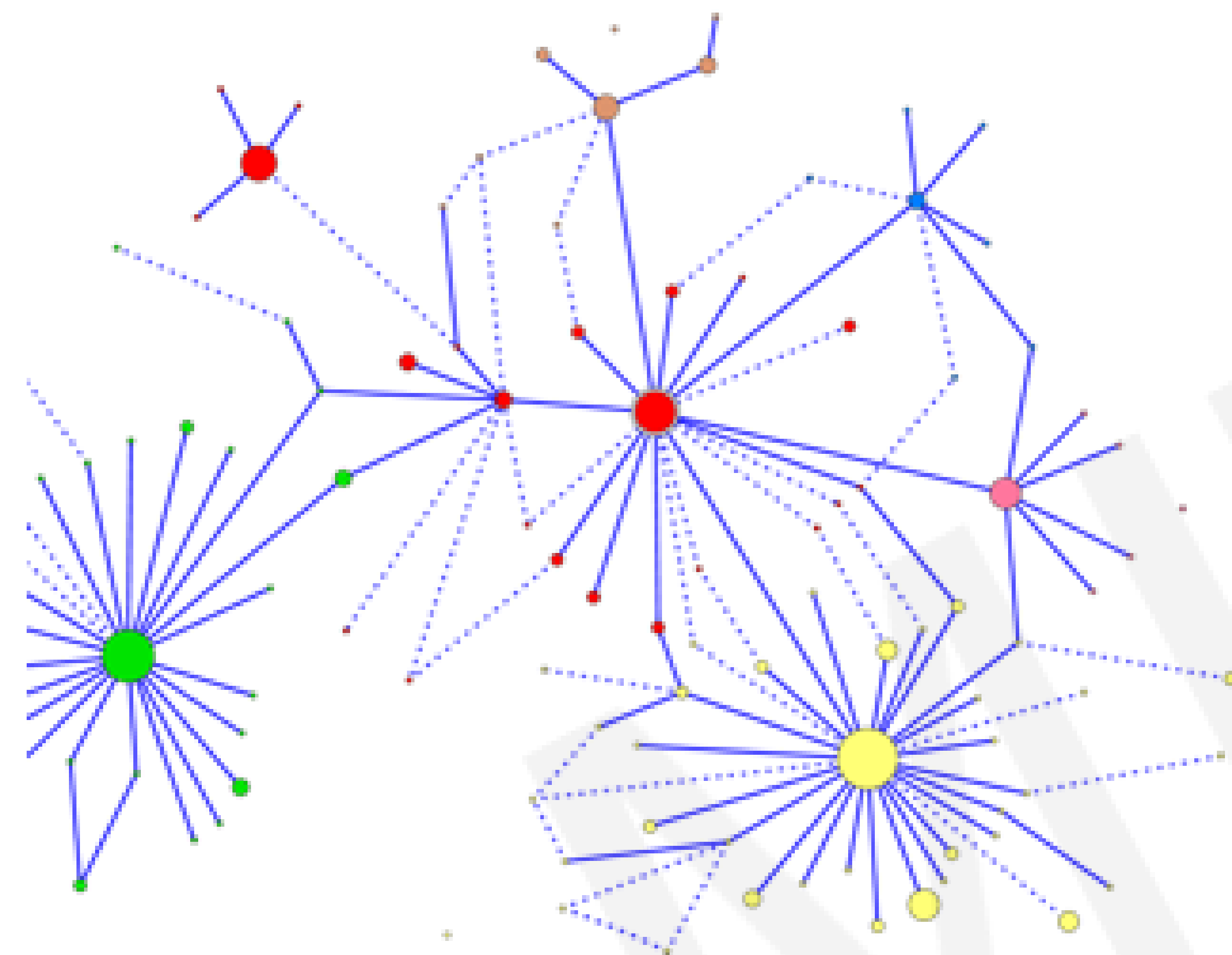
- 1) Apply a set of weights to extract **local features**
- 2) Use **multiple filters** to extract different features
- 3) **Spatially share** parameters of each filter



# Graphs as a Structure for Representing Data



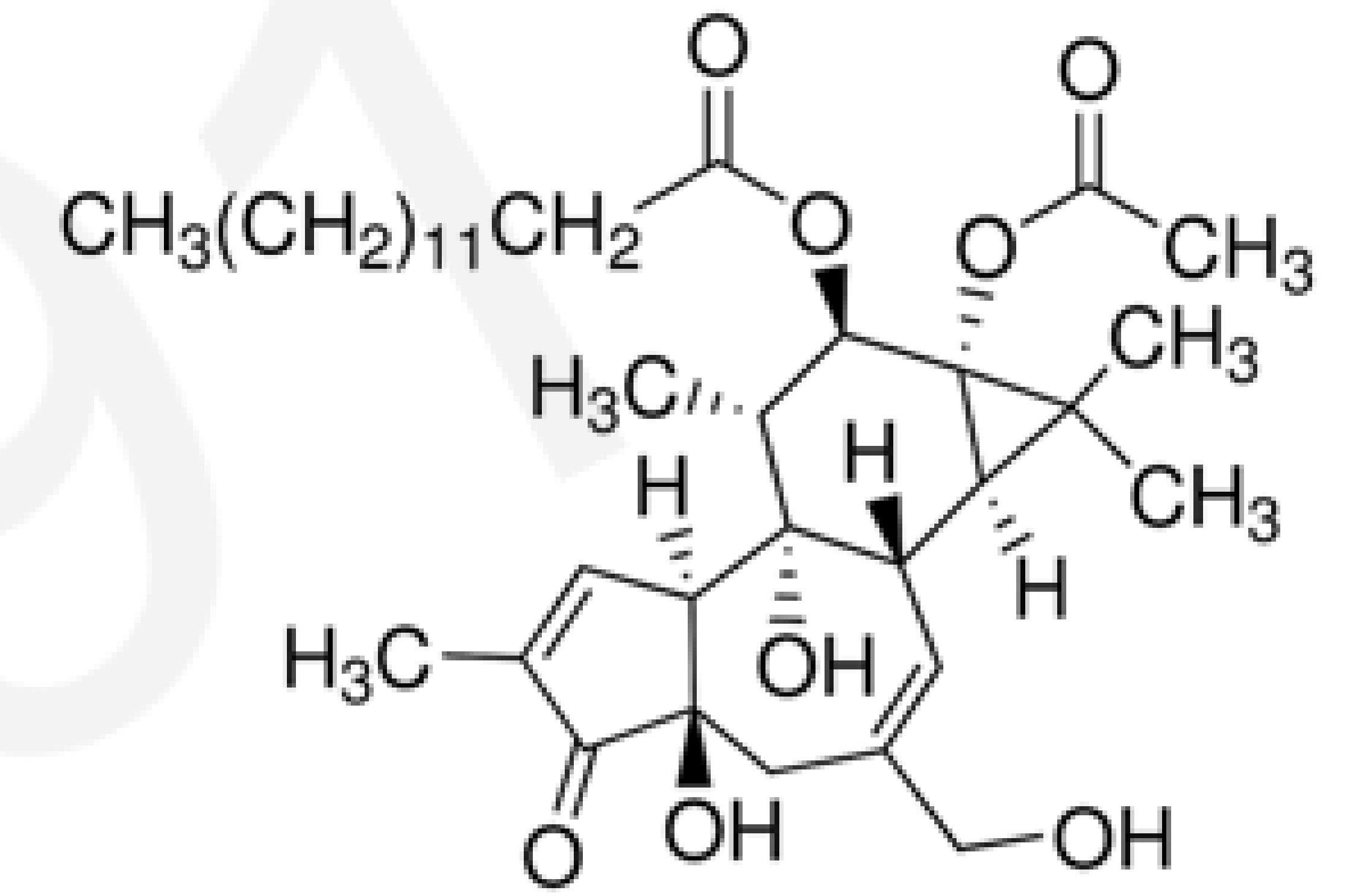
State Machines



Biological Networks



Social Networks



Molecules



Mobility & Transport

# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)





# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)



# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)



# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)



# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

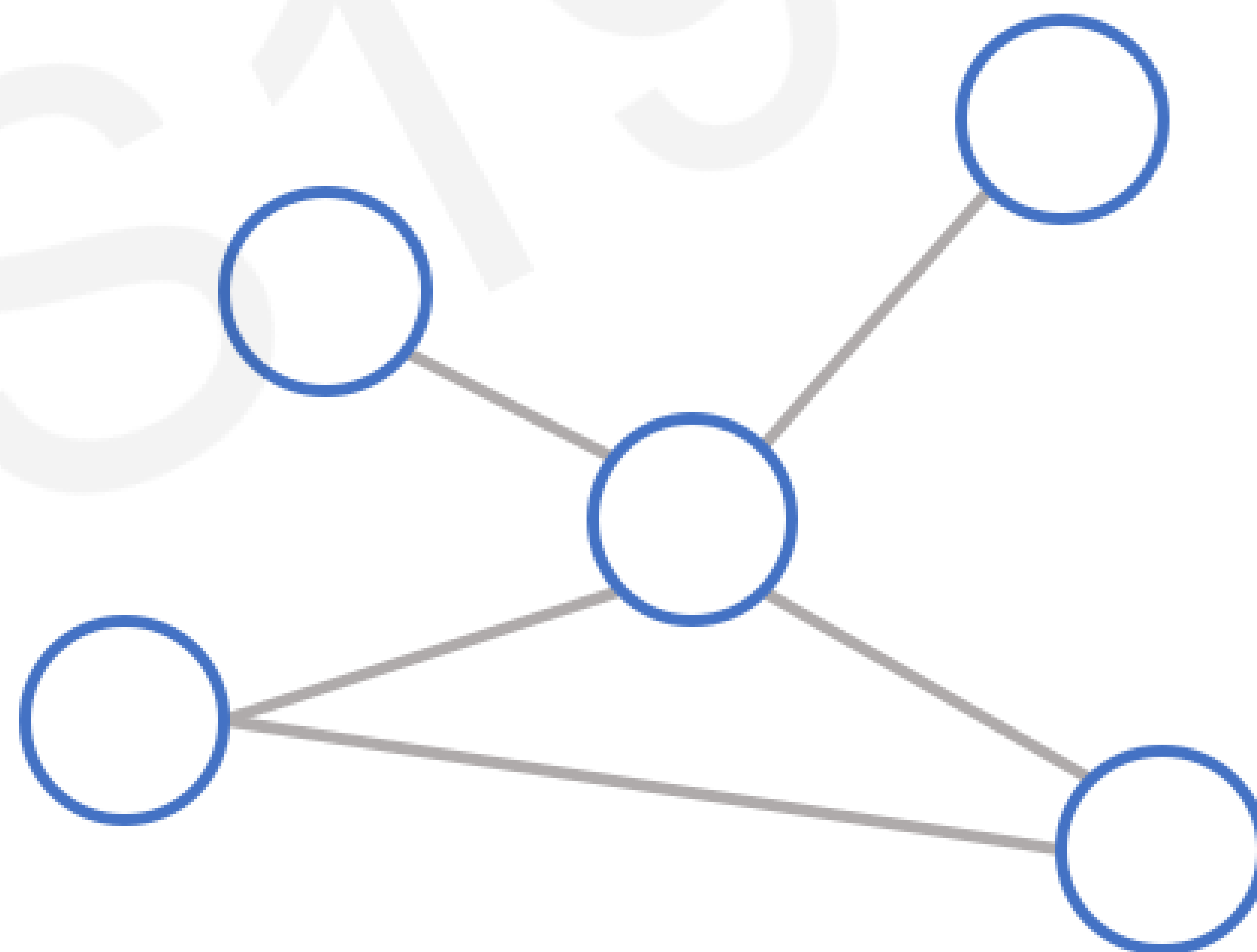


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

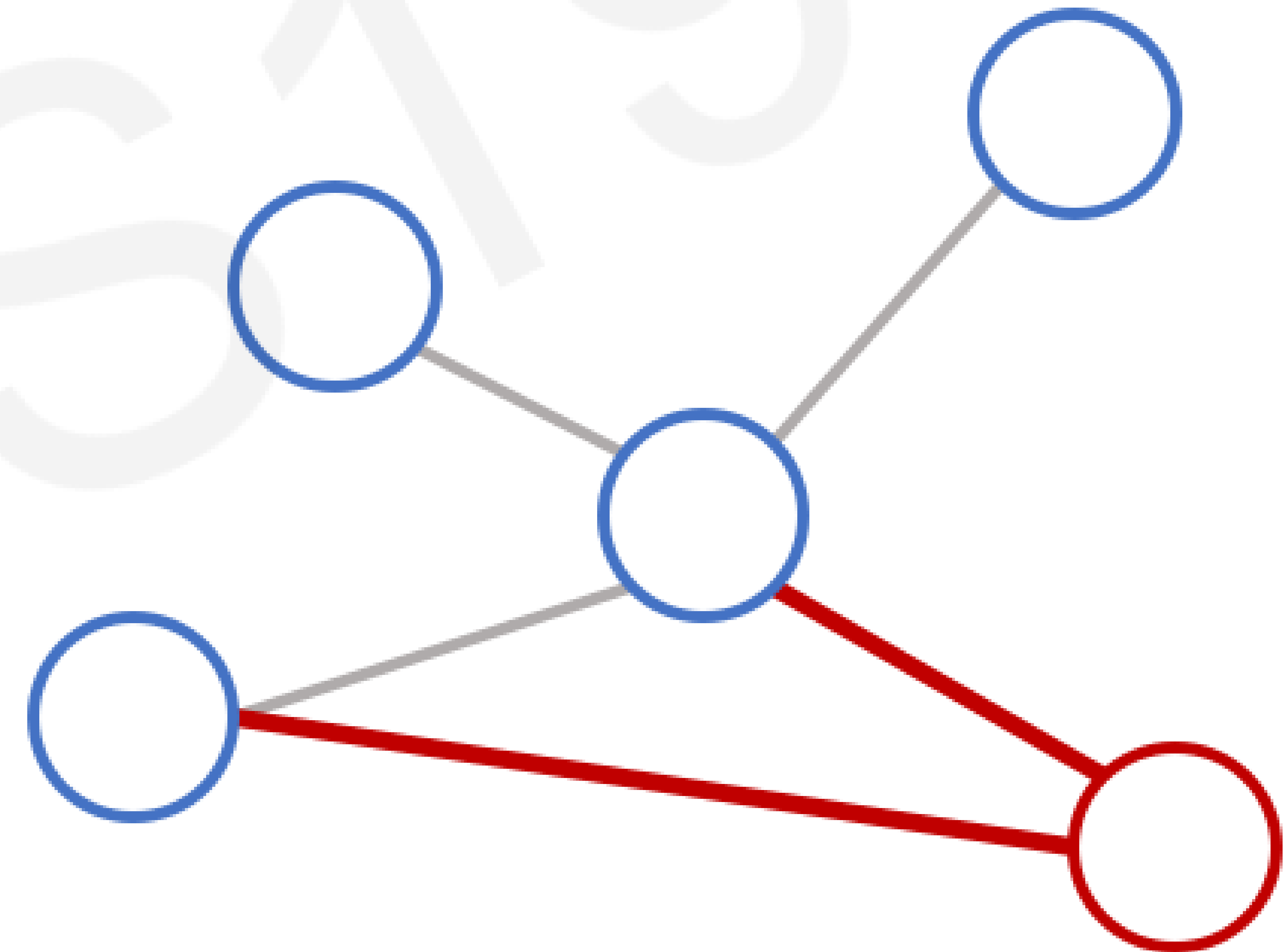


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

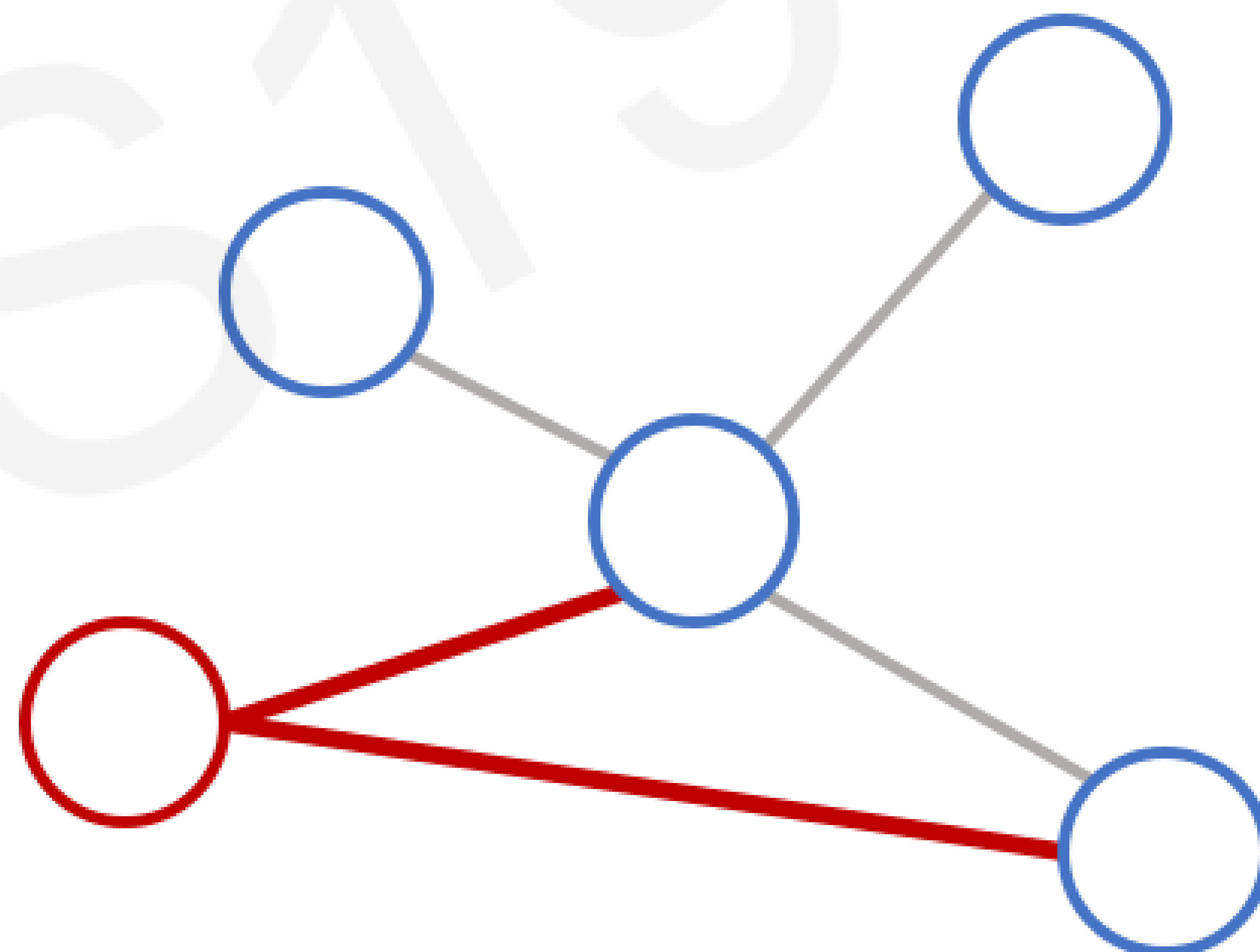


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)



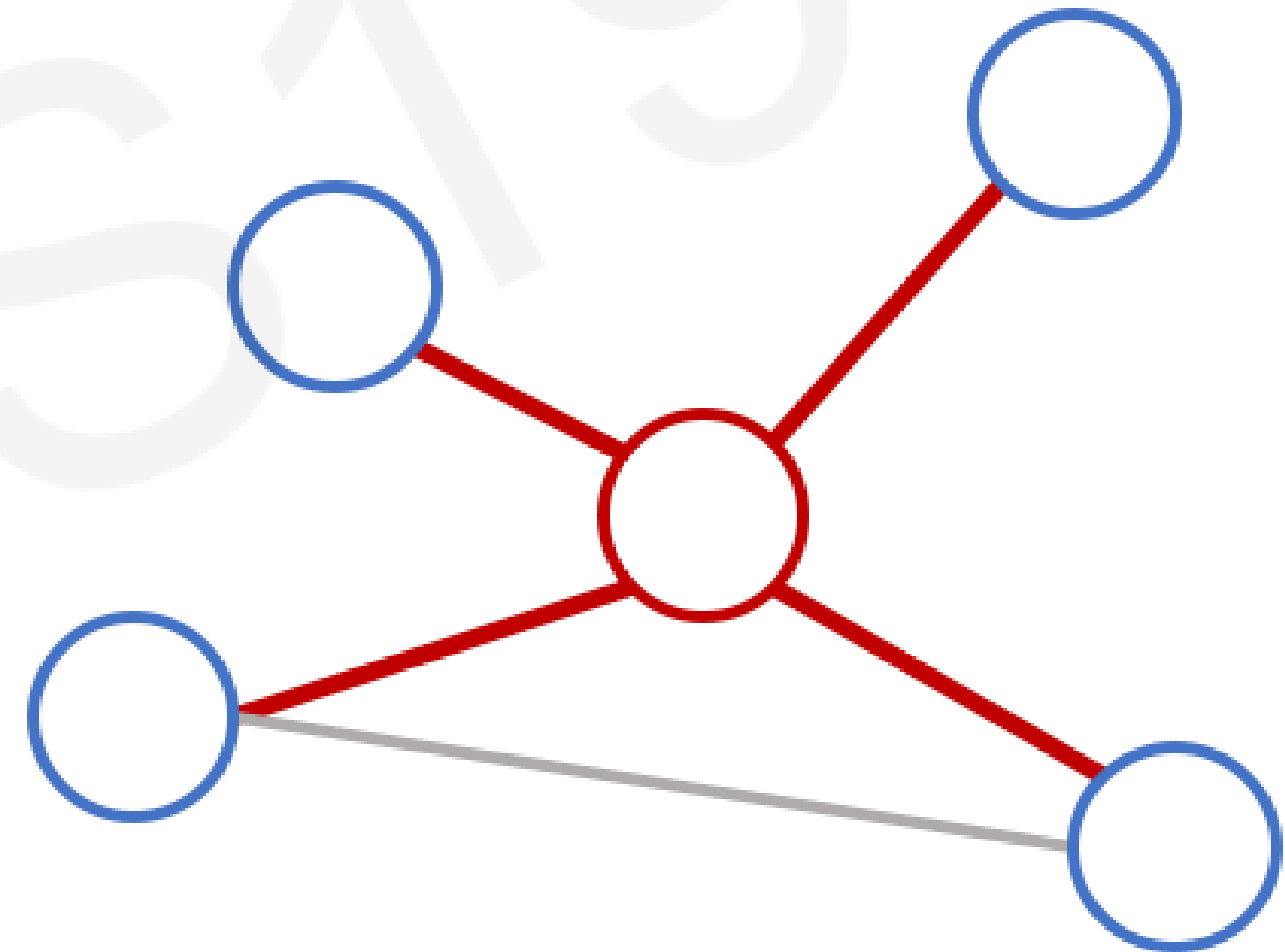


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

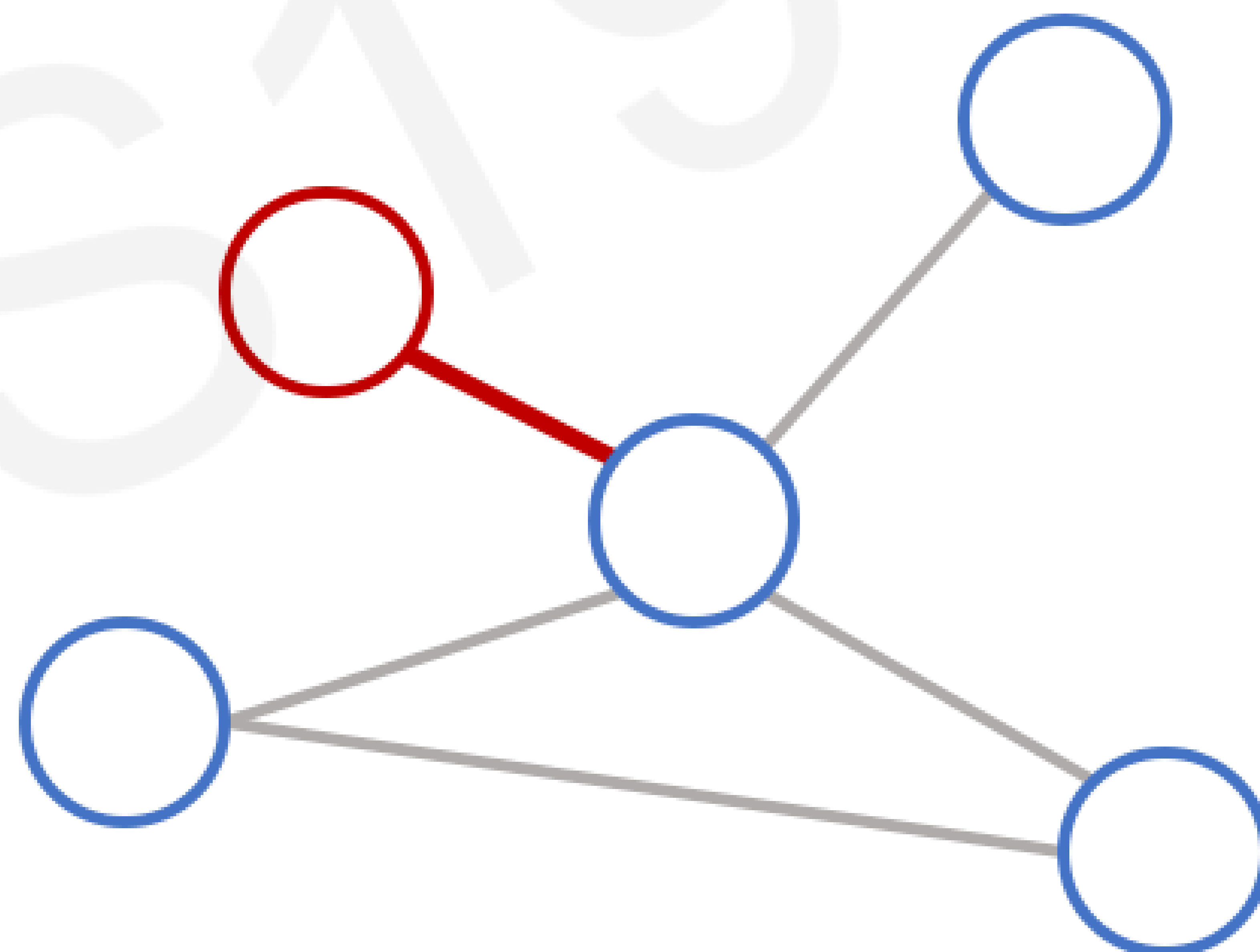


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

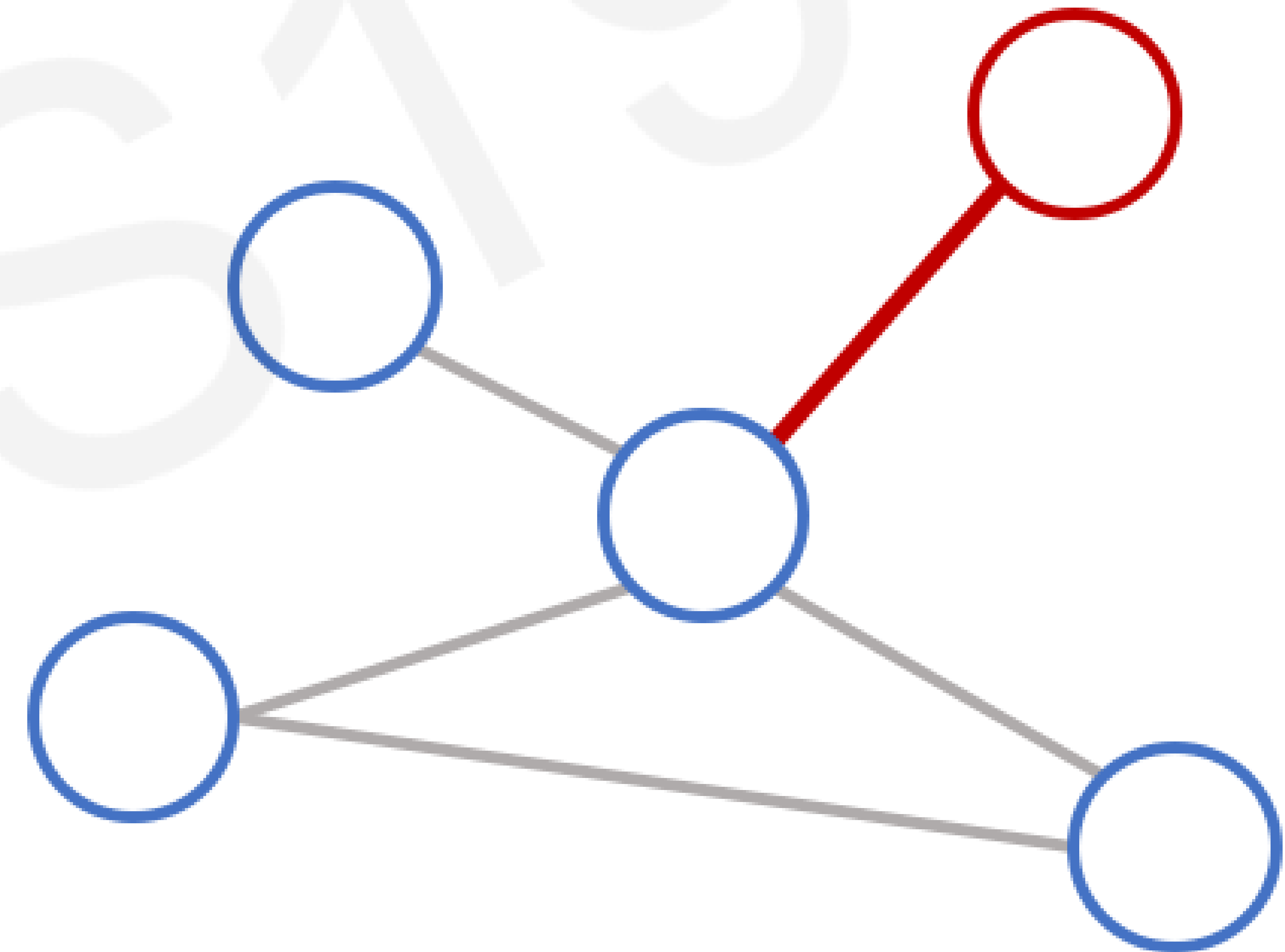


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

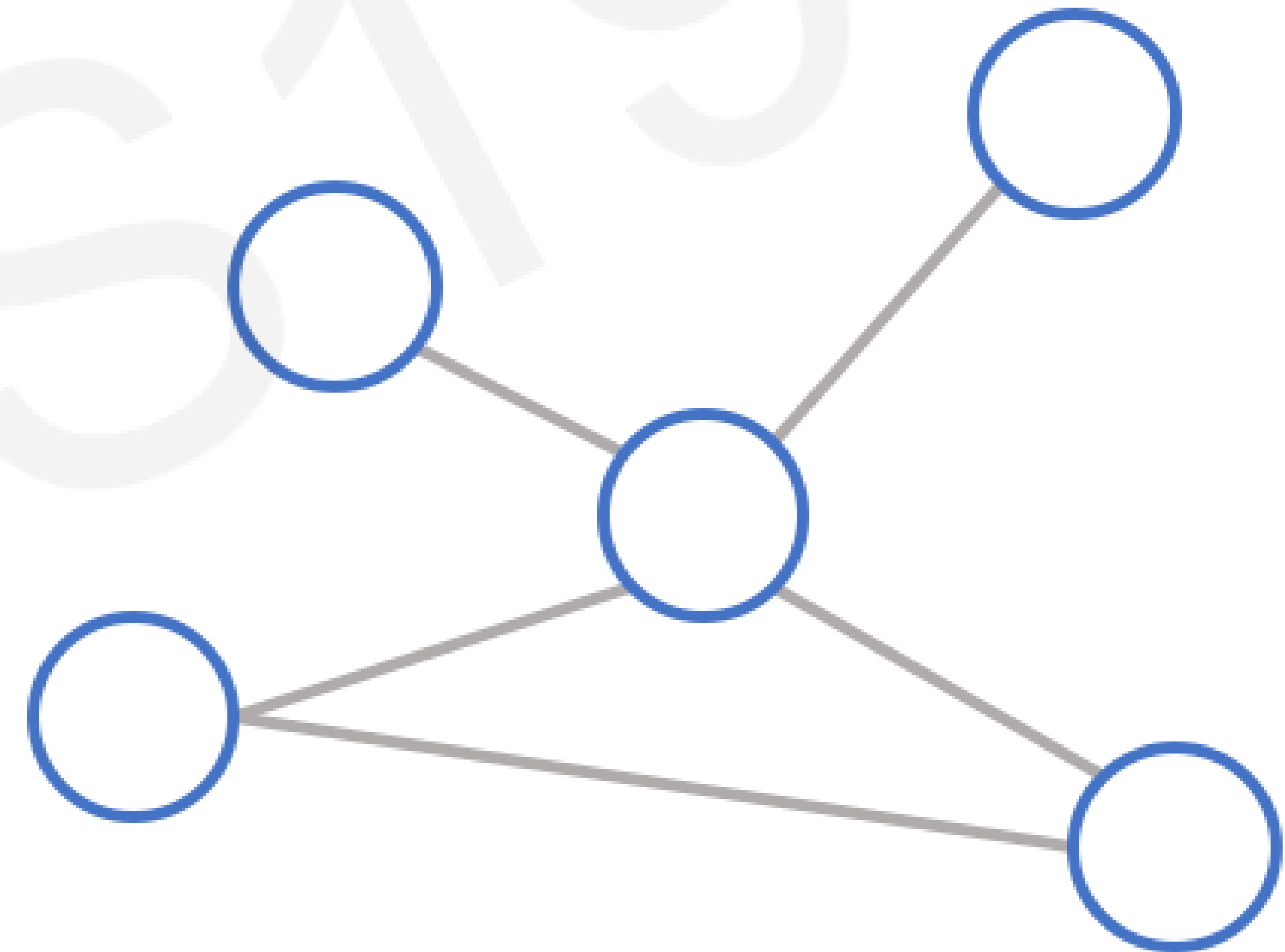


# Graph Convolutional Networks

Convolutional Networks



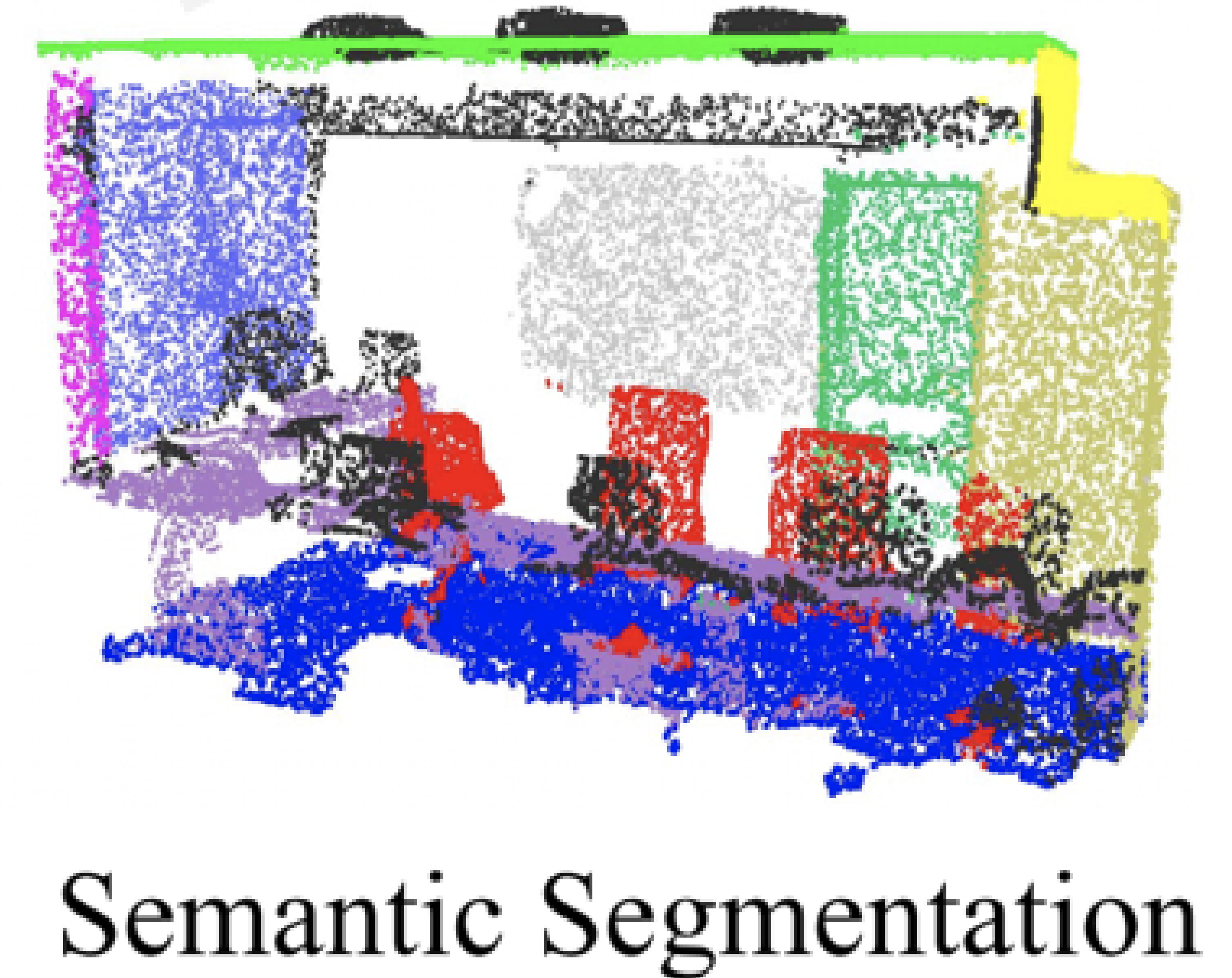
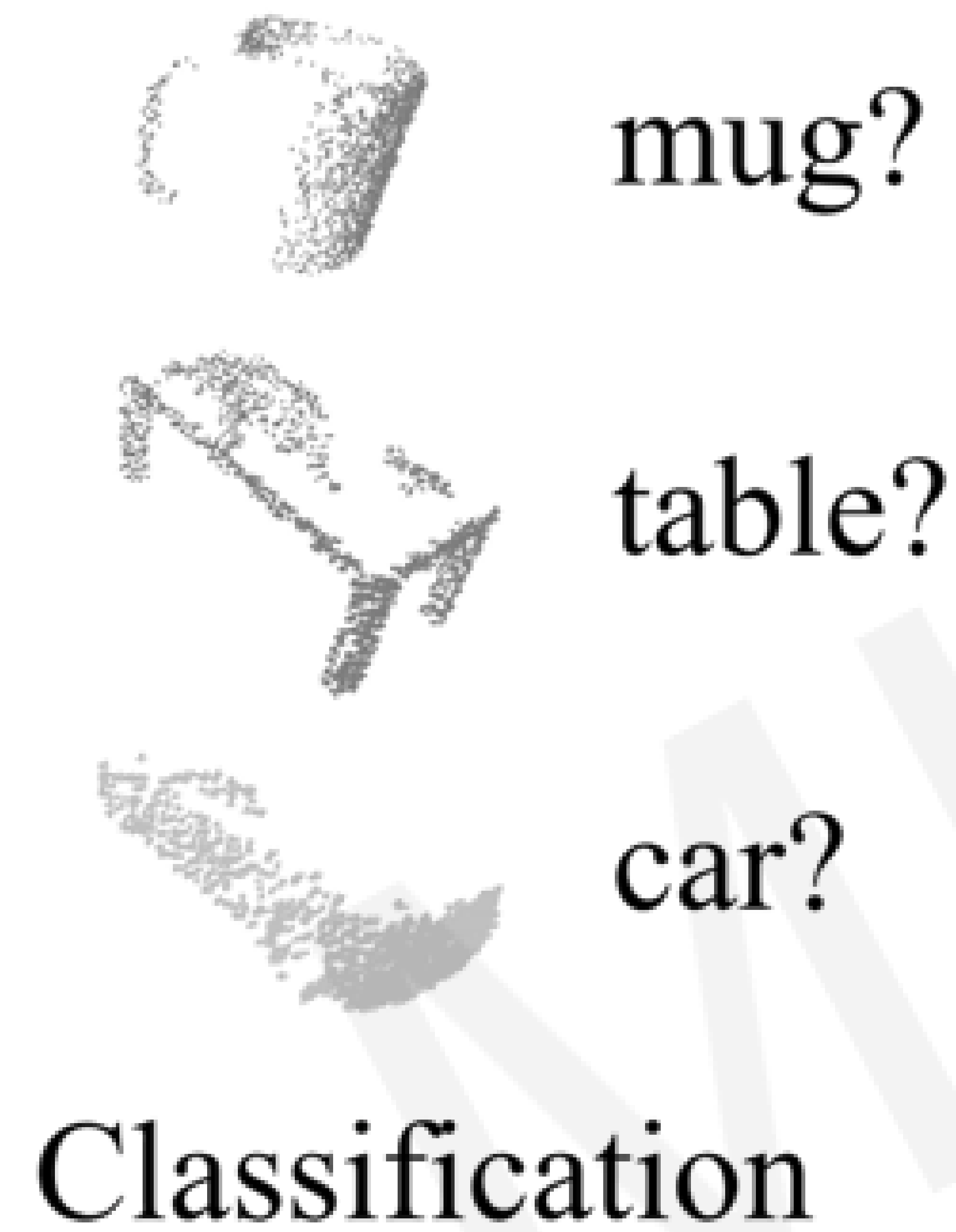
Graph Convolutional Networks (GCNs)



**Friday:** Graph neural networks for odor prediction  
Alex Wiltschko, Google Brain

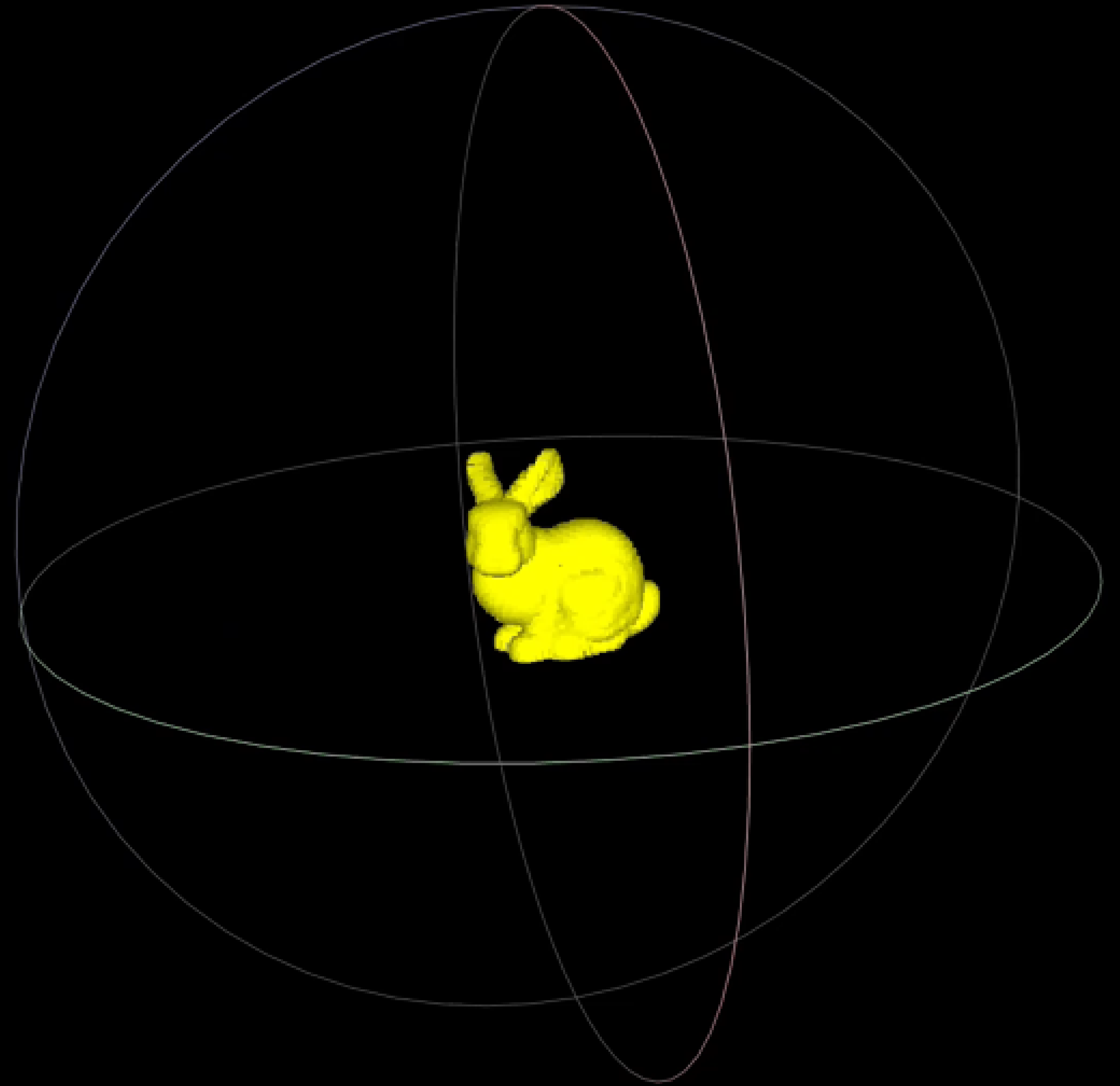
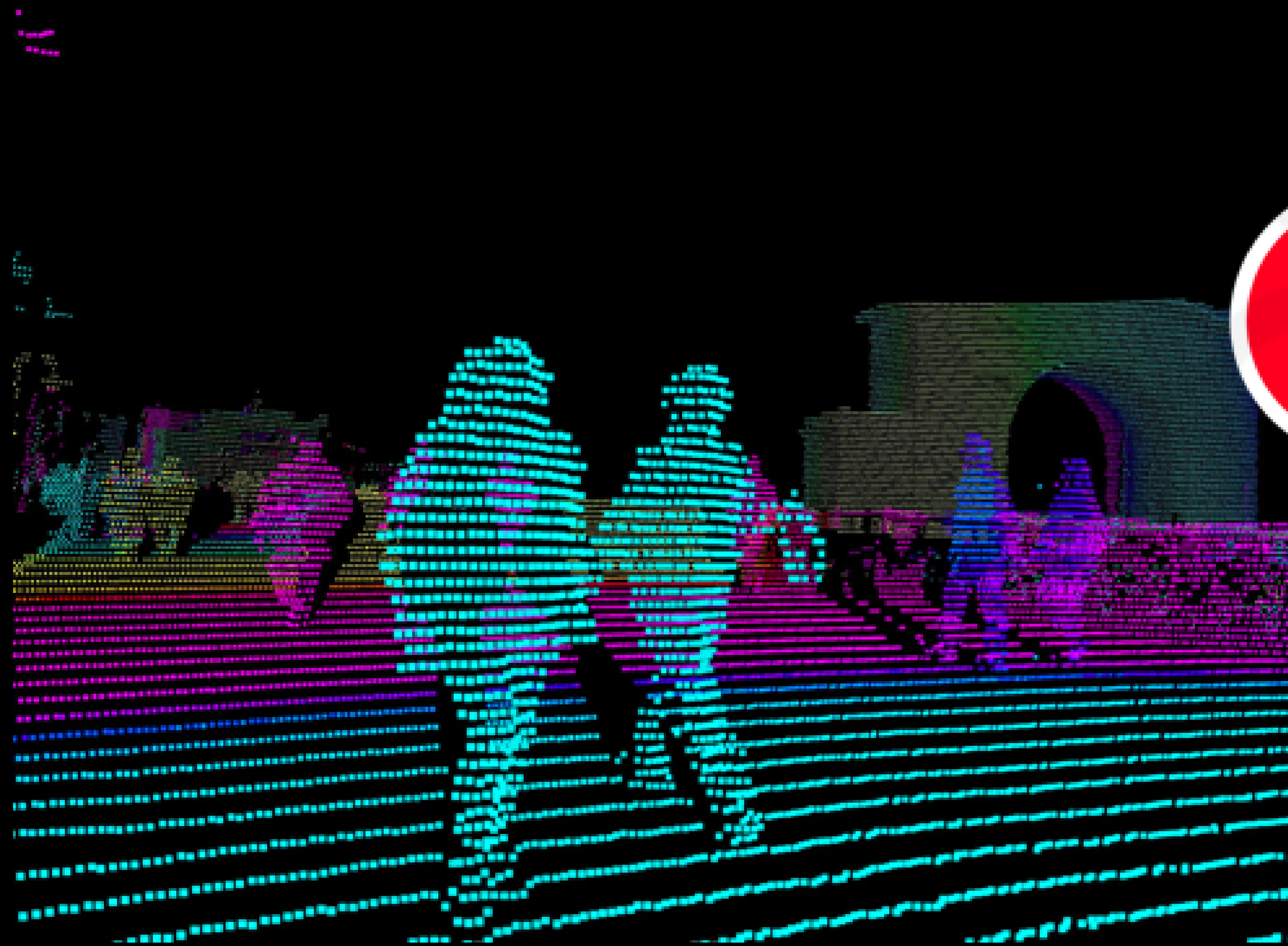
# Learning From 3D Data

Point clouds are **unordered sets** with **spatial dependence** between points



# Extending Graph CNNs to Pointclouds

Capture local geometric features of point clouds while maintaining order invariance



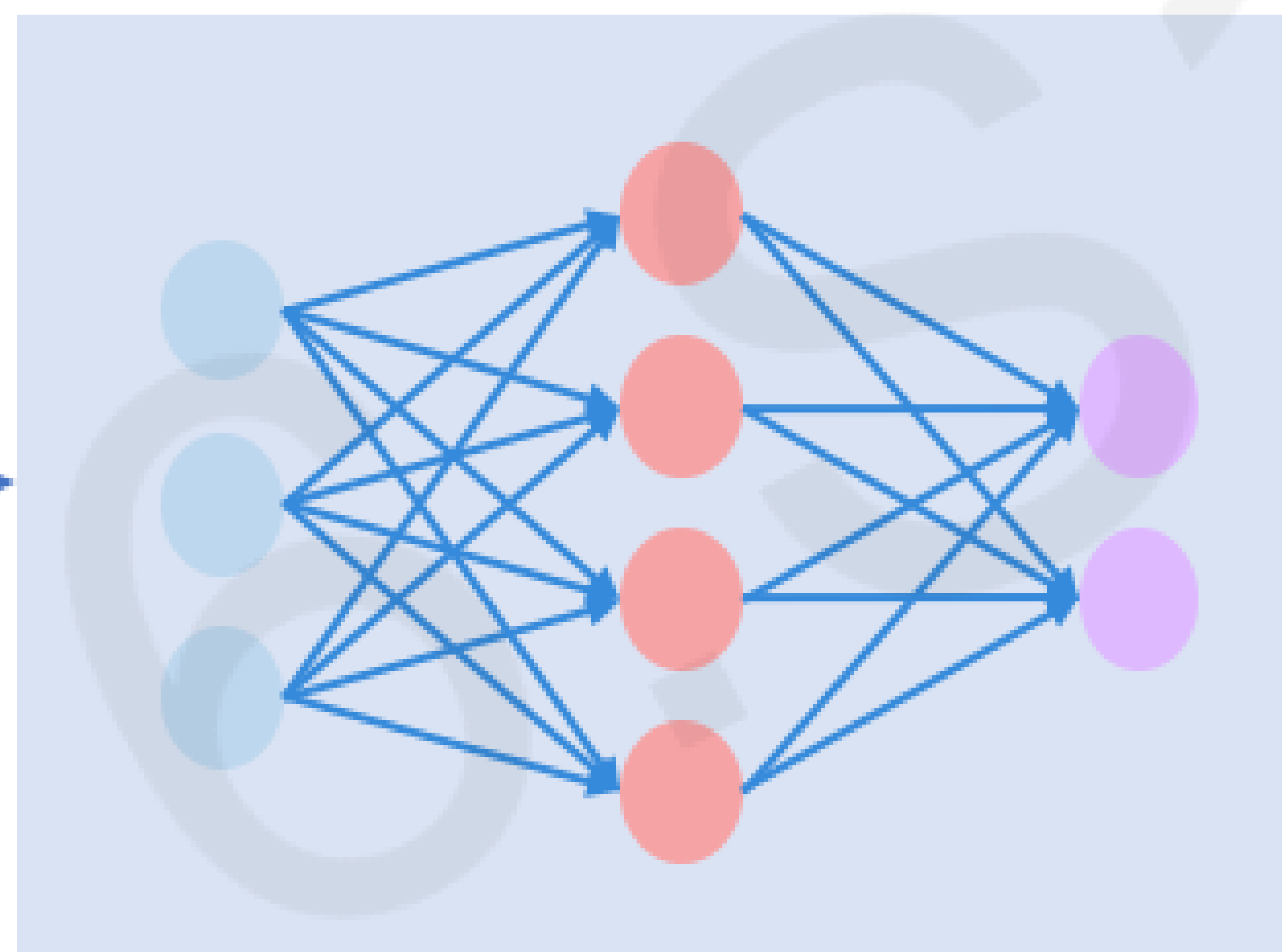
# New Frontiers II: Uncertainty Estimation & Bayesian Deep Learning



# Why care about uncertainty?



OR

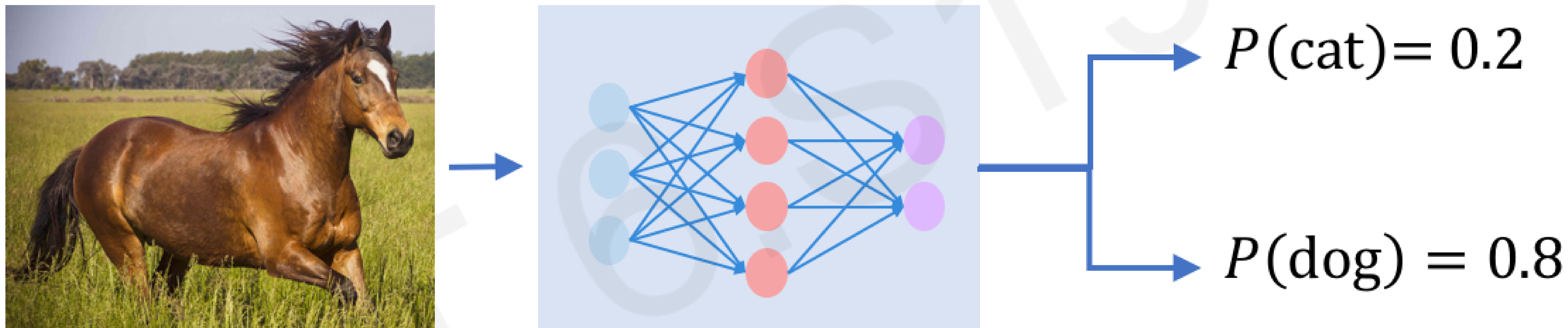


$P(\text{cat})$

$P(\text{dog})$

# Why care about uncertainty?

We need **uncertainty** metrics to assess the network's **confidence** in its predictions.



Remember:  $P(\text{cat}) + P(\text{dog}) = 1$

# Bayesian Deep Learning for Uncertainty

Network tries to learn output,  $\mathbf{Y}$ , directly from raw data,  $\mathbf{X}$

Find mapping,  $f$ , parameterized by weights  $\mathbf{W}$  such that

$$\min \mathcal{L}(\mathbf{Y}, f(\mathbf{X}; \mathbf{W}))$$

Bayesian neural networks aim to learn a posterior over weights,

$$P(\mathbf{W} | \mathbf{X}, \mathbf{Y}):$$

$$P(\mathbf{W} | \mathbf{X}, \mathbf{Y}) = \frac{P(\mathbf{Y} | \mathbf{X}, \mathbf{W}) P(\mathbf{W})}{P(\mathbf{Y} | \mathbf{X})}$$

# Bayesian Deep Learning for Uncertainty

Network tries to learn output,  $\mathbf{Y}$ , directly from raw data,  $\mathbf{X}$

Find mapping,  $f$ , parameterized by weights  $\mathbf{W}$  such that

$$\min \mathcal{L}(\mathbf{Y}, f(\mathbf{X}; \mathbf{W}))$$

Bayesian neural networks aim to learn a posterior over weights,

$$P(\mathbf{W} | \mathbf{X}, \mathbf{Y}):$$

Intractable!  $P(\mathbf{W} | \mathbf{X}, \mathbf{Y}) = \frac{P(\mathbf{Y} | \mathbf{X}, \mathbf{W}) P(\mathbf{W})}{P(\mathbf{Y} | \mathbf{X})}$

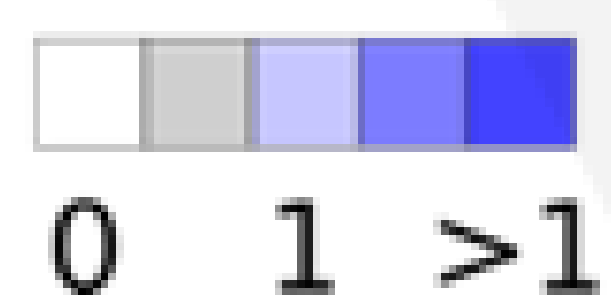
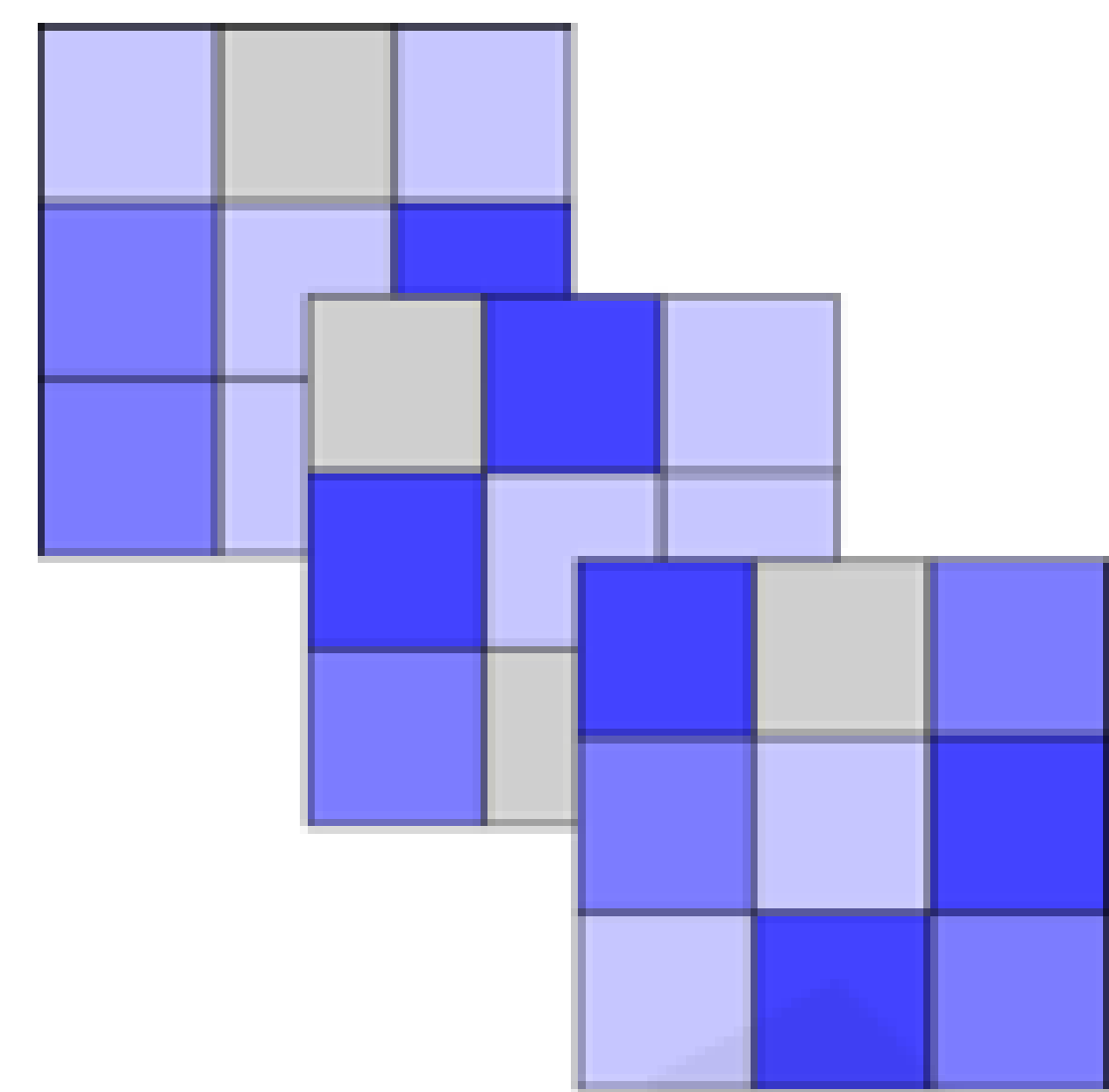
# Dropout for Uncertainty

Evaluate  $T$  stochastic forward passes through the network  $\{\mathbf{W}_t\}_{t=1}^T$

Dropout as a form of stochastic sampling  $z_{w,t} \sim \text{Bernoulli}(p) \quad \forall w \in \mathbf{W}$

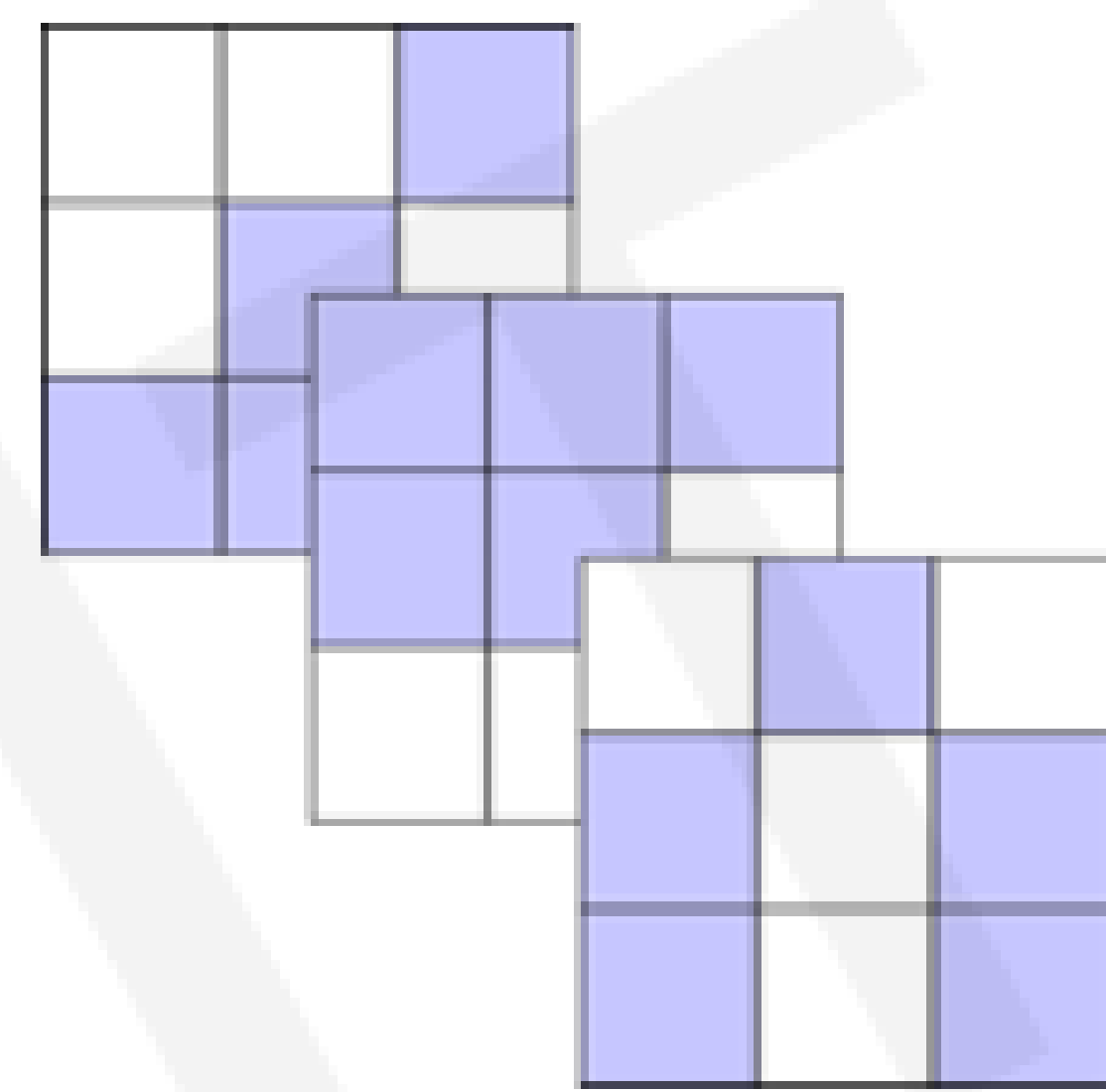
Unregularized Kernel

$\mathbf{W}$



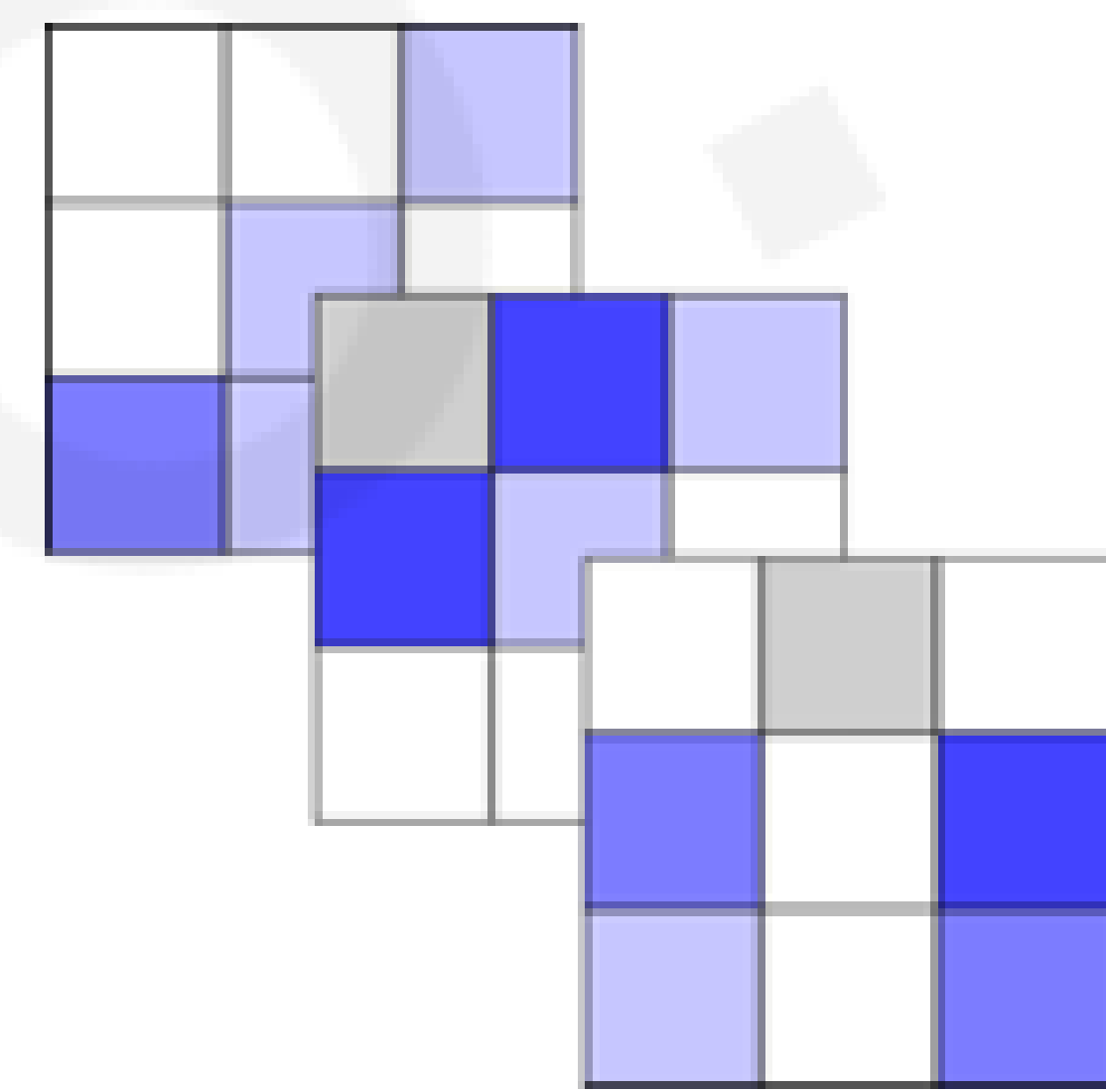
Bernoulli Dropout

$z_{w,t}$



Stochastic Sampled

$\mathbf{W}_t$



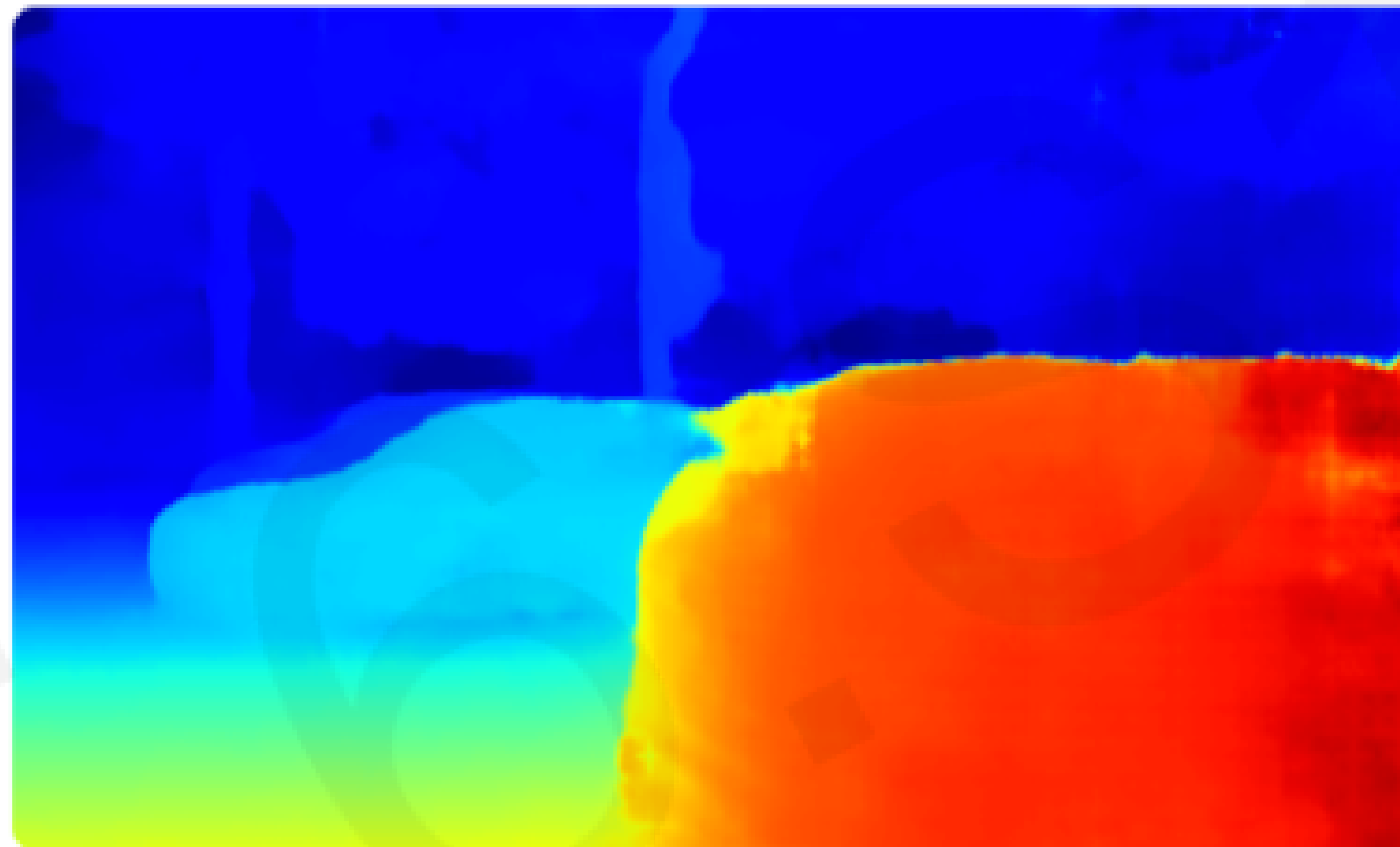
$$\mathbb{E}(\hat{\mathbf{Y}}|\mathbf{X}) = \frac{1}{T} \sum_{t=1}^T f(\mathbf{X}|\mathbf{W}_t)$$

$$\text{Var}(\hat{\mathbf{Y}}|\mathbf{X}) = \frac{1}{T} \sum_{t=1}^T f(\mathbf{X})^2 - \mathbb{E}(\hat{\mathbf{Y}}|\mathbf{X})^2$$

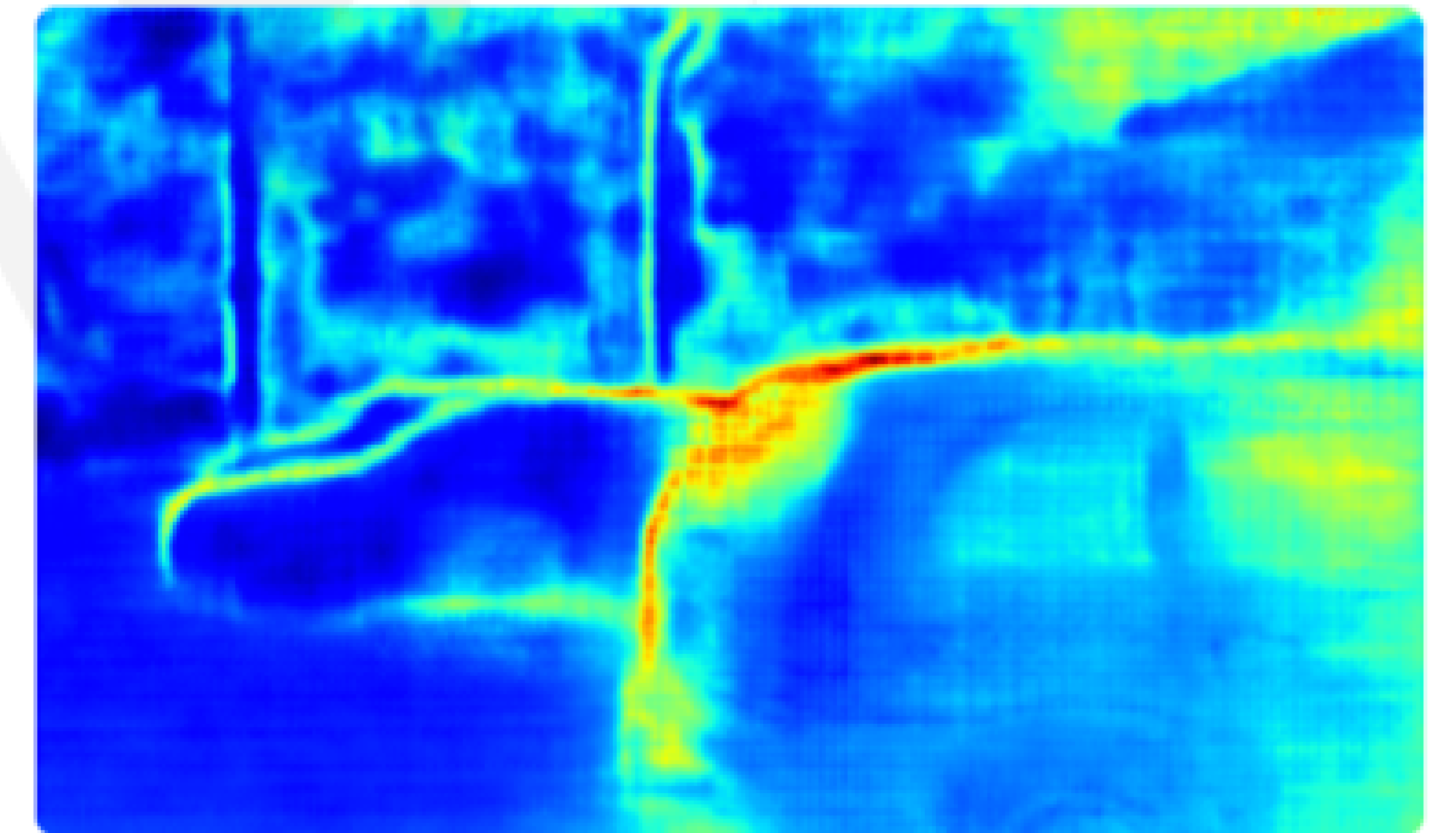
# Model Uncertainty Application



Input Image



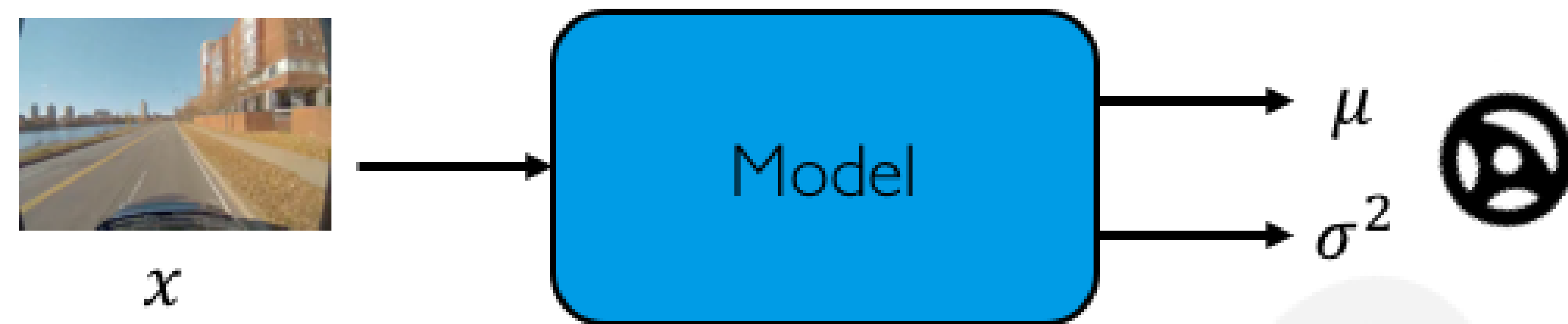
Predicted Depth



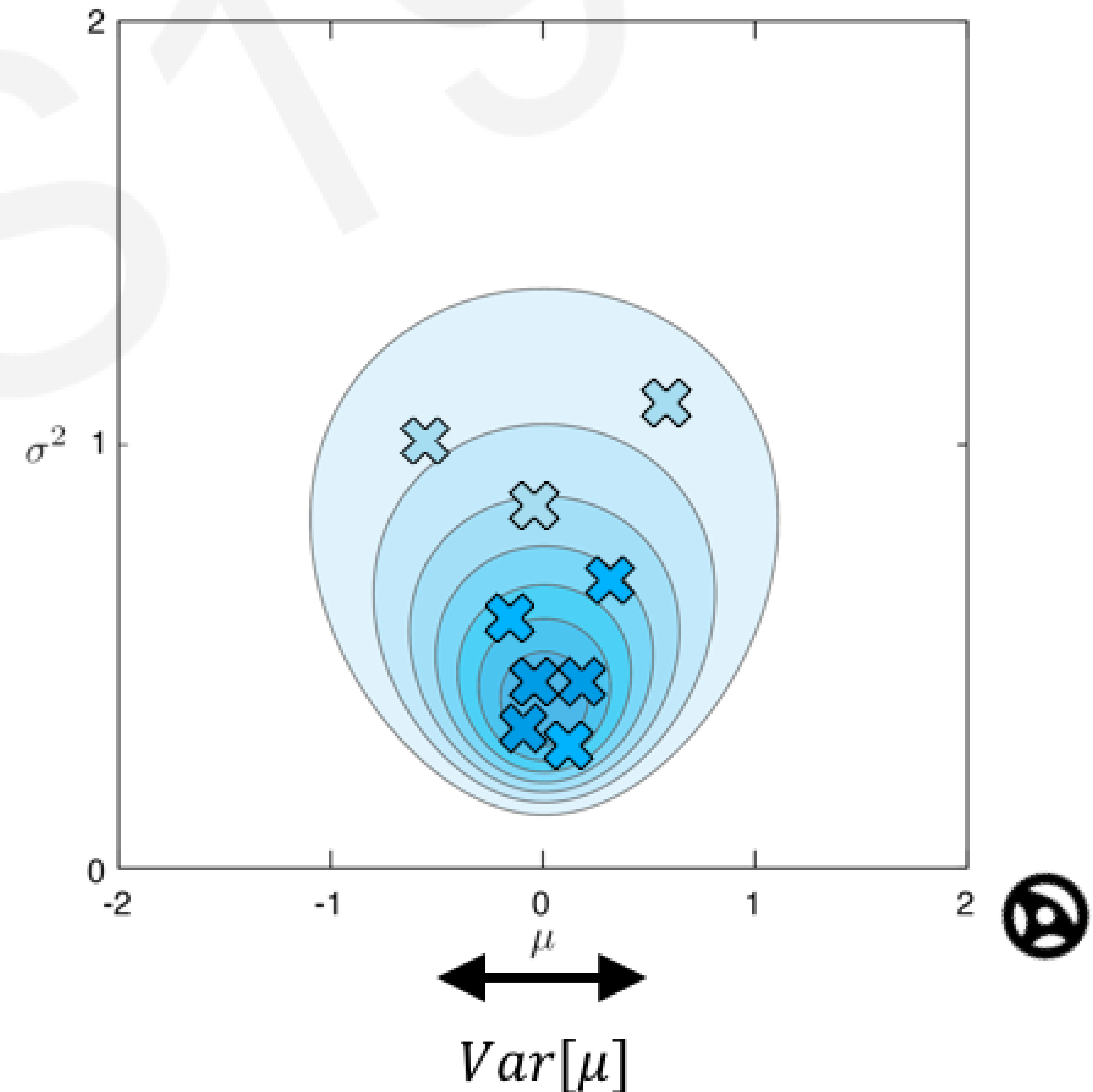
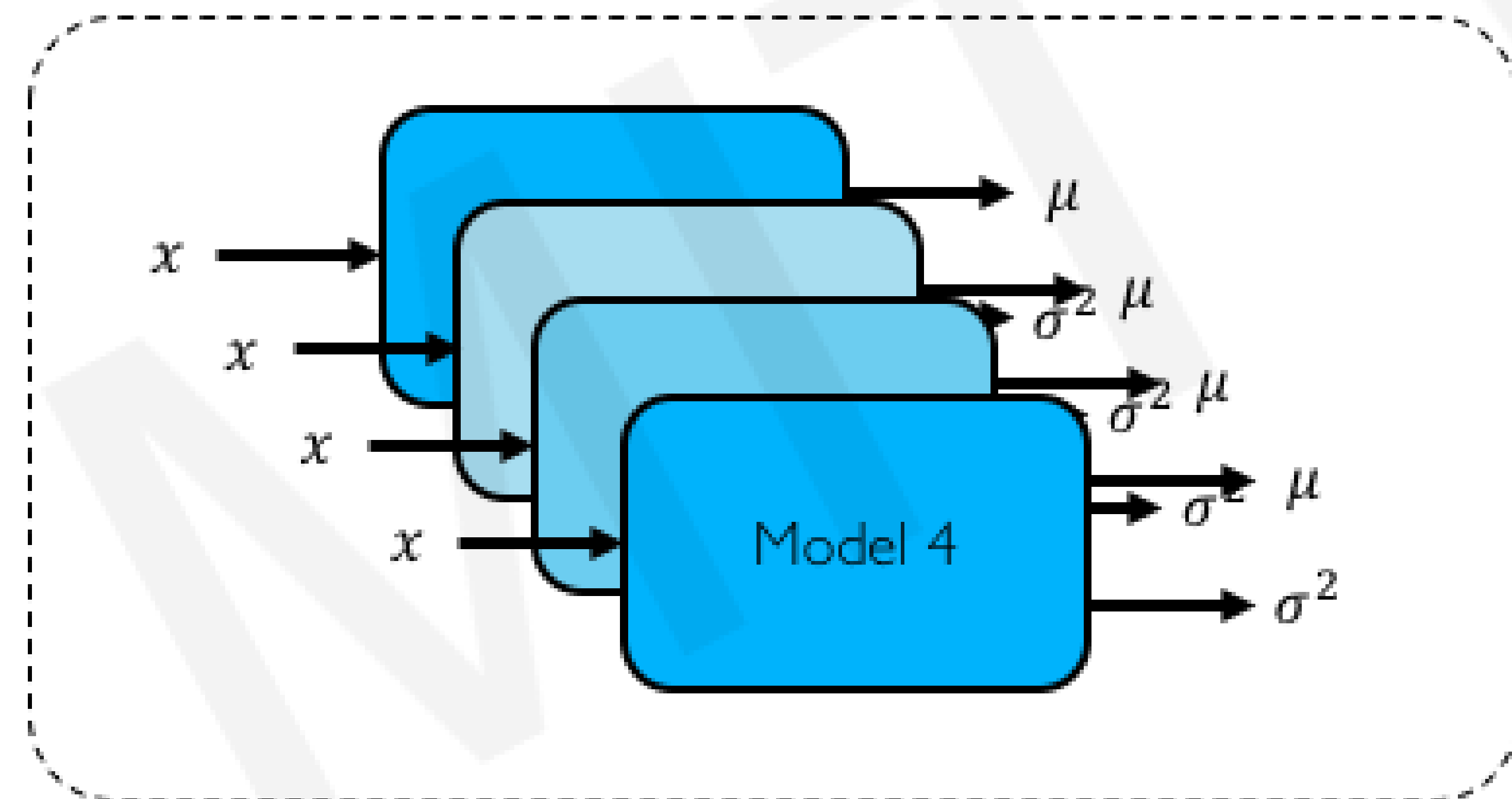
Model Uncertainty

# Uncertainty Estimation via Ensembling

Model ensembling for estimating uncertainty



Ensemble

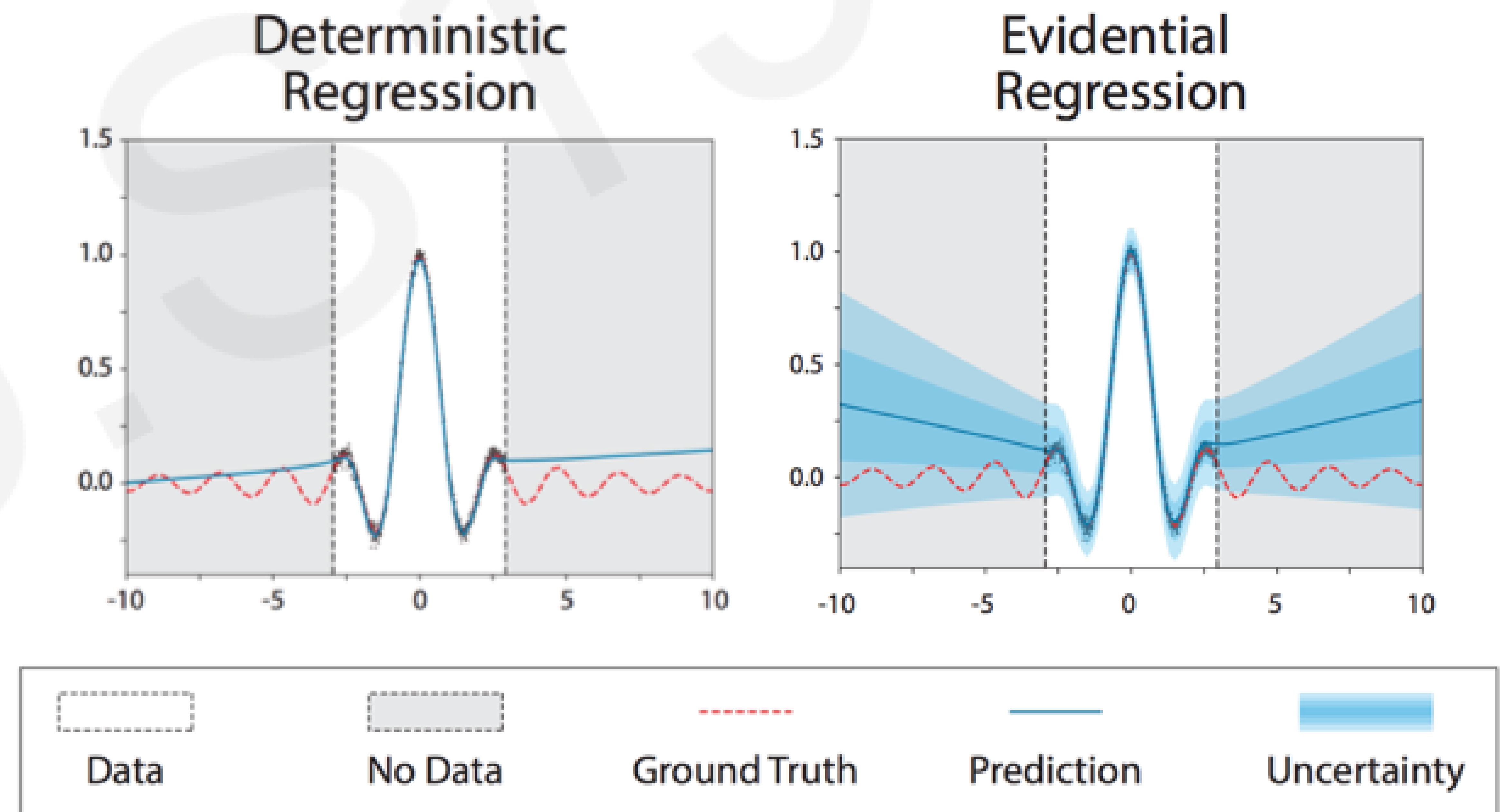
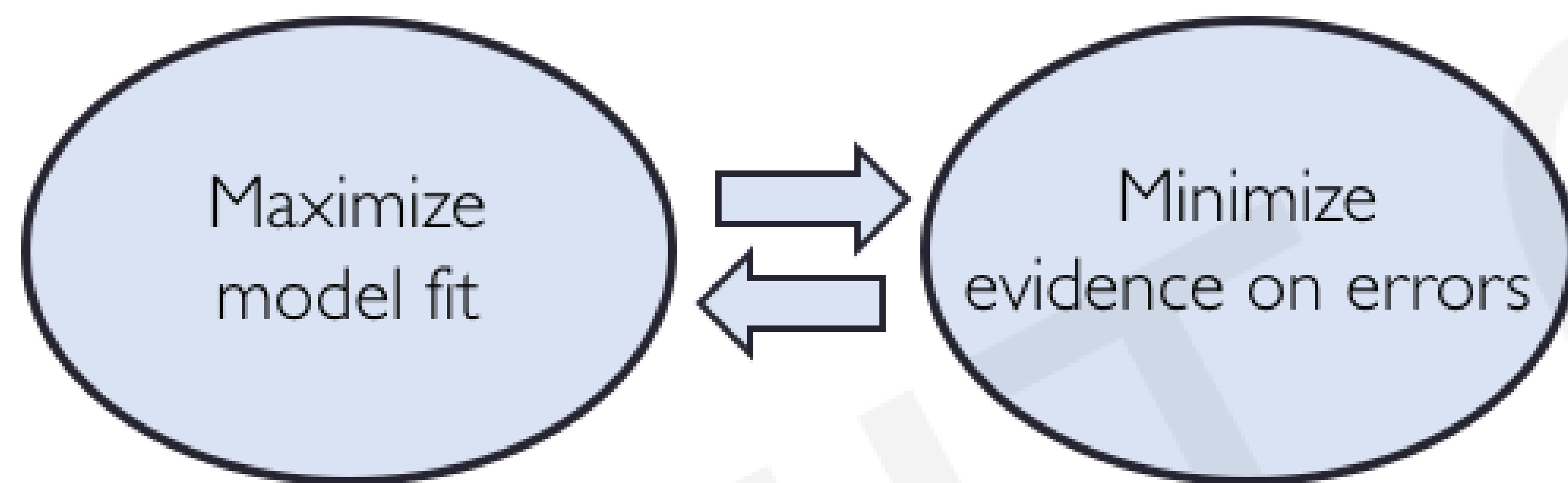




# Evidential Deep Learning

Directly learn the underlying uncertainties using **evidential distributions**

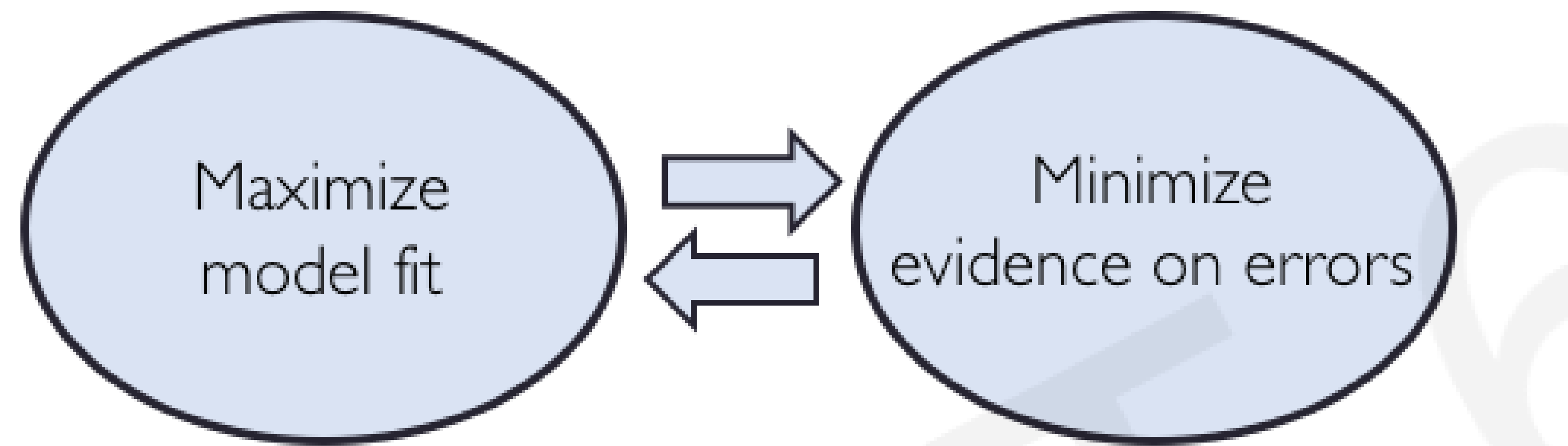
Competing loss training:



# Evidential Deep Learning

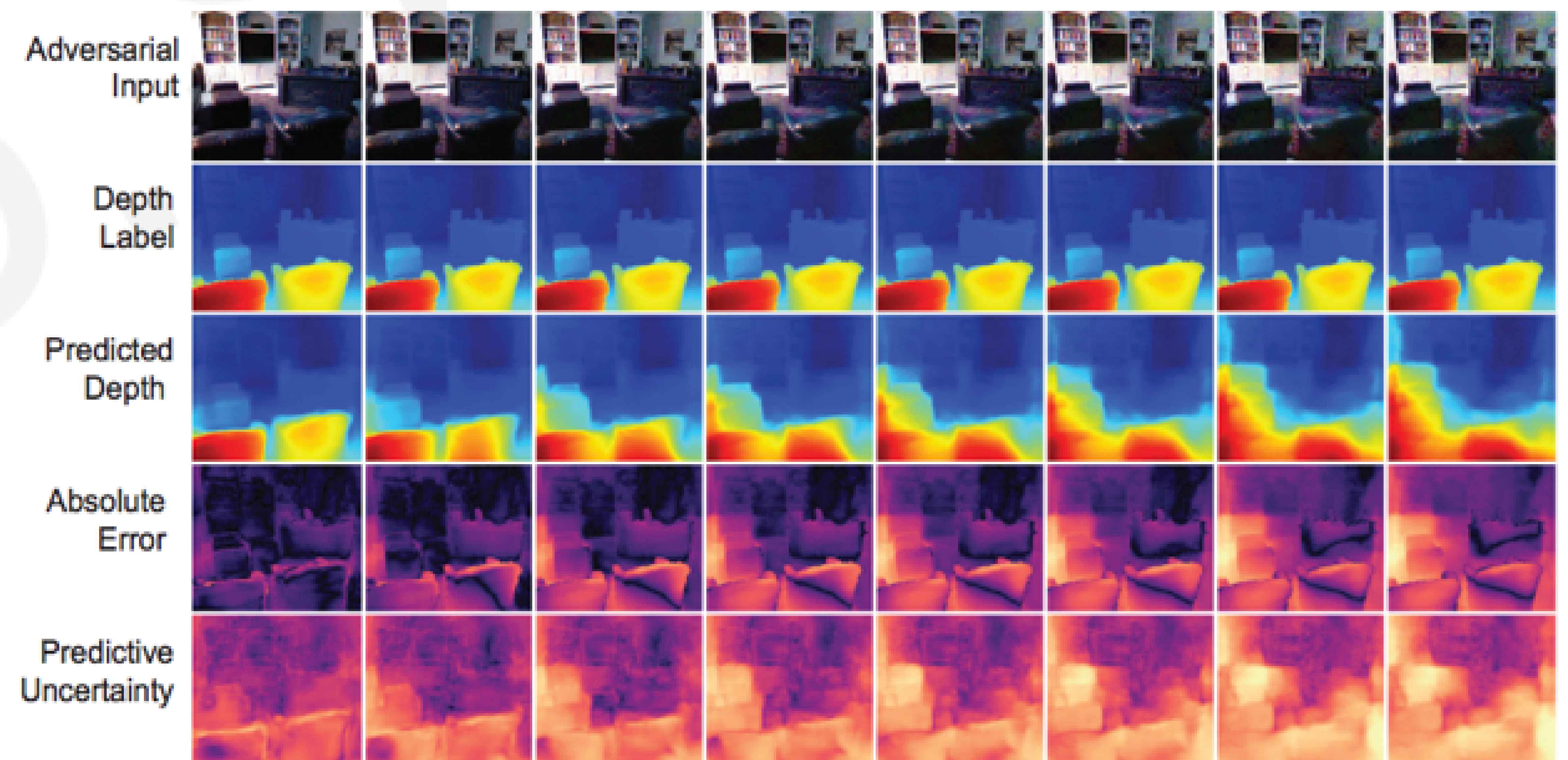
Directly learn the underlying uncertainties using evidential distributions

### Competing loss training:

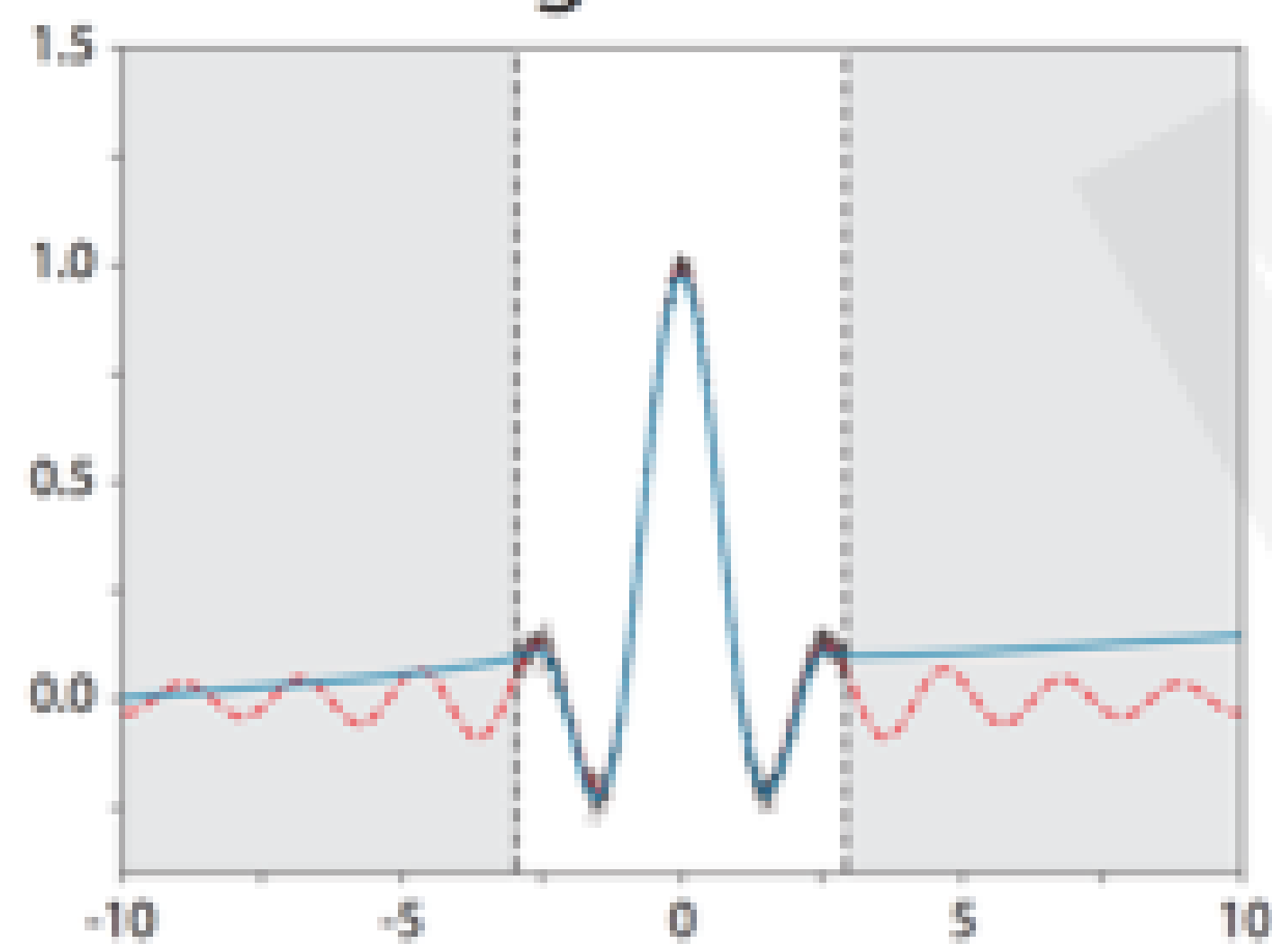


### Robustness to adversarial perturbation

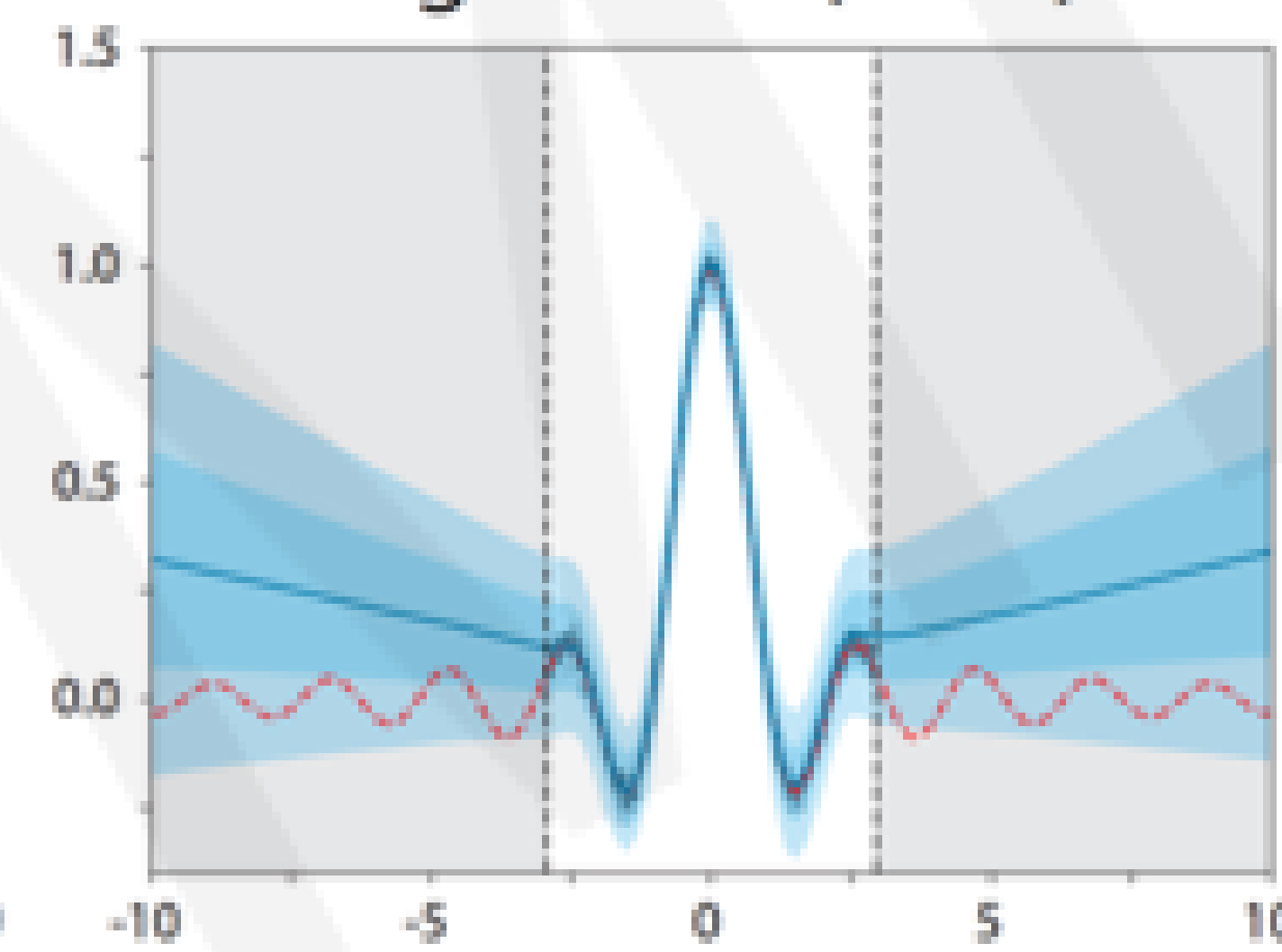
$\epsilon = \frac{1}{80}$     $\epsilon = \frac{2}{80}$     $\epsilon = \frac{3}{80}$     $\epsilon = \frac{4}{80}$     $\epsilon = \frac{5}{80}$     $\epsilon = \frac{6}{80}$     $\epsilon = \frac{7}{80}$     $\epsilon = \frac{8}{80}$



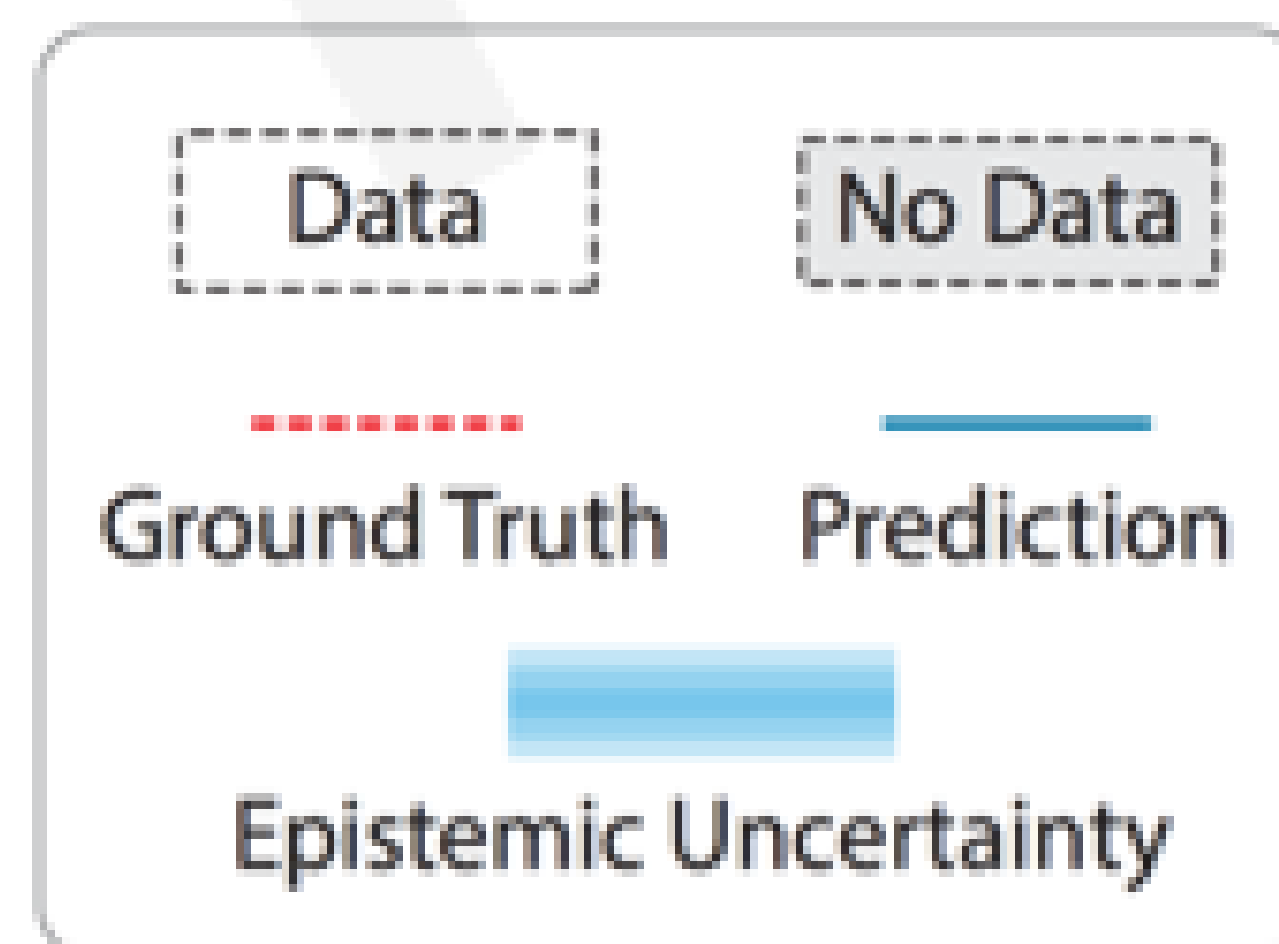
### Deterministic Regression



### Evidential Regression (Ours)

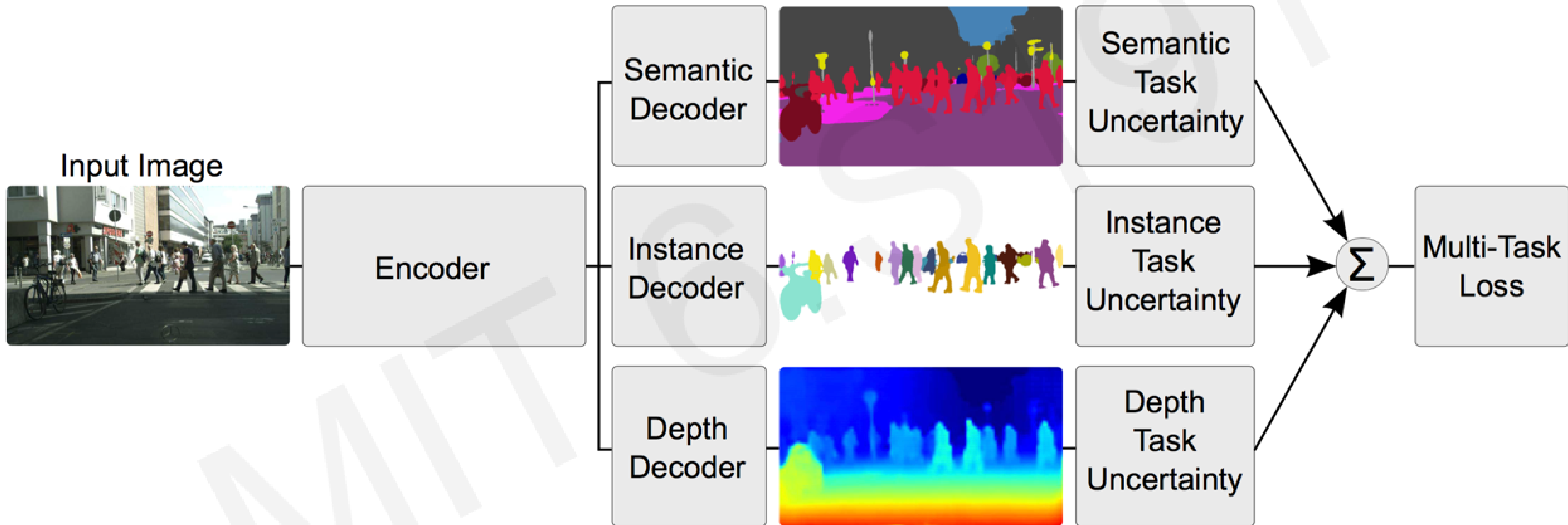


### Legend

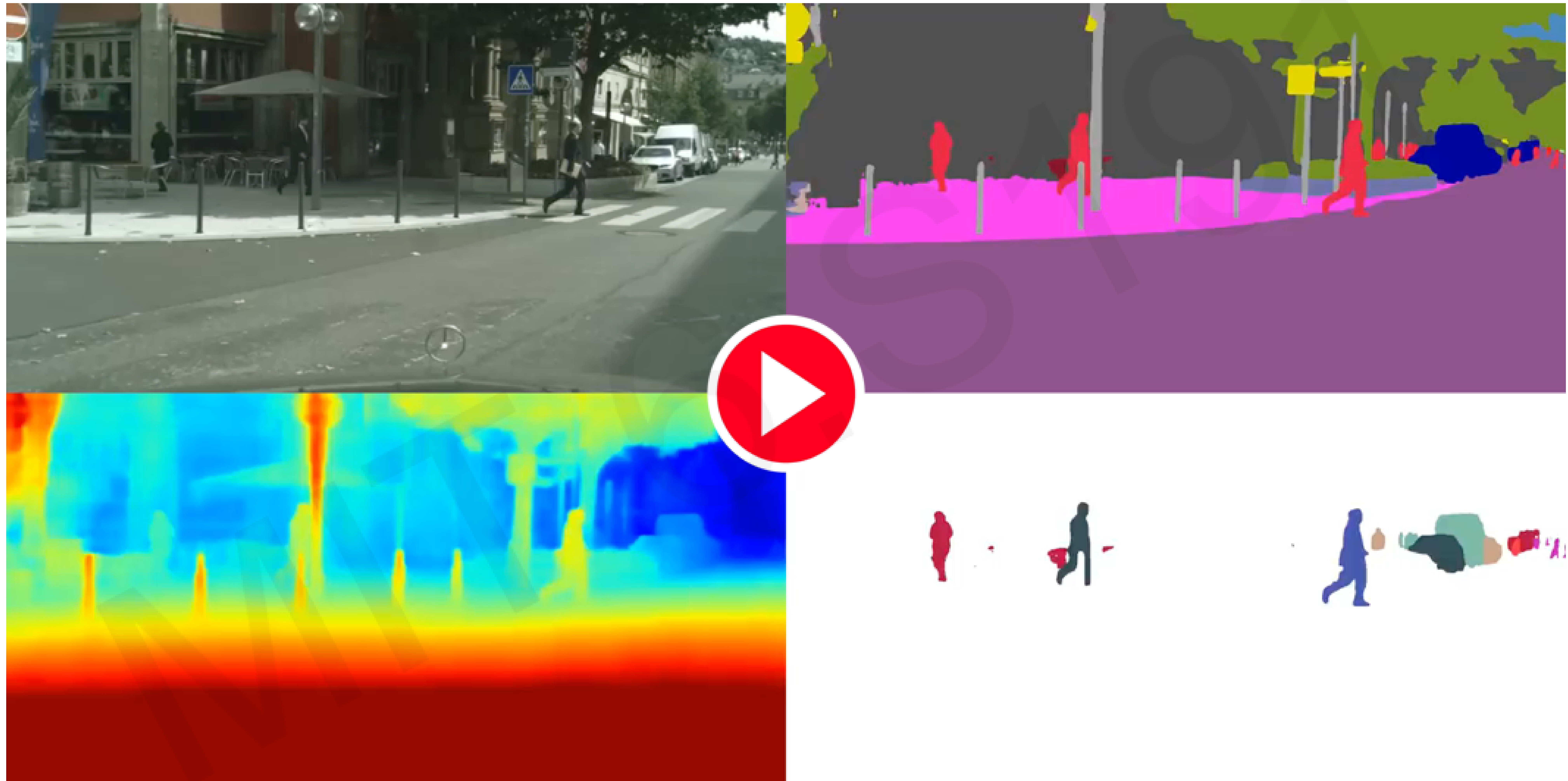


Increasing Adversarial Perturbation

# Multi-Task Learning Using Uncertainty



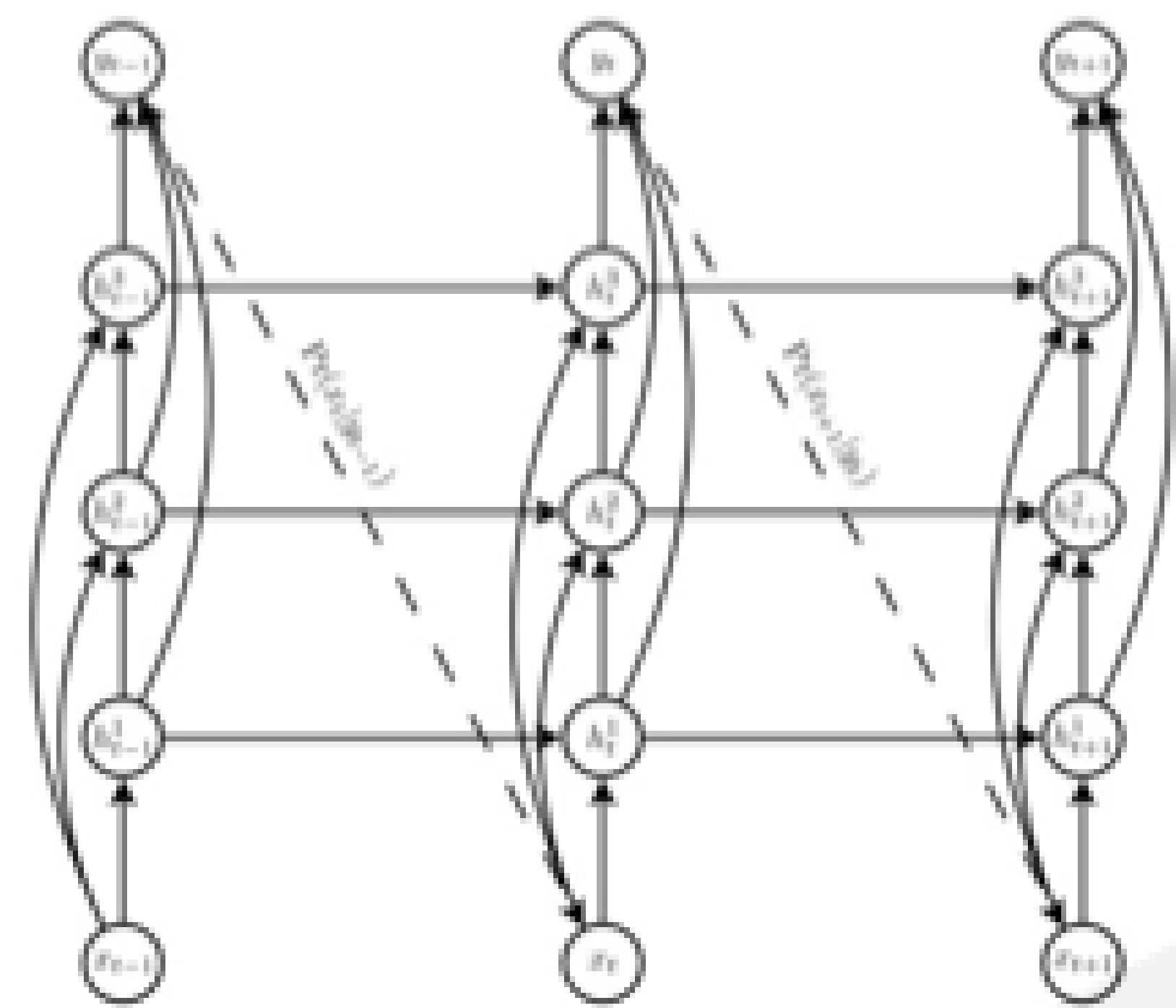
# Multi-Task Learning Using Uncertainty



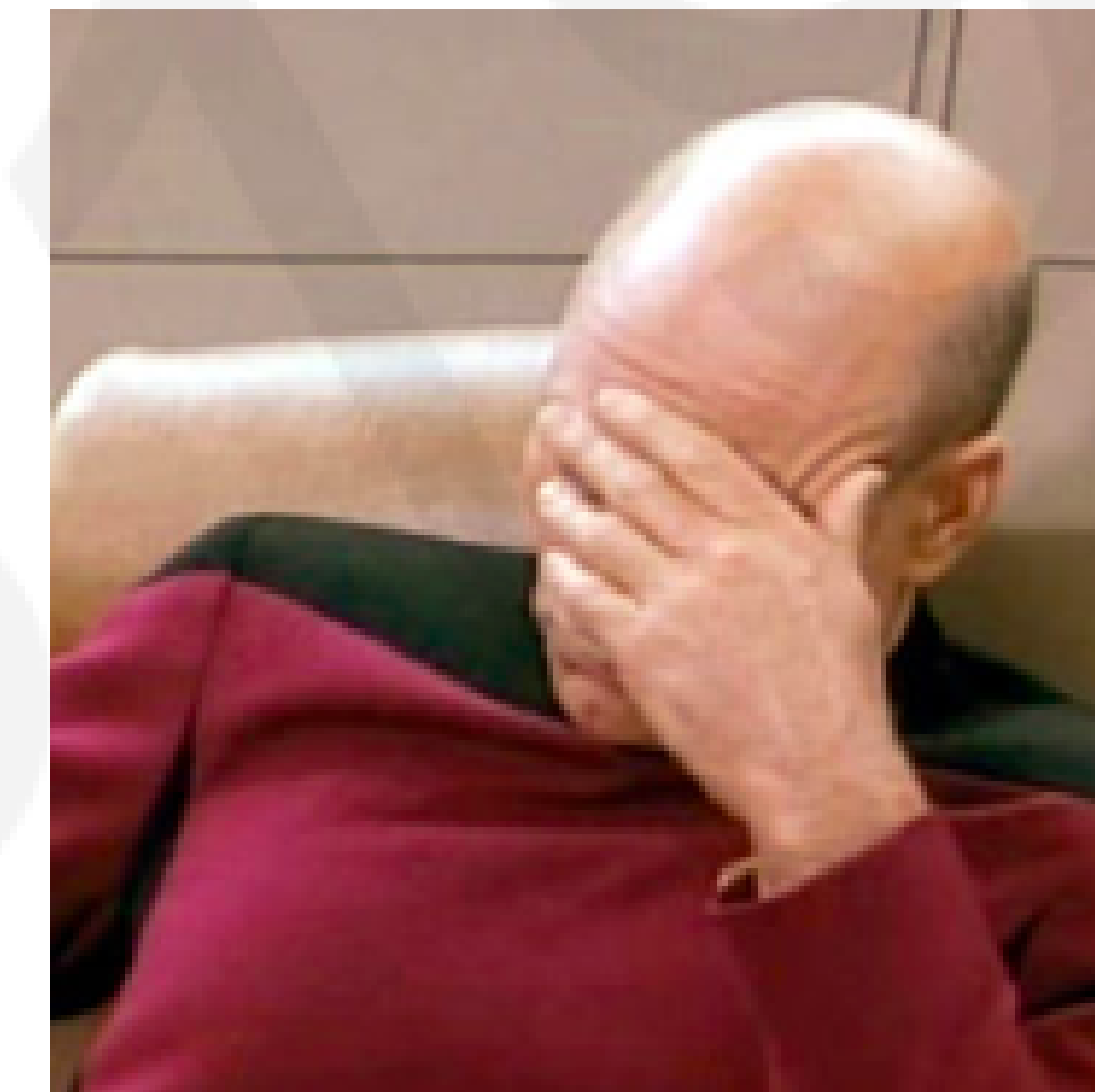
# New Frontiers III: Automated Machine Learning

# Motivation: Automated Machine Learning

Standard deep neural networks are optimized for **a single task**



Complexity of models increases



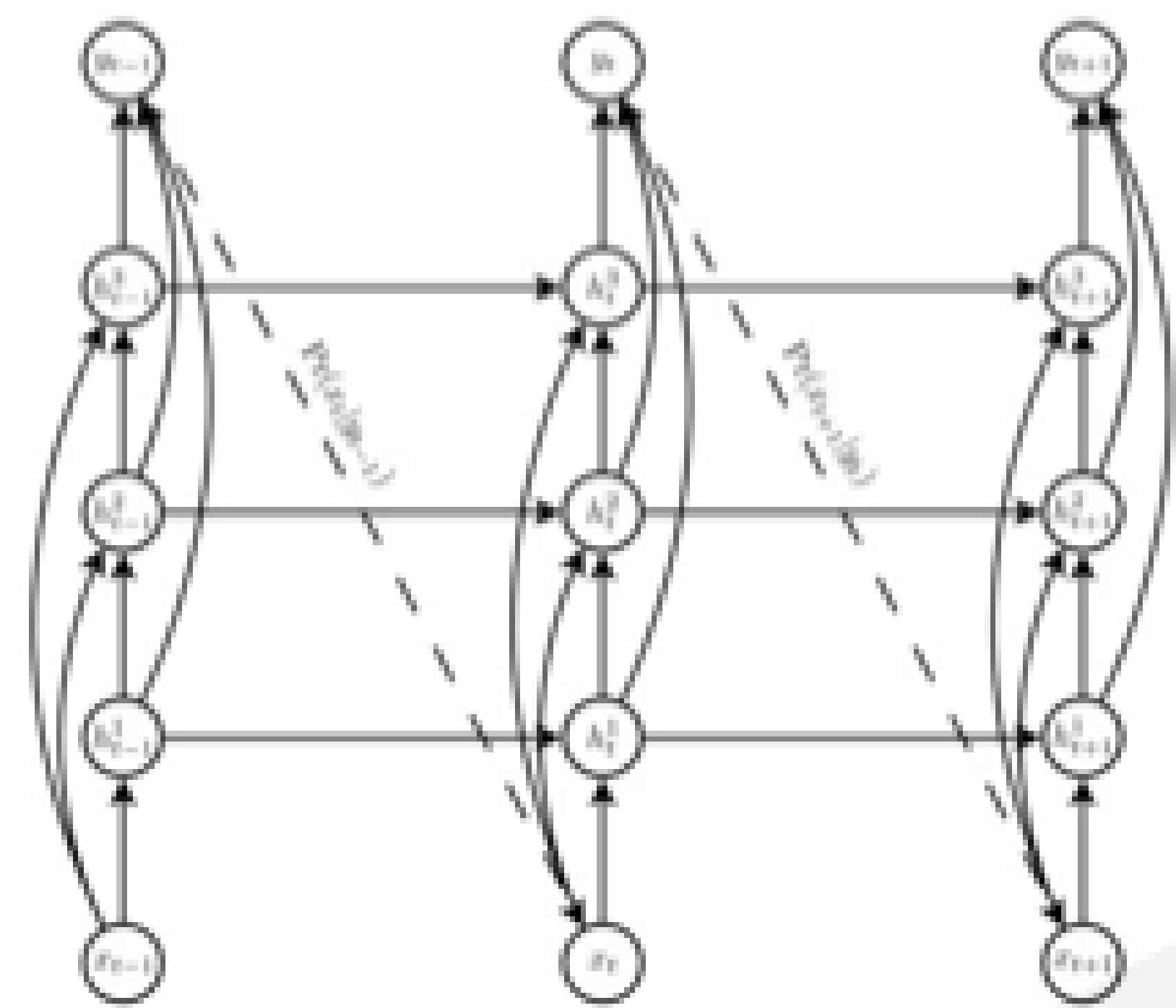
Greater need for specialized engineers

Often require **expert knowledge** to build an architecture for a given task

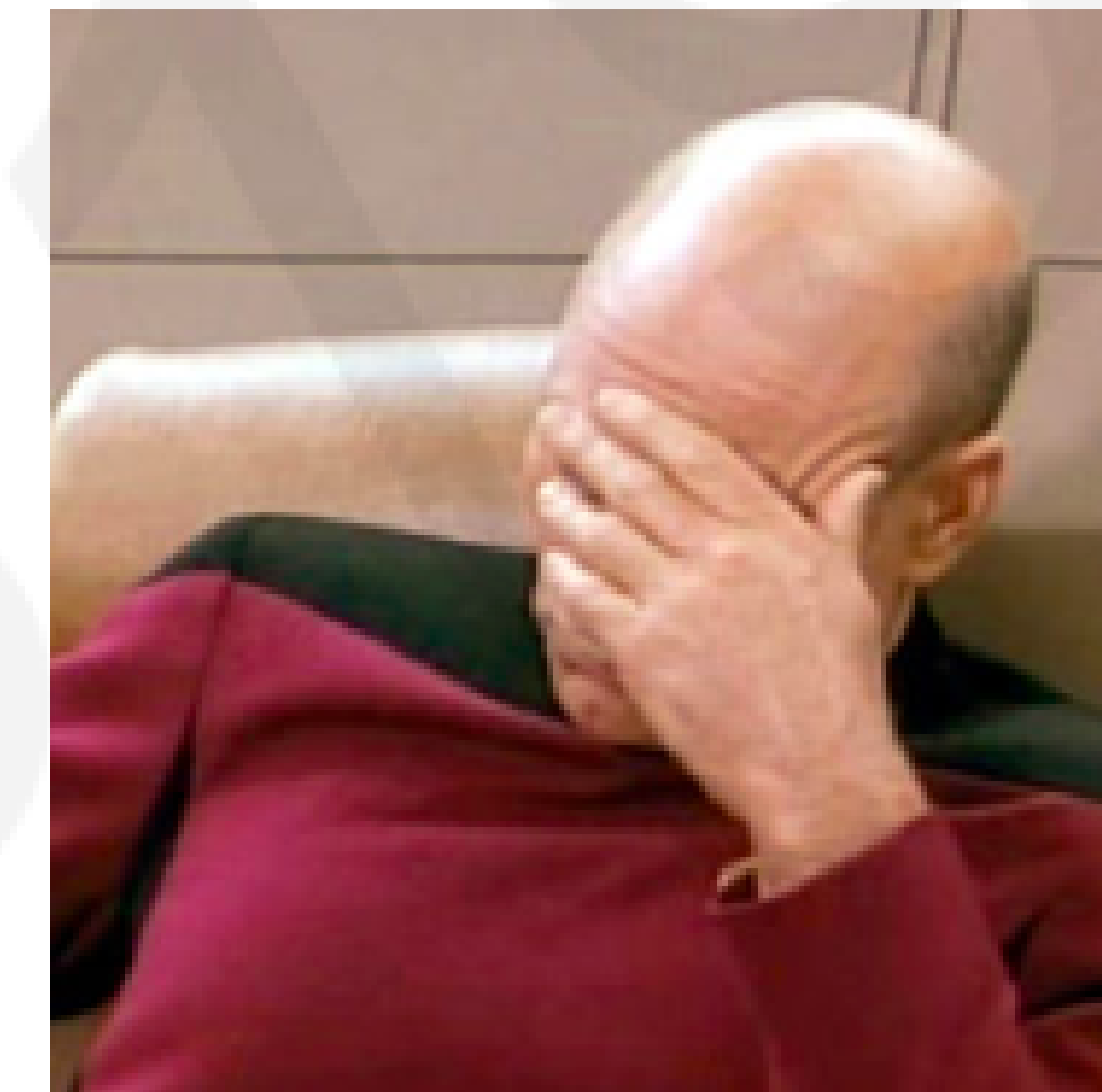


# Motivation: Automated Machine Learning

Standard deep neural networks are optimized for **a single task**



Complexity of models increases



Greater need for specialized engineers

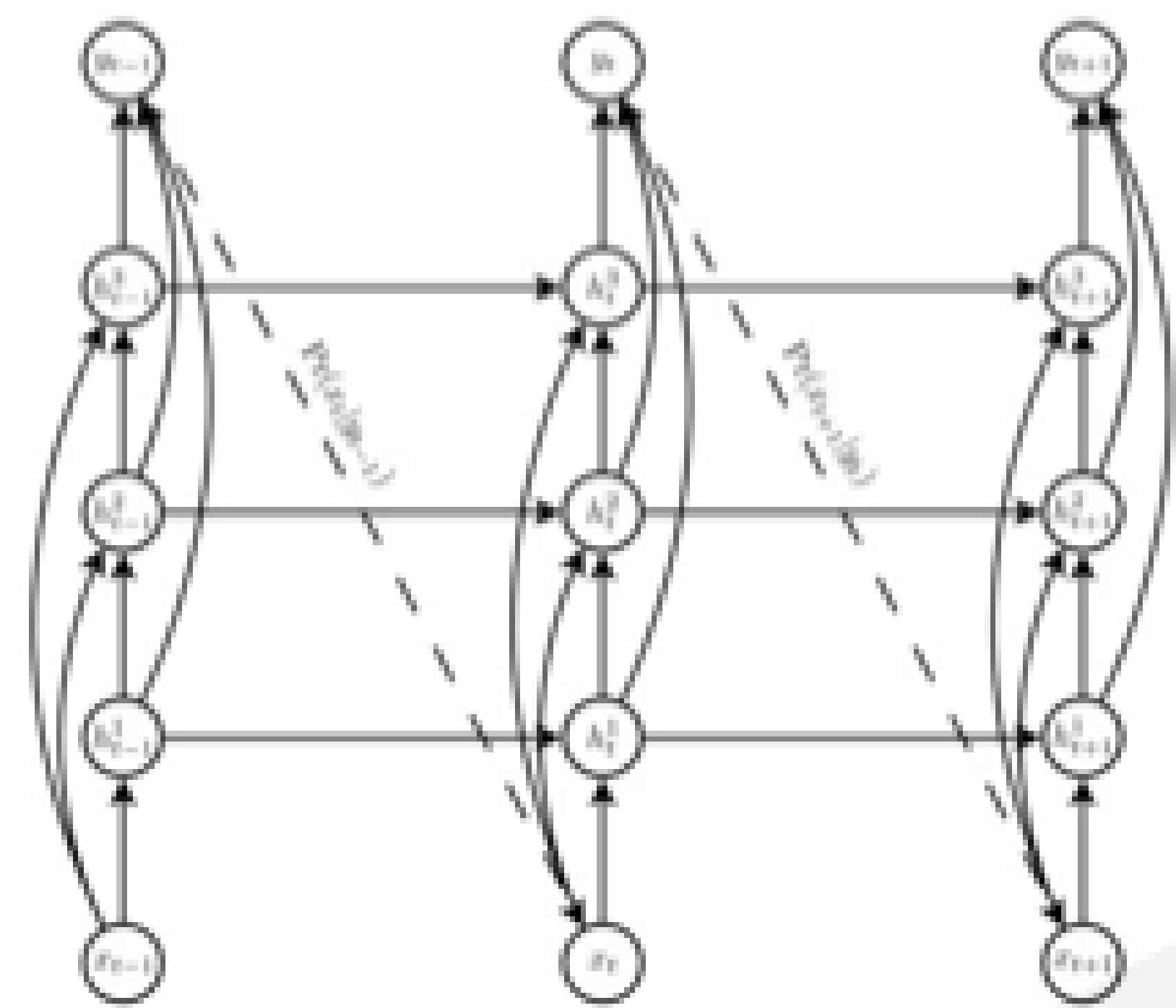
Often require **expert knowledge** to build an architecture for a given task

Build a learning algorithm that **learns which model** to use to solve a given problem

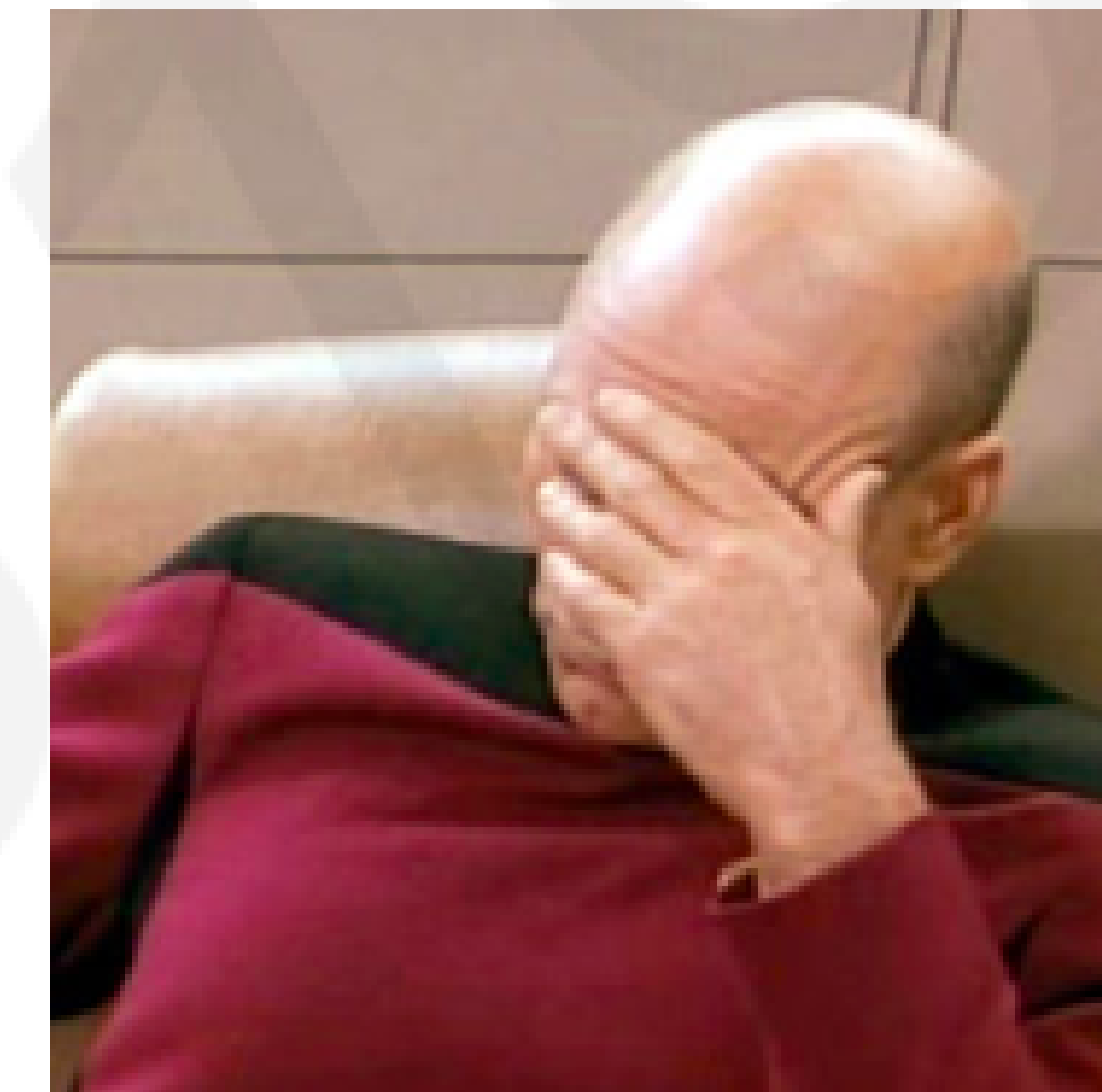


# Motivation: Automated Machine Learning

Standard deep neural networks are optimized for **a single task**



Complexity of models increases



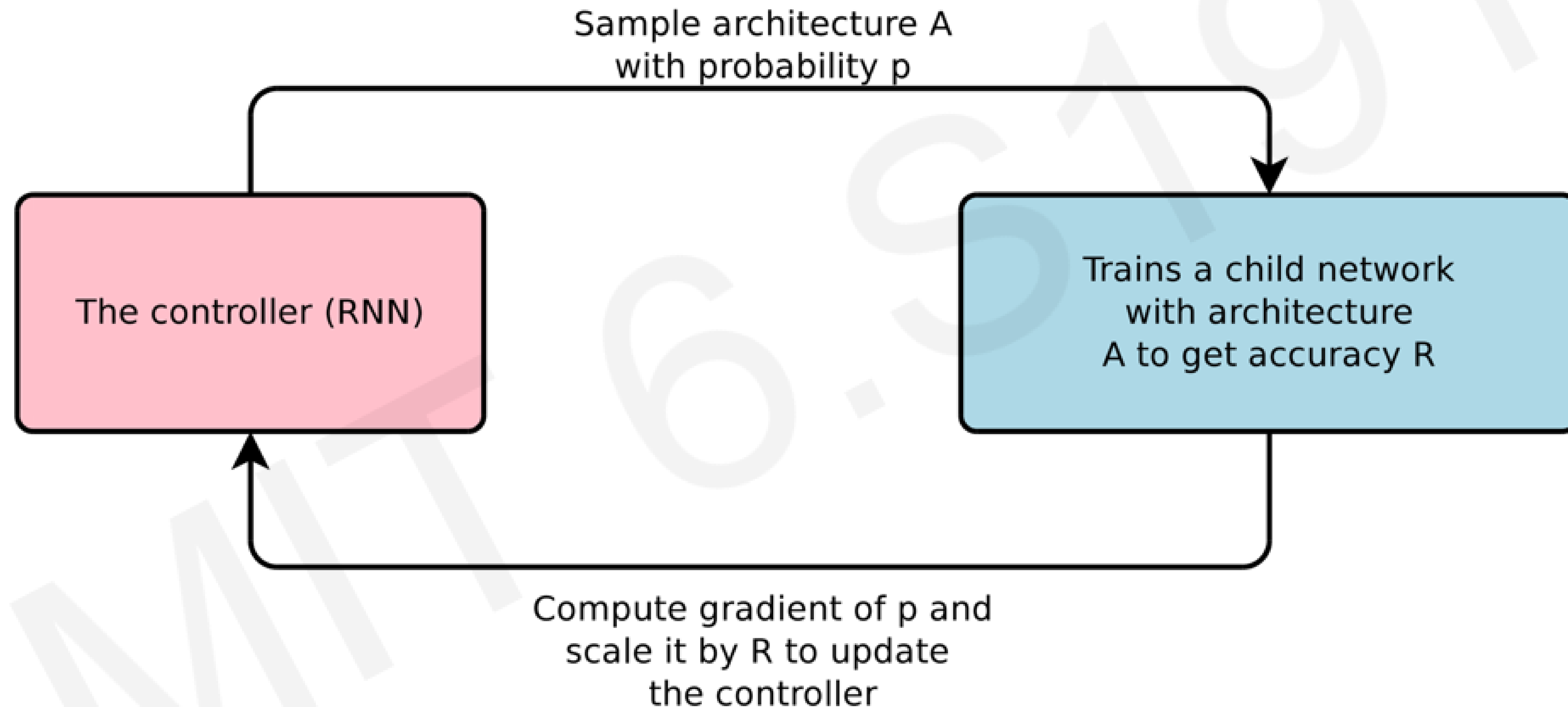
Greater need for specialized engineers

Often require **expert knowledge** to build an architecture for a given task

Build a learning algorithm that **learns which model** to use to solve a given problem

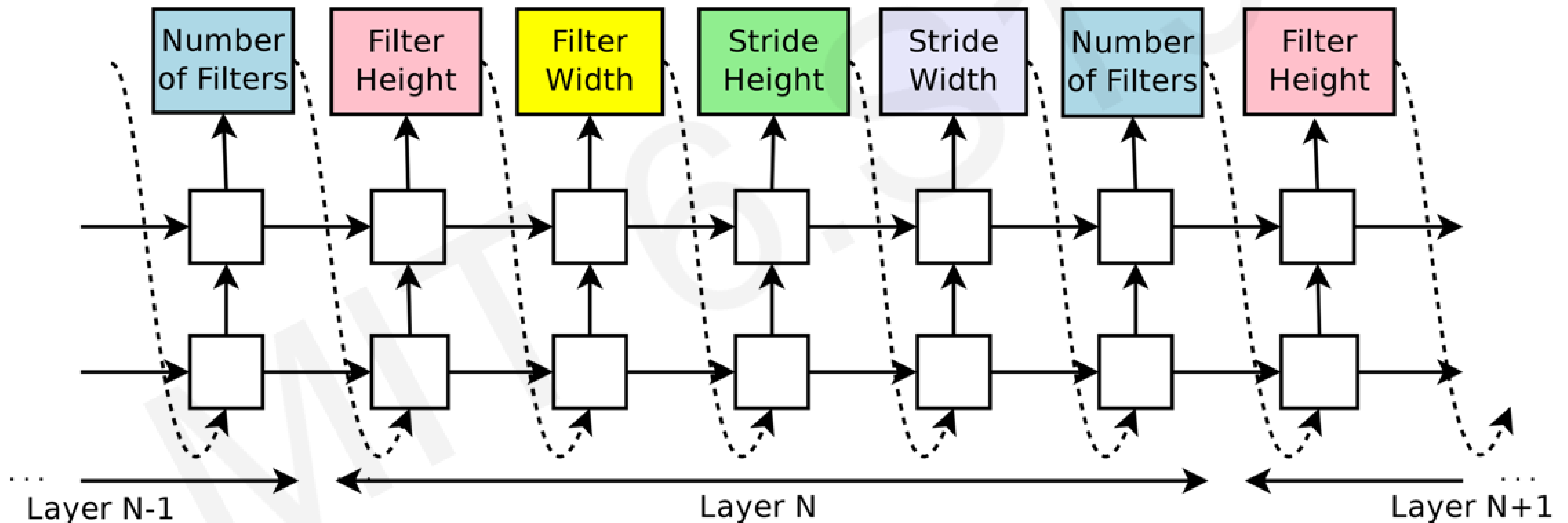
## AutoML

# Automated Machine Learning (AutoML)

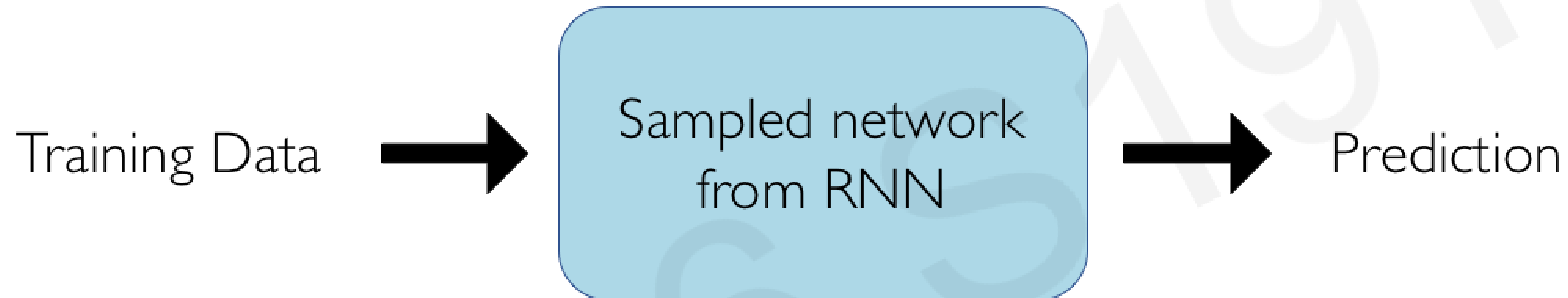


# AutoML: Model Controller

At each step, the model samples a brand new network



# AutoML: The Child Network



Compute final accuracy on this dataset.

Update RNN controller based on the accuracy of the child network after training.

# AutoML on the Cloud



## AutoML Vision<sup>BETA</sup>

Start with as little as a few dozen photographic samples, and Cloud AutoML will do the rest.



## AutoML Natural Language<sup>BETA</sup>

Automatically predict text categories through either single or multi-label classification.



## AutoML Translation<sup>BETA</sup>

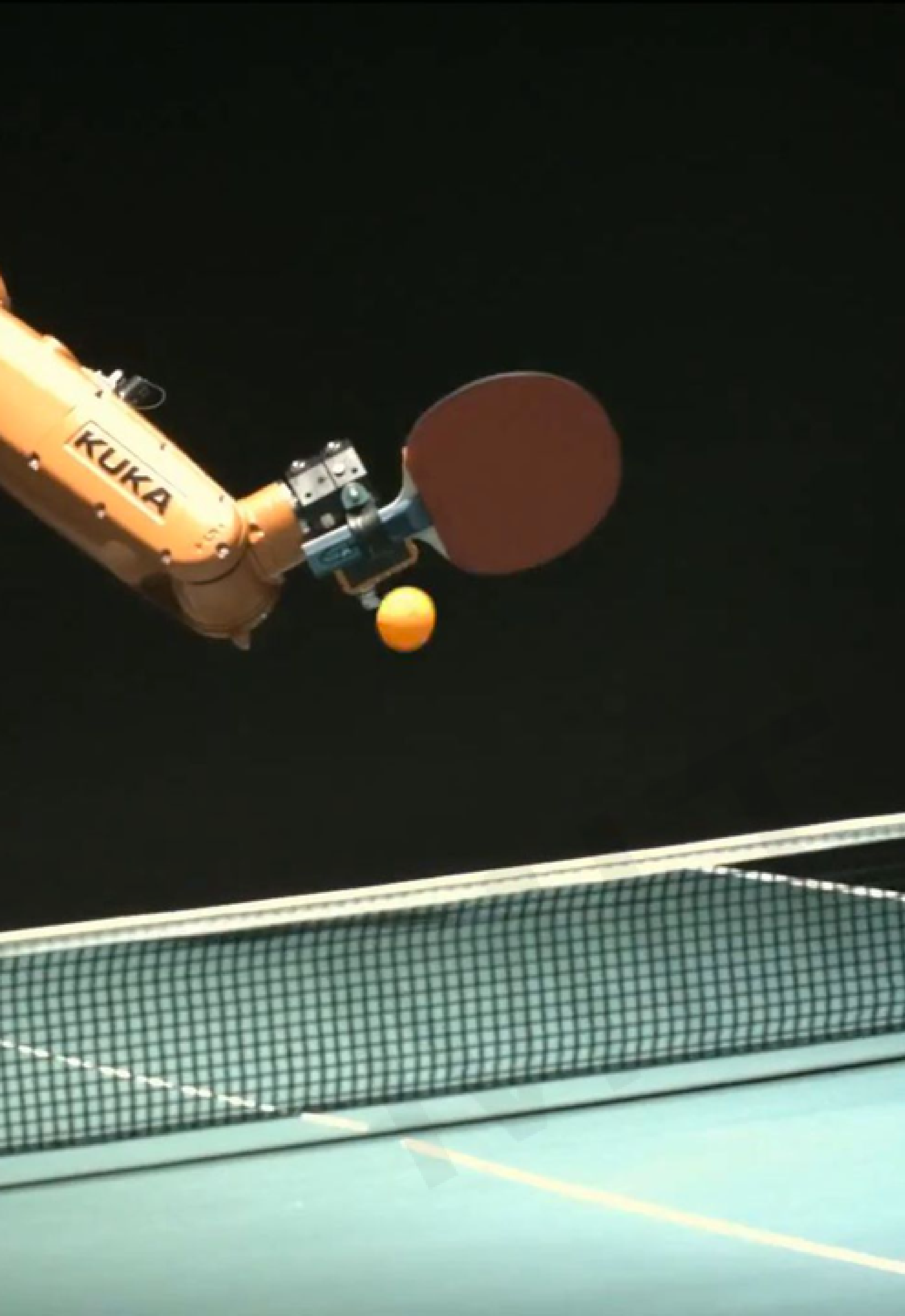
Upload translated language pairs to train your own custom model.



# AutoML Spawns a Powerful Idea

- Design an AI algorithm that can build new models capable of solving a task
- Reduces the need for experienced engineers to design the networks
- Makes deep learning more accessible to the public

Connections and distinctions  
between artificial and human  
intelligence



# 6.S191: Introduction to Deep Learning

## Lab 3: Reinforcement Learning

Link to download labs:

<http://introtodeeplearning.com#schedule>

1. Open the lab in Google Colab
2. Start executing code blocks and filling in the #TODOs
3. Need help? Find a TA or come to the front!!