# Deep Learning Limitations and New Frontiers

Ava Soleimany

MIT 6.S191

January 25, 2021

# T-shirts! To be delivered!

# Lecture Schedule

**Intro to Deep Learning**
Lecture 1
*Jan. 18, 2021*
[Slides] [Video] *coming soon!*

**Deep Sequence Modeling**
Lecture 2
*Jan. 19, 2021*
[Slides] [Video] *coming soon!*

**Intro to TensorFlow; Music Generation**
Software Lab 1
*Due Jan. 20, 2021*
[Code] *coming soon!*

**Deep Computer Vision**
Lecture 3
*Jan. 20, 2021*
[Slides] [Video] *coming soon!*

**Deep Generative Modeling**
Lecture 4
*Jan. 21, 2021*
[Slides] [Video] *coming soon!*

**De-biasing Facial Recognition Systems**
Software Lab 2
*Due Jan. 22, 2021*
[Paper] [Code] *coming soon!*

**Deep Reinforcement Learning**
Lecture 5
*Jan. 22, 2021*
[Slides] [Video] *coming soon!*

**Limitations and New Frontiers**
Lecture 6
*Jan. 25, 2021*
[Slides] [Video] *coming soon!*

**Pixels-to-Control Learning**
Software Lab 3
*Due Jan. 26, 2021*
[Code] *coming soon!*

**Evidential Deep Learning**
Lecture 7
*Jan. 26, 2021*
[Slides] [Video] *coming soon!*

**Bias and Fairness**
Lecture 8
*Jan. 26, 2021*
[Slides] [Video] *coming soon!*

**Project Proposals**
Work on Final Proposals
*Due Jan. 27, 2021*

**Guest Lecture**
Lecture 9
*Jan. 27, 2021*
[Info] [Slides] [Video] *coming soon!*

**Guest Lecture**
Lecture 10
*Jan. 27, 2021*
[Info] [Slides] [Video] *coming soon!*

**Final Project**
Paper review
*Due Jan. 28, 2021*

**Guest Lecture**
Lecture 11
*Jan. 28, 2021*
[Info] [Slides] [Video] *coming soon!*

**Guest Lecture**
Lecture 12
*Jan. 28, 2021*
[Info] [Slides] [Video] *coming soon!*

**Final Project**
Work on Final Projects

**Project Presentations**
Project Pitches
*Jan. 29, 2021*

**Project Presentations**
Project Pitches
*Jan. 29, 2021*

**Awards Ceremony**
Winners announced!
*Jan. 29, 2021*

- Remaining lectures
- Graded P/D/F; 6 Units
- Lab 3 submission: 1/26/21
- Paper review: 1/28/21
- Final projects: 1/29/21

# 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 29**
- Submit groups by **Wed 1/27 11:59pm ET** to be eligible
- Submit slide by **Thu 1/28 11:59pm ET** to be eligible
- Instructions: syllabus/Canvas

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


NVIDIA 3080 GPU

4x Google Home Max

3x Display Monitors

# 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 29**
- Submit groups by **Wed 1/27 11:59pm ET** to be eligible
- Submit slide by **Thu 1/28 11:59pm ET** to be eligible
- Instructions: syllabus/Canvas

## Proposal Logistics

- Prepare slides on Google Slides
- **Group submit by Wed 1/27 11:59pm ET; info on syllabus/Canvas**
- In class project work: **Thu 1/28**
- **Slide submit by Thu 1/28 11:59 pm ET; info on syllabus/Canvas**
- Presentations on **Fri 1/29 1-3pm ET. Synchronous attendance required!**

# Final Class Project

## Option 1: Proposal Presentation

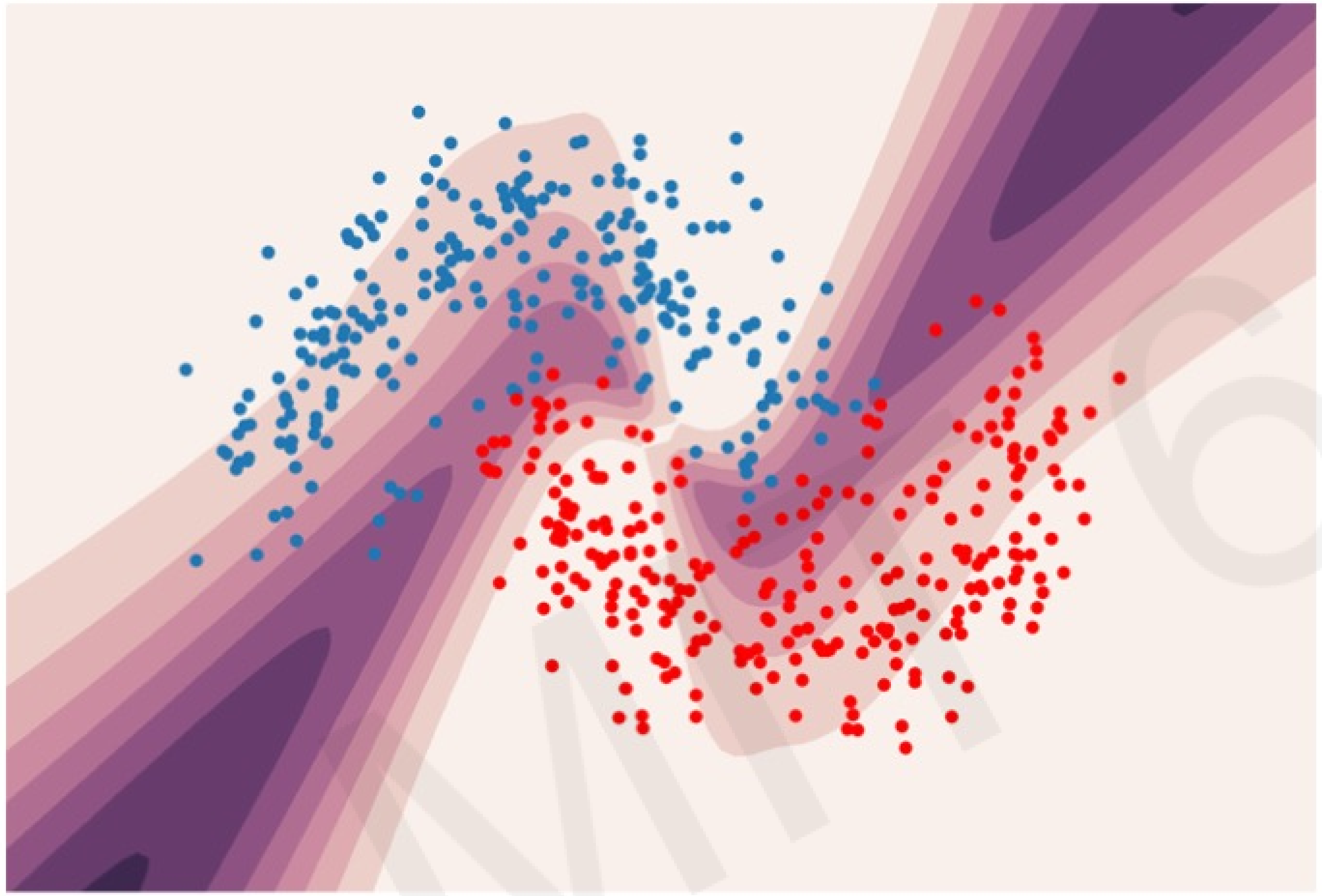
- At least 1 registered student to be prize eligible
- Present a novel deep learning research idea or application
- 3 minutes (strict)
- Presentations on Friday, Jan 29
- Submit groups by Wed 1/27 11:59pm ET to be eligible
- Submit slide by Thu 1/28 11:59pm ET to be eligible
- Instructions: syllabus/Canvas

## 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 **Thu Jan 28 11:59pm ET, by email**
- Further instructions on syllabus/Canvas

# Up Next: Hot Topic Spotlights

**Evidential Deep Learning**



Learning the uncertainty of neural networks

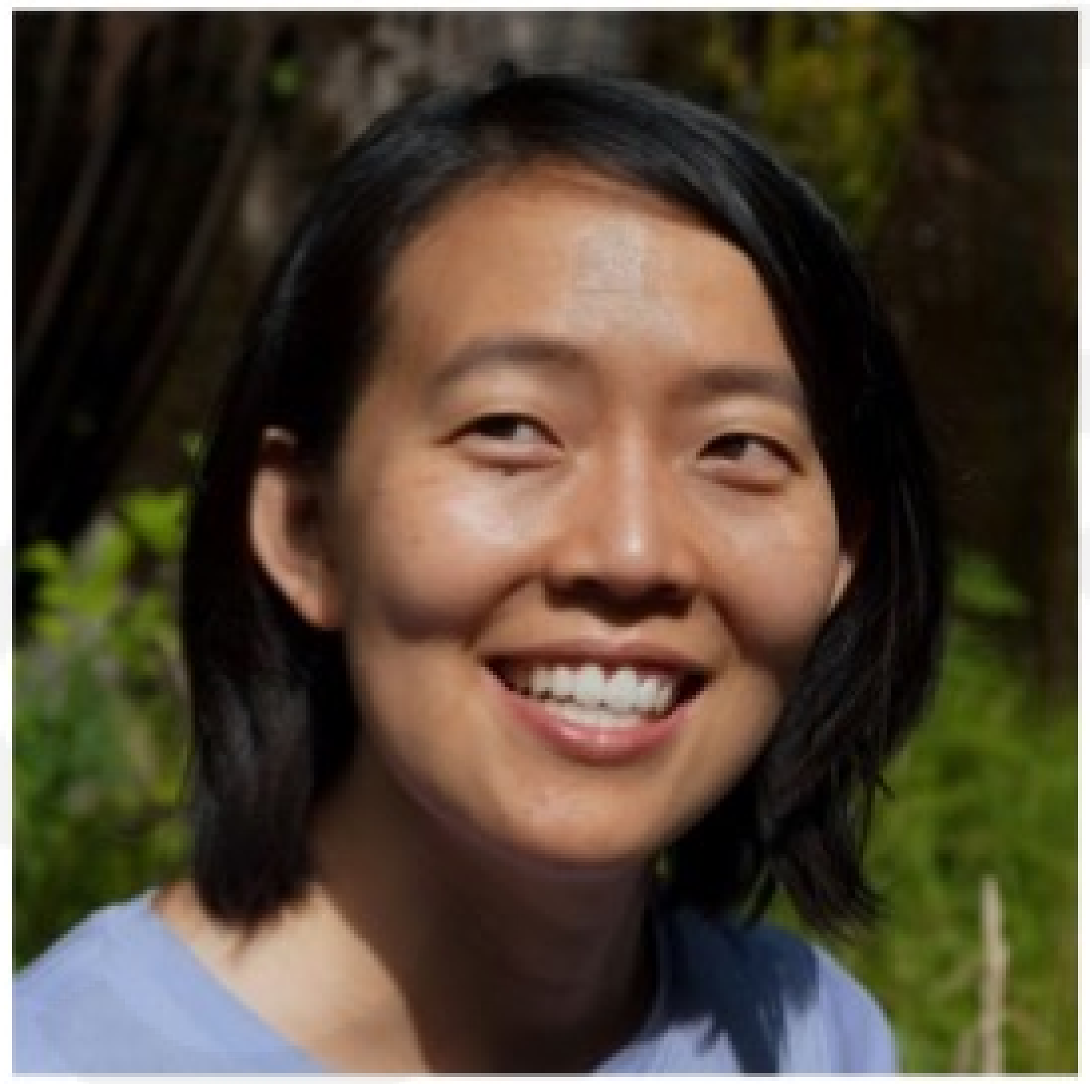
**Bias and Fairness**

Bias in deep learning: dangers and mitigation strategies

# Up Next: Guest Lectures

**Nigel Duffy**
Ernst & Young

**Kate Saenko**
MIT-IBM Watson AI Lab
Boston University



**Katherine Chou**
Google

**Sanja Fidler**
NVIDIA
U.Toronto

So far in 6.S191…

# The Rise of Deep Learning

# So far in 6.S191…

**Data**
- Signals
- Images
- Sensors

…

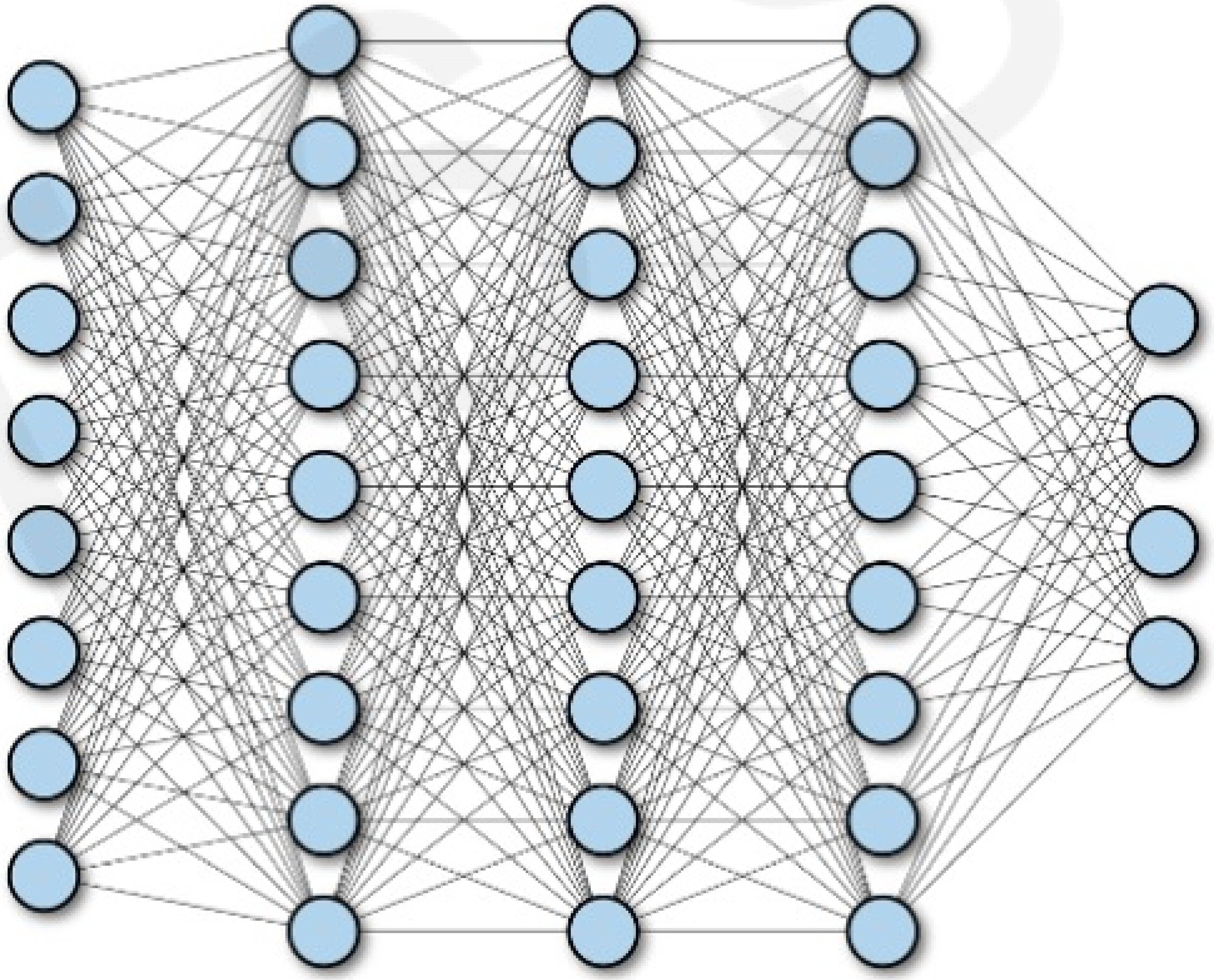
**Decision**
- Prediction
- Detection
- Action

…

# 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

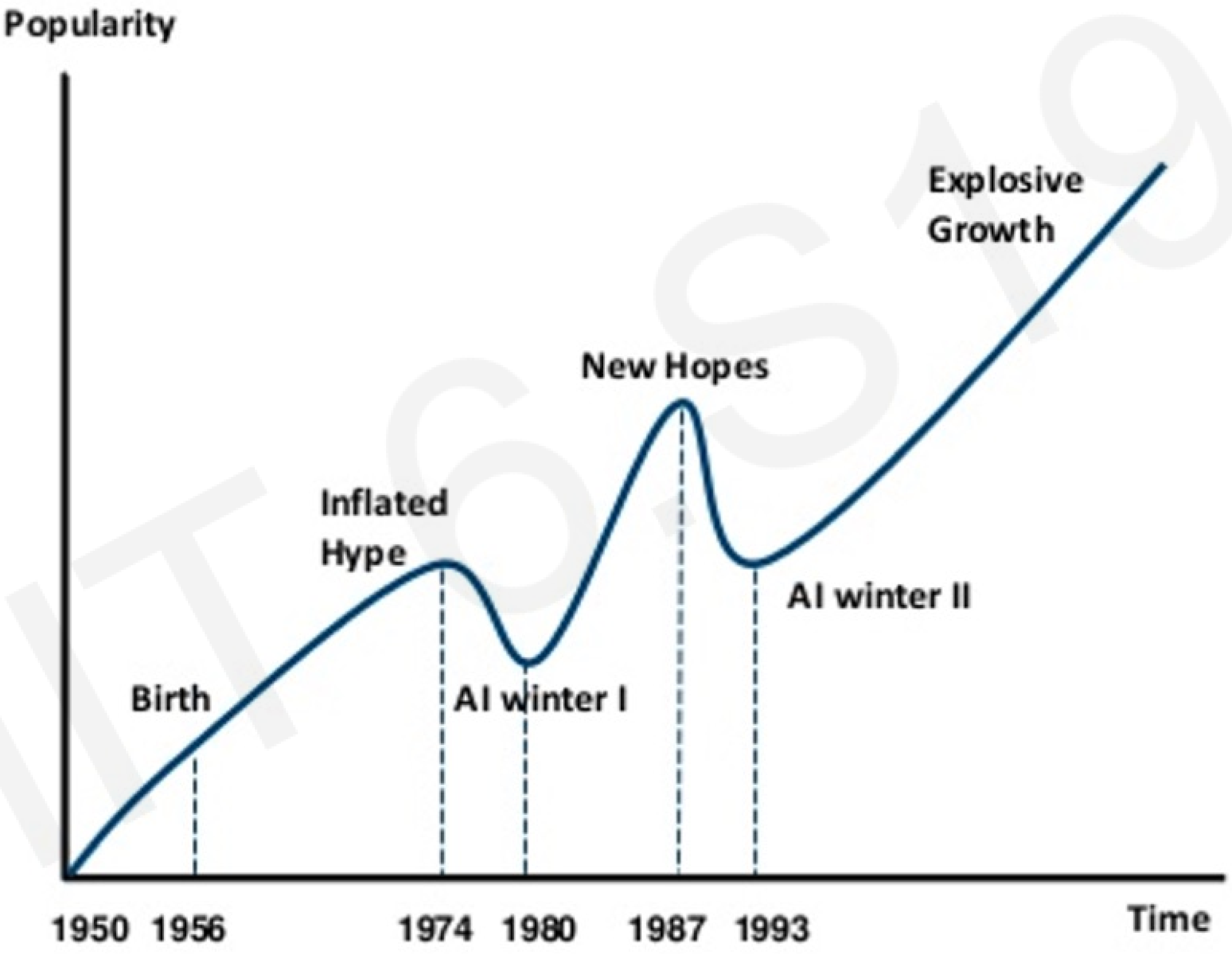
*A feedforward network with a single layer is sufficient to approximate, to an arbitrary precision, any continuous function.*

**Caveats:**

*The number of hidden units may be infeasibly large*

*The resulting model may not generalize*

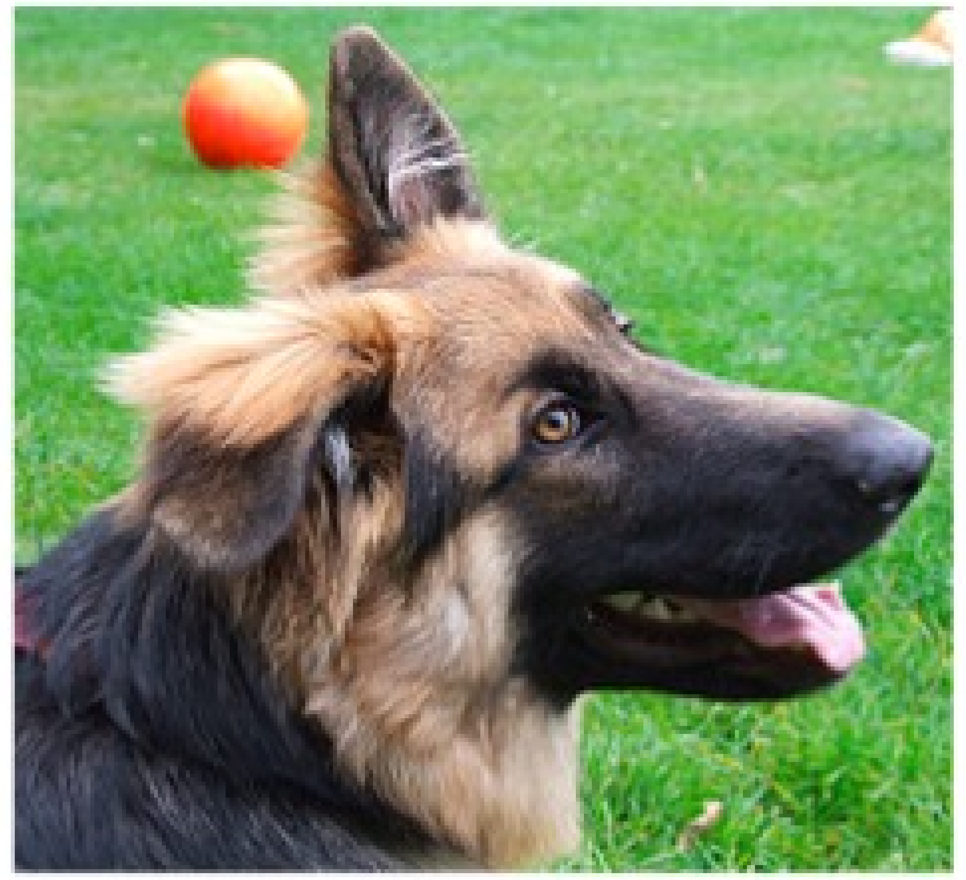
# Artificial Intelligence "Hype": Historical Perspective



# Limitations

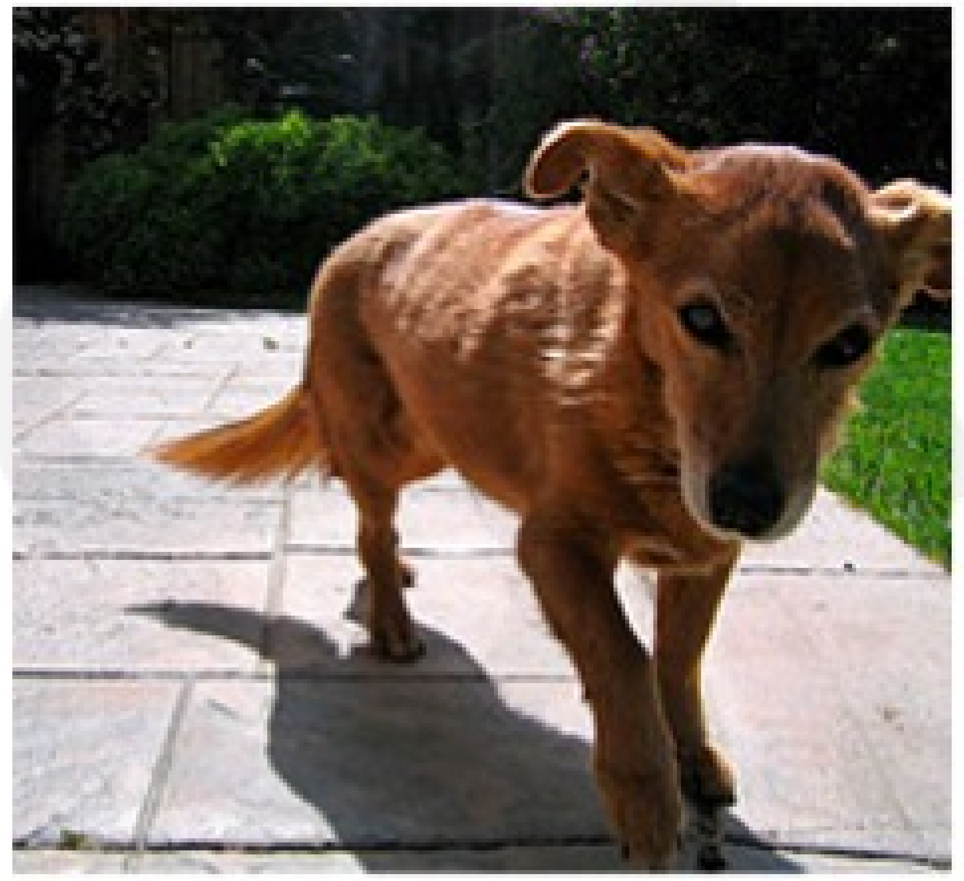
# Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"



dog

banana

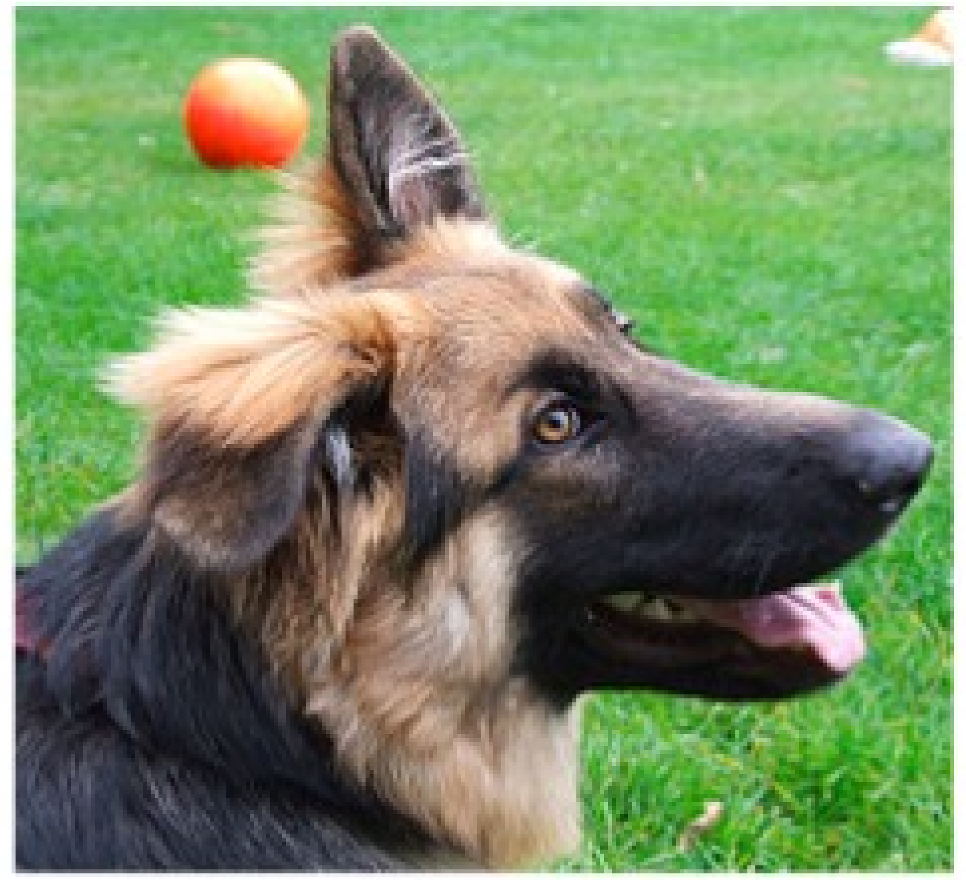
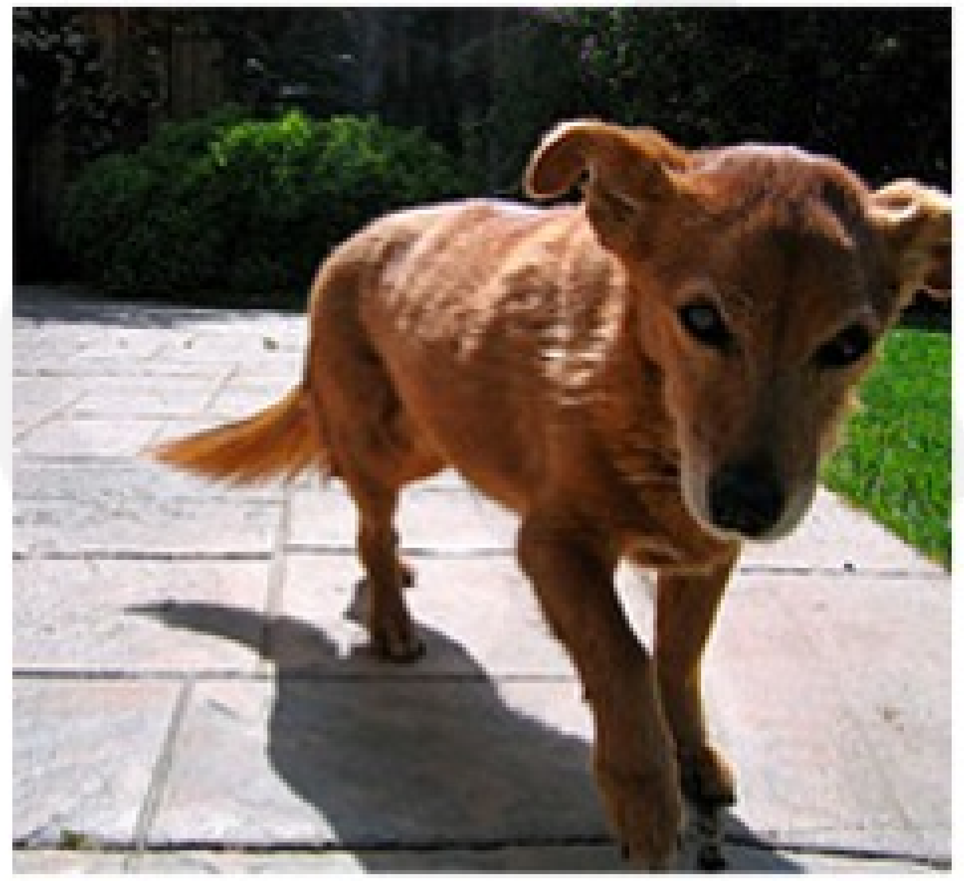


dog

tree

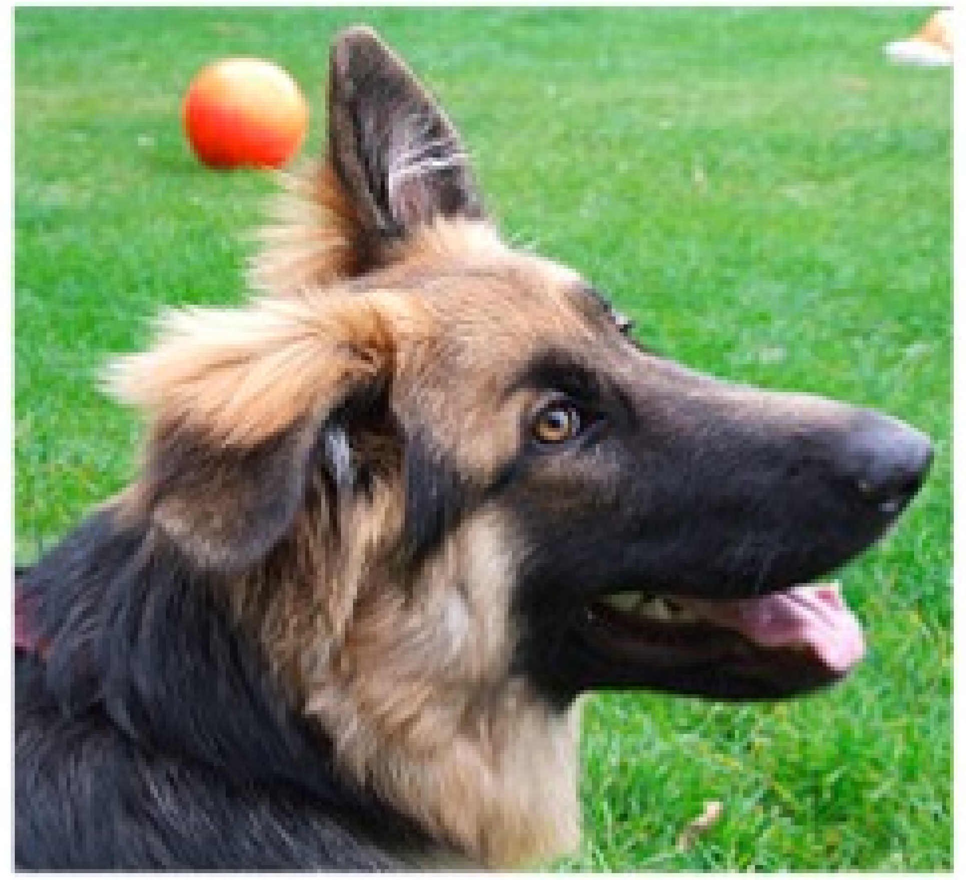
# Rethinking Generalization

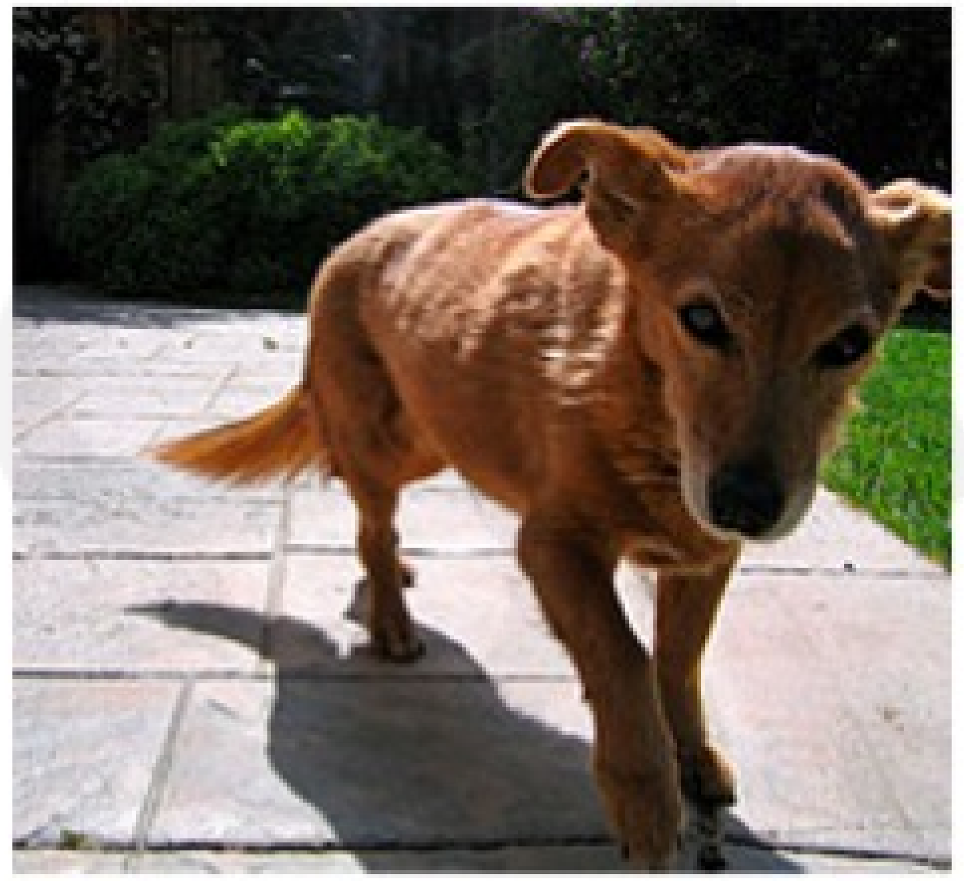
"Understanding Deep Neural Networks Requires Rethinking Generalization"

| | | | |
|---|---|---|---|
|  |  |  |  |
| dog | banana | dog | tree |
|  |  |  |  |

# Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"





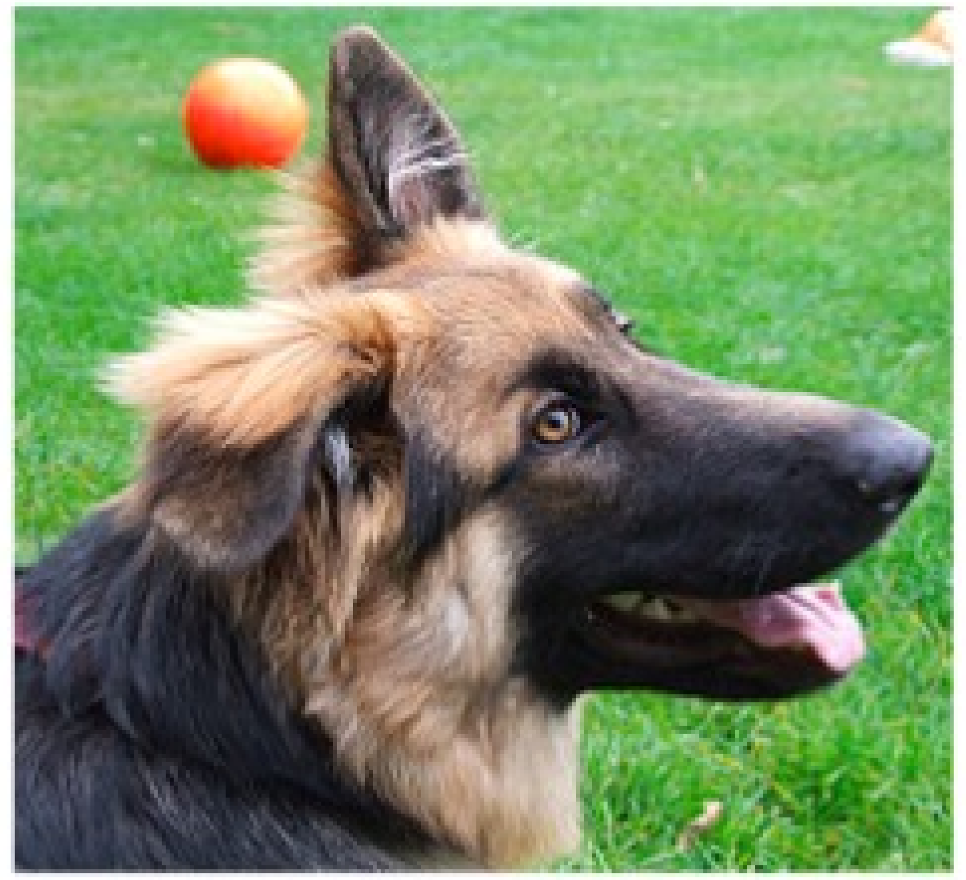

dog | banana | dog | tree




banana | dog | tree | dog

# Rethinking Generalization

"Understanding Deep Neural Networks Requires Rethinking Generalization"

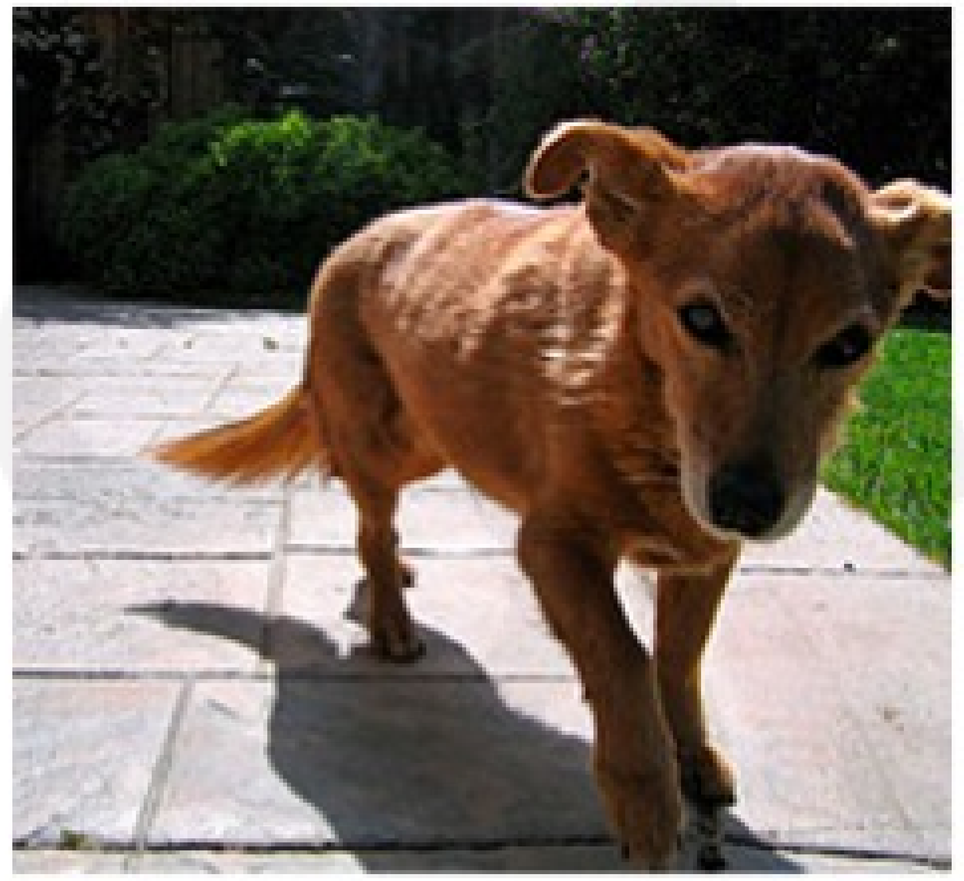


~~dog~~

banana

~~banana~~

dog



~~dog~~

tree

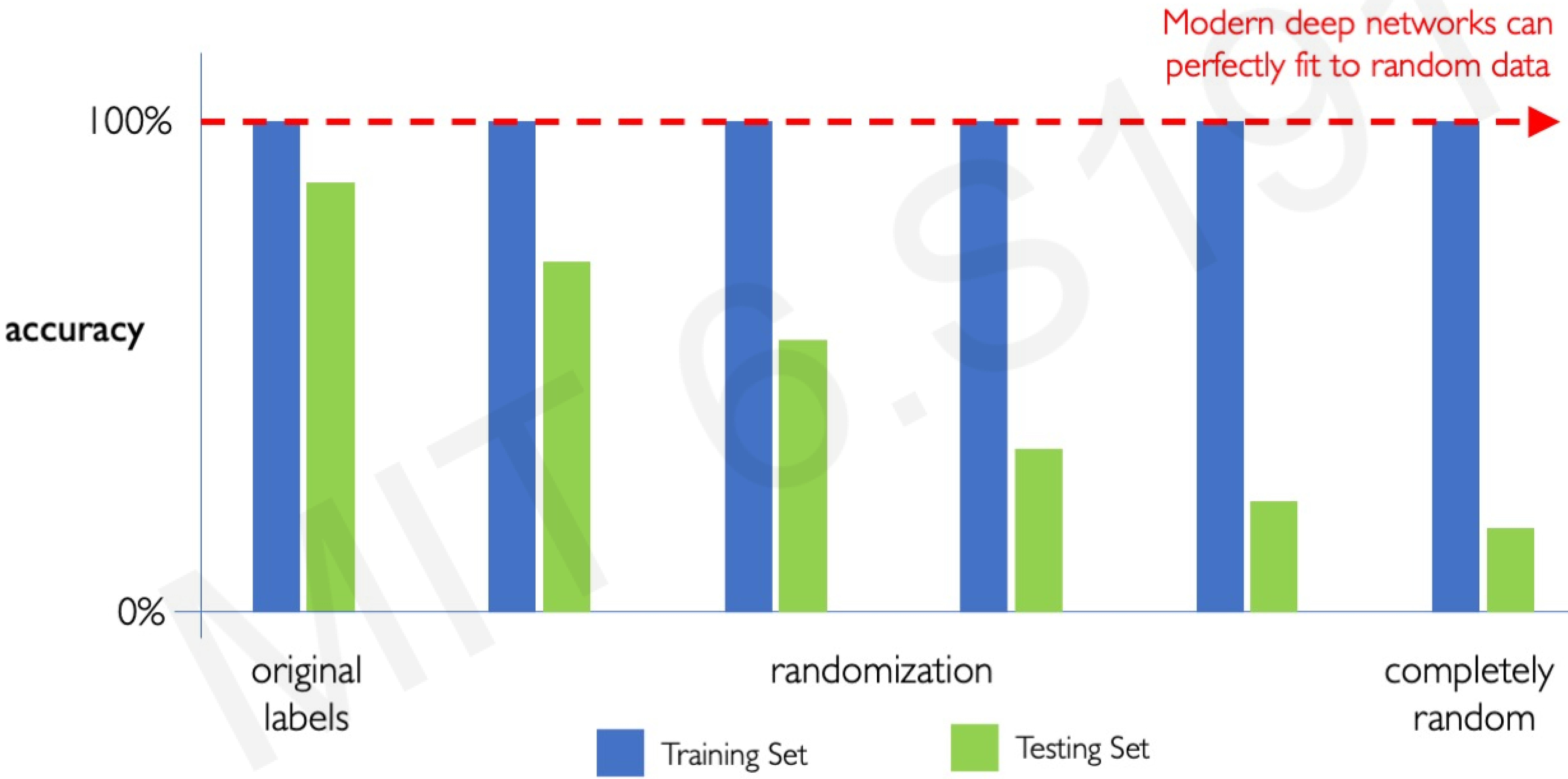
~~tree~~

dog

# Capacity of Deep Neural Networks

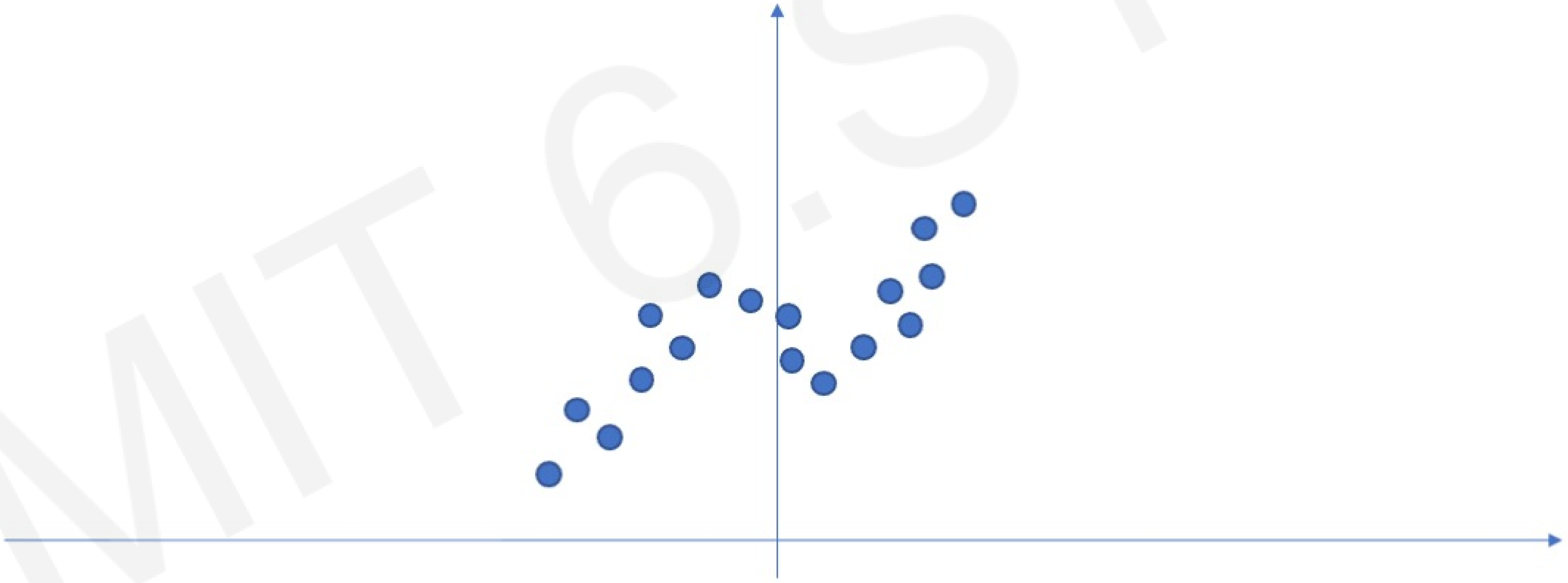
# Capacity of Deep Neural Networks

# Capacity of Deep Neural Networks



# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators

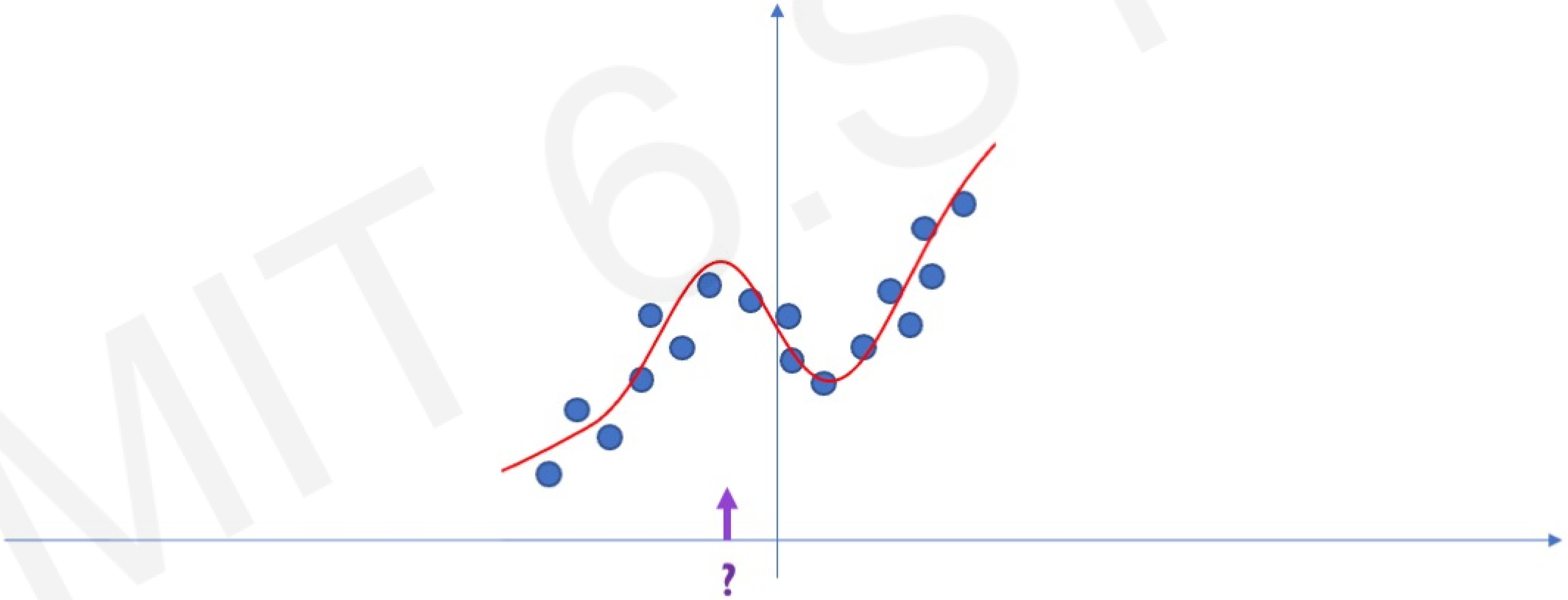


# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators

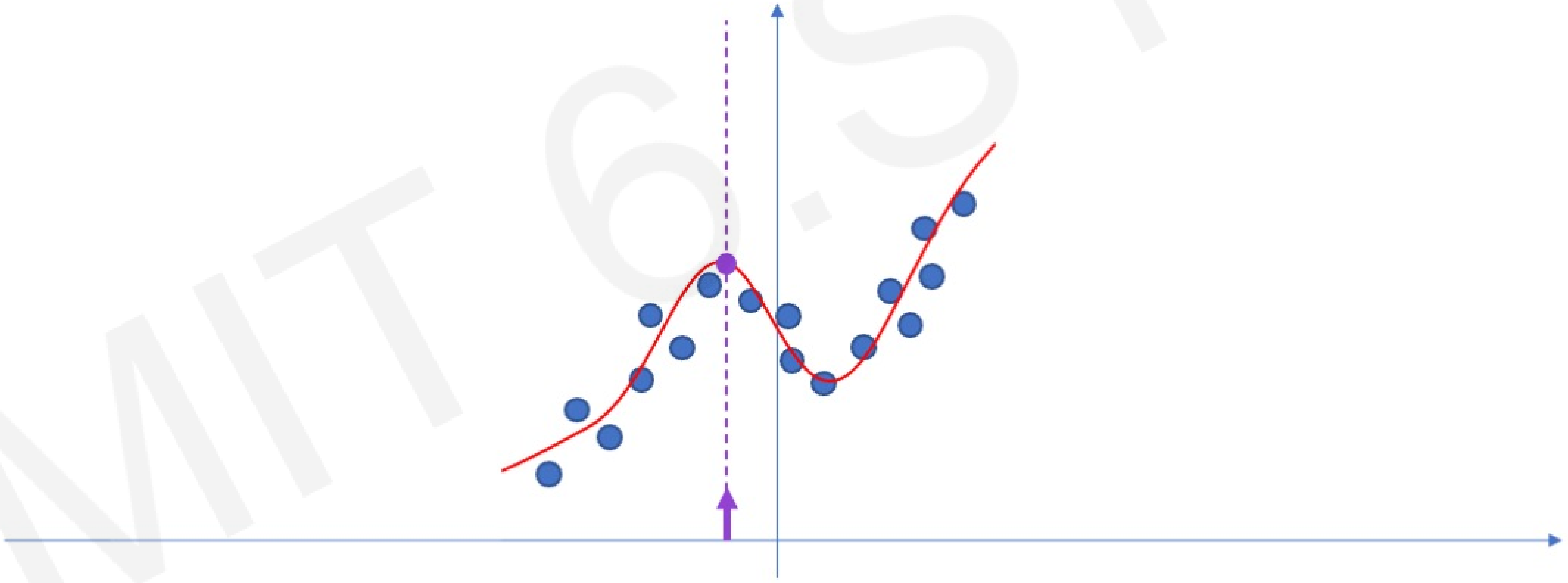
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



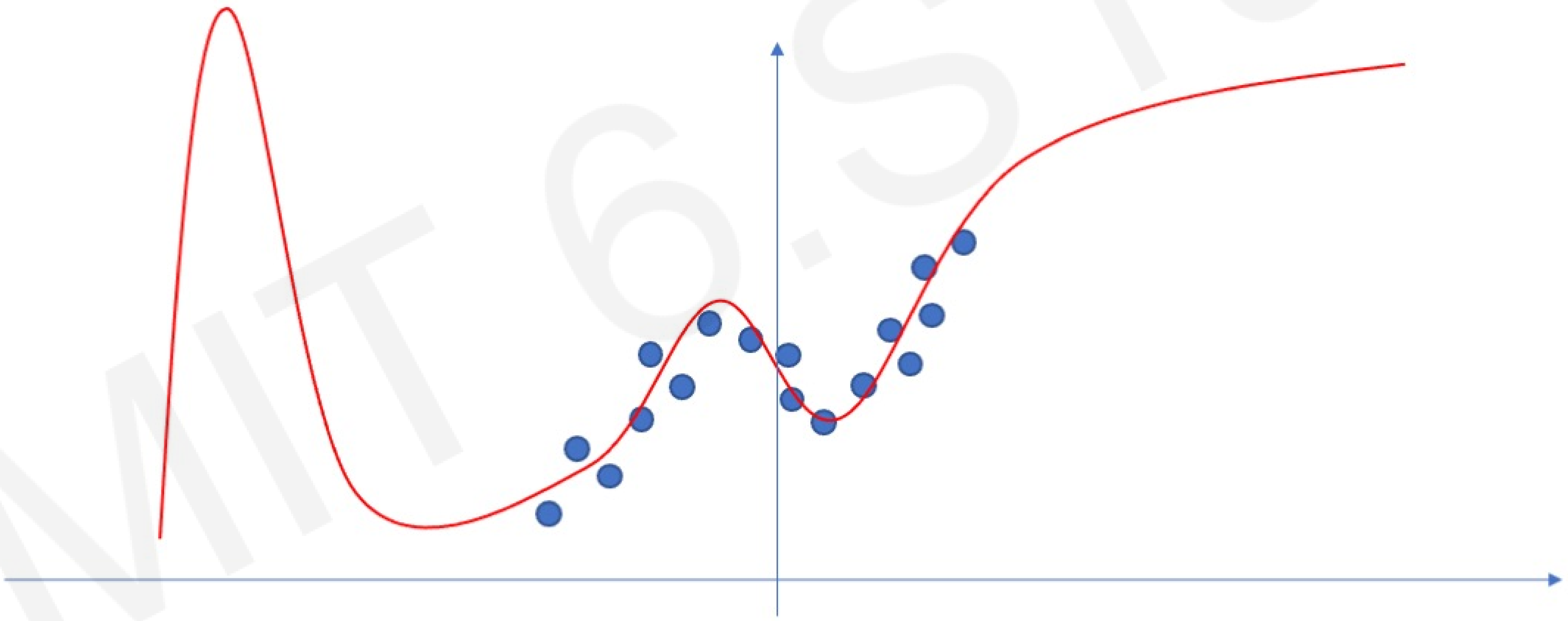
# Neural Networks as Function Approximators

Neural networks are **excellent** function approximators



# Neural Networks as Function Approximators

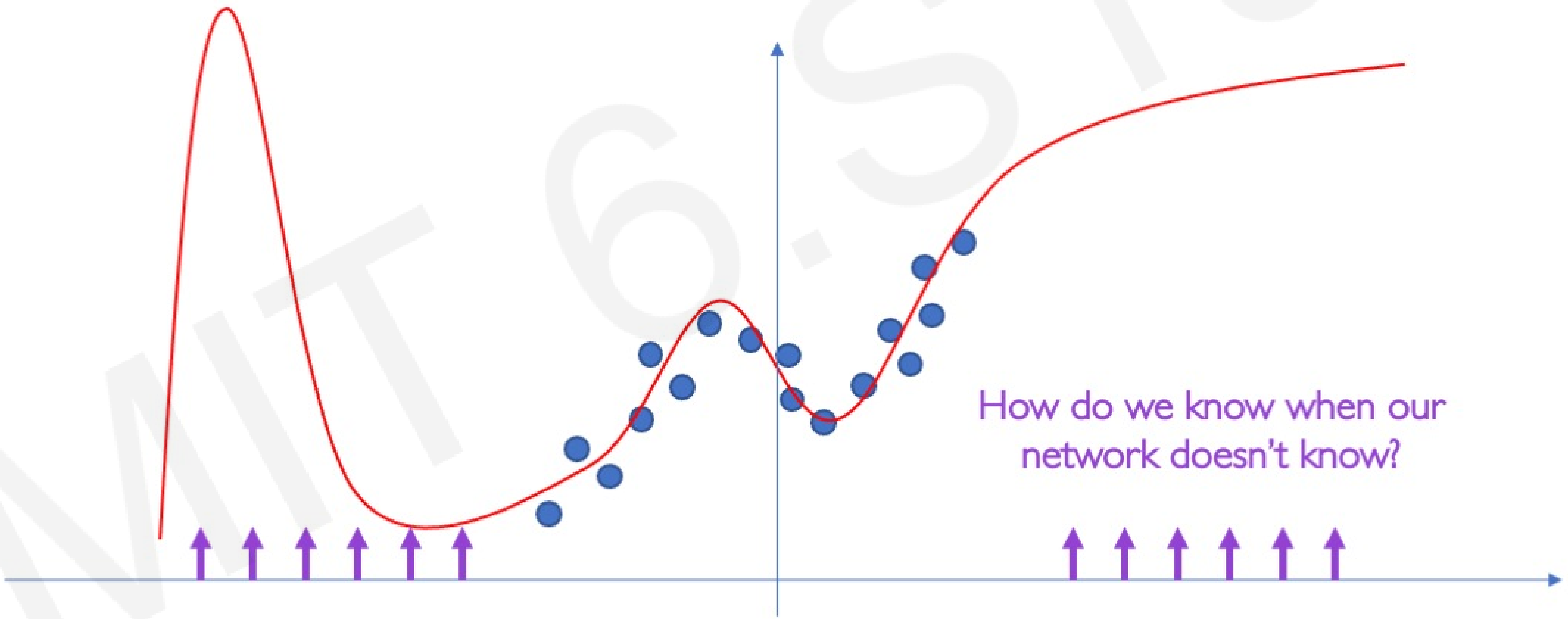
Neural networks are **excellent** function approximators



# Neural Networks as Function Approximators

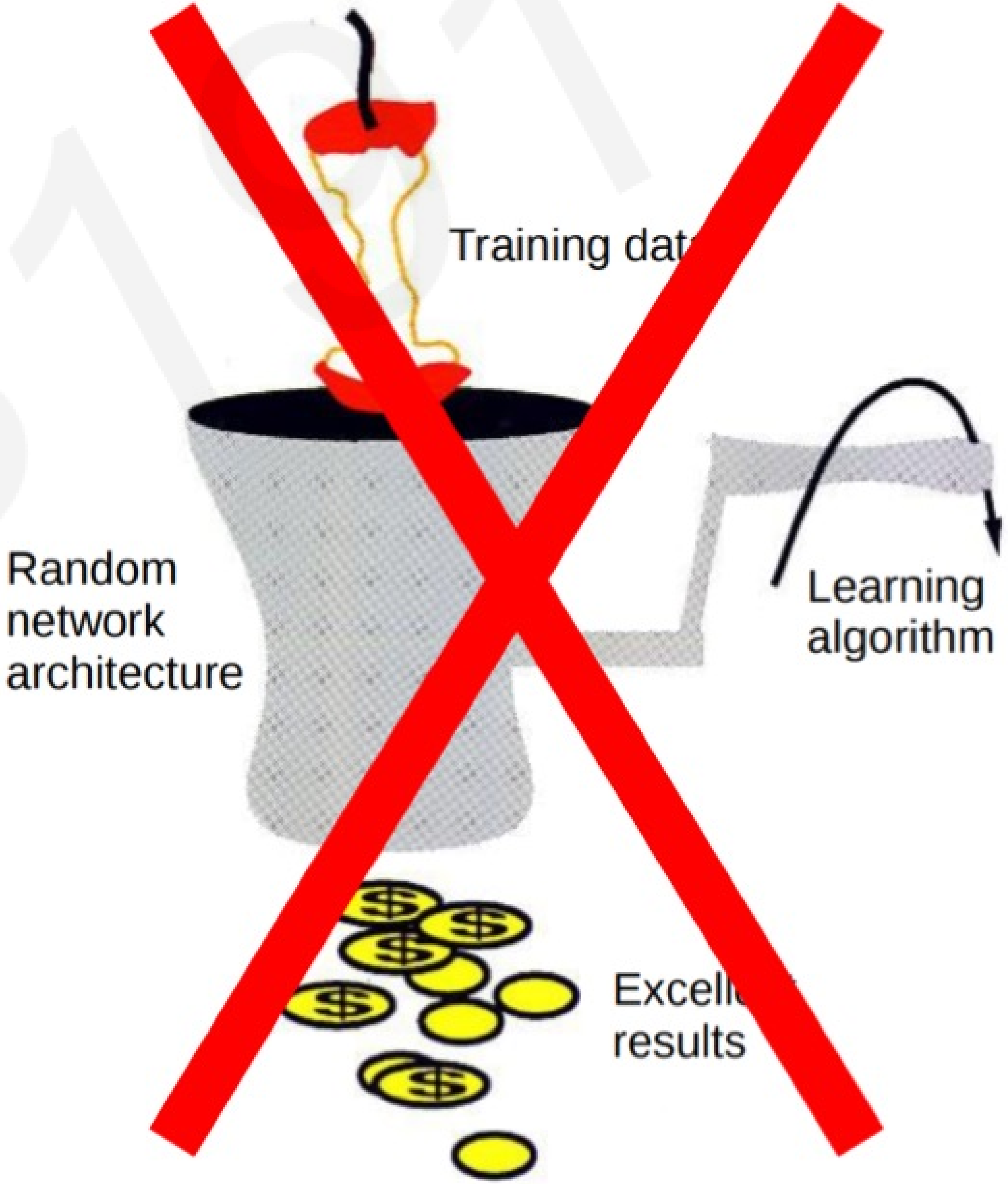
Neural networks are **excellent** function approximators

…when they have training data

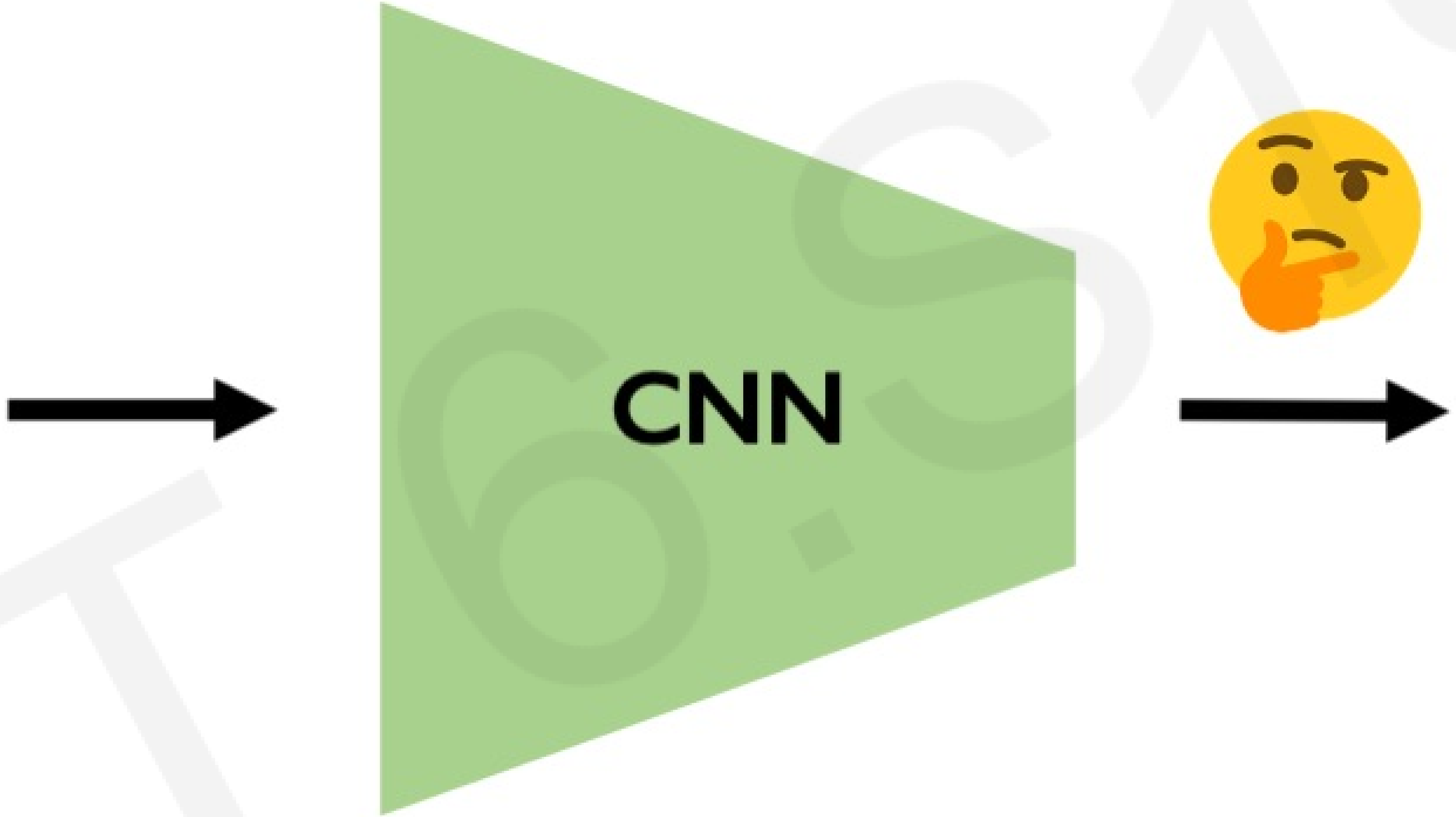


# Deep Learning = Alchemy?

NO!



# Neural Network Failure Modes, Part 1




Train network to
colorize BW images.

**Why could this be the case?**

# What Happens During Training…


P. Isola 6.869. 1/25/21

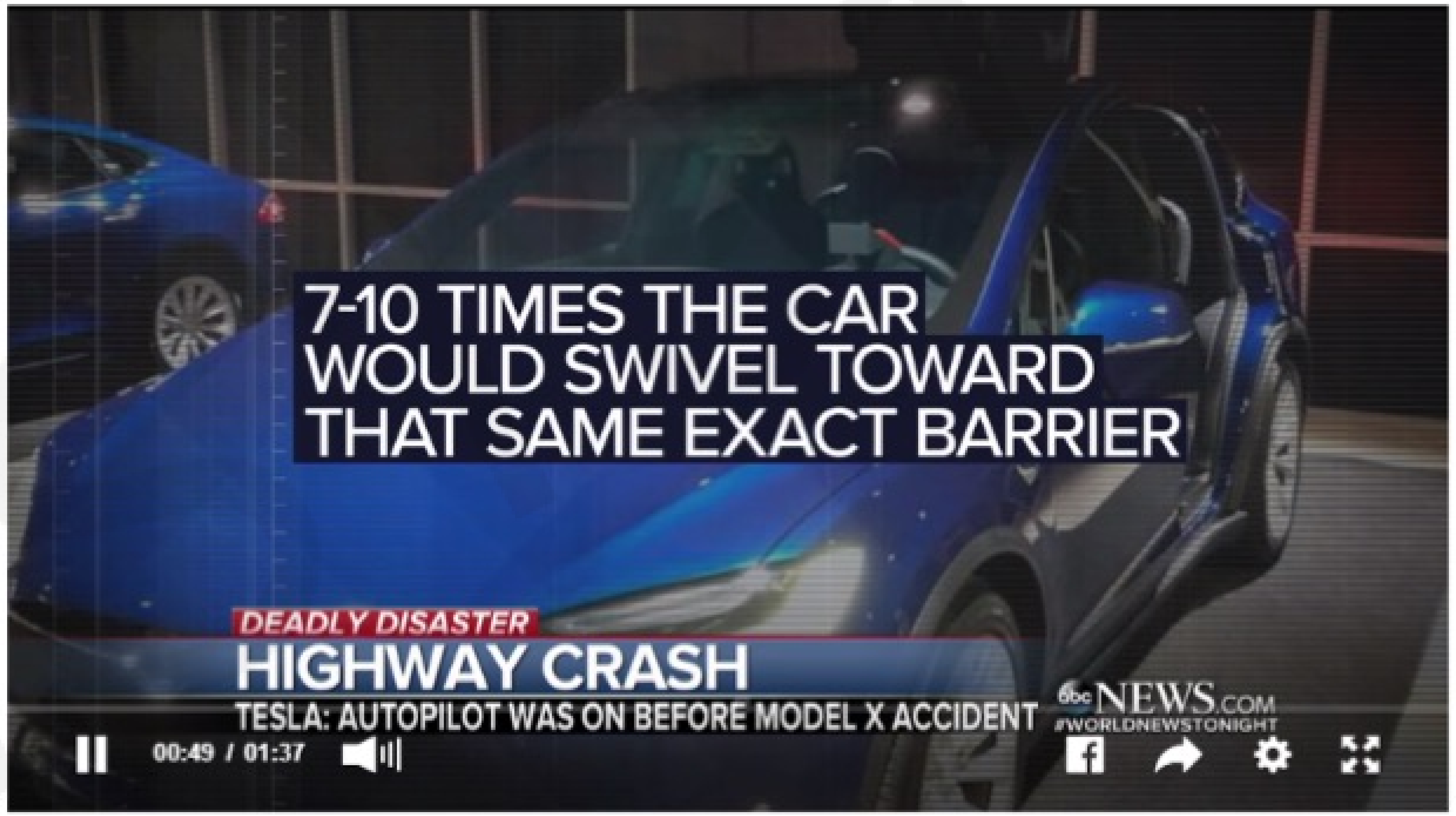
# Neural Network Failure Modes, Part II

## Tesla car was on autopilot prior to fatal crash in California, company says

*The crash near Mountain View, California, last week killed the driver.*

By **Mark Osborne**
March 31, 2018, 1:57 AM • 5 min read

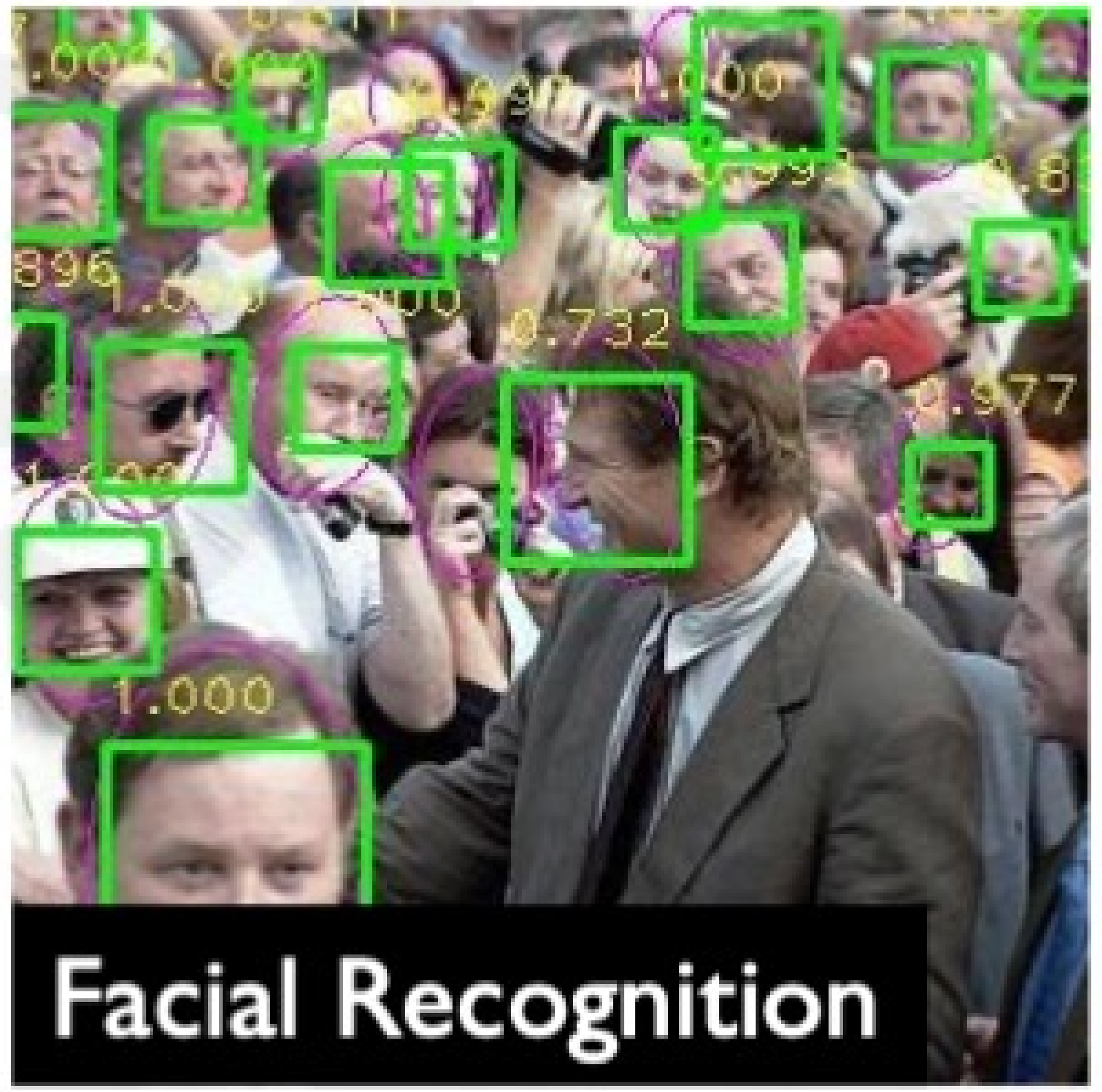
abc NEWS



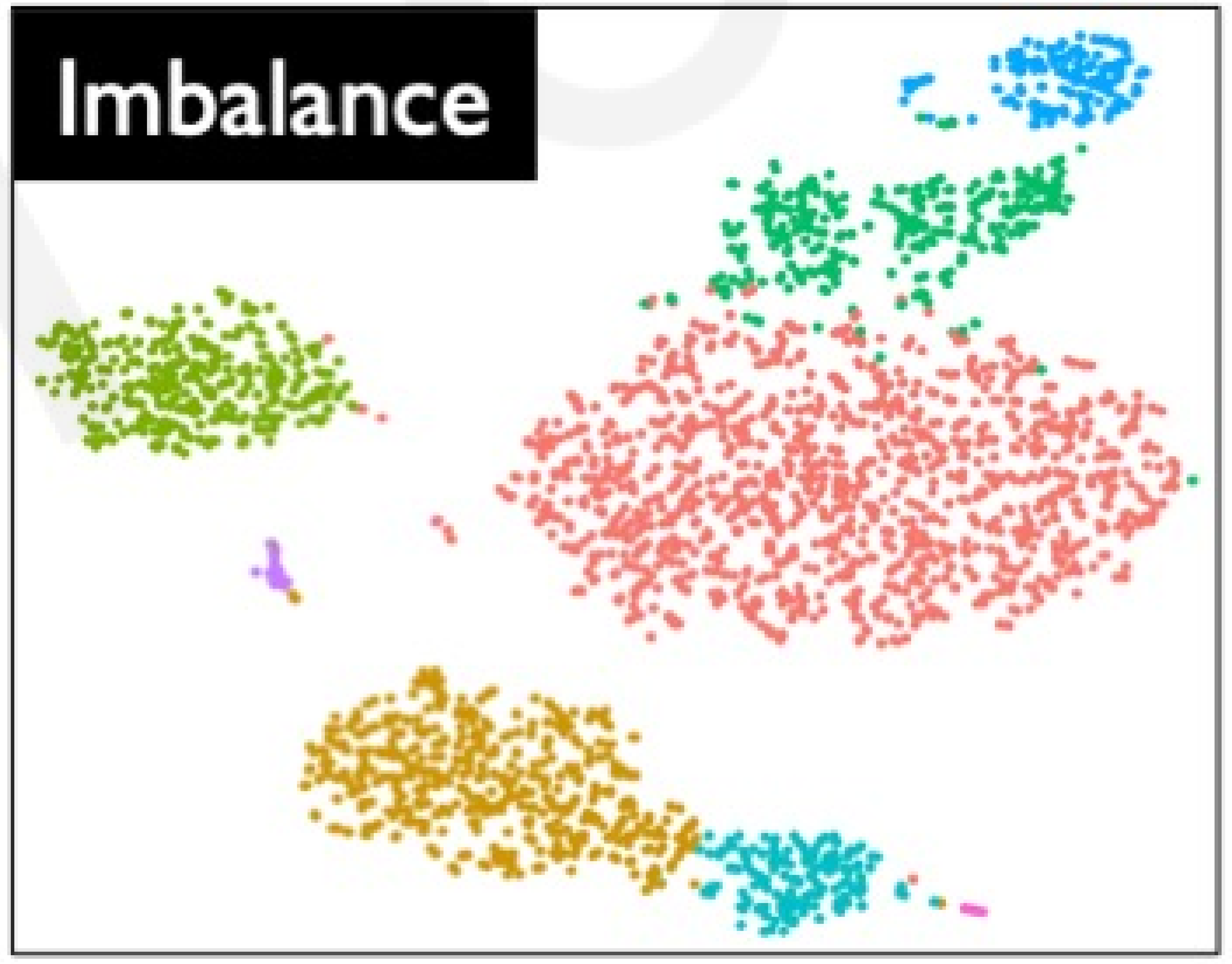

# Uncertainty in Deep Learning

**Safety-critical applications**





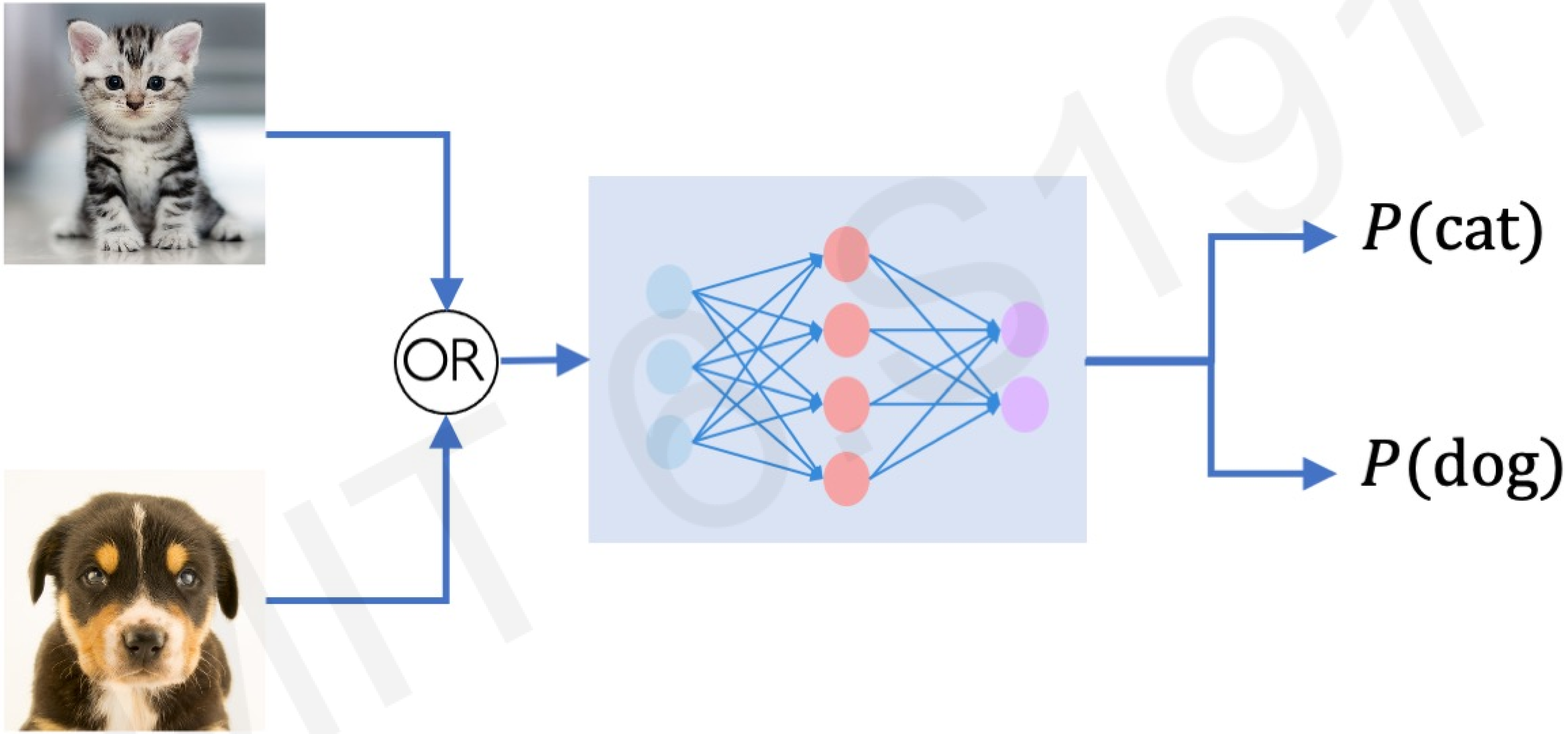

**Sparse and/or noisy datasets**





**6.S191 Lecture**

# What uncertainties do we need?



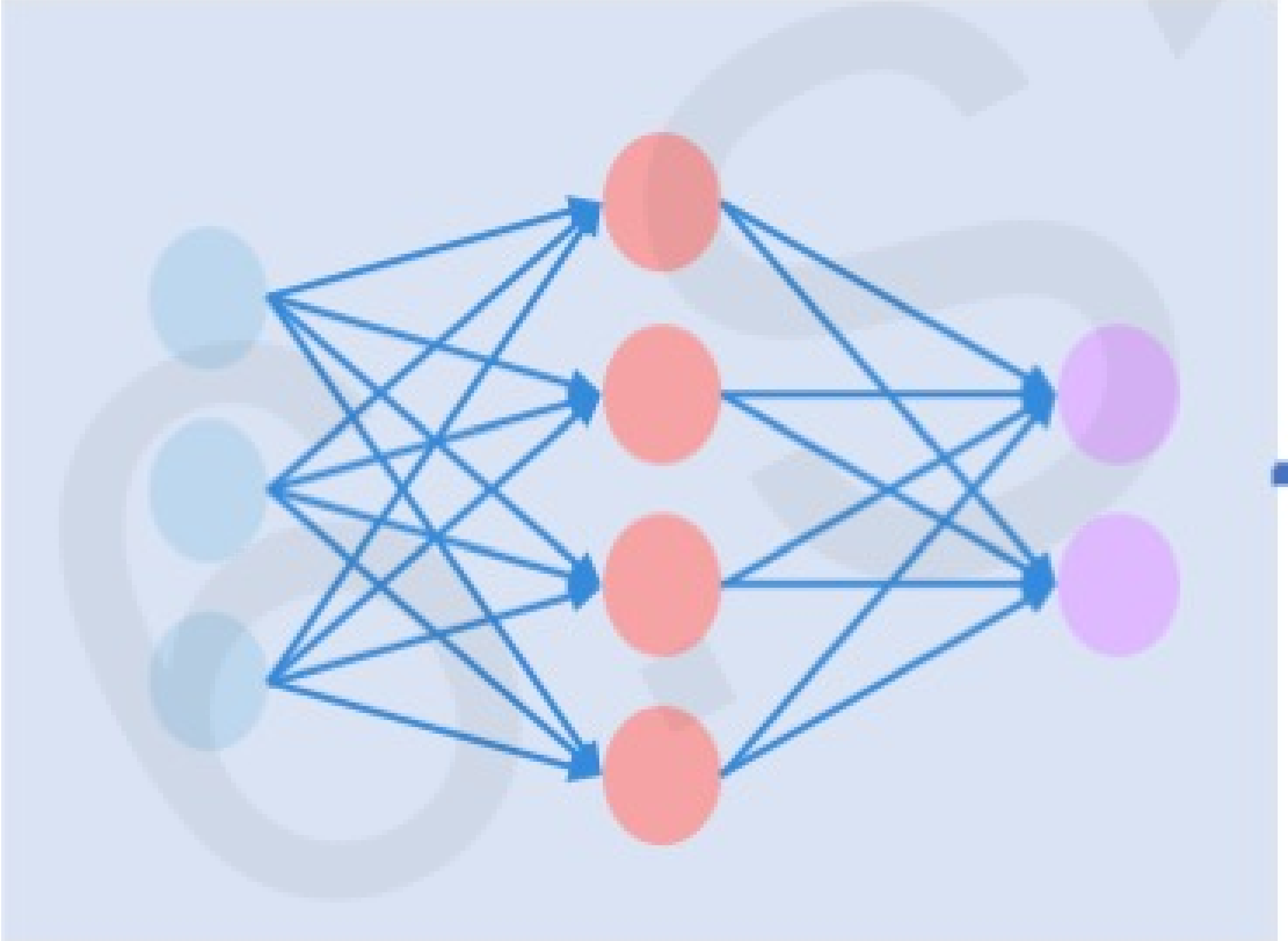



6.S191 Lecture

# What uncertainties do we need?

We need uncertainty metrics to assess the noise inherent to the data.

**aleatoric uncertainty**



$P(\text{cat}) = 0.5$

$P(\text{dog}) = 0.5$

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

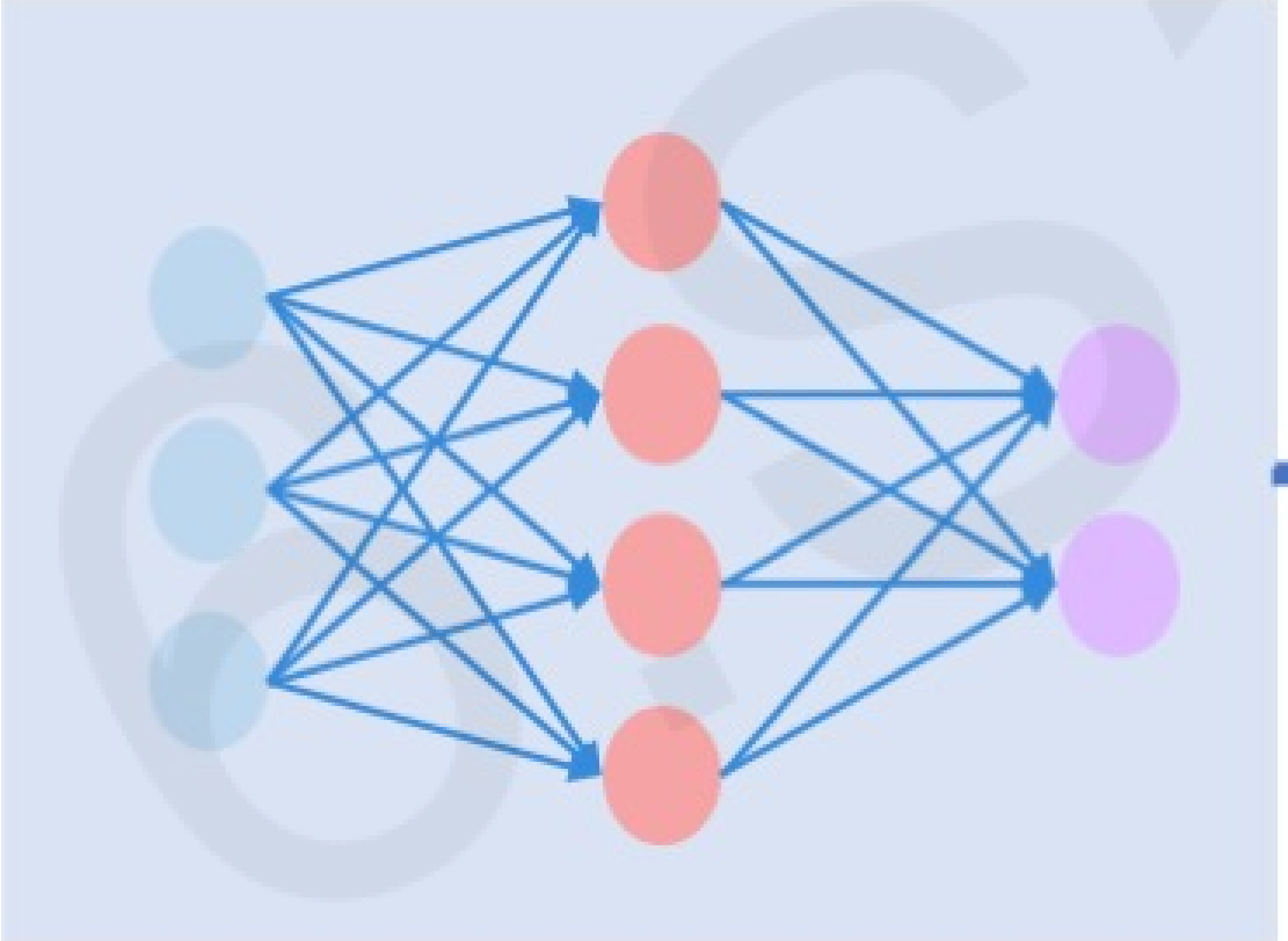


**6.S191 Lecture**

# What uncertainties do we need?

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

**epistemic uncertainty**



$P(\text{cat}) = 0.2$

$P(\text{dog}) = 0.8$

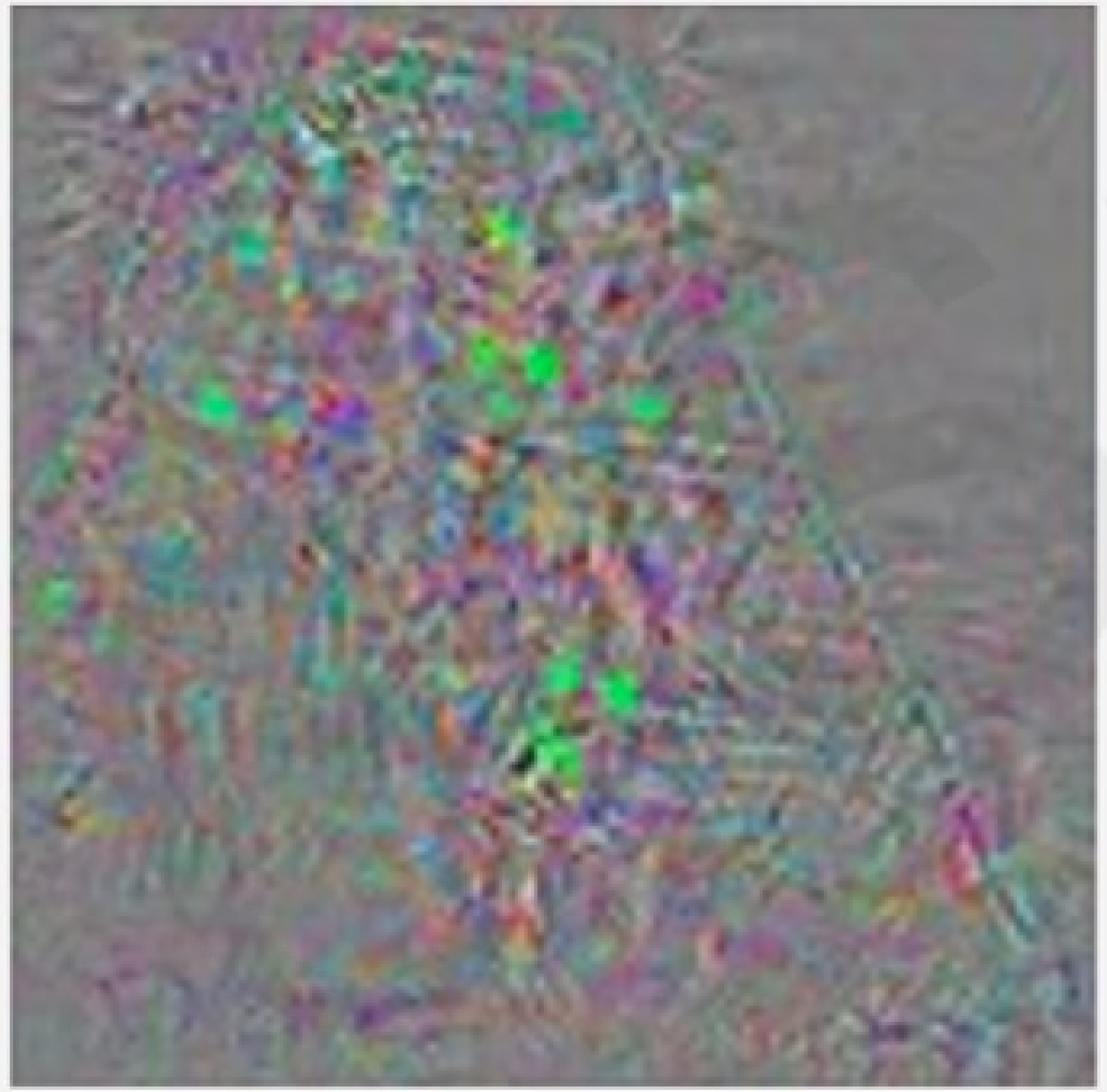
Remember: $P(\text{cat}) + P(\text{dog}) = 1$

6.S191 Lecture

# Neural Network Failure Modes, Part III

**Original image**

Temple (97%)



**Perturbations**

**Adversarial example**

Ostrich (98%)





Despois "Adversarial examples" 2017.

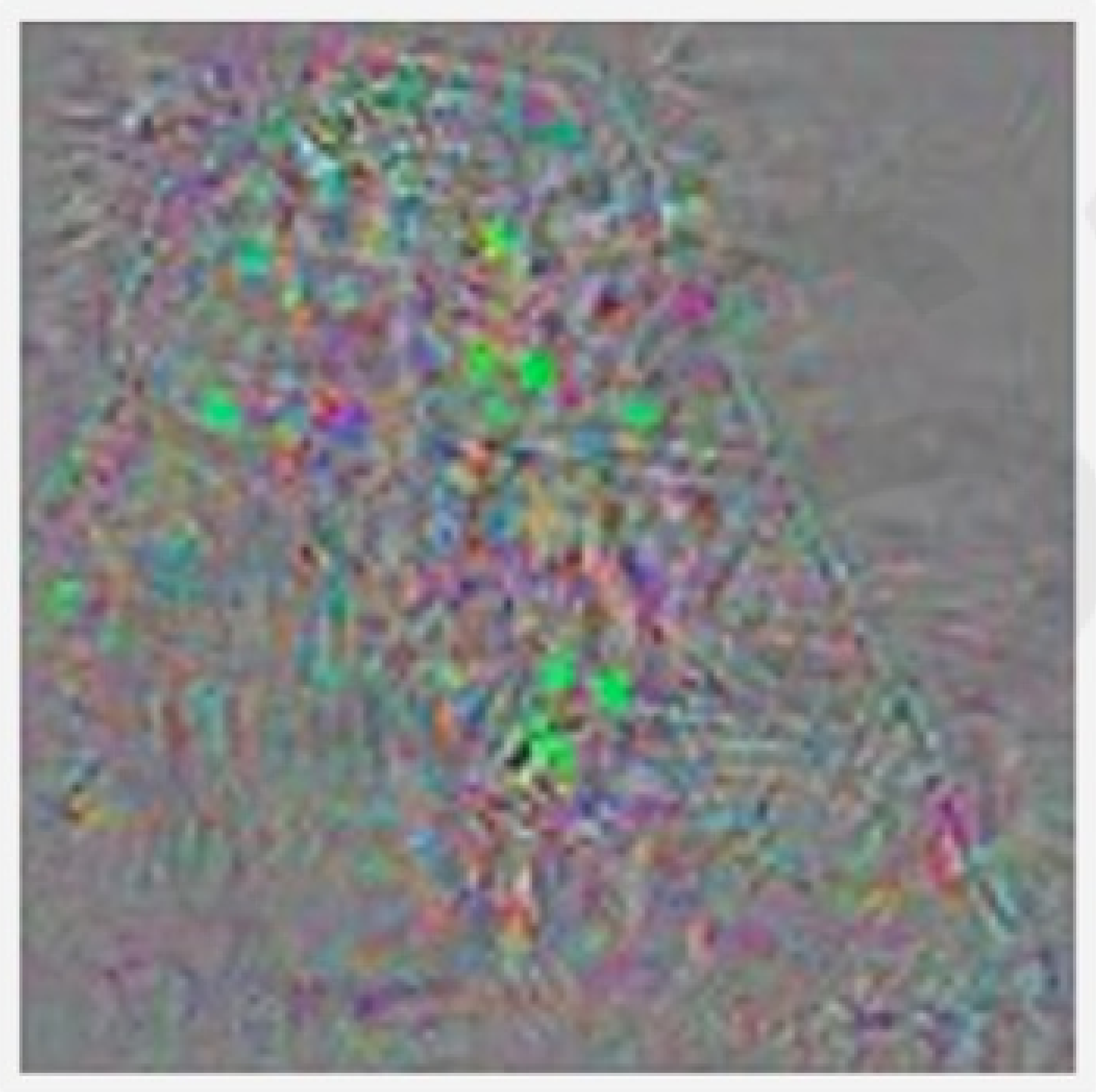
# Adversarial Attacks on Neural Networks



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

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

*"How does a small change in weights decrease our loss"*

# Adversarial Attacks on Neural Networks

**Remember:**

We train our networks with gradient descent

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

Fix your image $\boldsymbol{x}$, and true label $\boldsymbol{y}$

*"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


Athalye+ *ICML* 2018.

# Algorithmic Bias

Overcoming Racial Bias In AI Systems And Startlingly Even In AI Self-Driving Cars

Racial bias in a medical algorithm favors white patients over sicker black patients

AI expert calls for end to UK use of 'racially biased' algorithms

AI Bias Could Put Women's Lives At Risk – A Challenge For Regulators

Gender bias in AI: building fairer algorithms

Bias in AI: A problem recognized but still unresolved

Amazon, Apple, Google, IBM, and Microsoft worse at transcribing black people's voices than white people's with AI voice recognition, study finds

Millions of black people affected by racial bias in health-care algorithms

Study reveals rampant racism in decision-making software used by US hospitals – and highlights ways to correct it.

When It Comes to Gorillas, Google Photos Remains Blind

Google promised a fix after its photo-categorization software labeled black people as gorillas in 2015. More than two years later, it hasn't found one.

*The Week in Tech: Algorithmic Bias Is Bad. Uncovering It Is Good.*

Google 'fixed' its racist algorithm by removing gorillas from its image-labeling tech

Artificial Intelligence has a gender bias problem – just ask Siri

The Best Algorithms Struggle to Recognize Black Faces Equally

US government tests find even top-performing facial recognition systems misidentify blacks at rates five to 10 times higher than they do whites.



6.S191 Lecture

# 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**
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- Difficult to **encode structure** and prior knowledge during learning
- **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**
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- Difficult to **encode structure** and prior knowledge during learning
- **Finicky to optimize**: non-convex, choice of architecture, learning parameters
- Often require **expert knowledge** to design, fine tune architectures

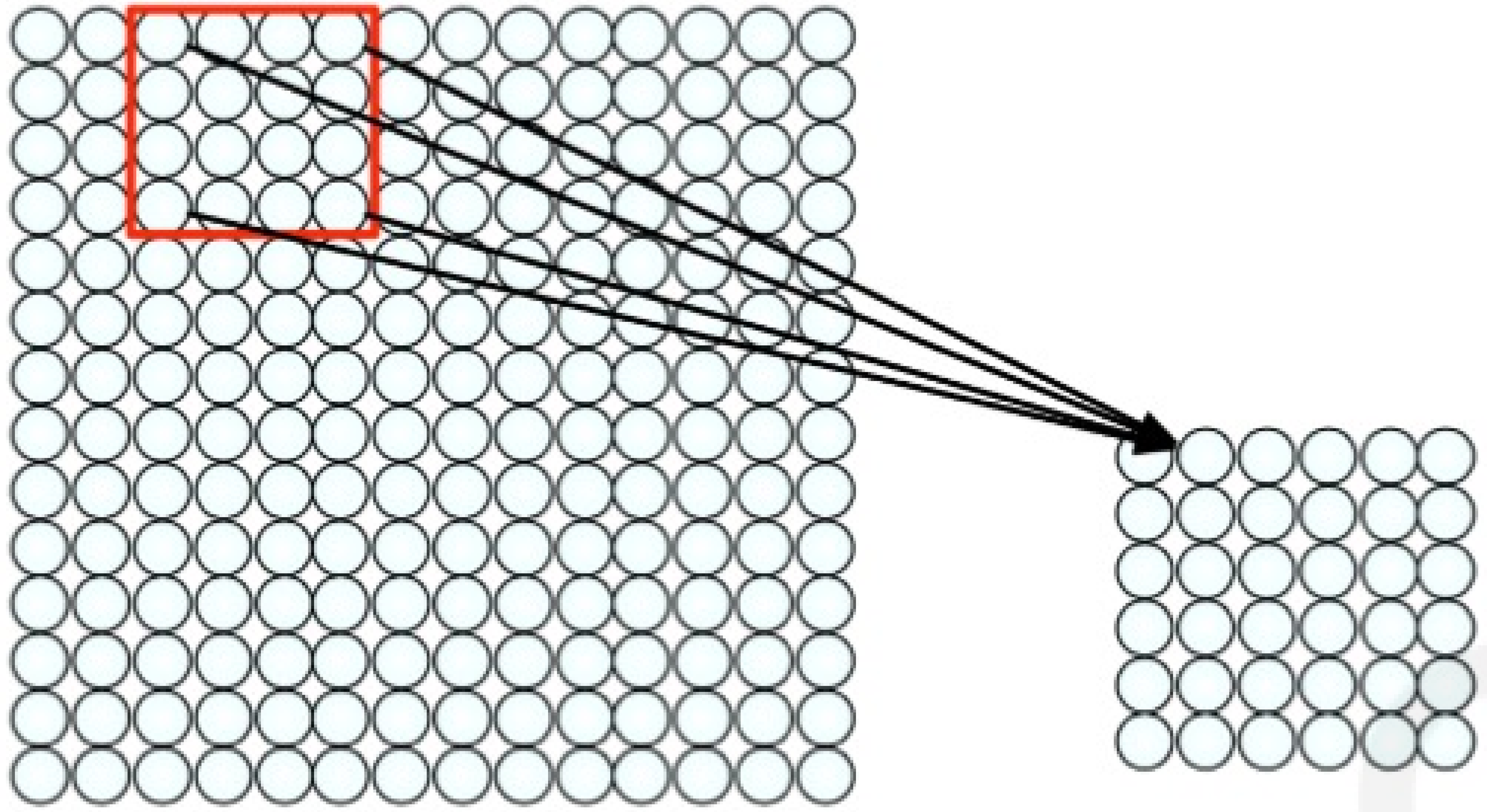
6.S191 Lecture

# 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**
- Poor at **representing uncertainty** (how do you know what the model knows?)
- Uninterpretable **black boxes**, difficult to trust
- Difficult to **encode structure** and prior knowledge during learning
- **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

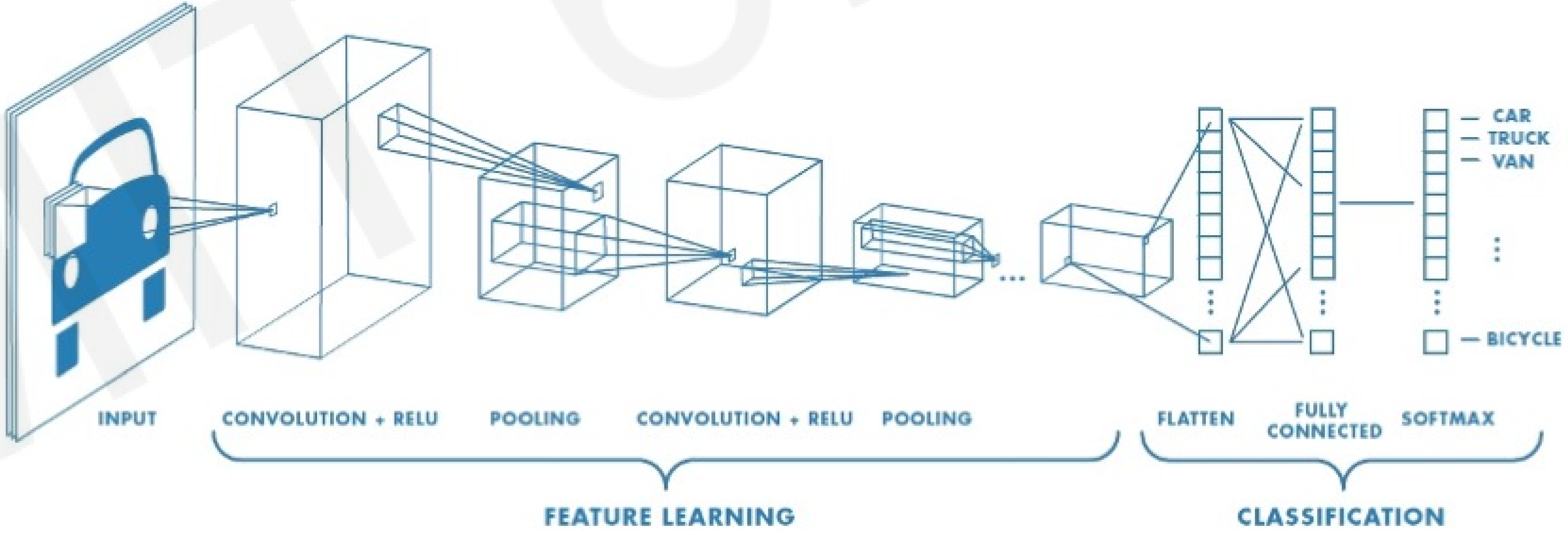
# CNNs: Using Spatial Structure



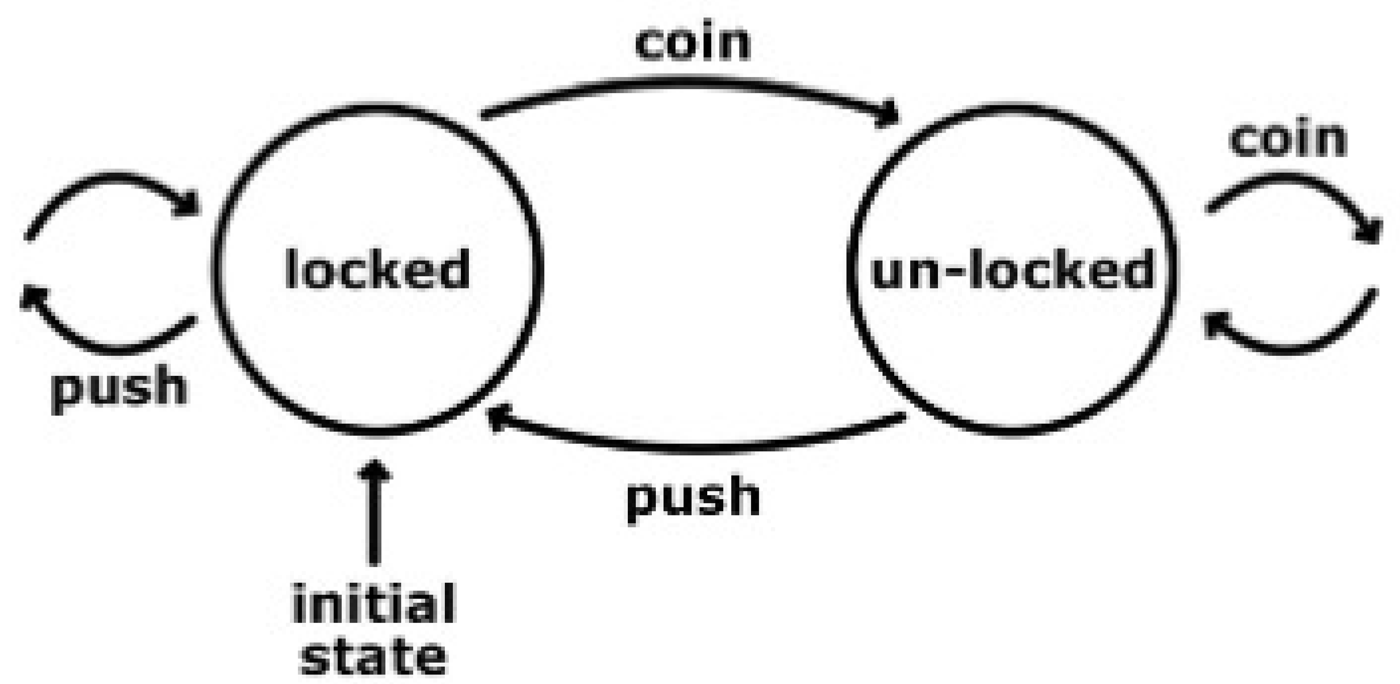
1) Apply a set of weights to extract **local features**

2) Use **multiple filters** to extract different features

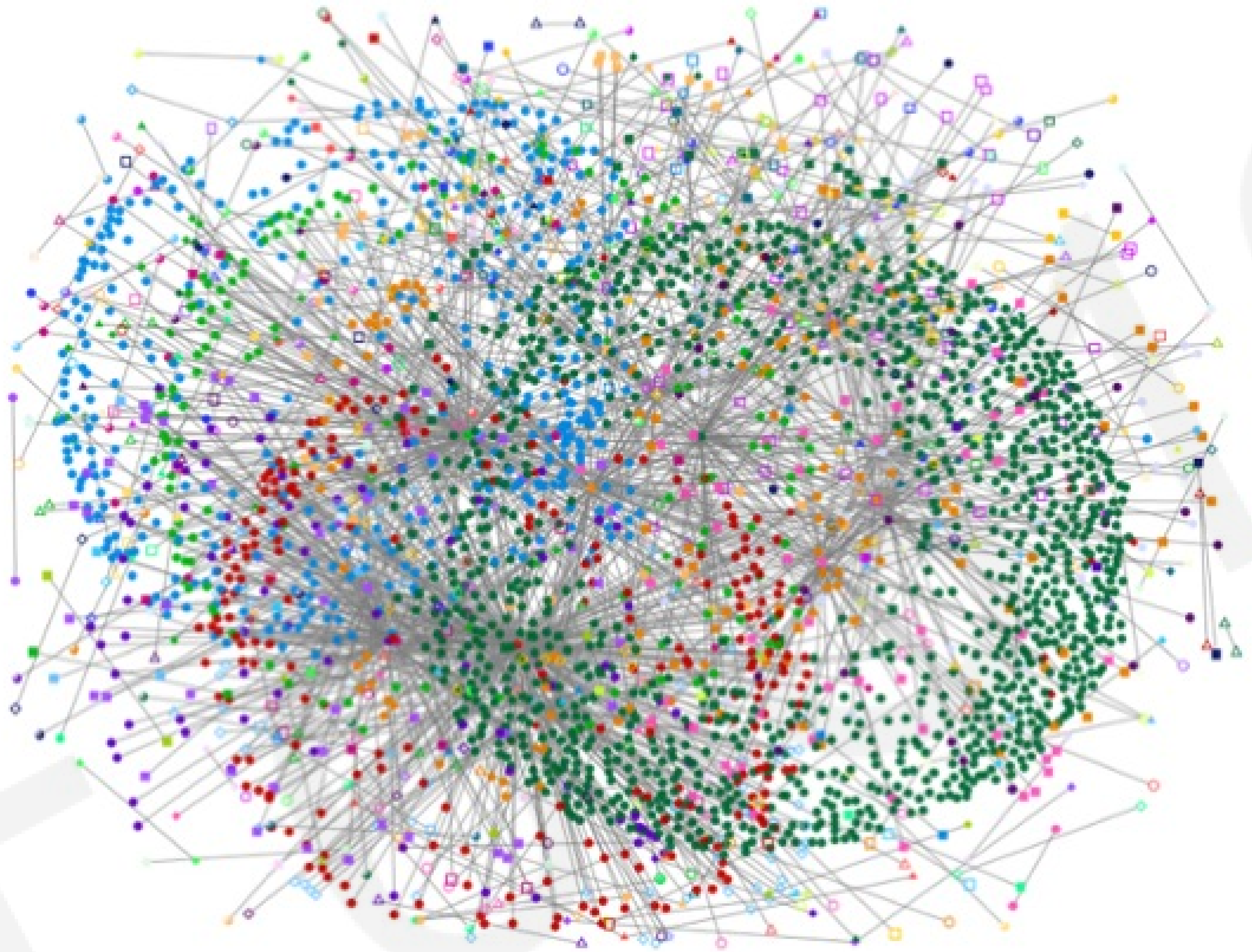
3) **Spatially share** parameters of each filter



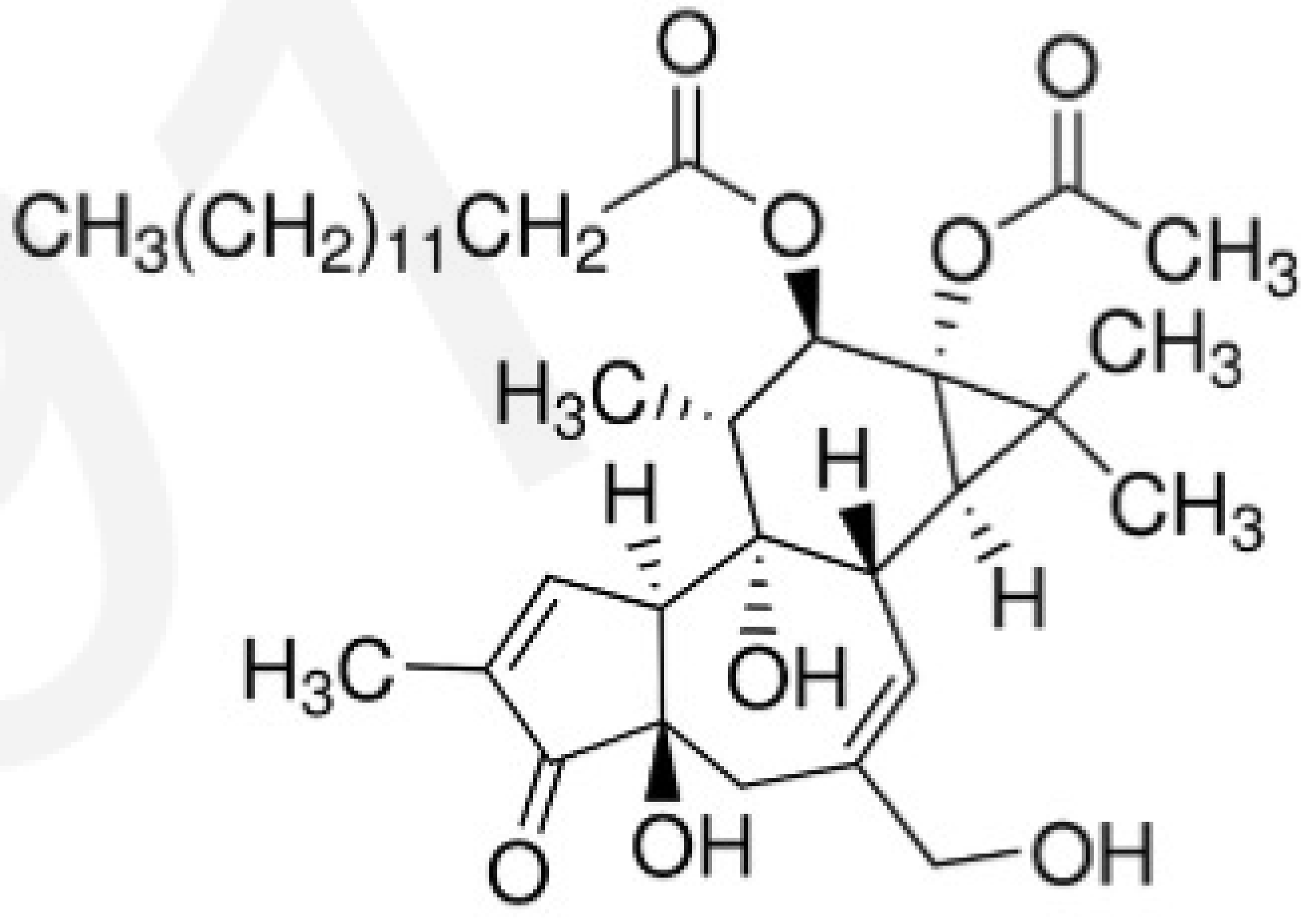
# Graphs as a Structure for Representing Data



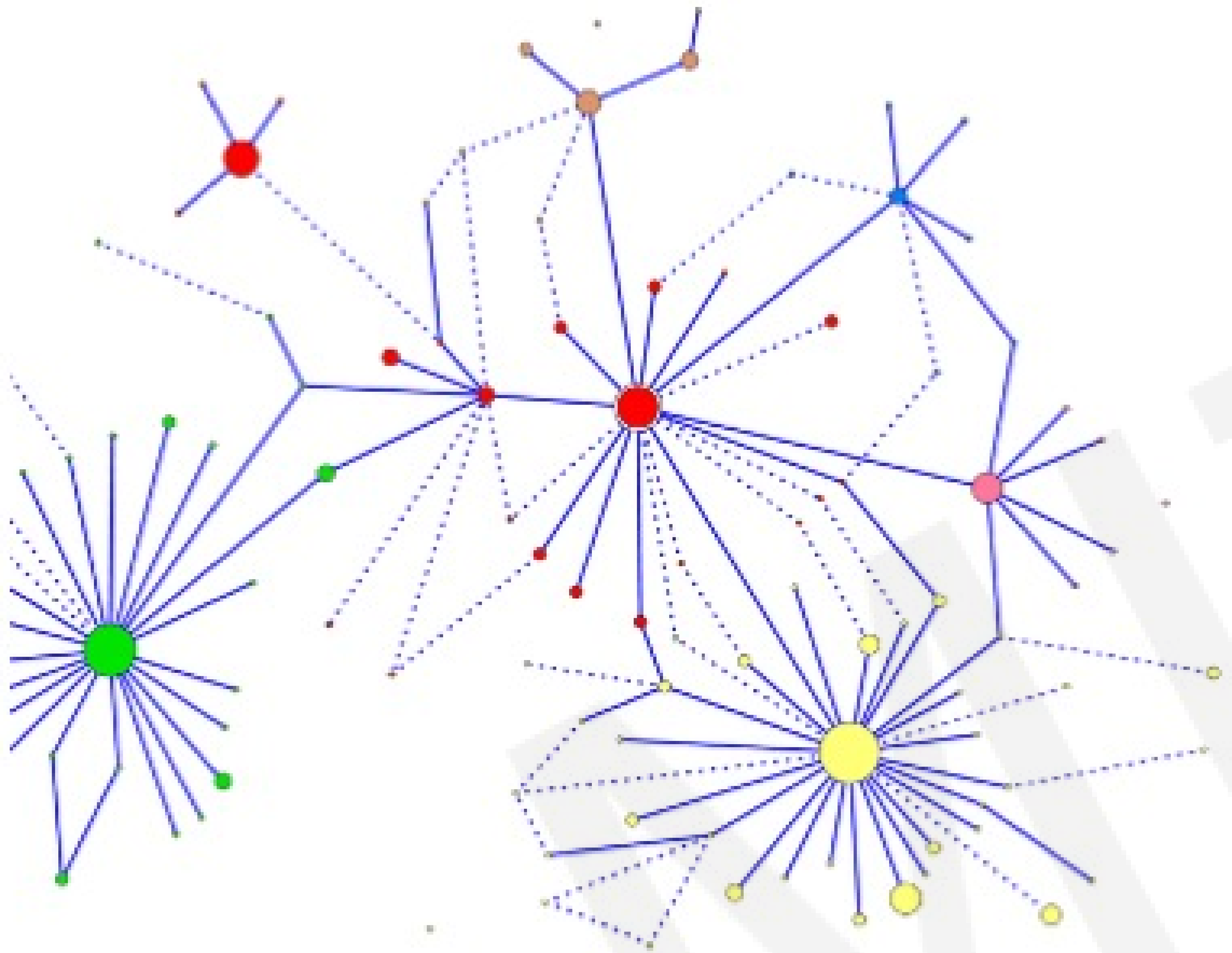

State Machines



Social Networks




Molecules



Biological Networks

**Many real-world data – such as networks – cannot be captured by "standard" encodings or Euclidean geometries**

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

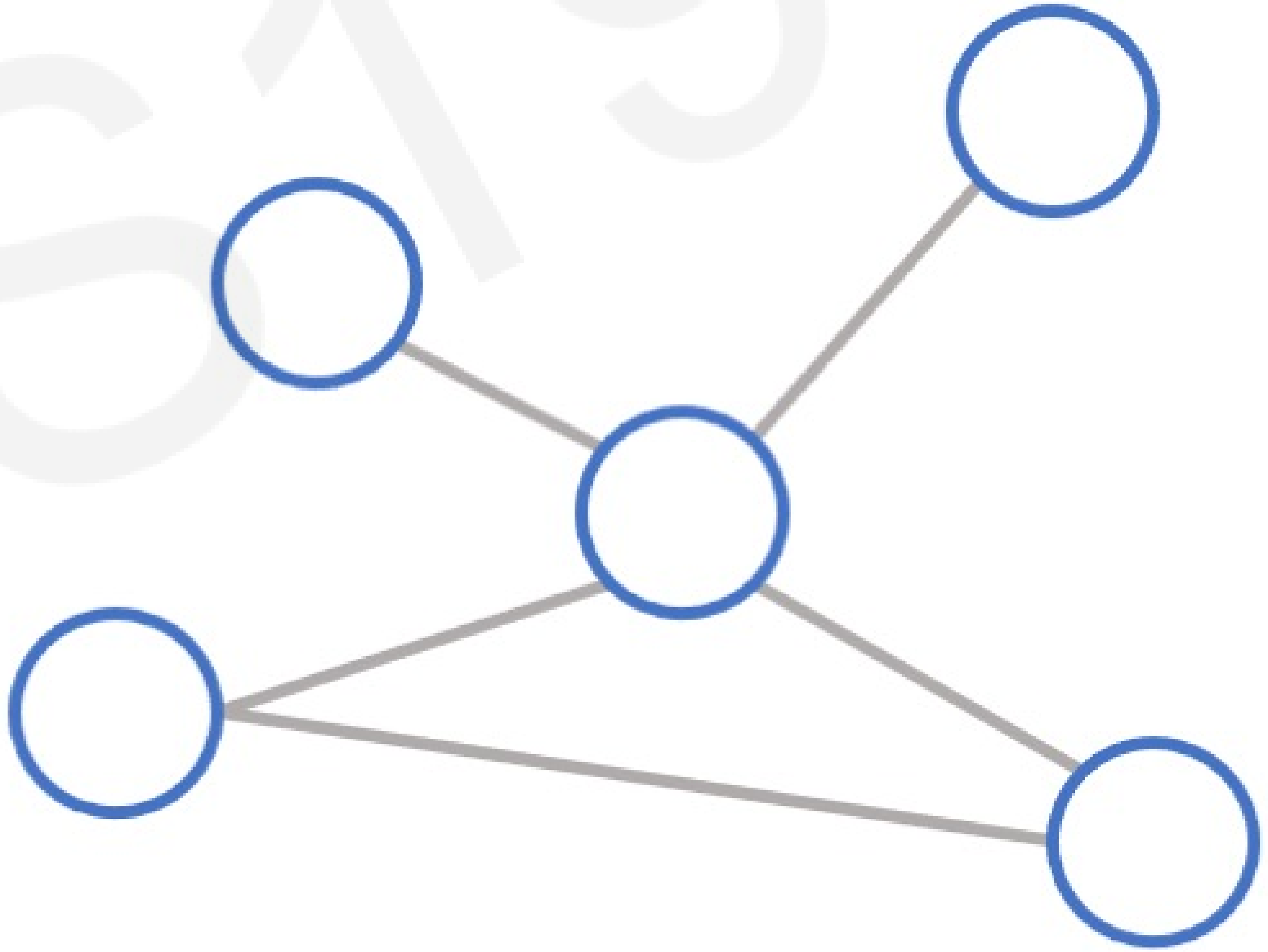
Graph Convolutional Networks (GCNs)

# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)





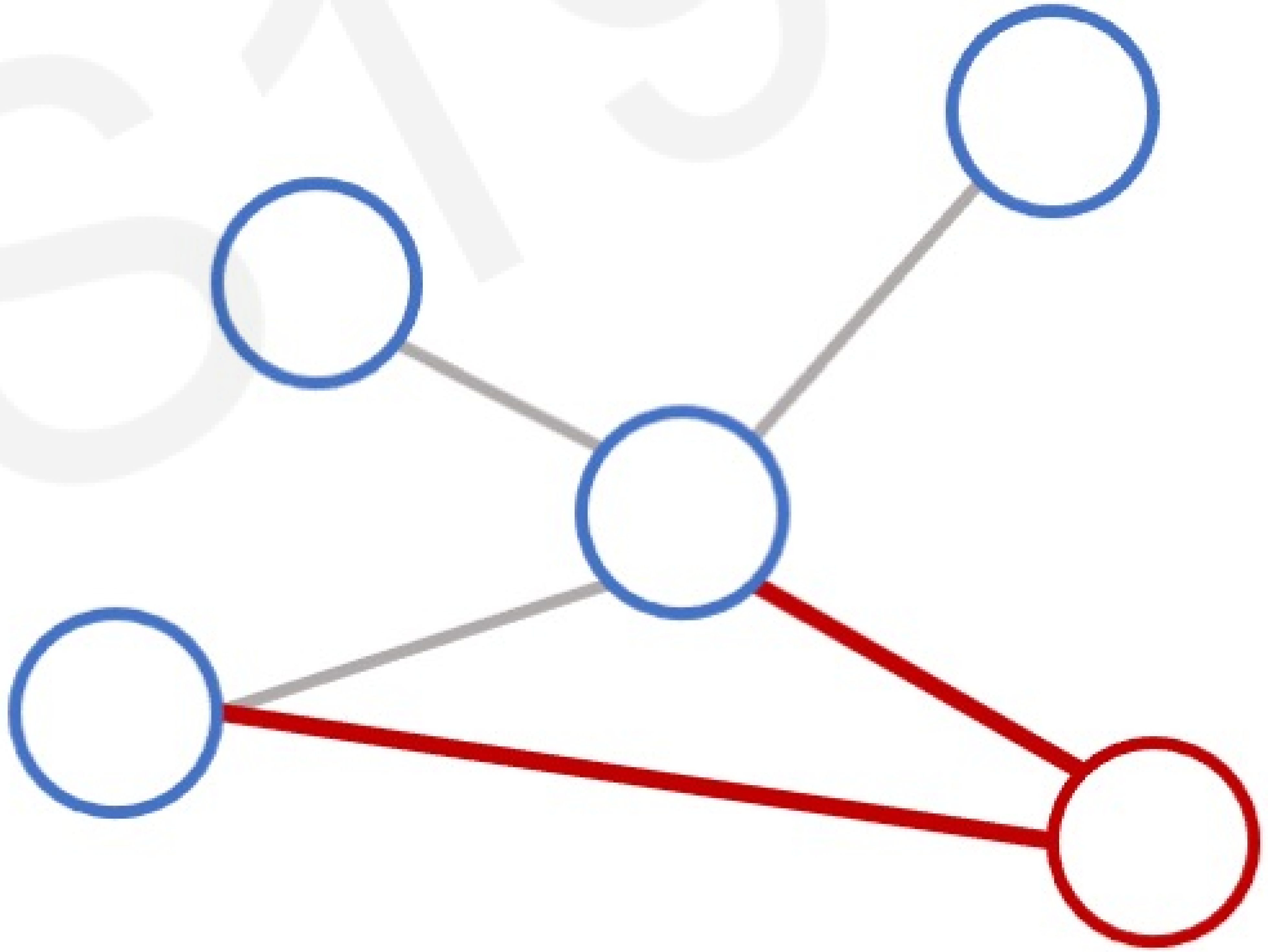
Duvenaud+ *NeurIPS* 2015; Kipf & Welling *ICLR* 2017.

# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

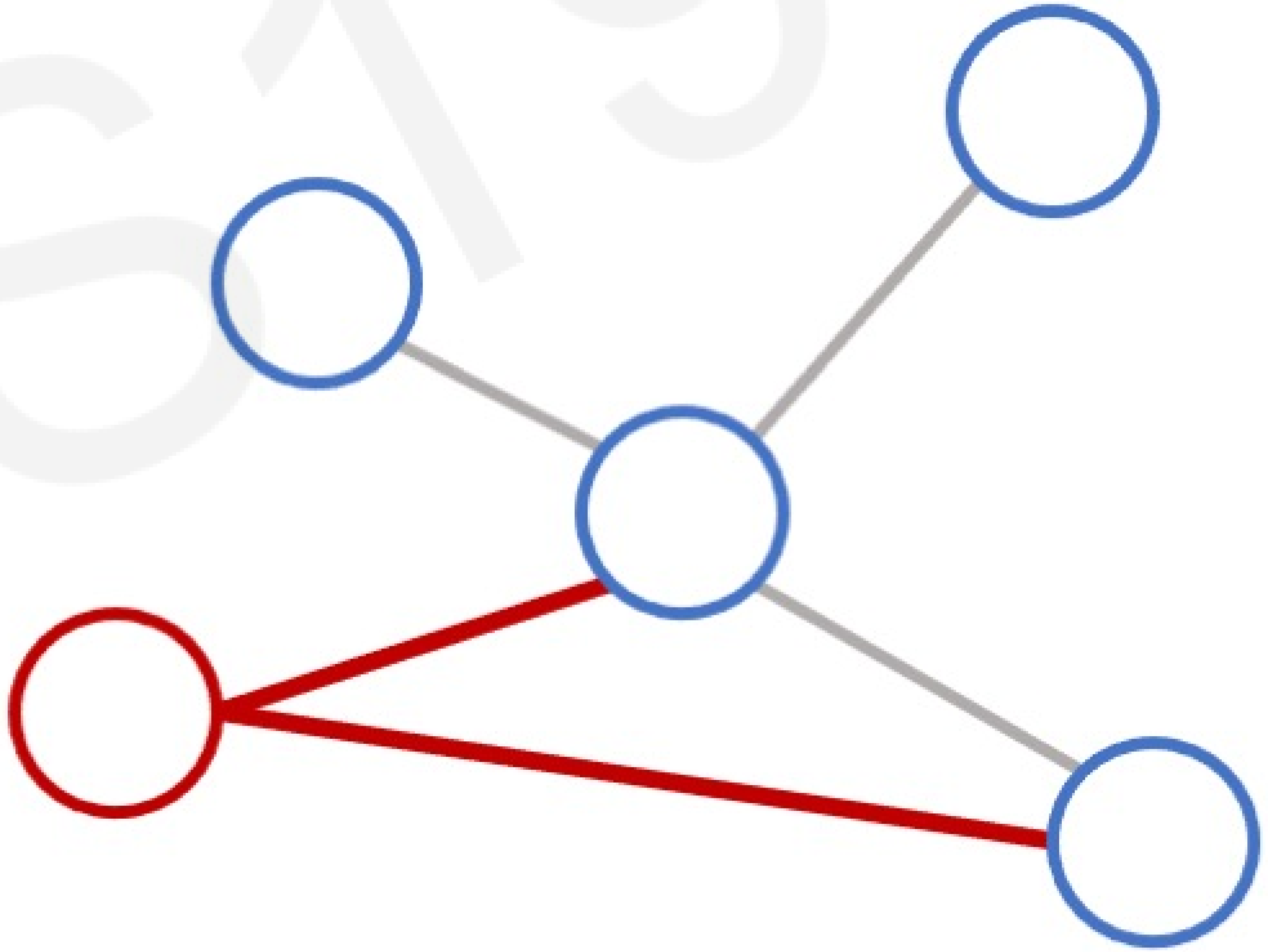


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

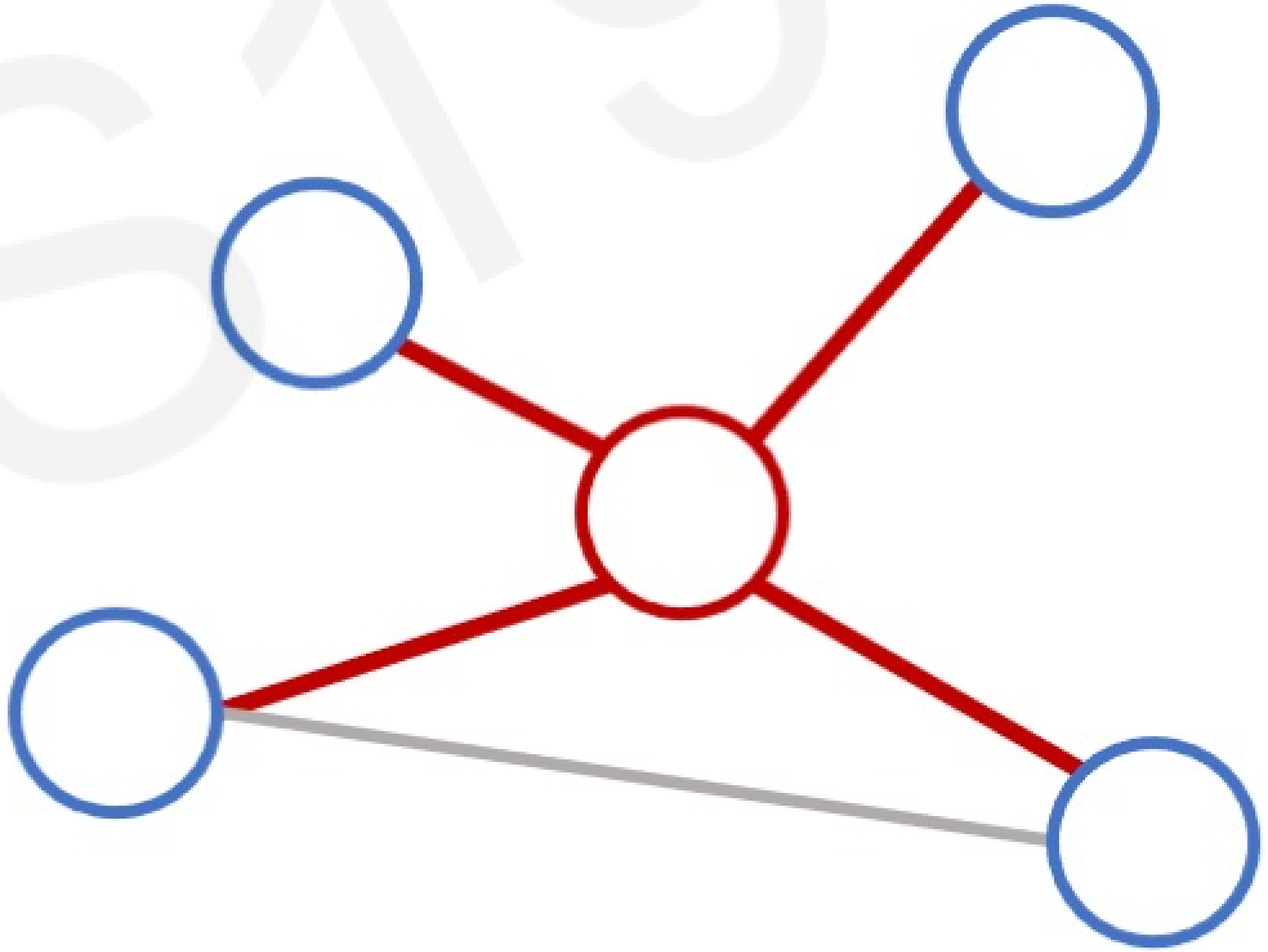


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

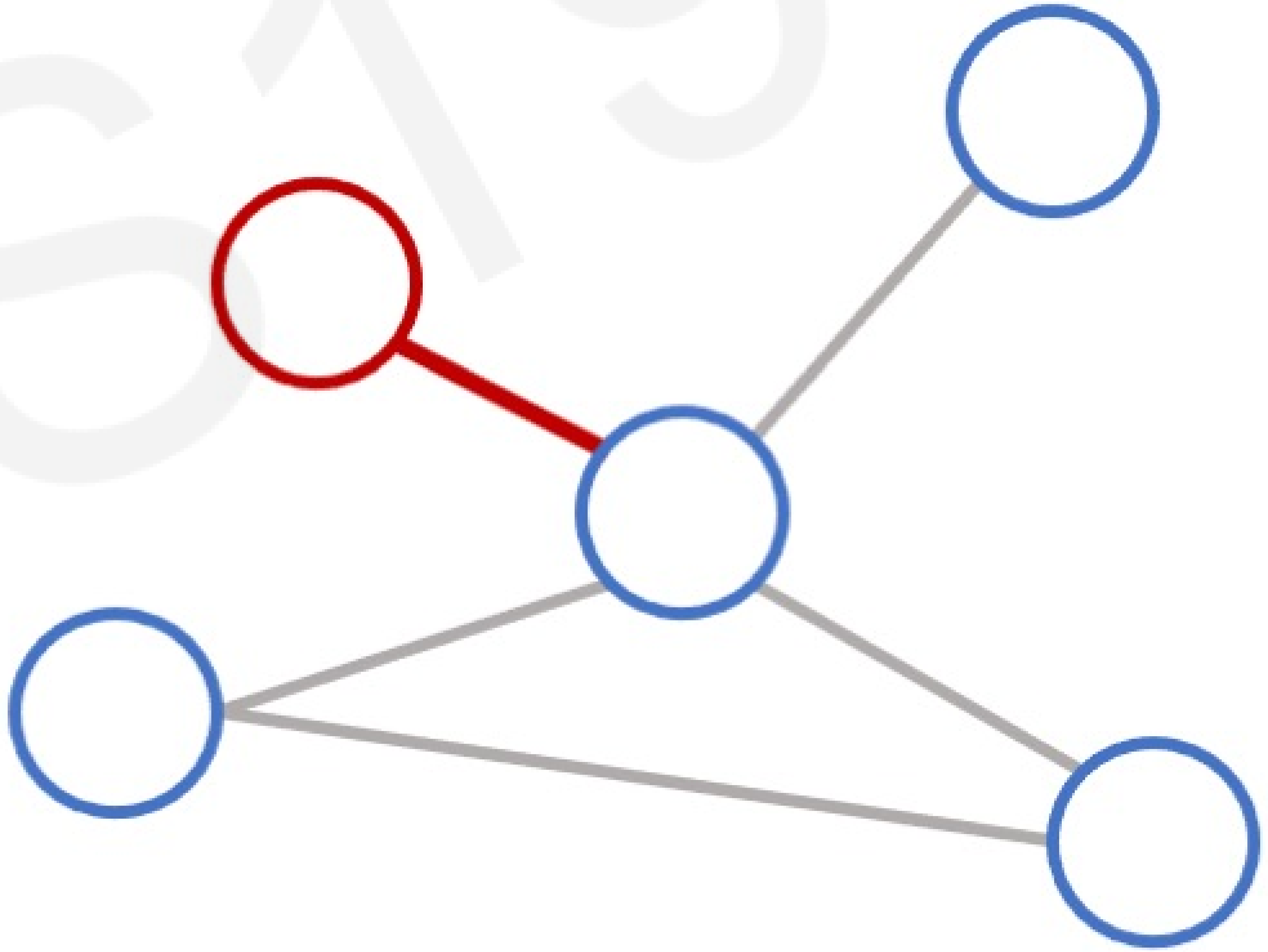


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)







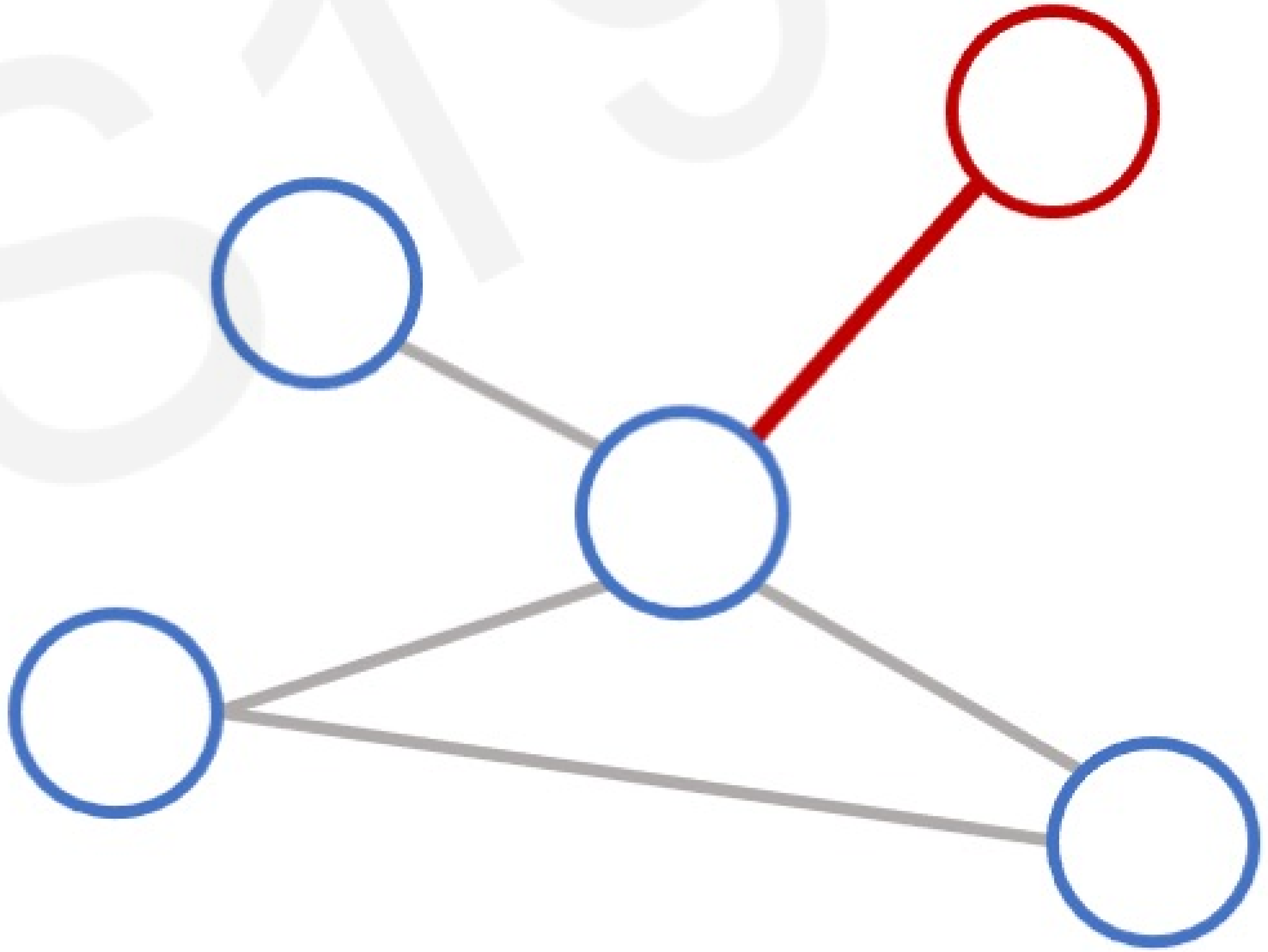
Duvenaud+ *NeurIPS* 2015; Kipf & Welling *ICLR* 2017.

# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)

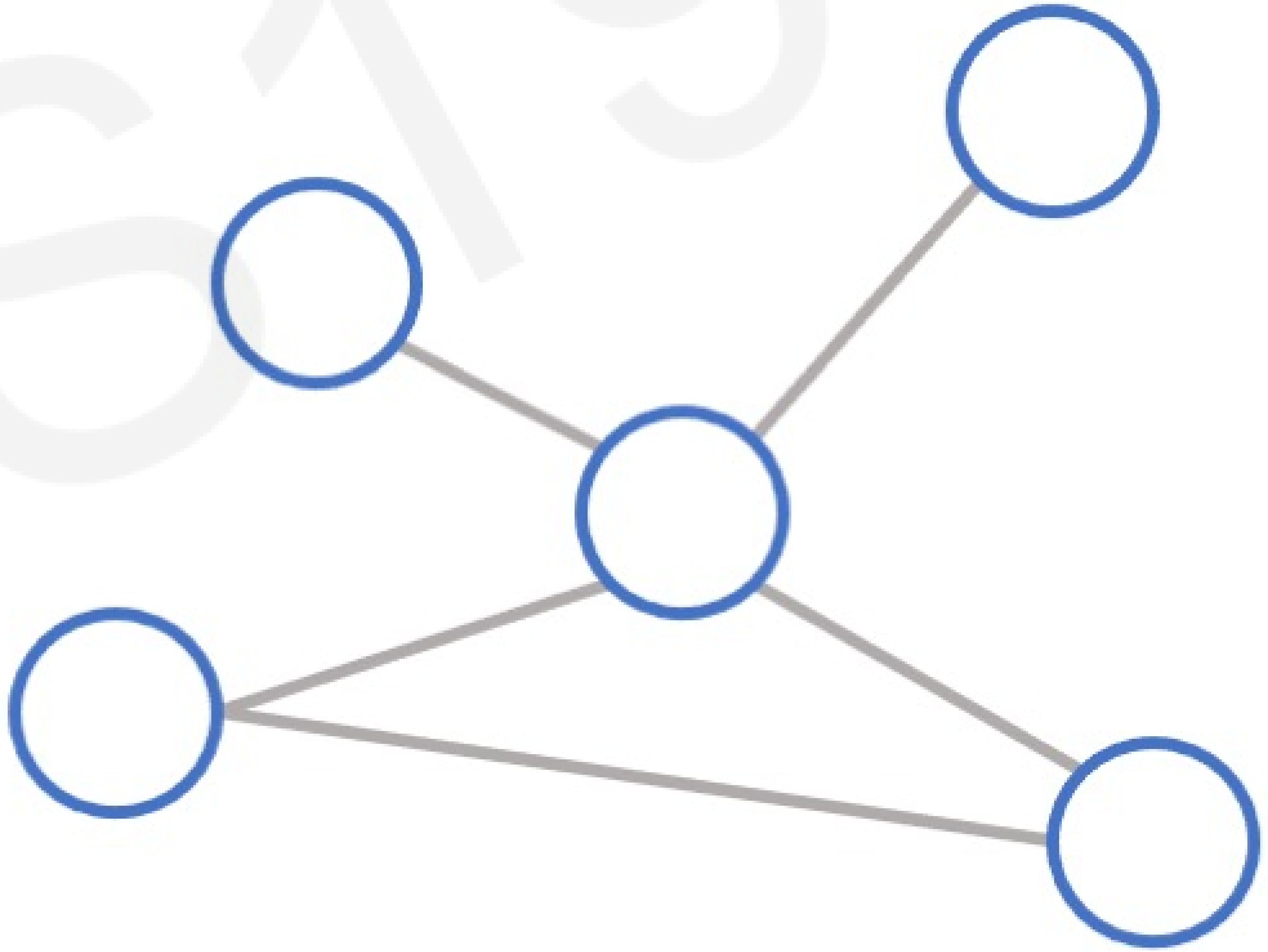


# Graph Convolutional Networks

Convolutional Networks



Graph Convolutional Networks (GCNs)



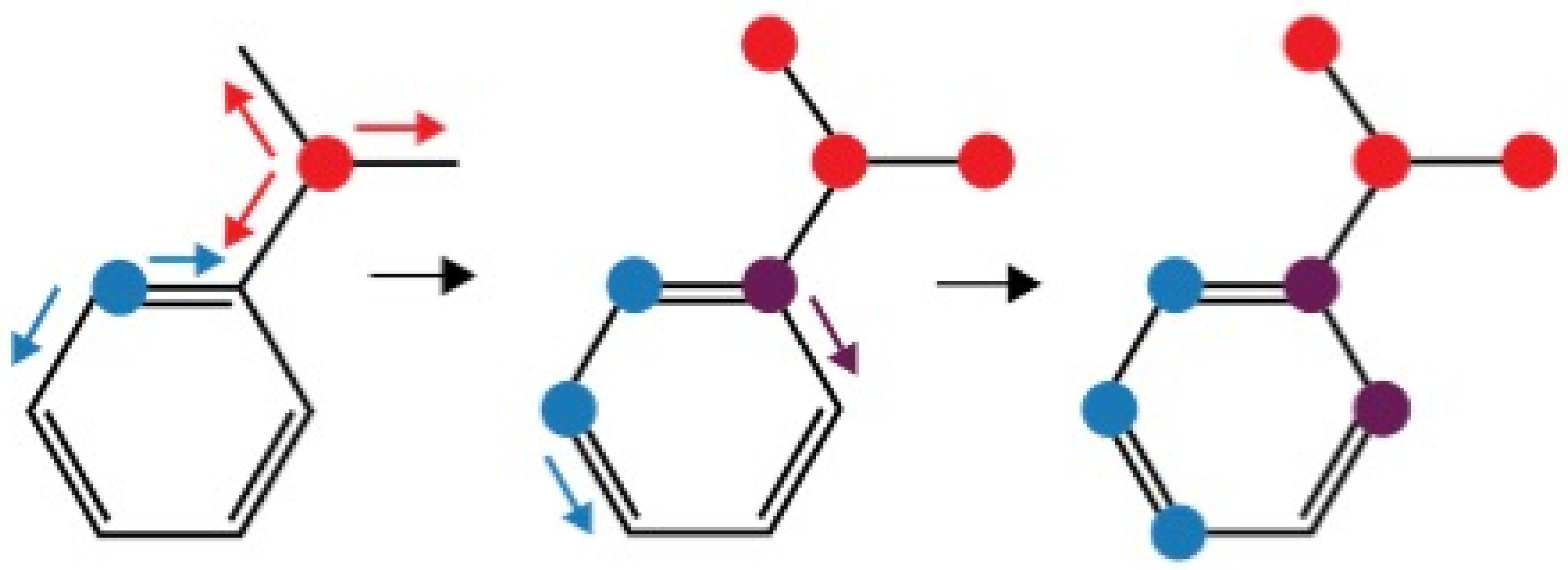




Duvenaud+ *NeurIPS* 2015; Kipf & Welling *ICLR* 2017. 1/25/21

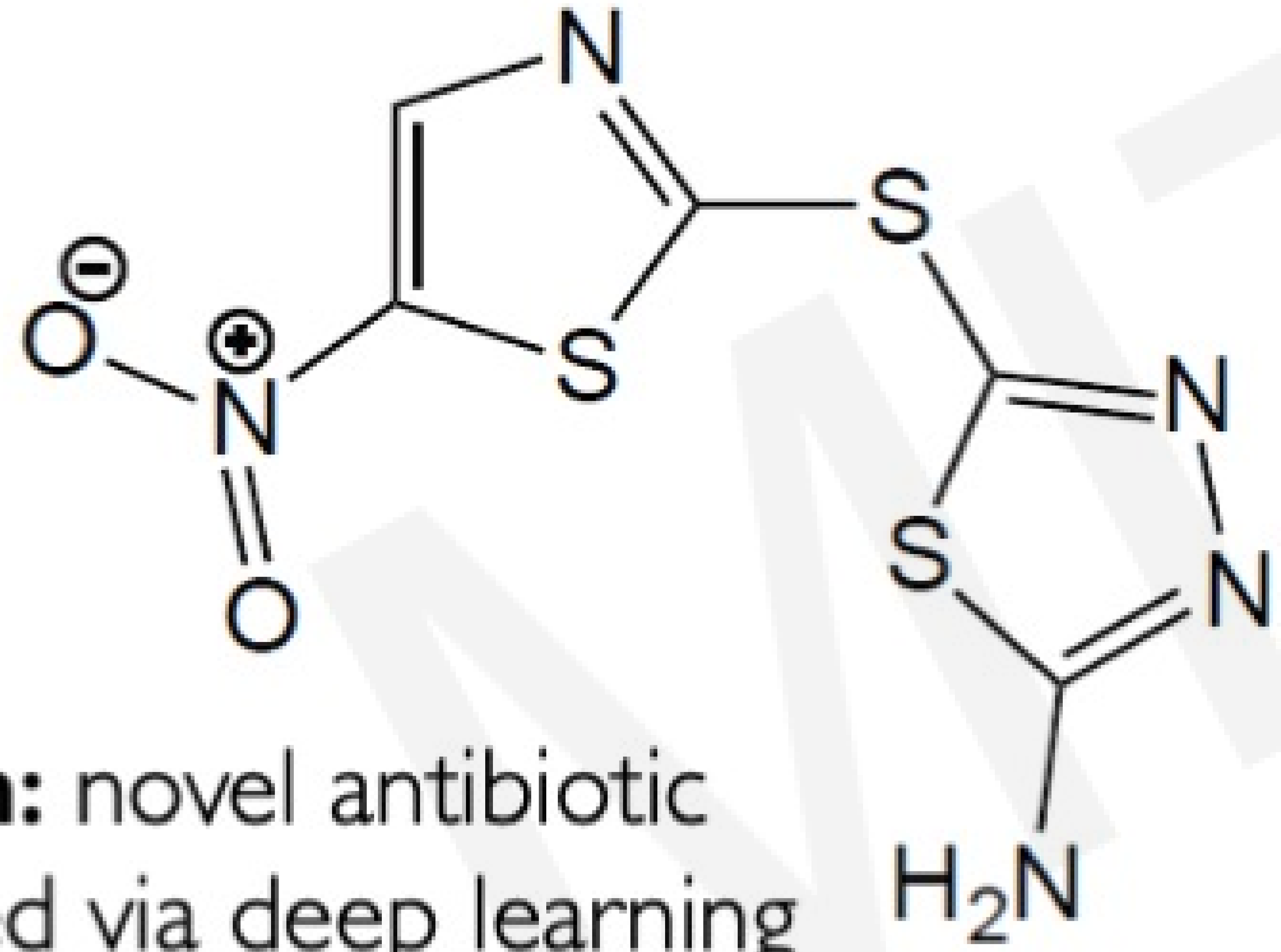
# Applications of Graph Neural Networks

## Molecular Discovery



Message-passing neural network

Jin+ *JCIM* 2019; Soleimany+ *ML4Mol* 2020

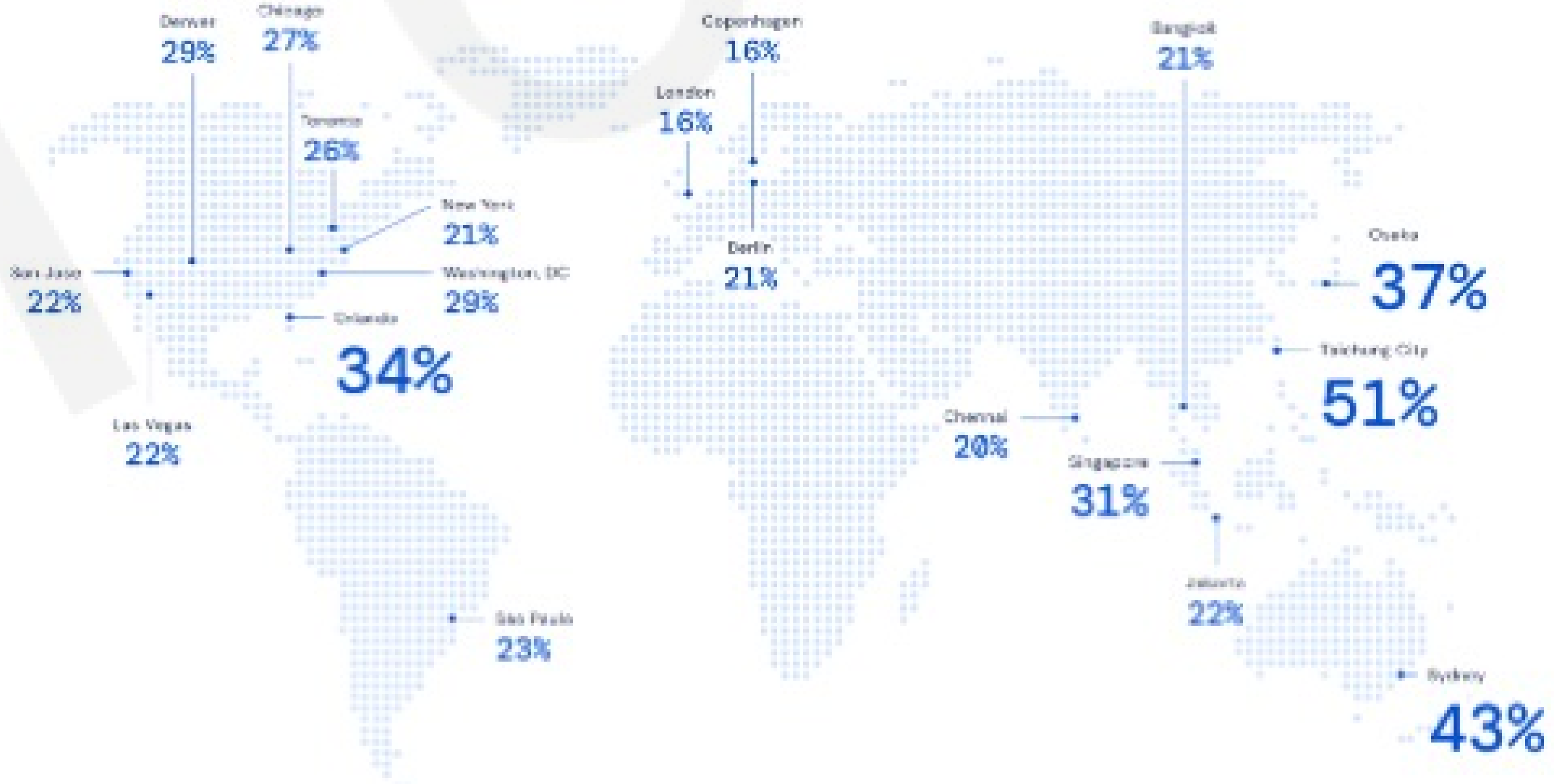



**Halicin:** novel antibiotic discovered via deep learning

Stokes+ *Cell* 2020

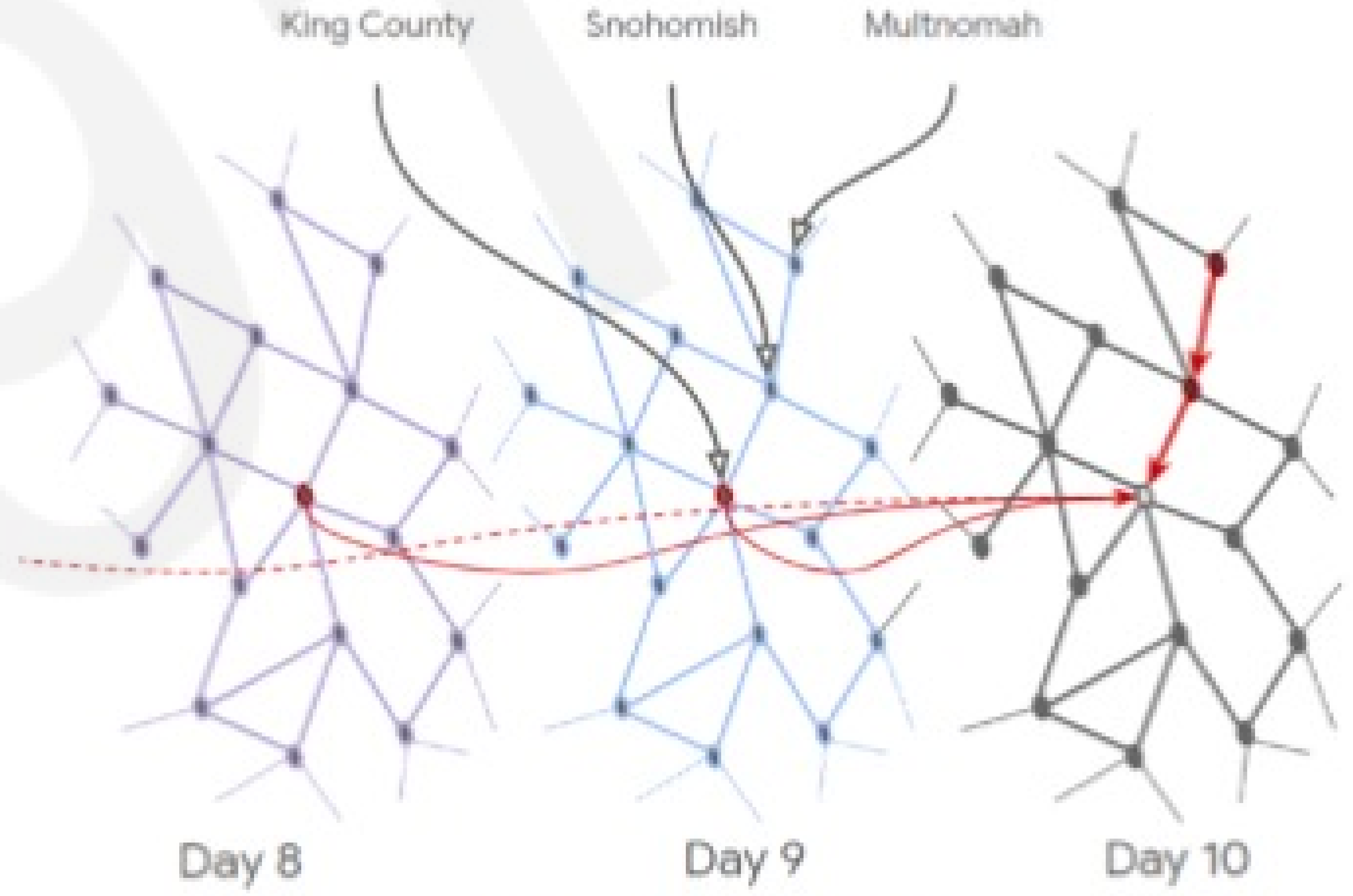
## Traffic Prediction

ETA Improvements with GoogleMaps

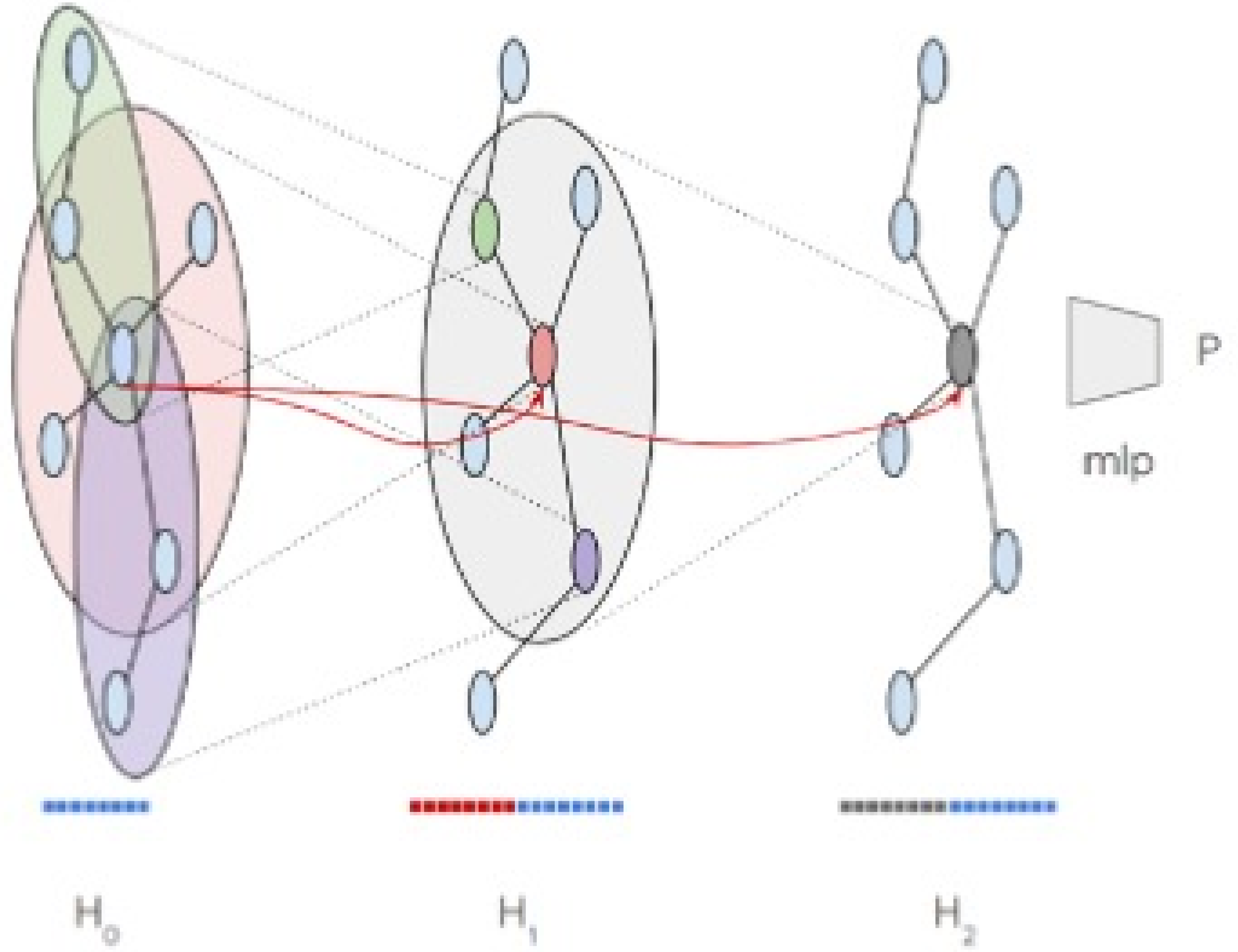



DeepMind + GoogleMaps

## COVID-19 Forecasting




Spatio-temporal data




Graph network + temporal embedding

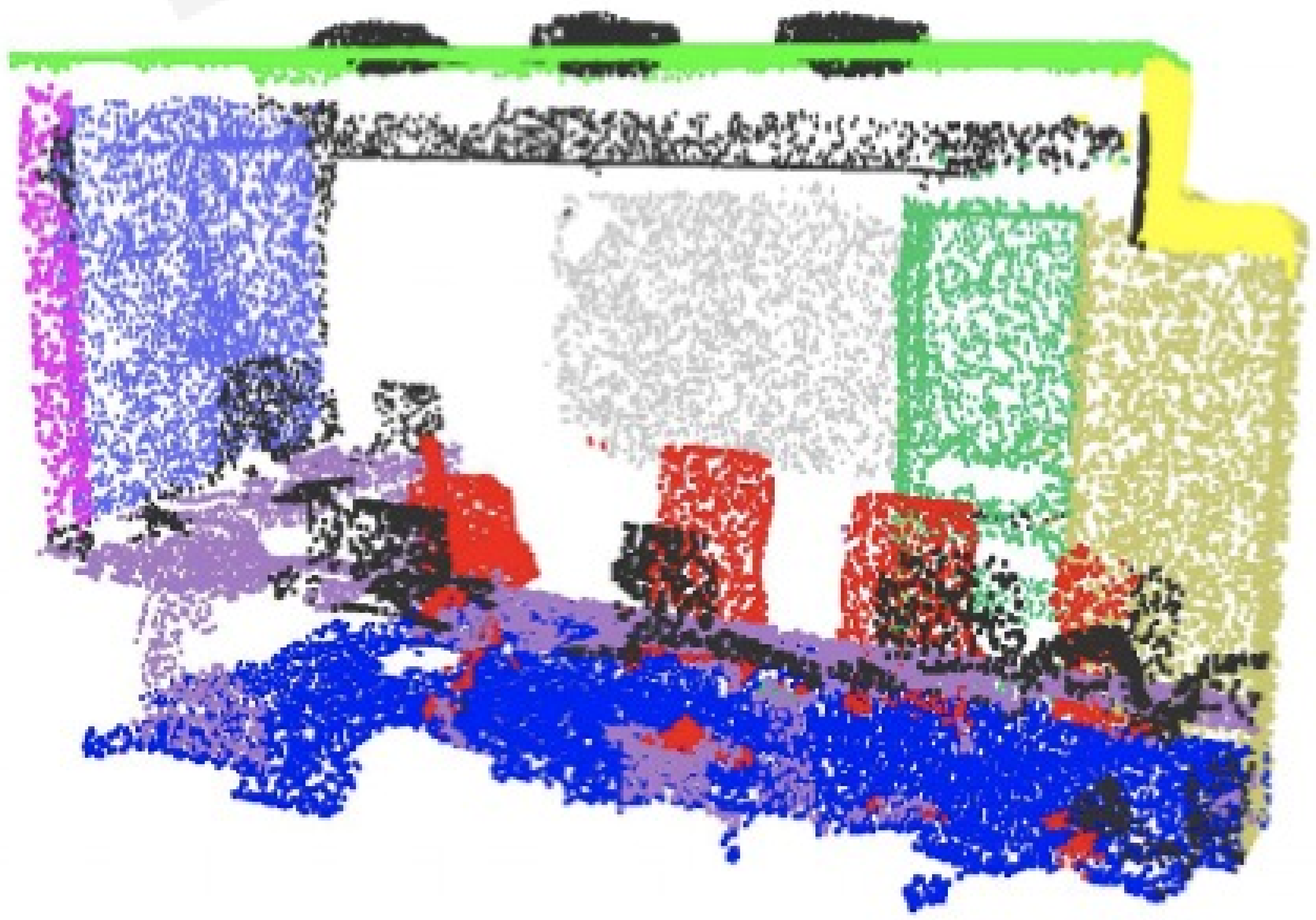
Kapoor+ *KDD* 2020

# Learning From 3D Data

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


Classification

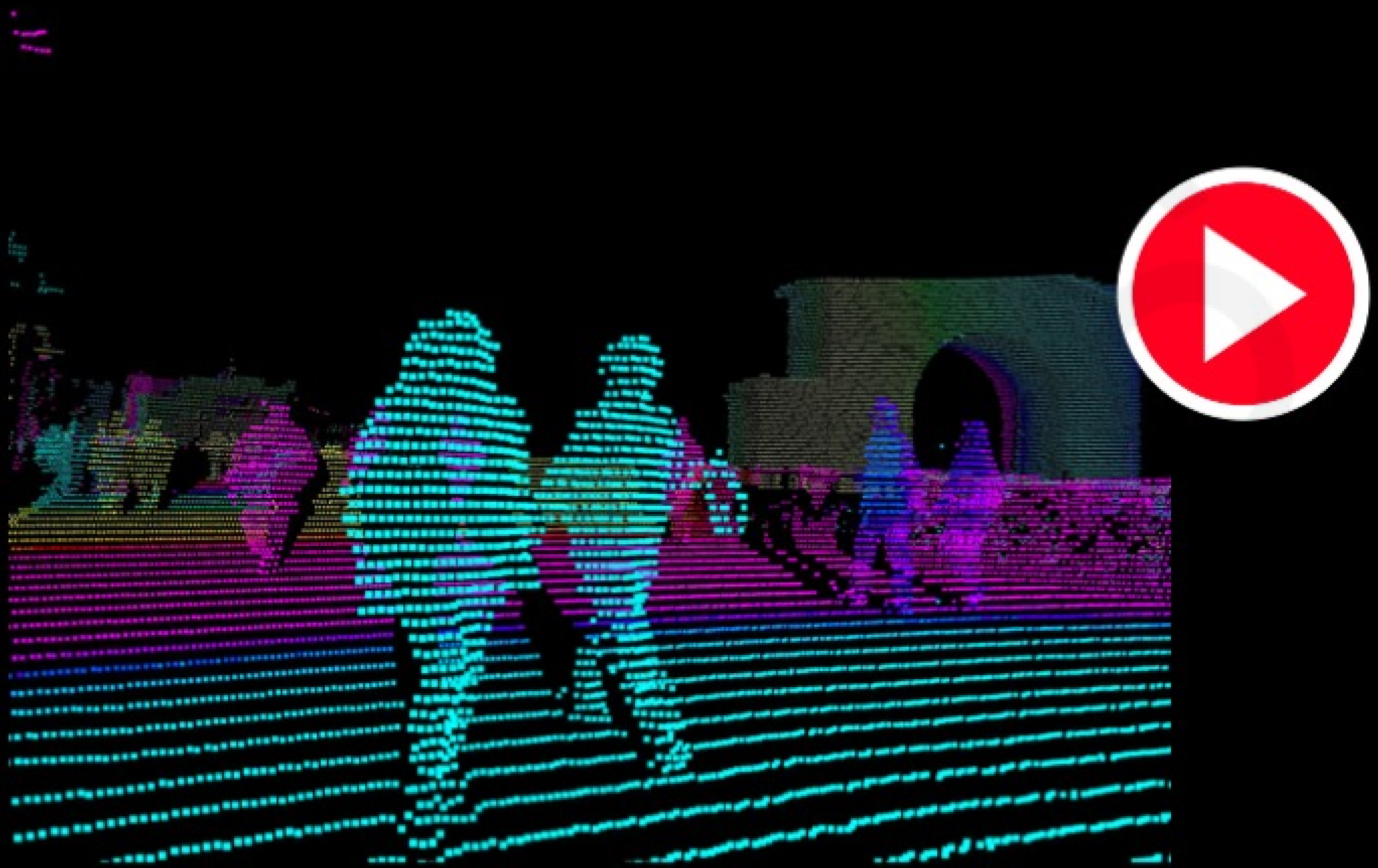
Part Segmentation

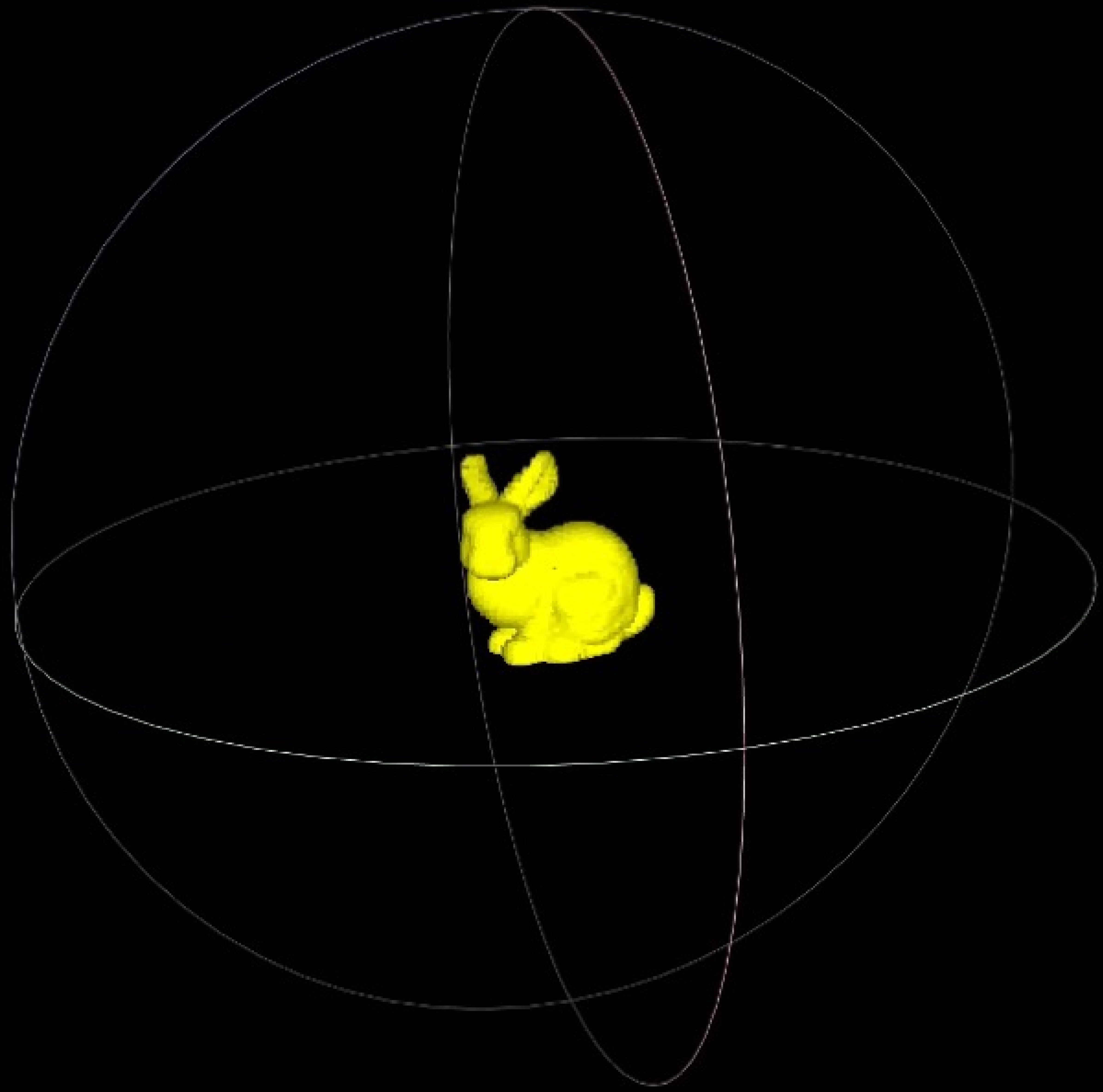


Semantic Segmentation

# Extending Graph CNNs to Pointclouds

**Capture local geometric features of point clouds while maintaining order invariance**

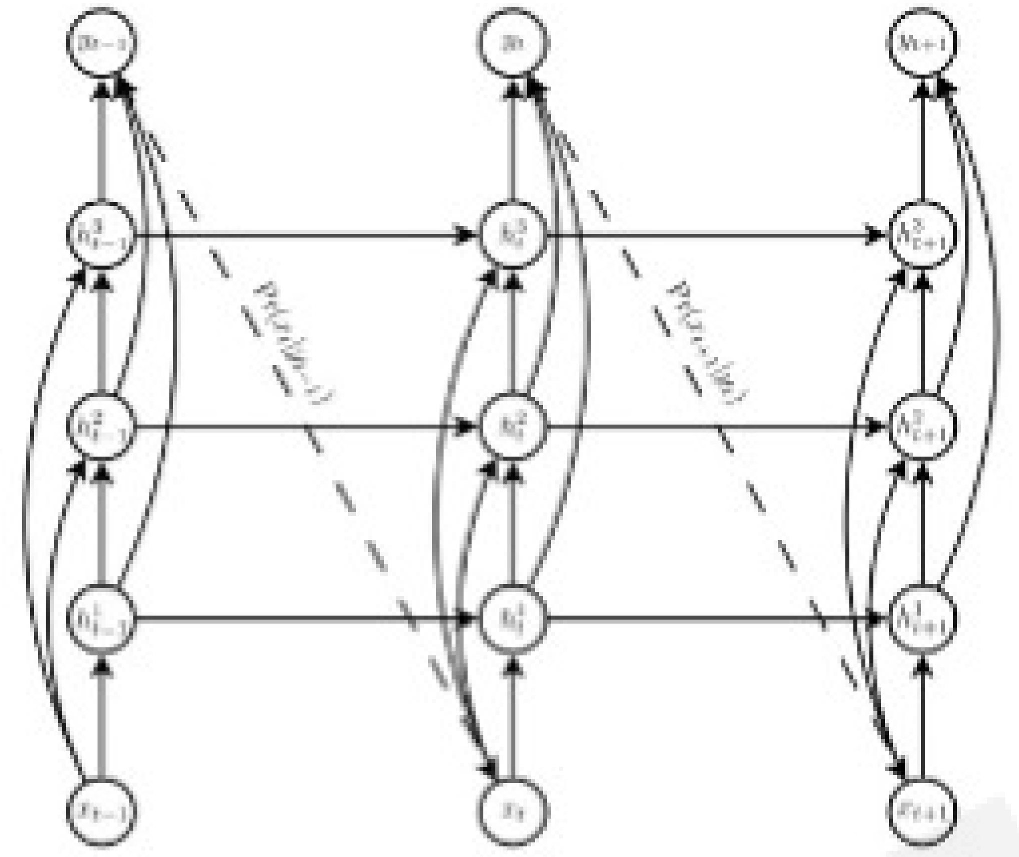




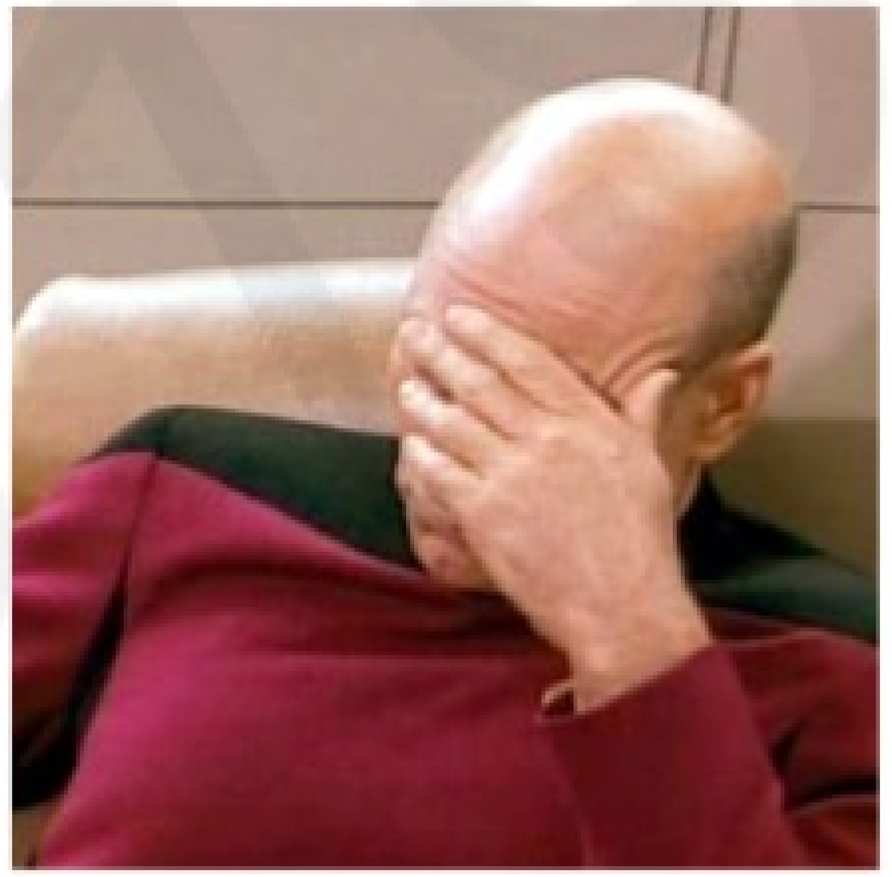
# New Frontiers II:
## Automated Machine Learning & AI

# Motivation: Automated Machine Learning

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



Complexity of models increases



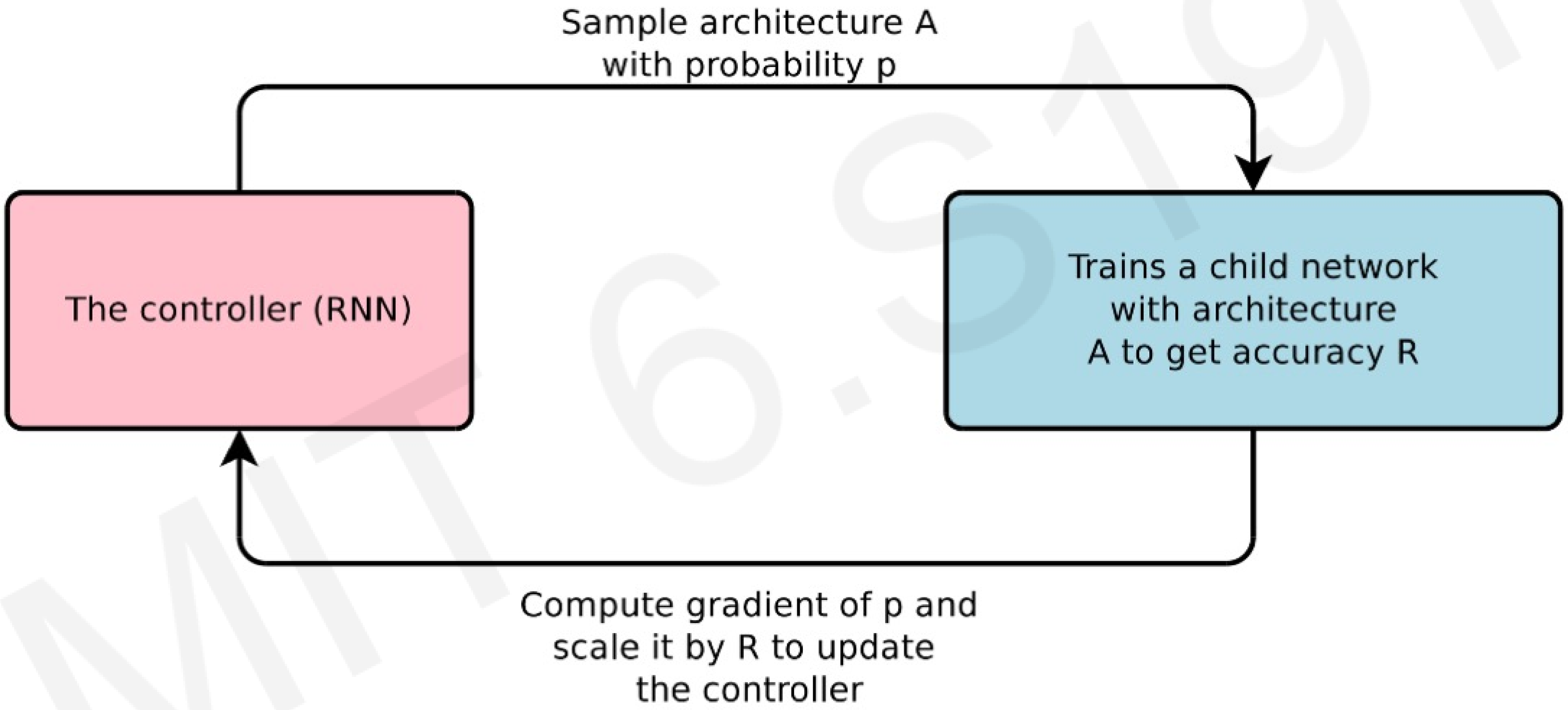
Greater need for specialized engineers

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

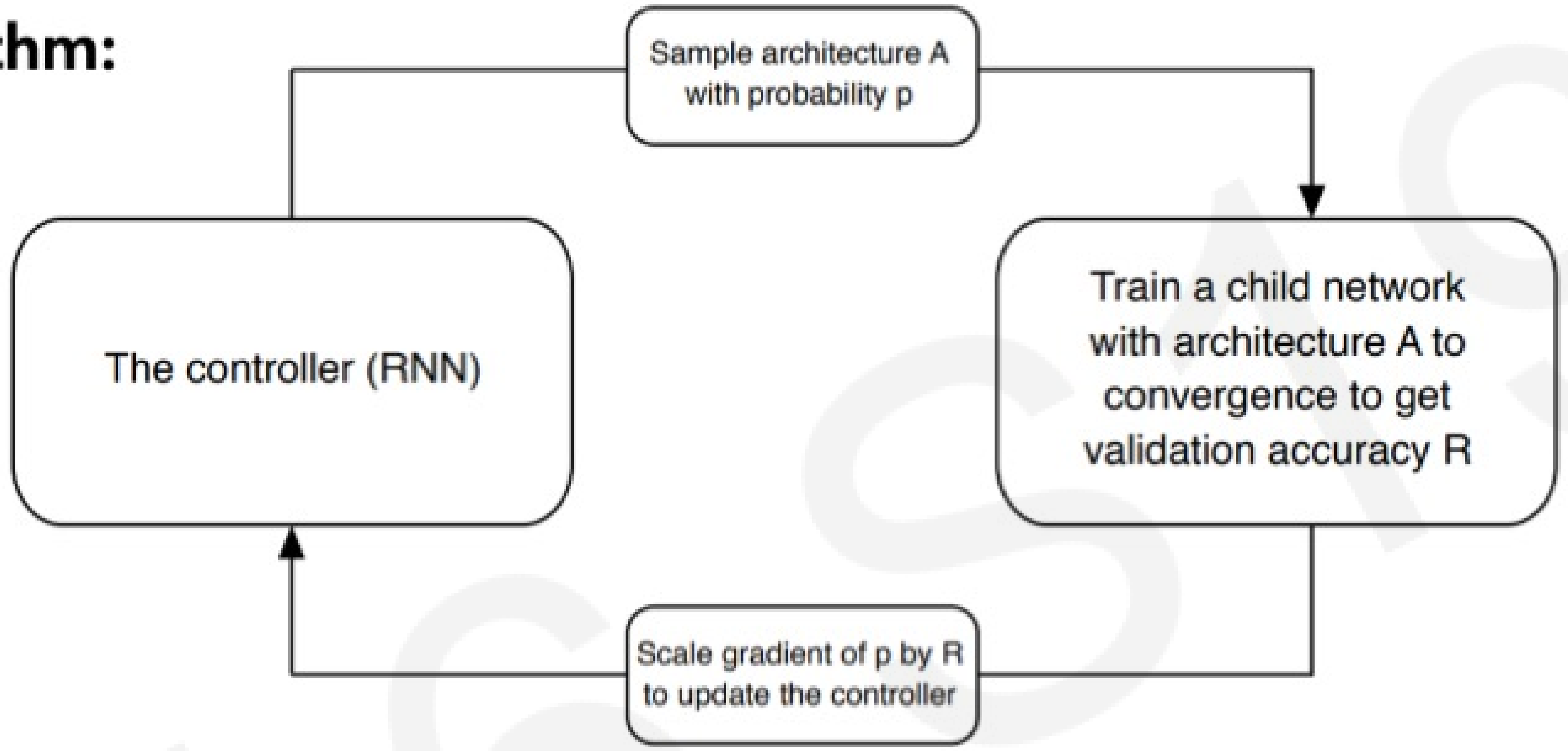
# AutoML: The Child Network

Training Data → Sampled network from RNN → Prediction

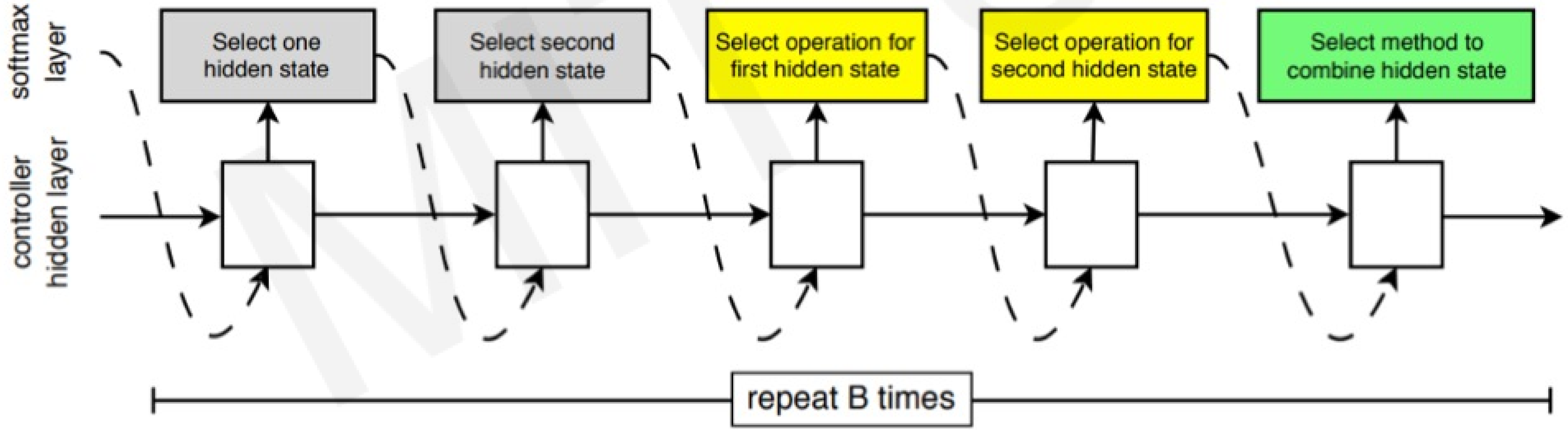
Compute final accuracy on this dataset.
Update RNN controller based on the accuracy of the child network after training.

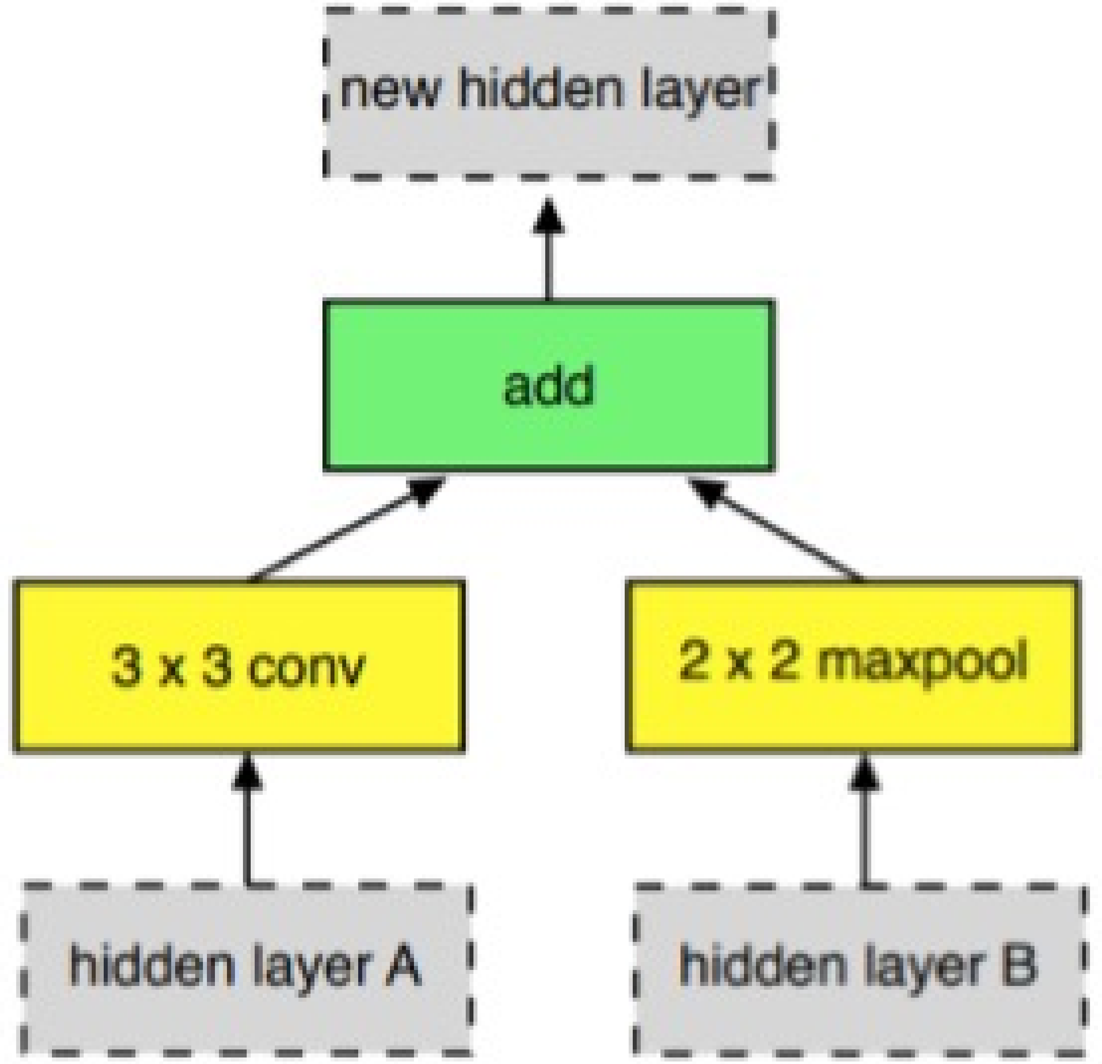
# Learning Architectures for Image Recognition

**Neural architecture search algorithm:**



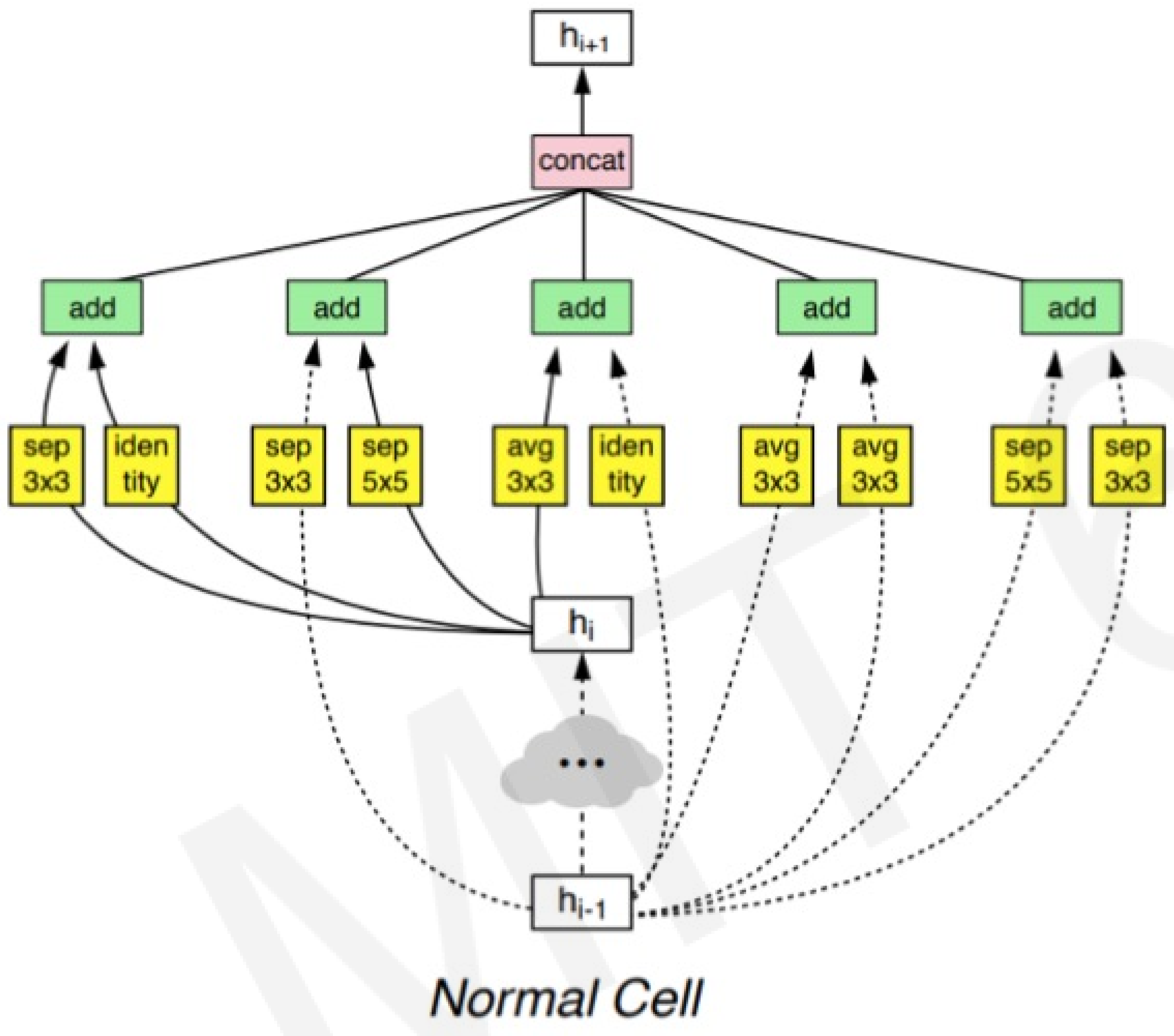

**Controller architecture for constructing convolutional layers:**



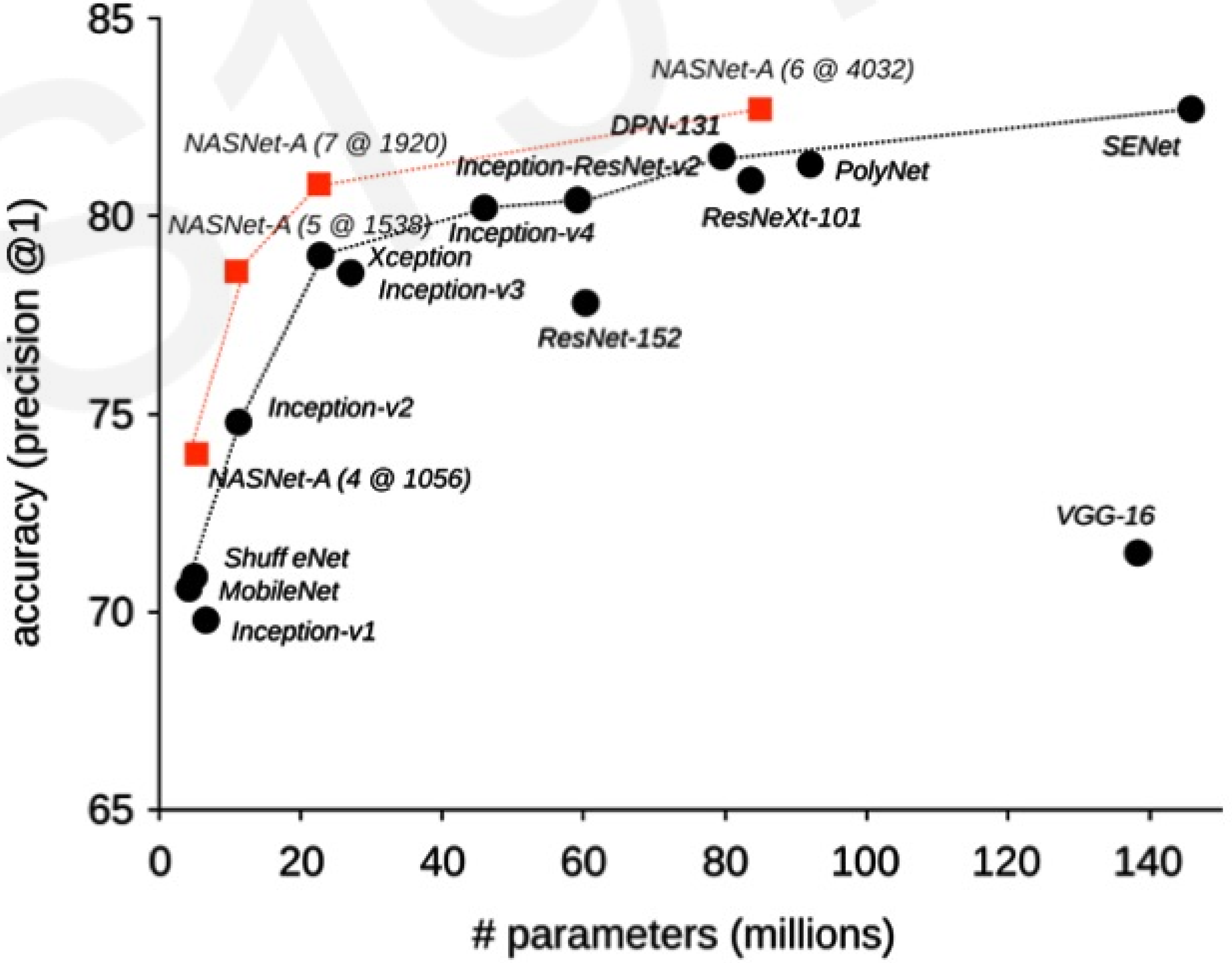



# Learning Architectures for Image Recognition

**Learned architecture for convolutional cell**




**Model performance on ImageNet**



# From AutoML to AutoAI

## AutoAI

**Provide data in a CSV file** → **Prepare data** → **Select model type** → **Generate and rank model pipelines** → **Save and deploy a model**

**Prepare data**
- Feature type detection
- Missing values imputation
- Feature encoding and scaling

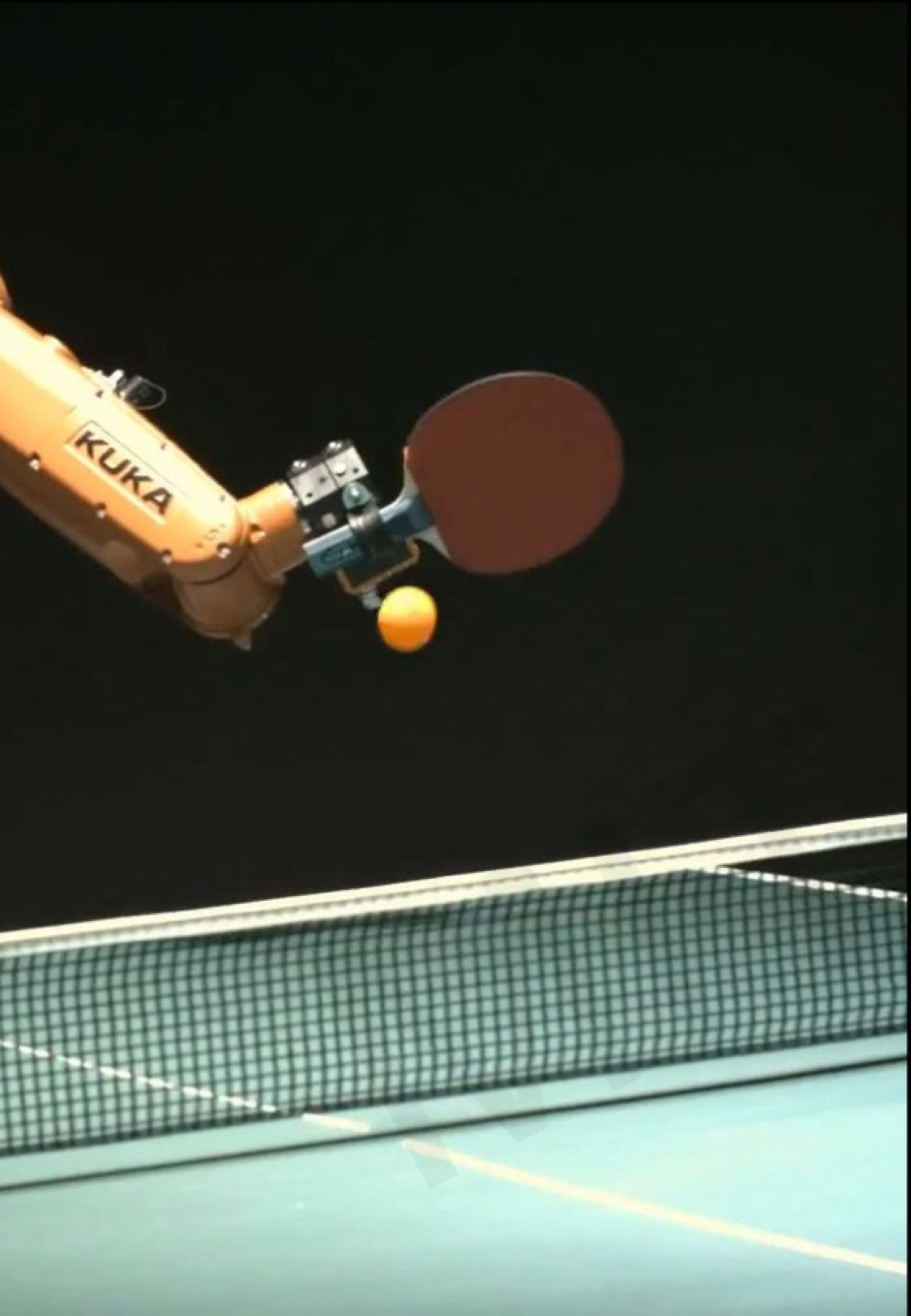
**Select model type**
- Selection of the best algorithm for the data

**Generate and rank model pipelines**
- Hyper-parameter optimization (HPO)
- Optimized feature engineering

# AutoAI Spawns a Powerful Idea

- Design an AI pipeline 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? Come to class Gather.Town!