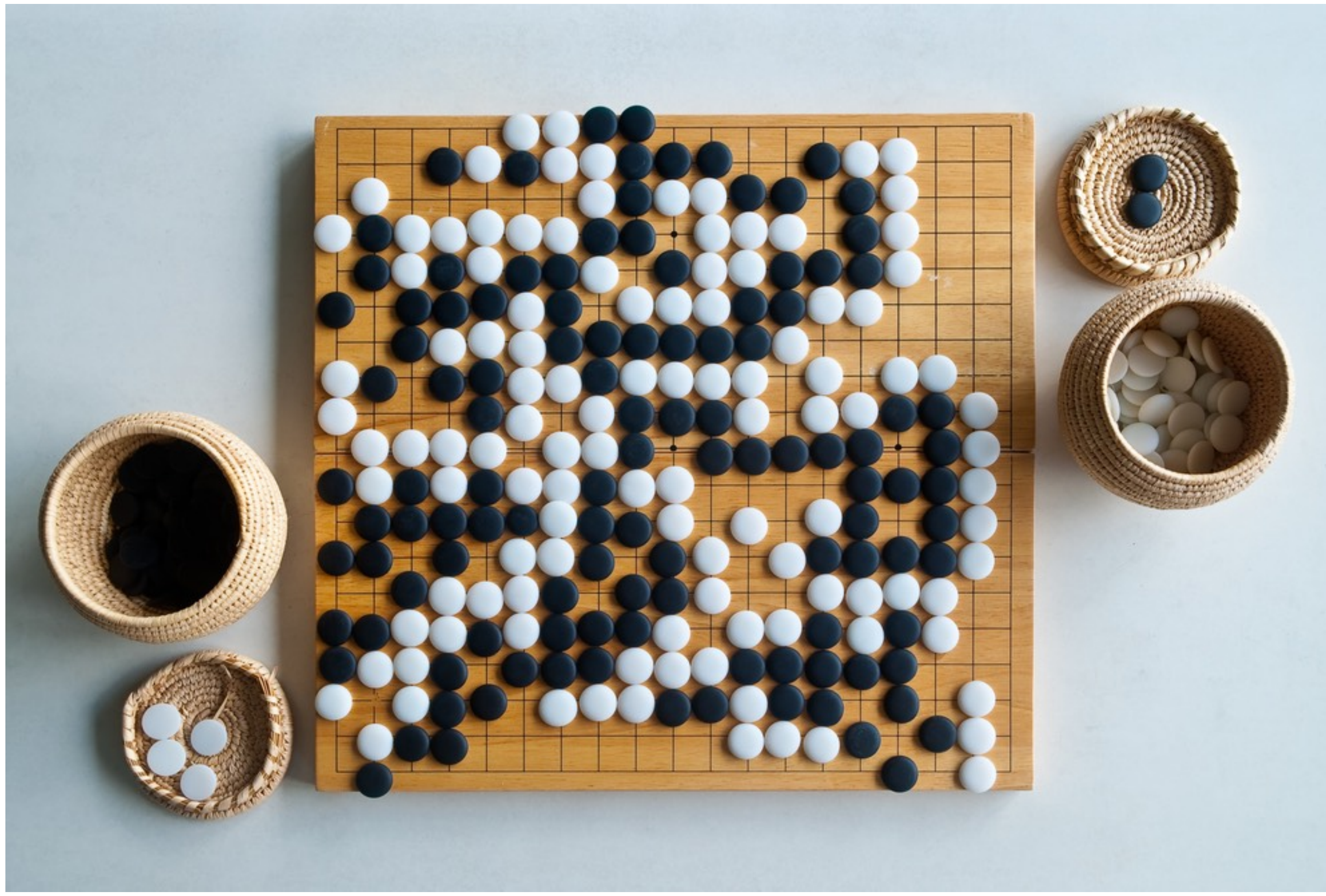
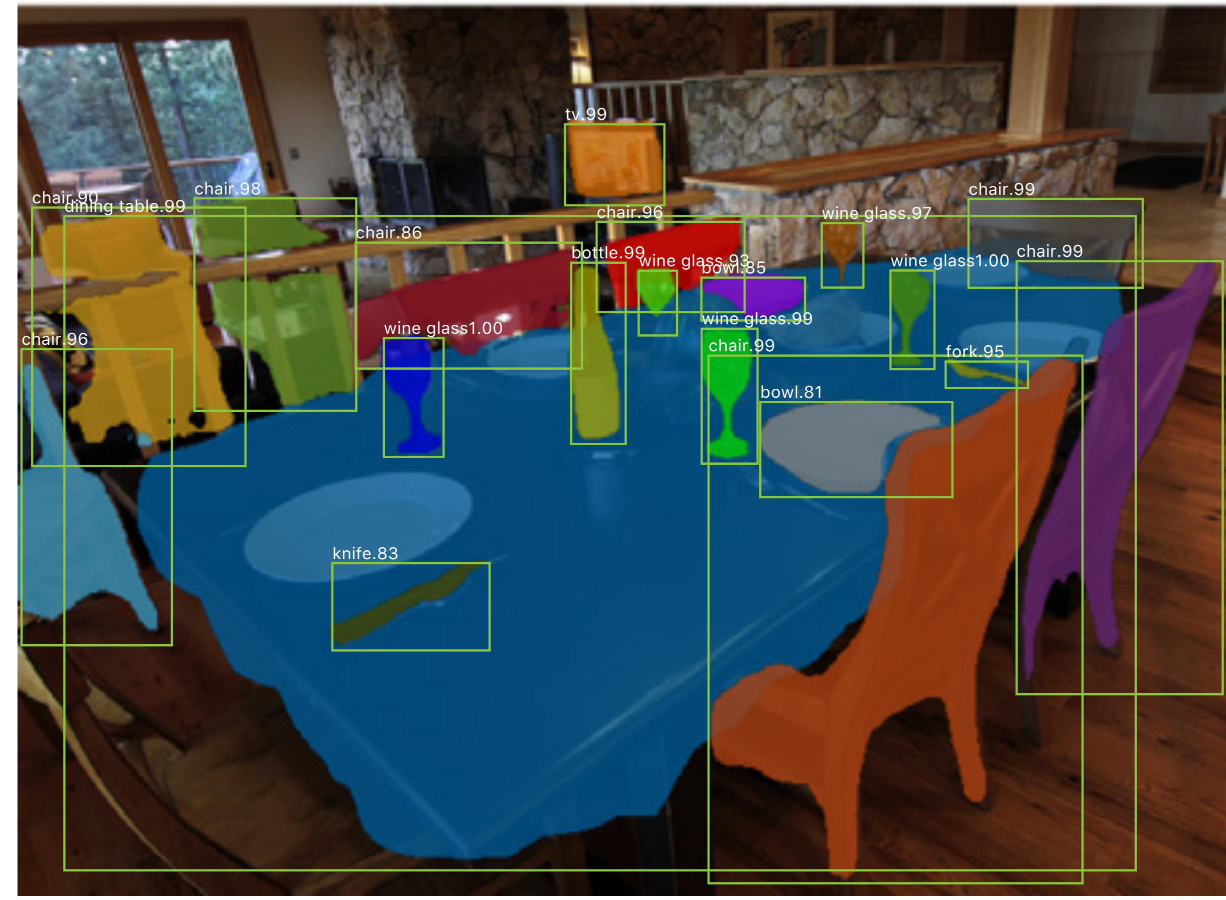
Musings on Continual Learning

Pulkit Agrawal





tv.99
chair.90
dining table.99
chair.98
chair.96
chair.86
wine glass.97
chair.99
bottle.99
wine glass.93
bowl.85
wine glass1.00
chair.99
chair.96
wine glass1.00
wine glass.99
chair.99
fork.95
bowl.81
knife.83

Boston Dynamics

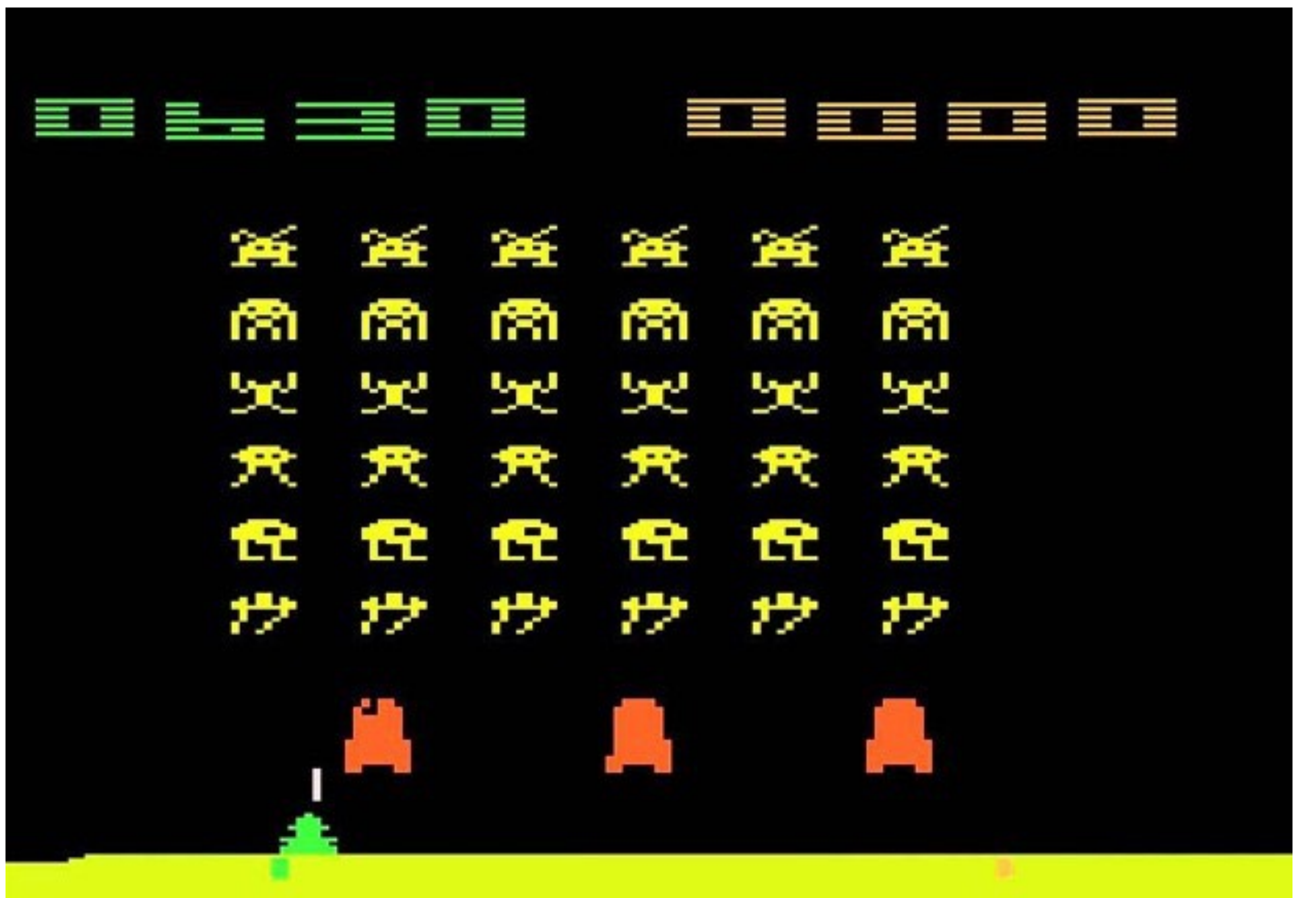
What is a zebra?

What is a zebra?

Success in Reinforcement Learning

ATARI Games

AlphaGo



~10-50 million interactions!

21 million games!

Impressive Specialists

Today's AI **AI we want**

Task Specific → Generalists

???

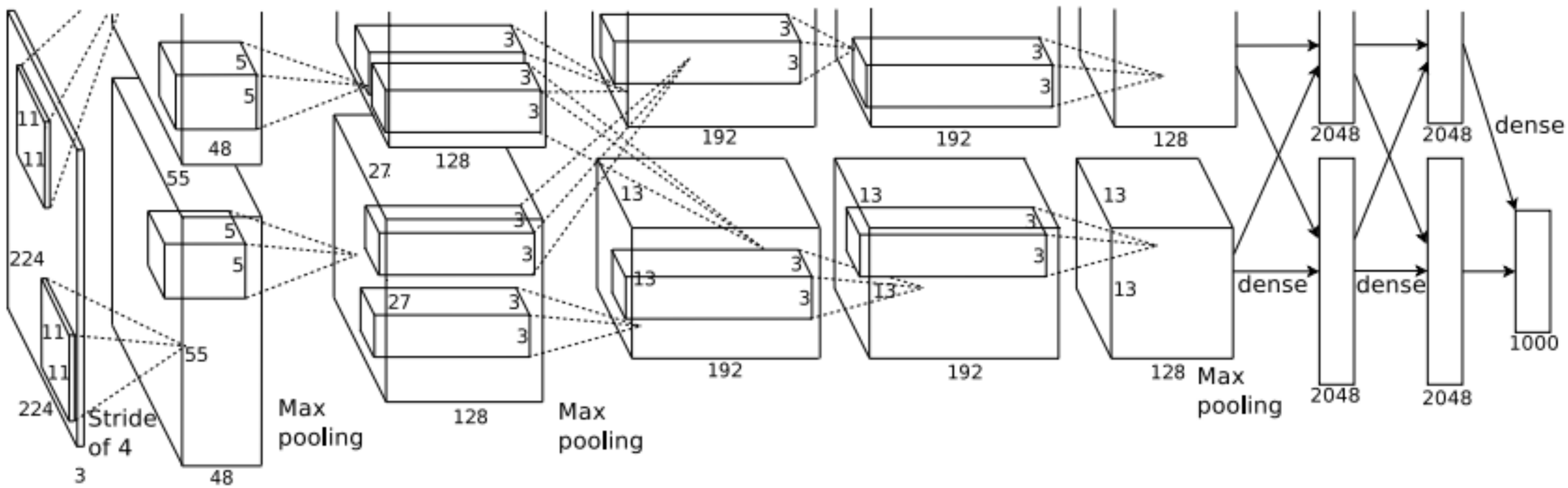
Core Characteristic: Reuse past knowledge to solve new tasks

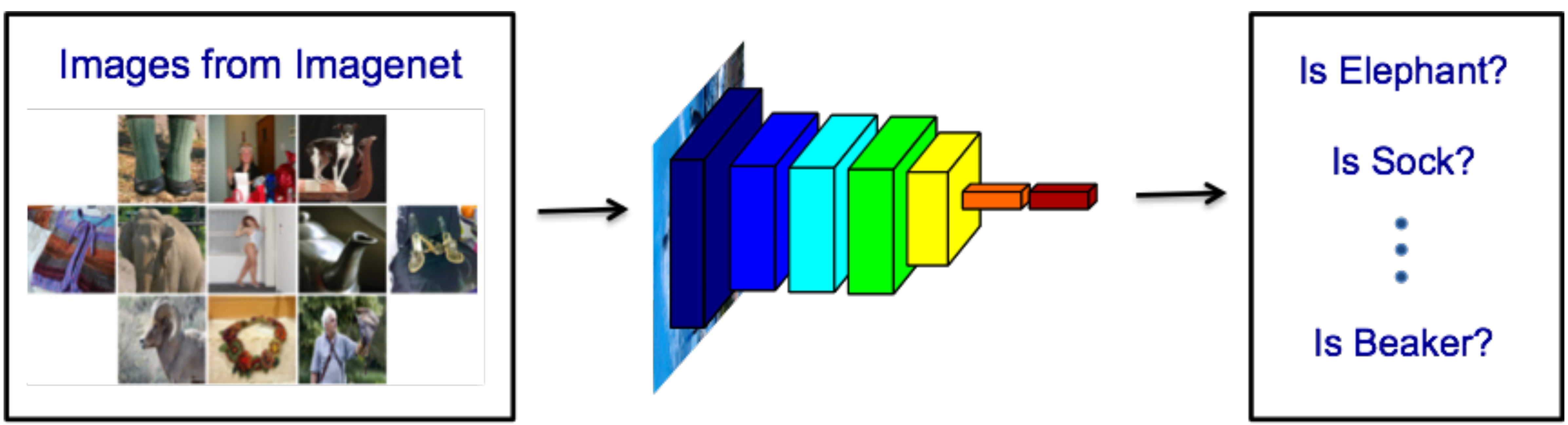
Learn to perform N tasks → Solve the (N+1)th task

faster

or,
more complex task

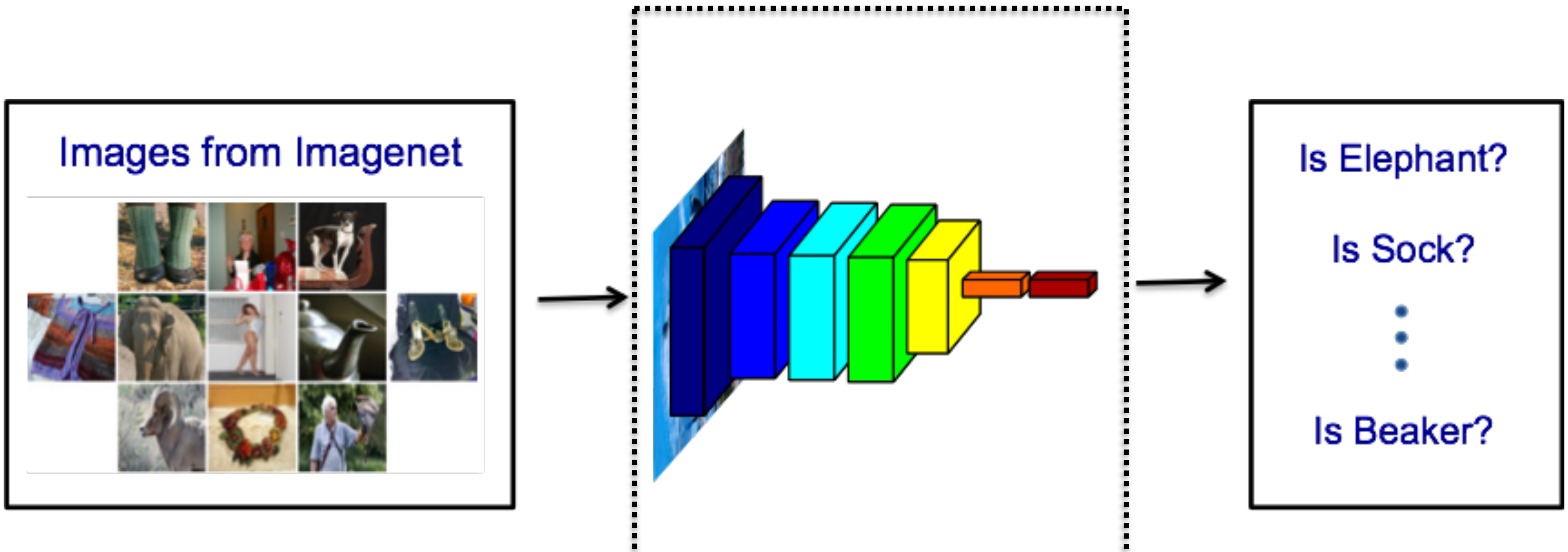
Success on Imagenet



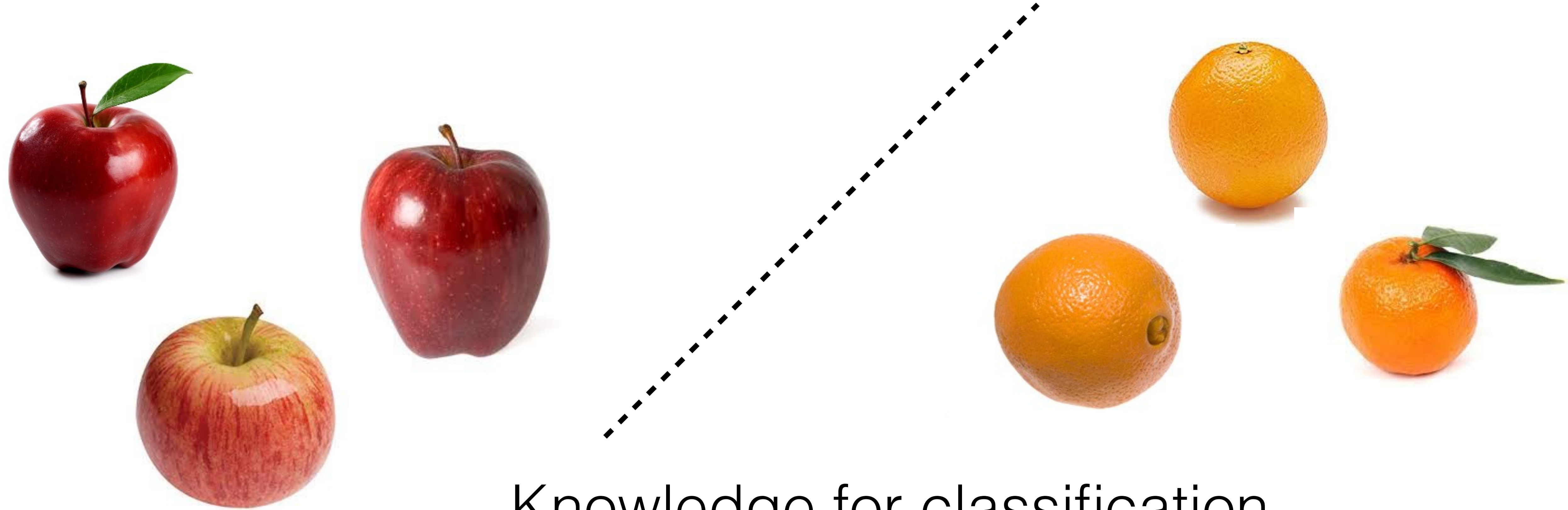


Training on N tasks —> Object classification knowledge

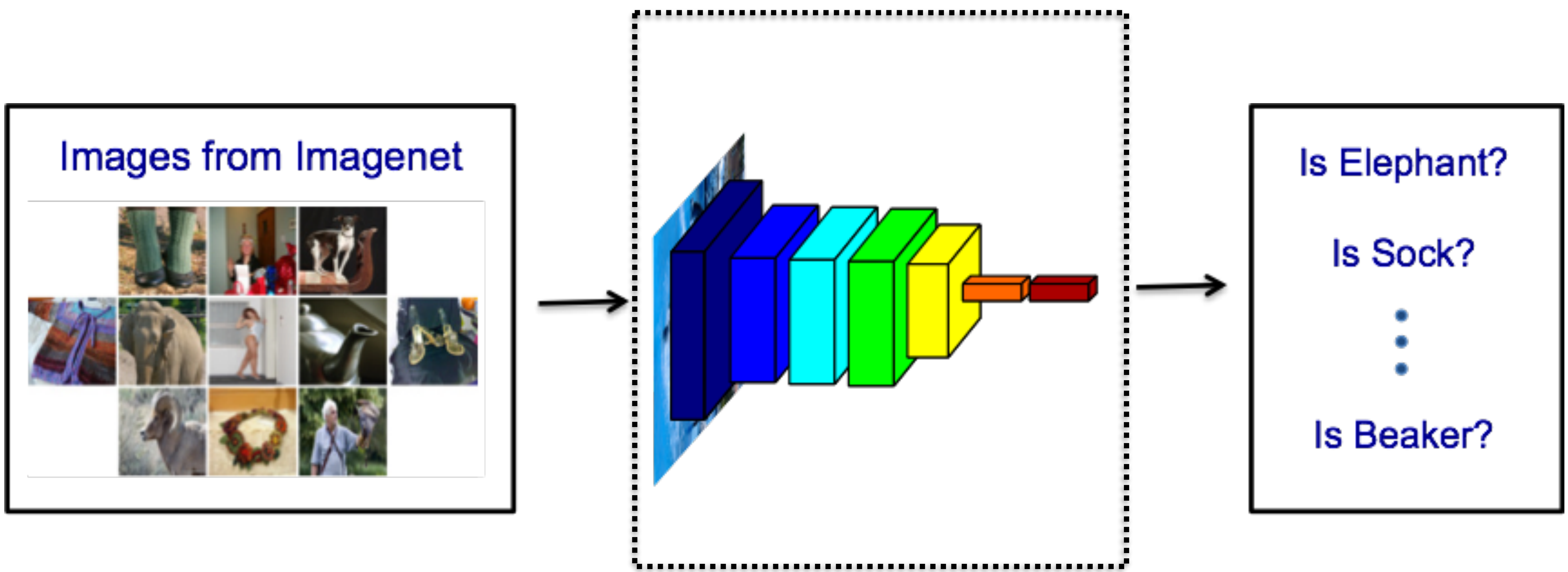
Knowledge for classification



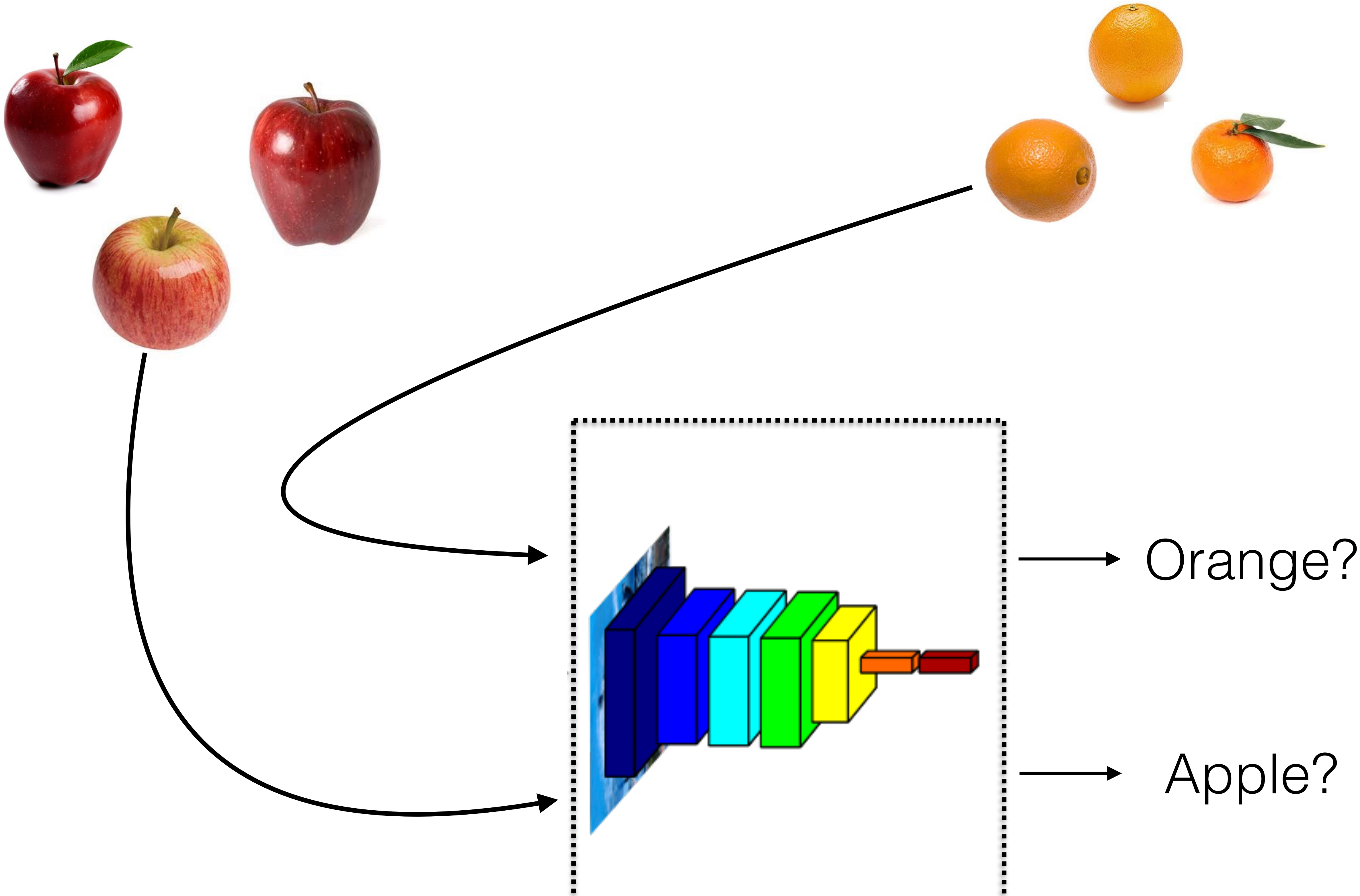
Training on N tasks —> Object classification knowledge



Knowledge for classification

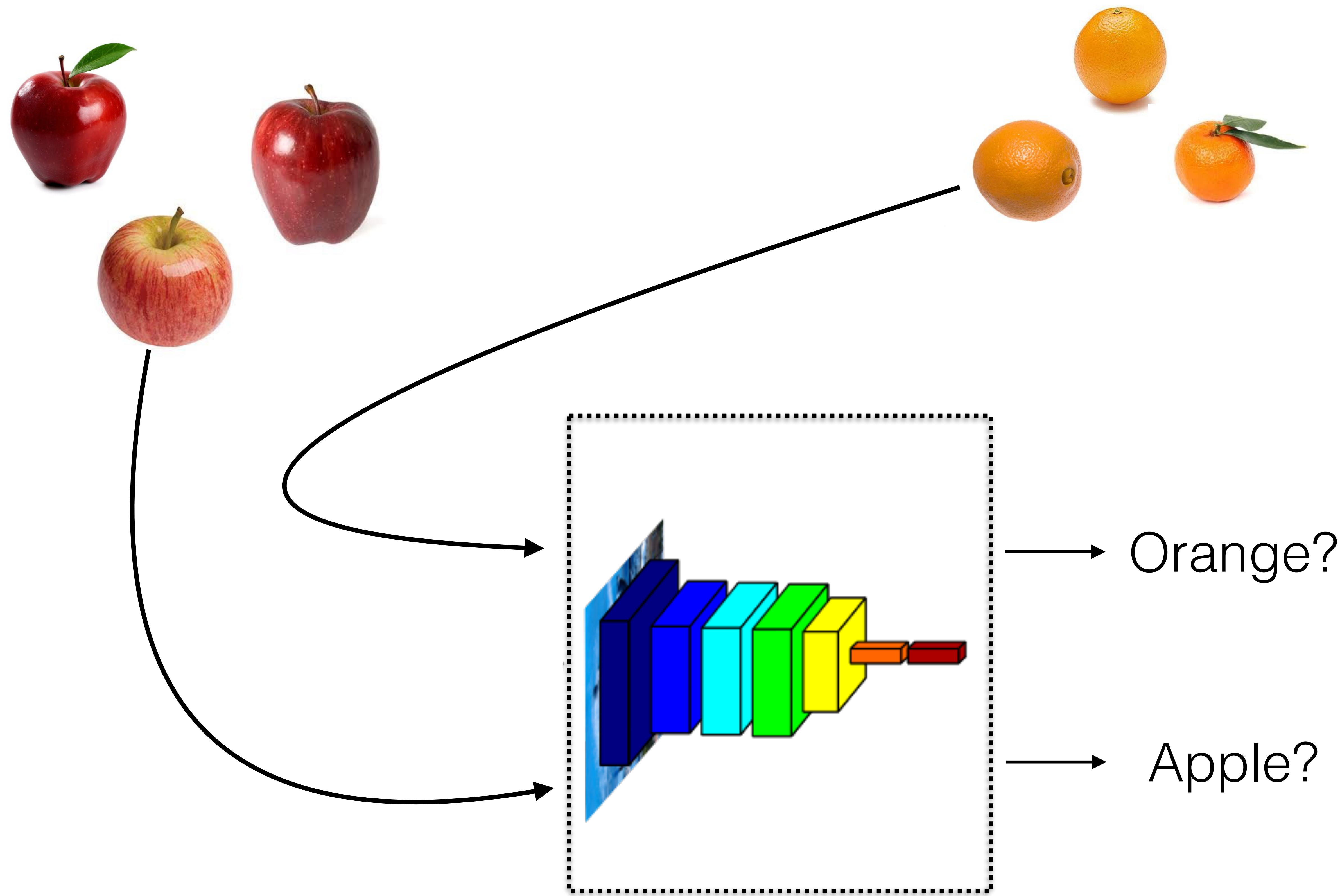


Reuse knowledge by fine-tuning



Imagenet: 1000 examples/class

New task: ~100 examples/class



Still need hundreds of “labelled” data points!

Fine-tuning with very few data points, won’t be effective!

Problem Setup

Training Set

Apple

x_1 y_1

Orange

x_2 y_2

Problem Setup

Training Set

Apple

x_1 y_1

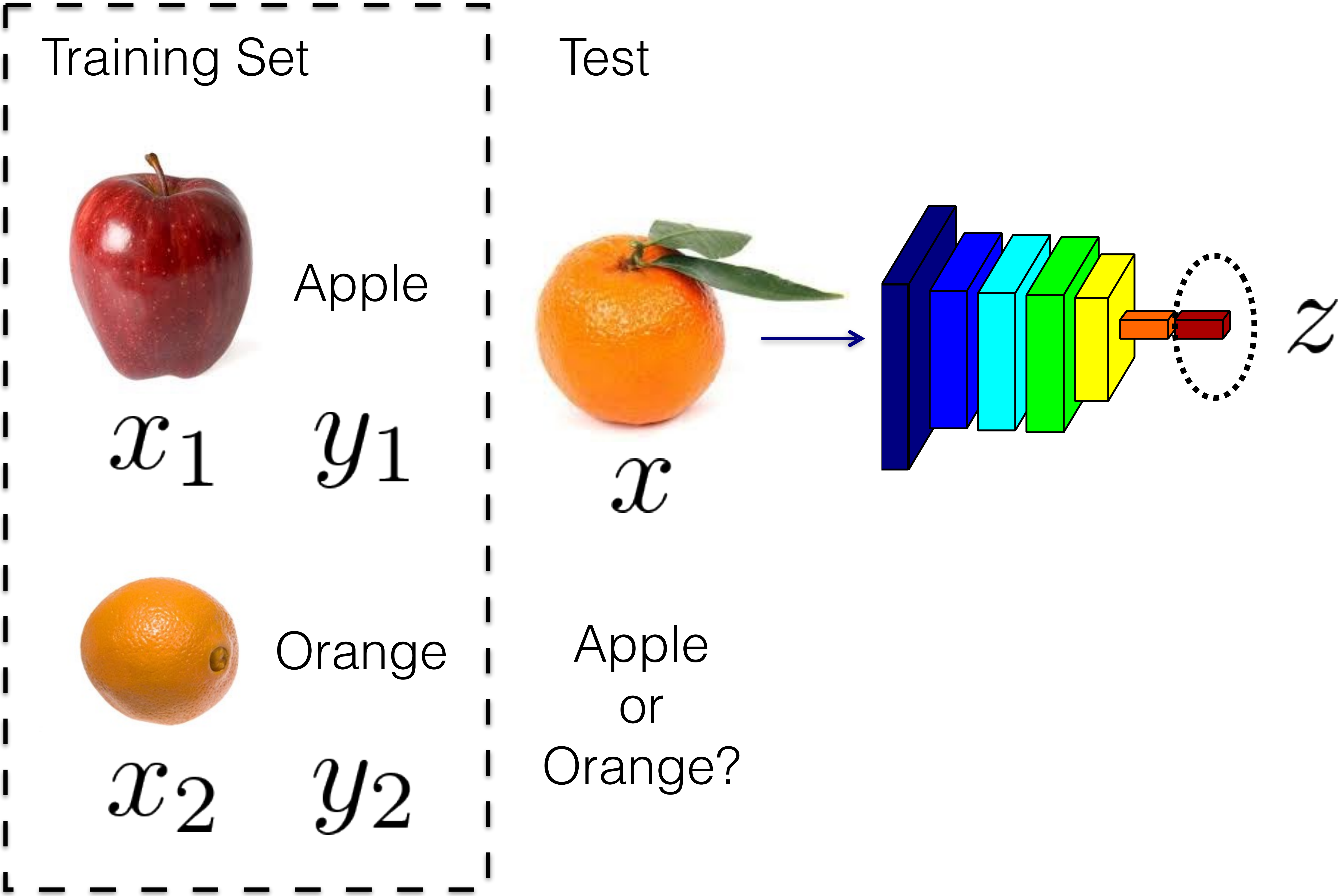
Orange

x_2 y_2

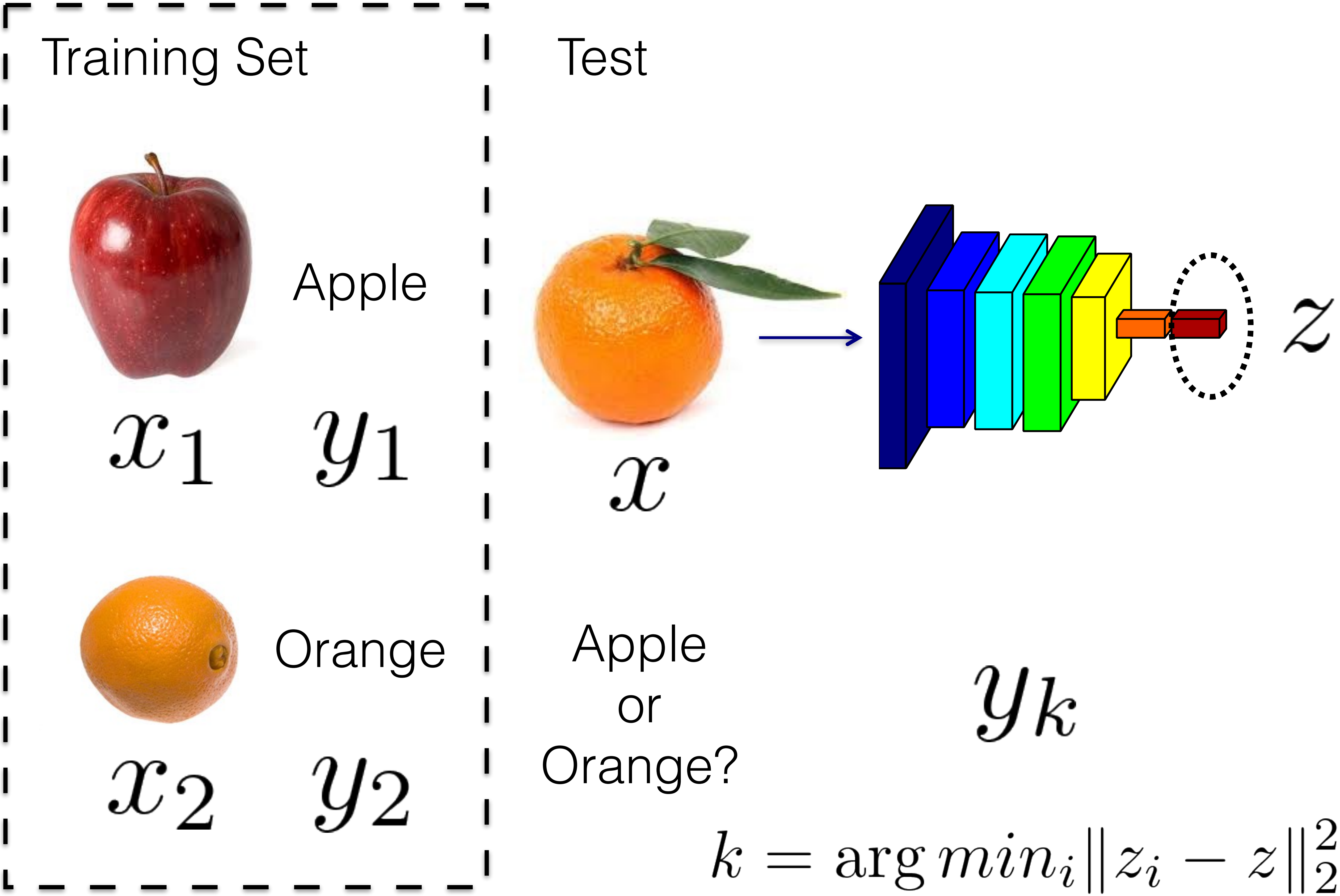
Test

Apple
or
Orange?

Use Nearest Neighbors



Use Nearest Neighbors



What does the performance depend on??

Training Set

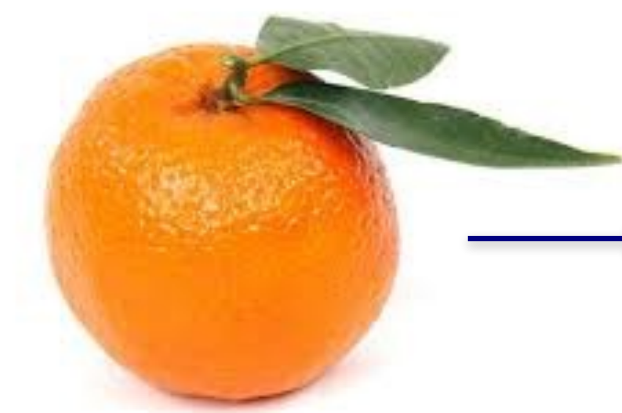
Apple

x_1 y_1

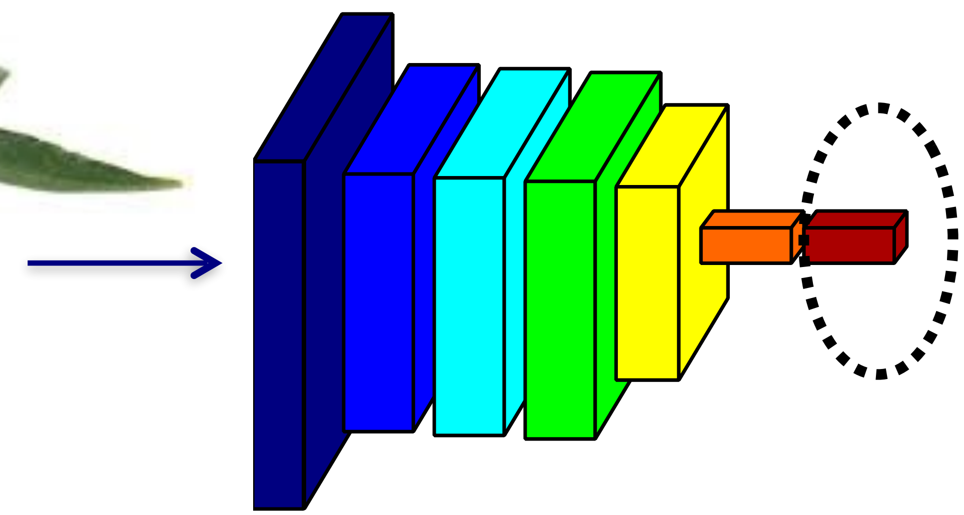
Orange

x_2 y_2

Test



x



z

Apple or Orange?

y_k

$$k = \arg min_i \|z_i - z\|_2^2$$

What does the performance depend on??

Training Set

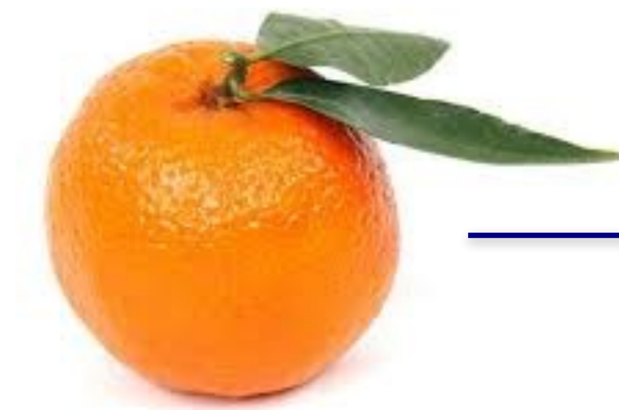
Apple

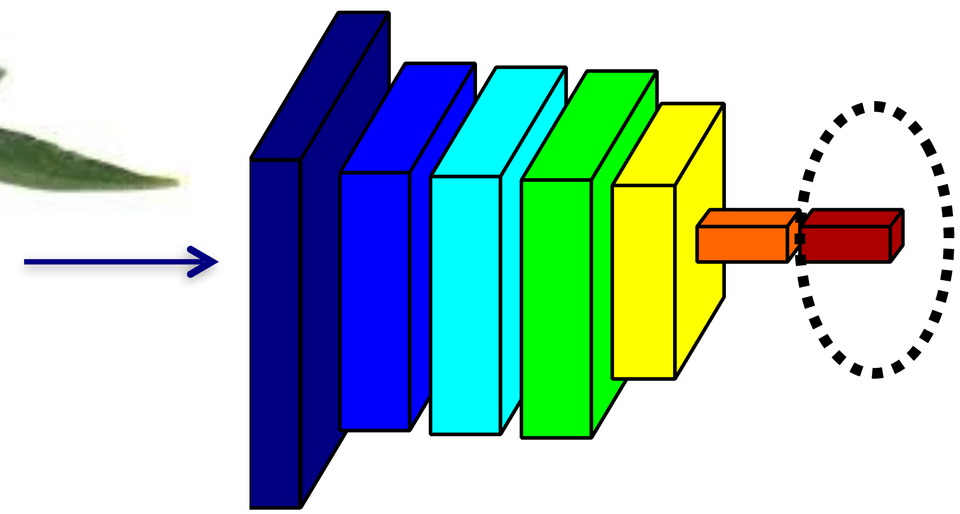
x_1 y_1

Orange

x_2 y_2

Features might not be optimized for matching!





z

x

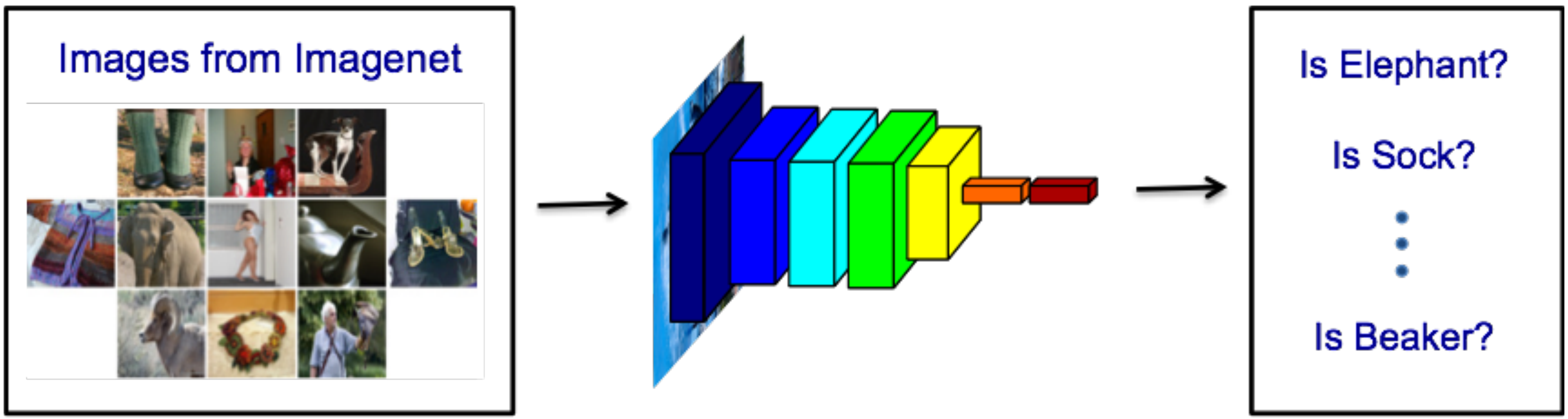
Apple or Orange?

y_k

$$k = \arg min_i \|z_i - z\|_2^2$$

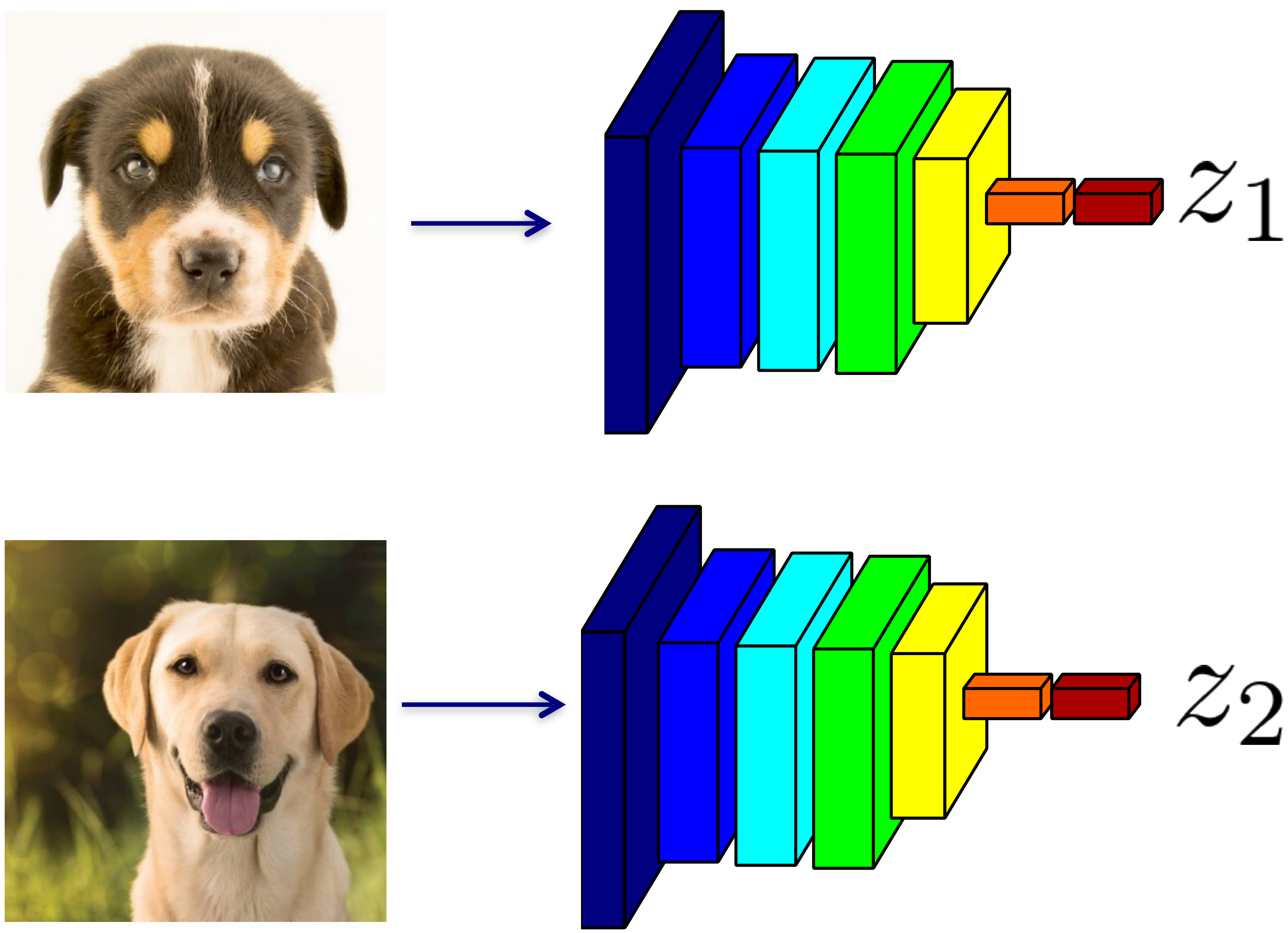
Metric Learning via Siamese Networks*

Instead of one v/s all classification



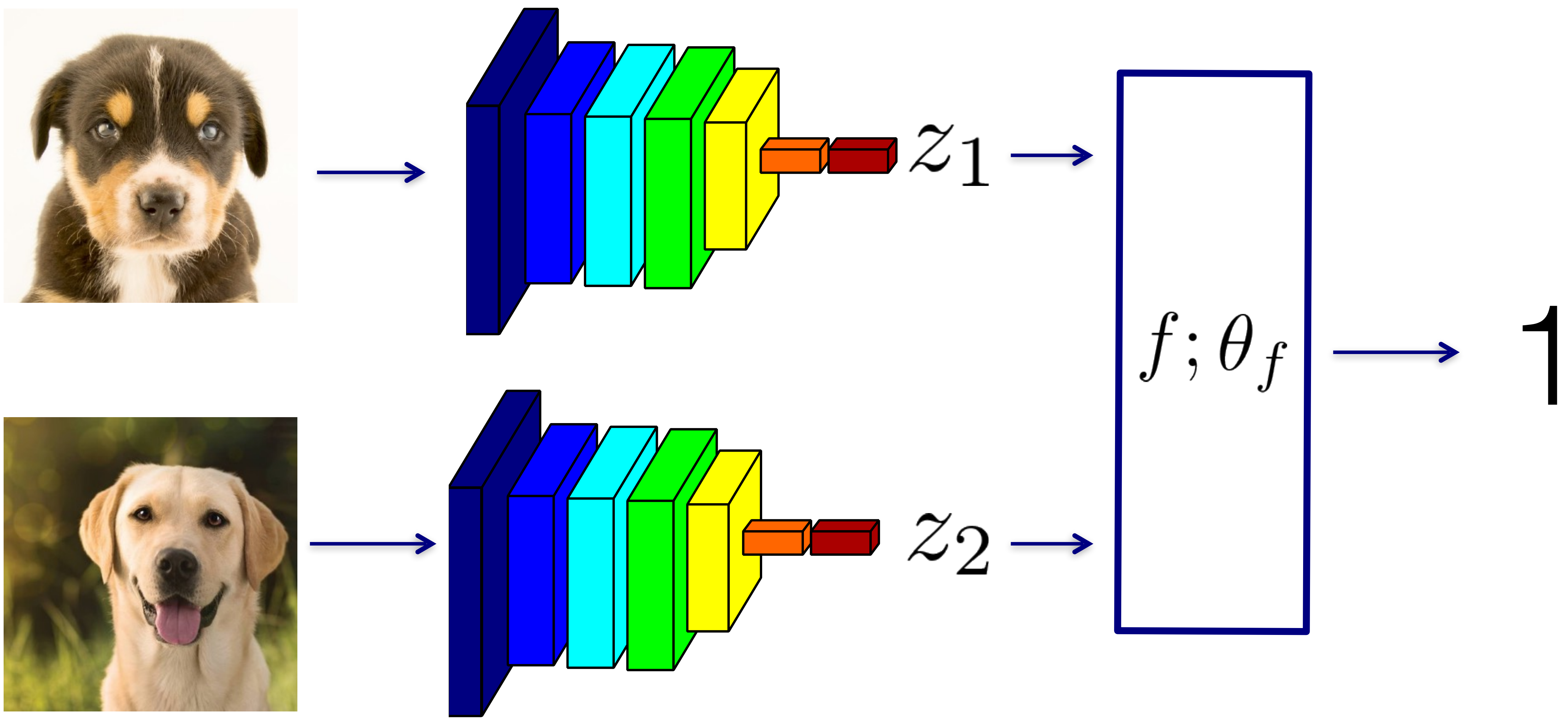
(*Hadsell et. al. 2006)

Metric Learning via Siamese Networks*



(*Hadsell et. al. 2006)

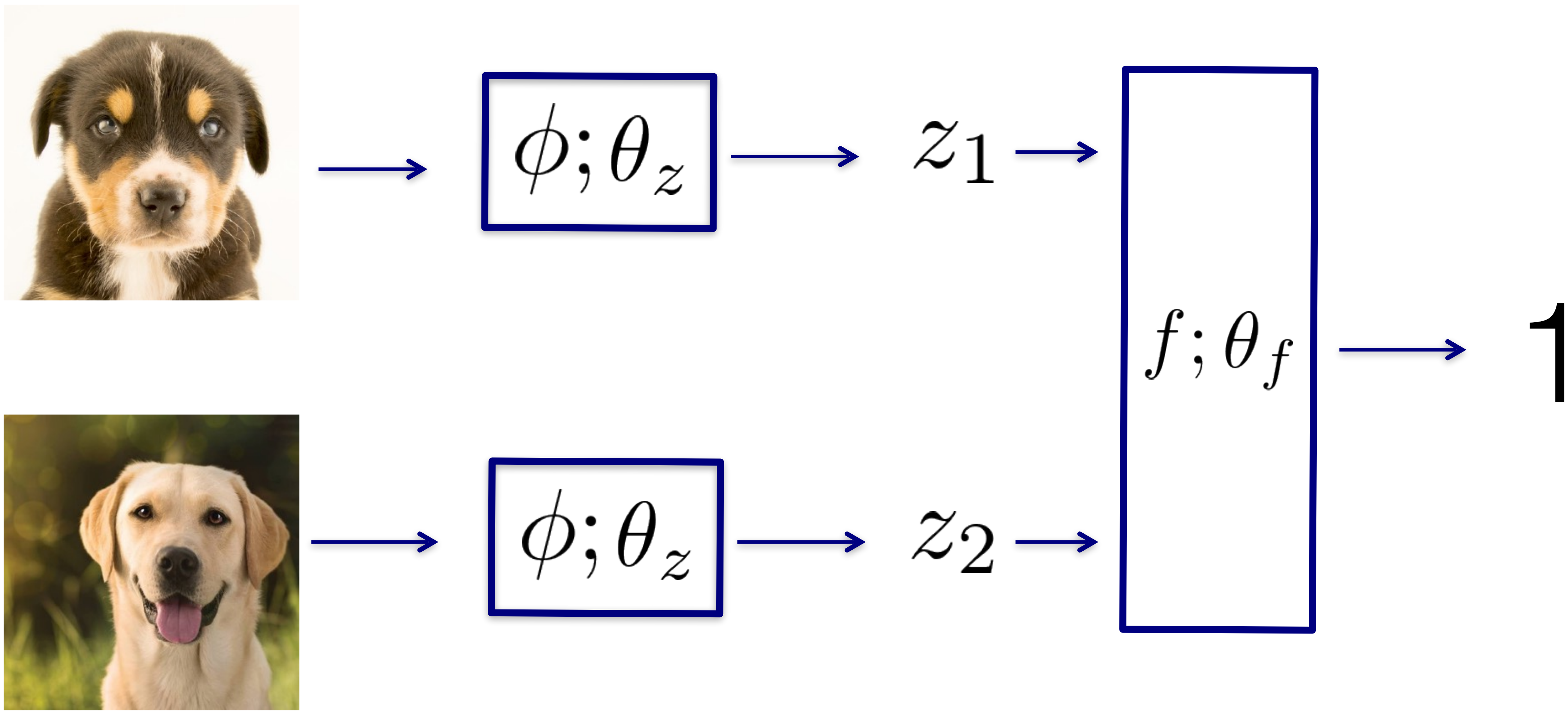
Metric Learning via Siamese Networks*



Same class: Output = 1

(*Hadsell et. al. 2006)

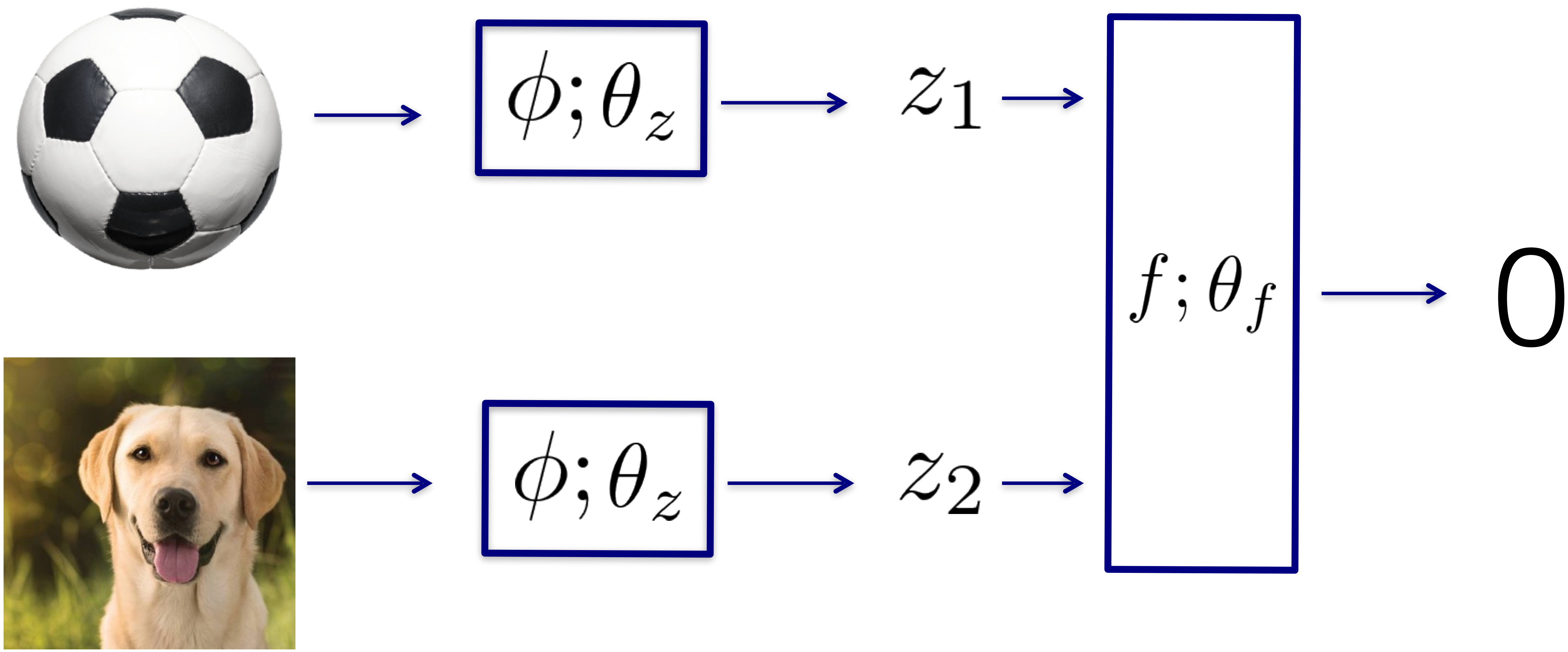
Metric Learning via Siamese Networks*



Same class: Output = 1

(*Hadsell et. al. 2006)

Metric Learning via Siamese Networks*

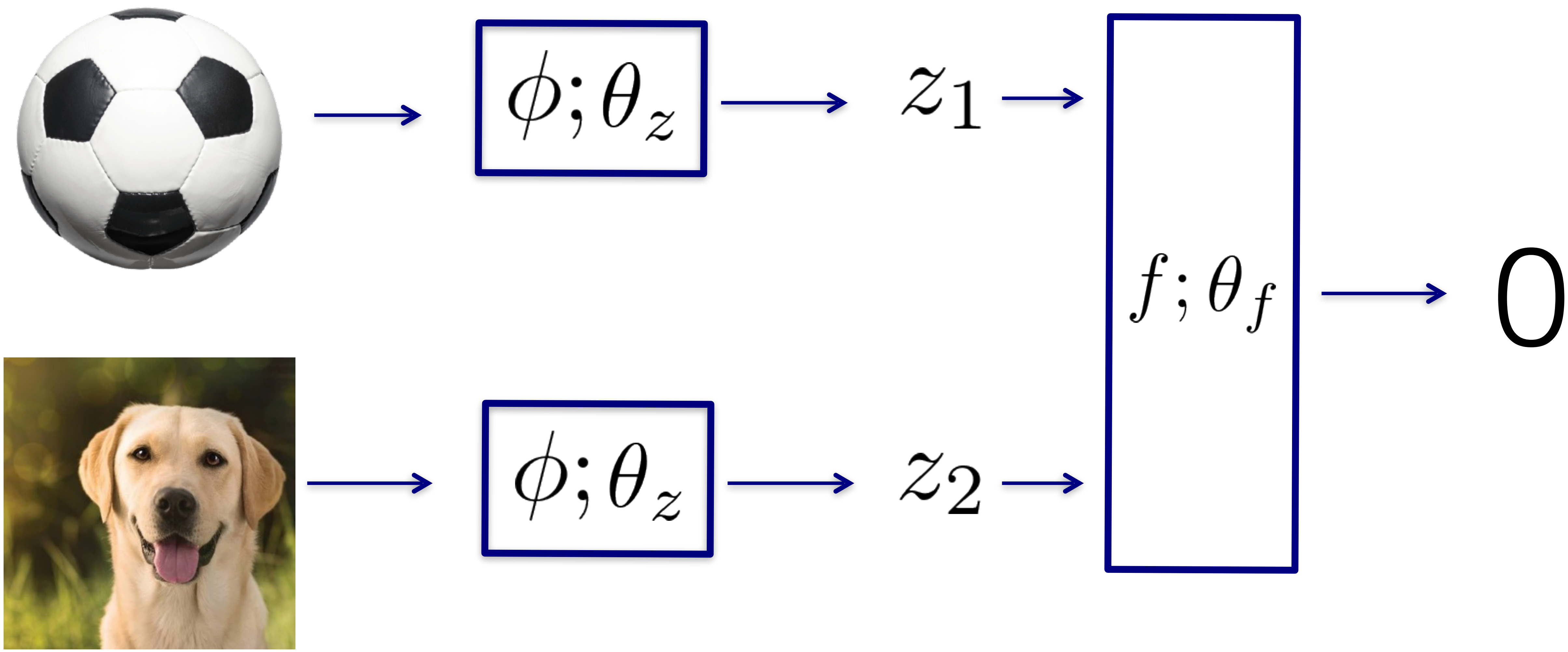


Same class: Output = 1

Different class: Output = 0

(*Hadsell et. al. 2006)

Metric Learning via Siamese Networks*



$$\min_{\theta_z,\theta_f}$$

Same class: Output = 1

Different class: Output = 0

(*Hadsell et. al. 2006)

Solving using Siamese Network

Training Set

Apple

x_1 y_1

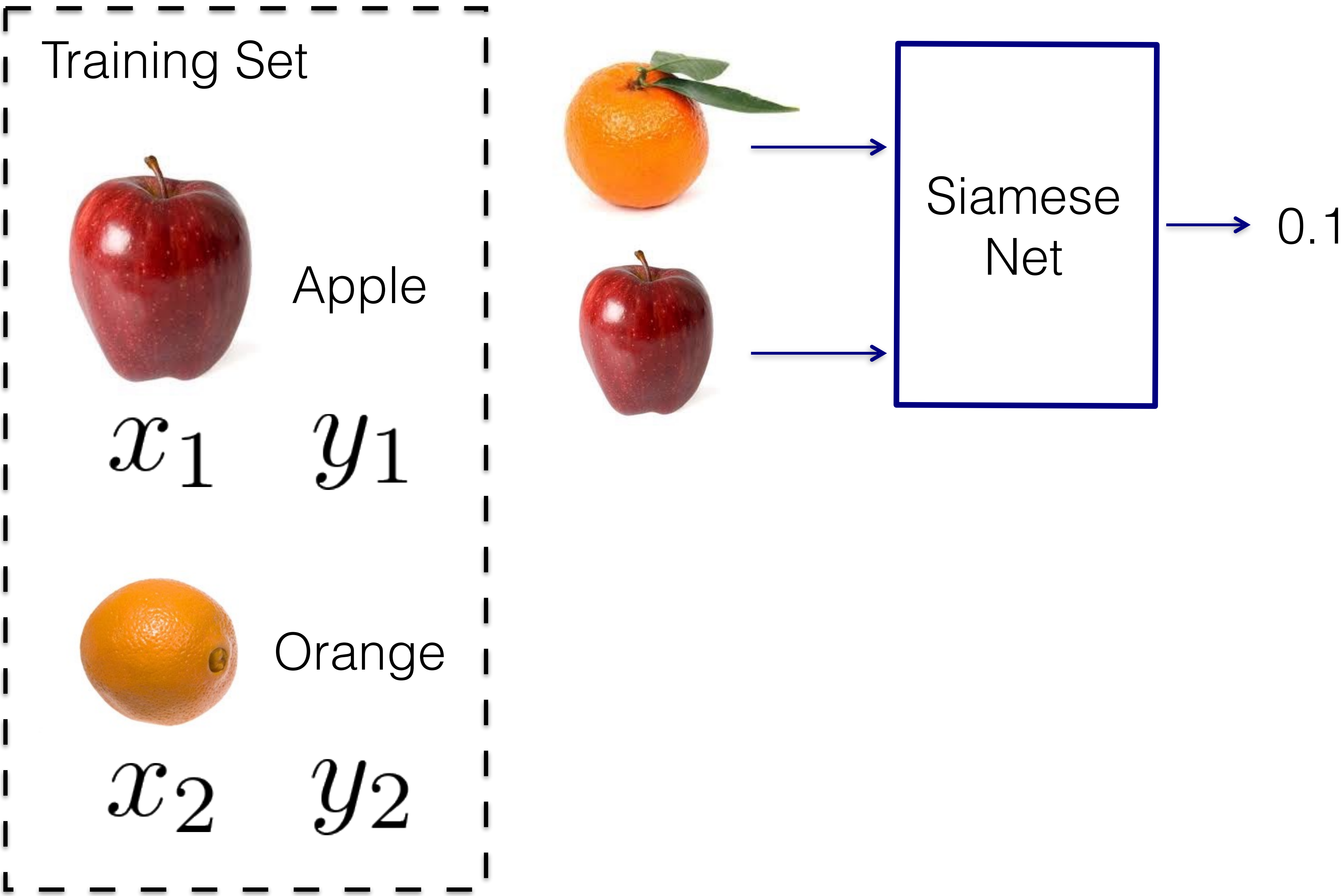
Orange

x_2 y_2

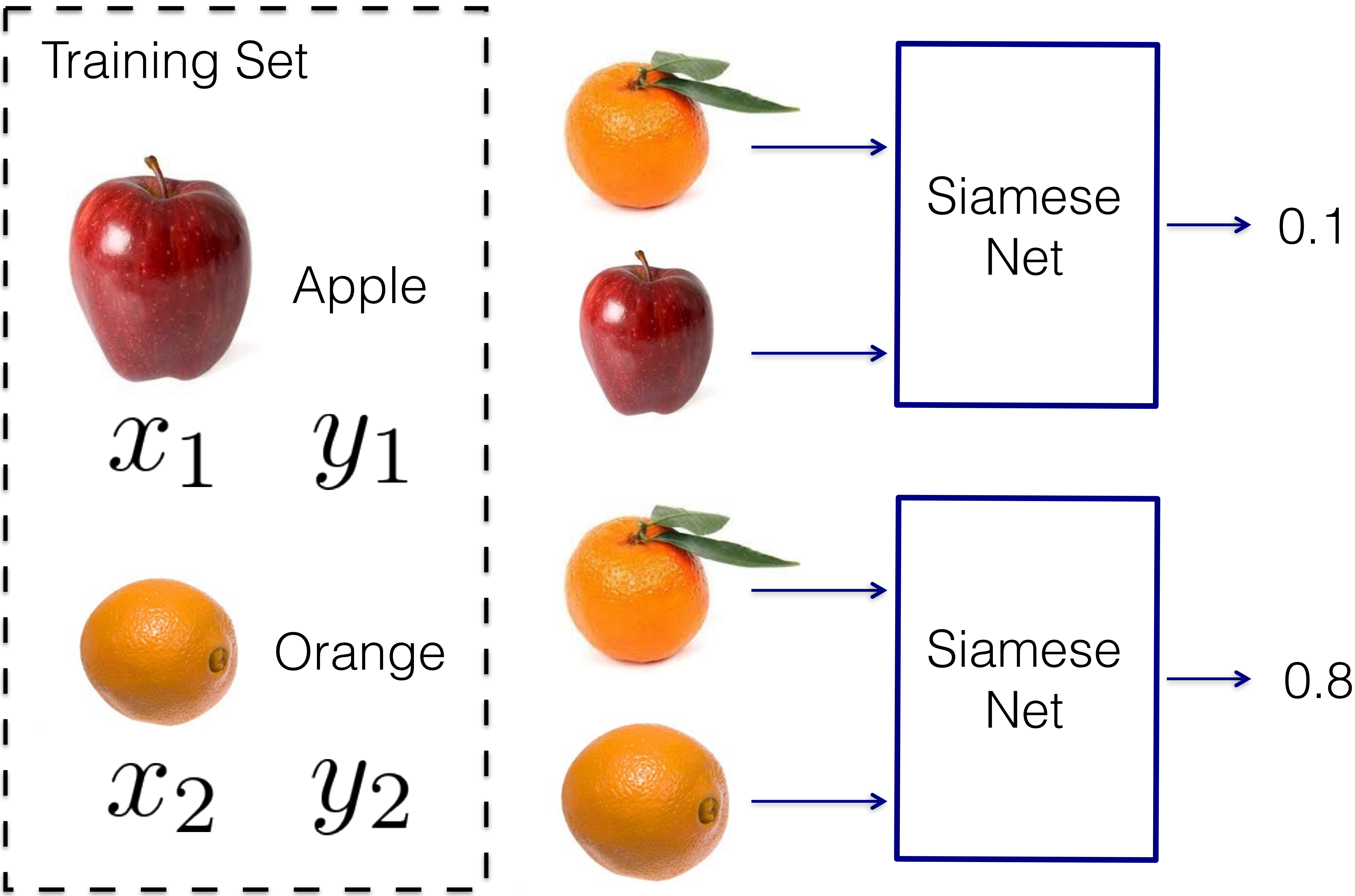
Test

Apple
or
Orange?

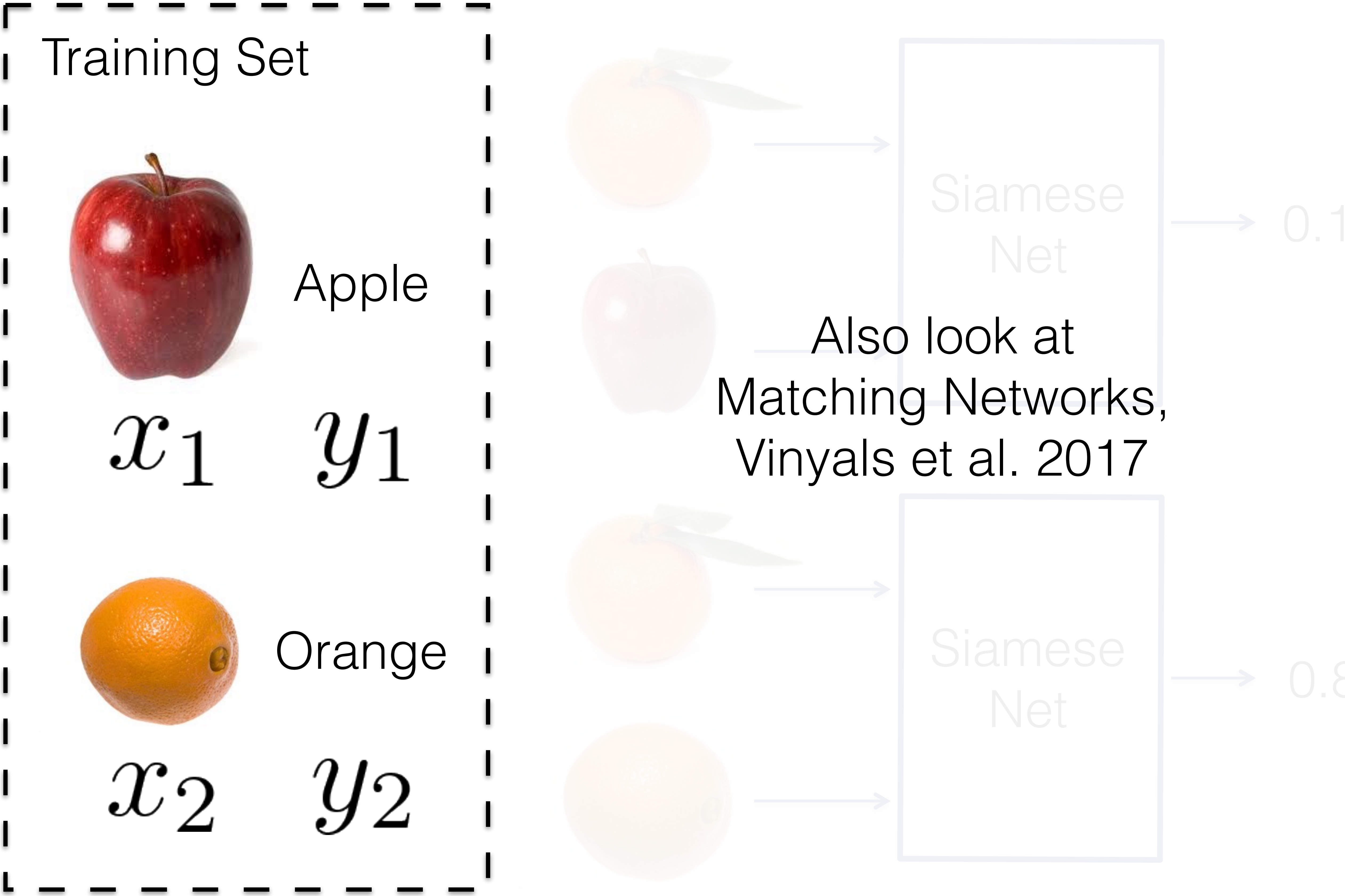
Solving using Siamese Network



Solving using Siamese Network

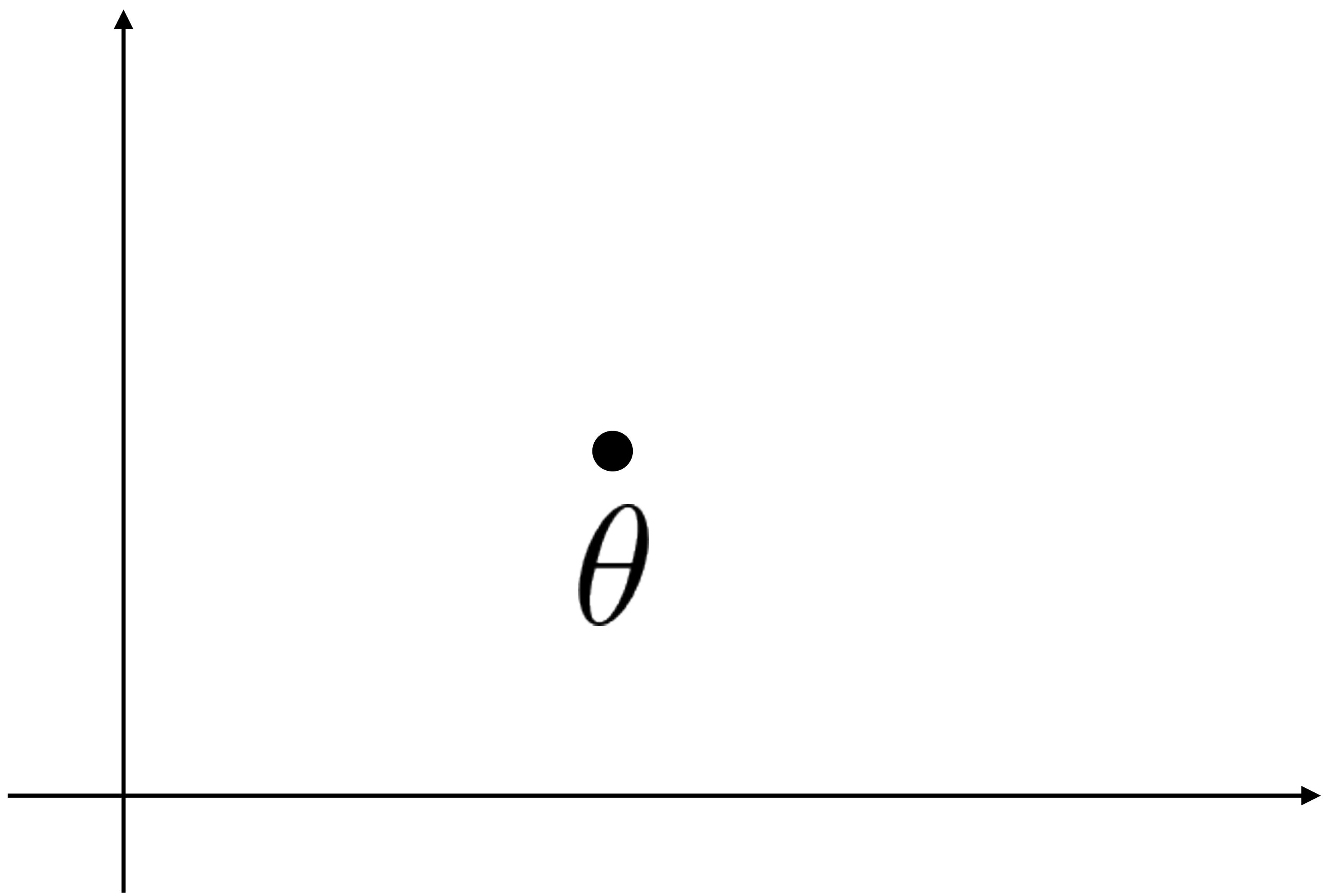


Solving using Siamese Network



Also look at Matching Networks, Vinyals et al. 2017

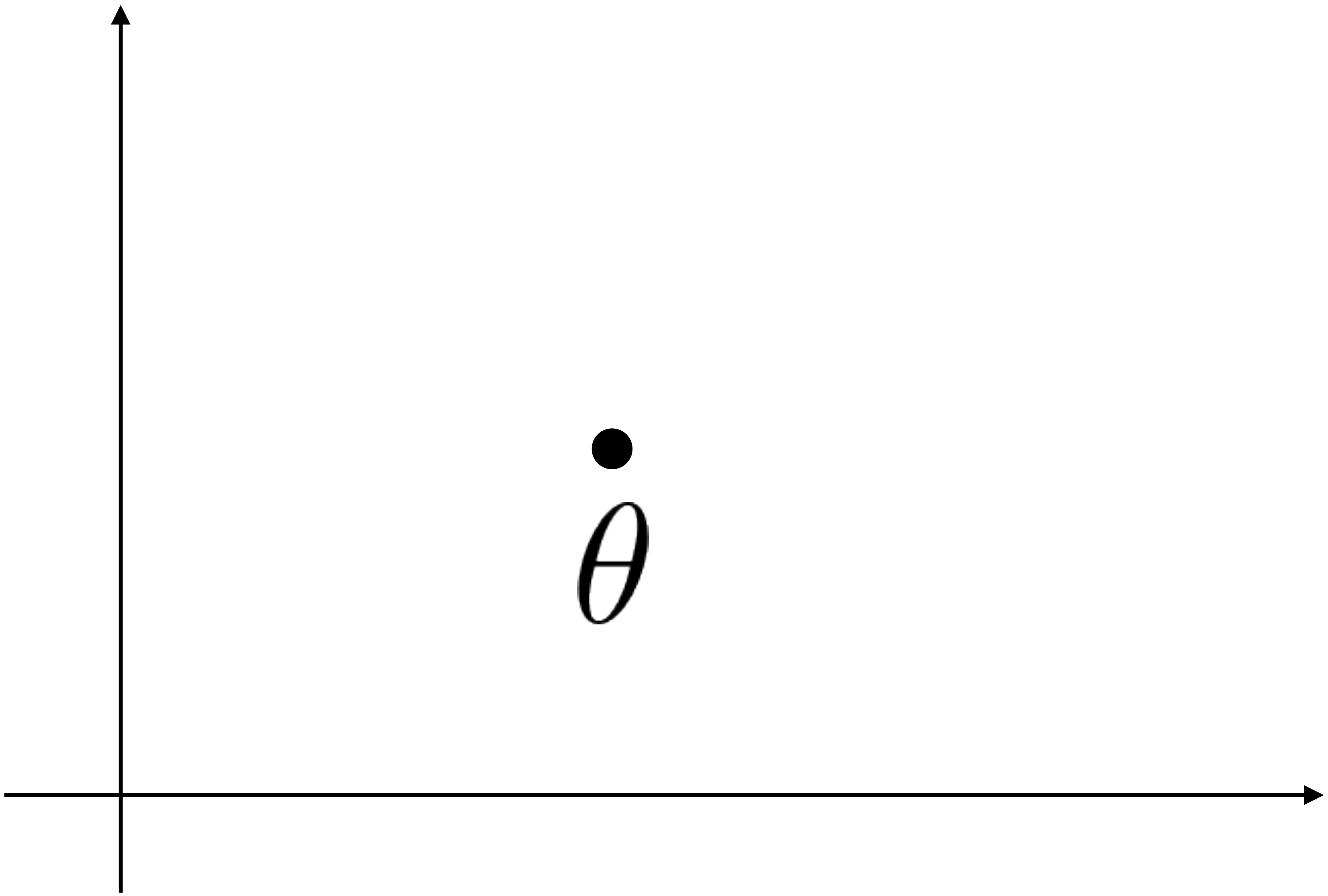
Another perspective



θ: parameters after training on say Imagenet

Another perspective

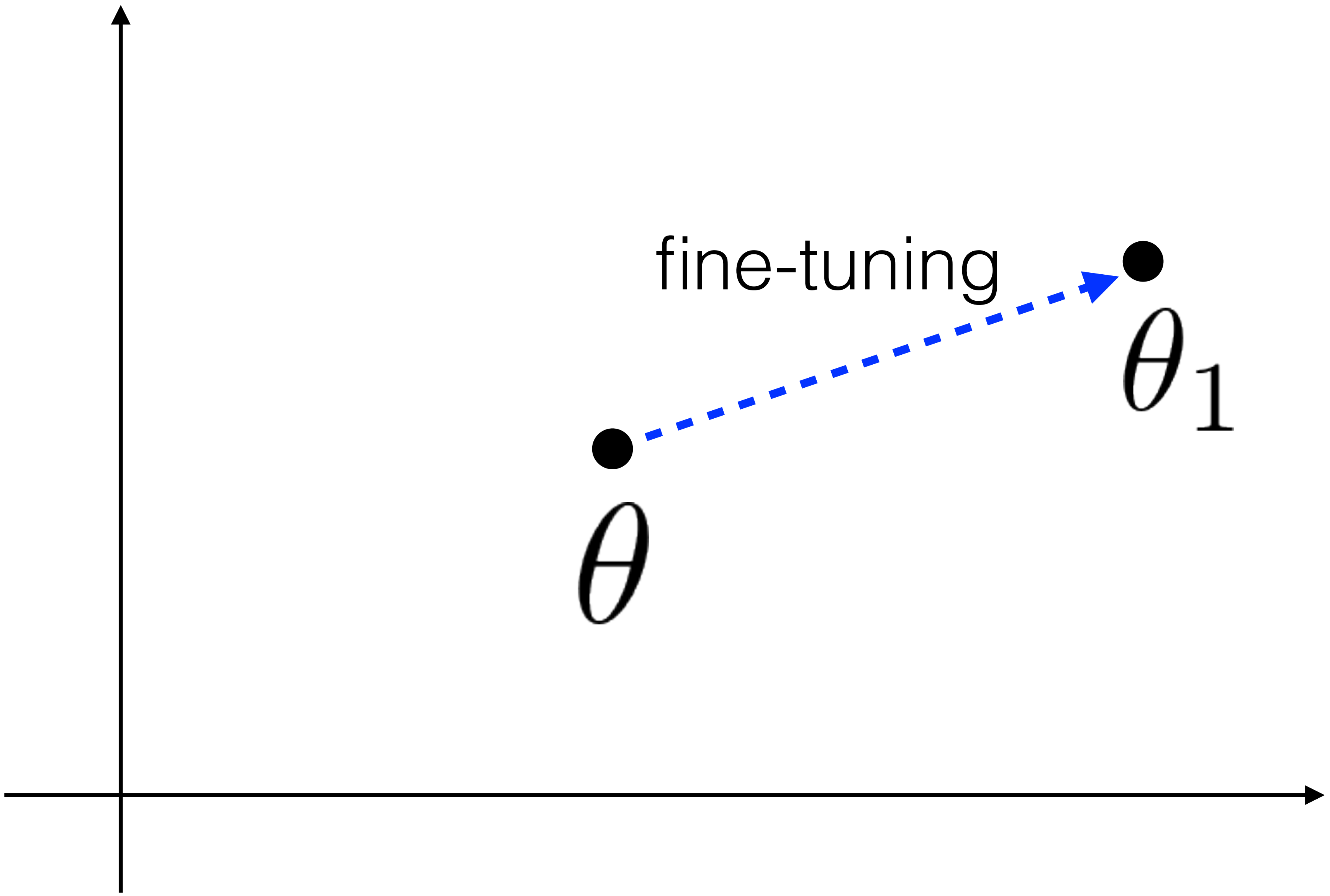
Task1: Apple v/s Orange



θ: parameters after training on say Imagenet

Another perspective

Task1: Apple v/s Orange

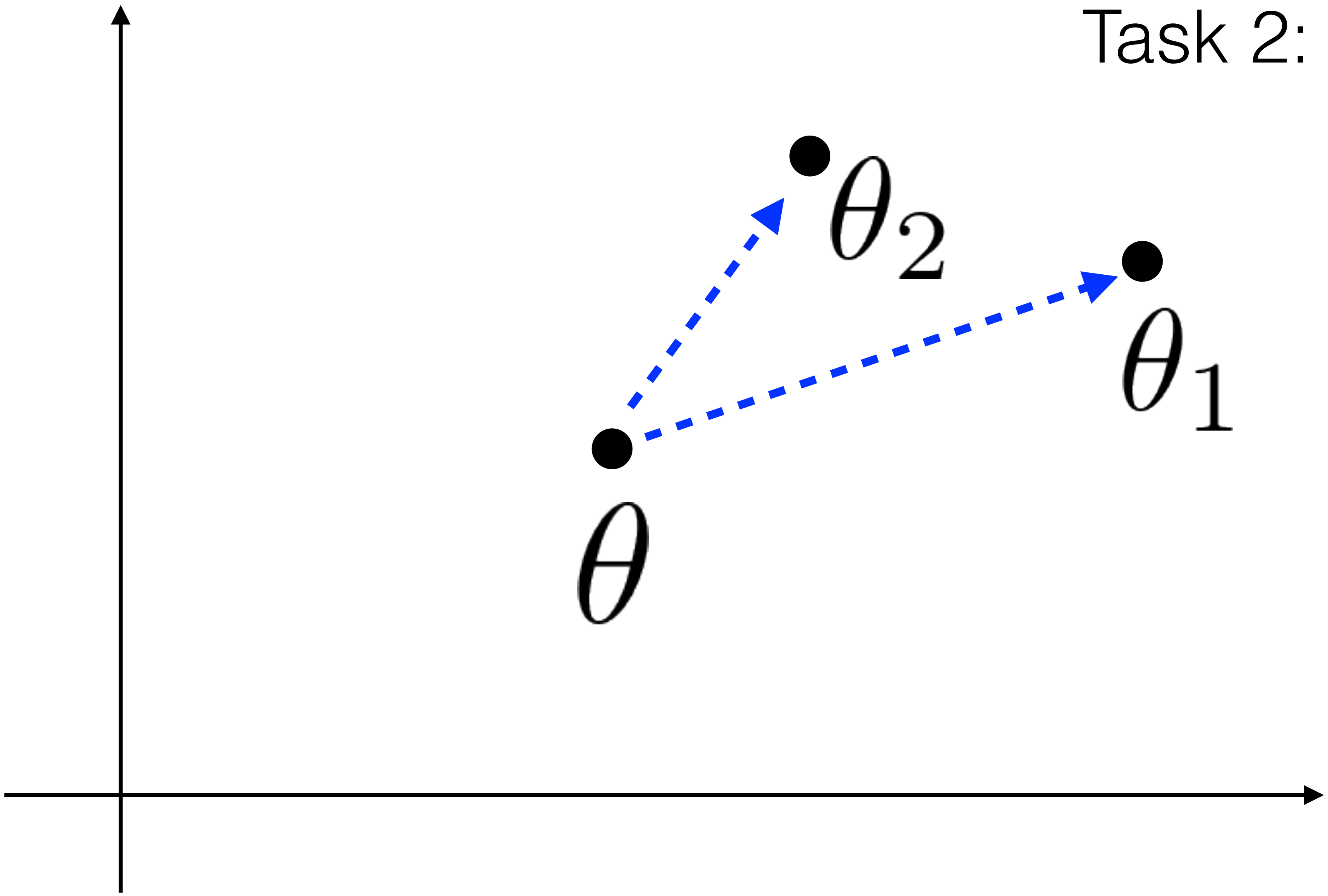


θ: parameters after training on say Imagenet

Another perspective

Task1: Apple v/s Orange

Task 2: Dog v/s Cat

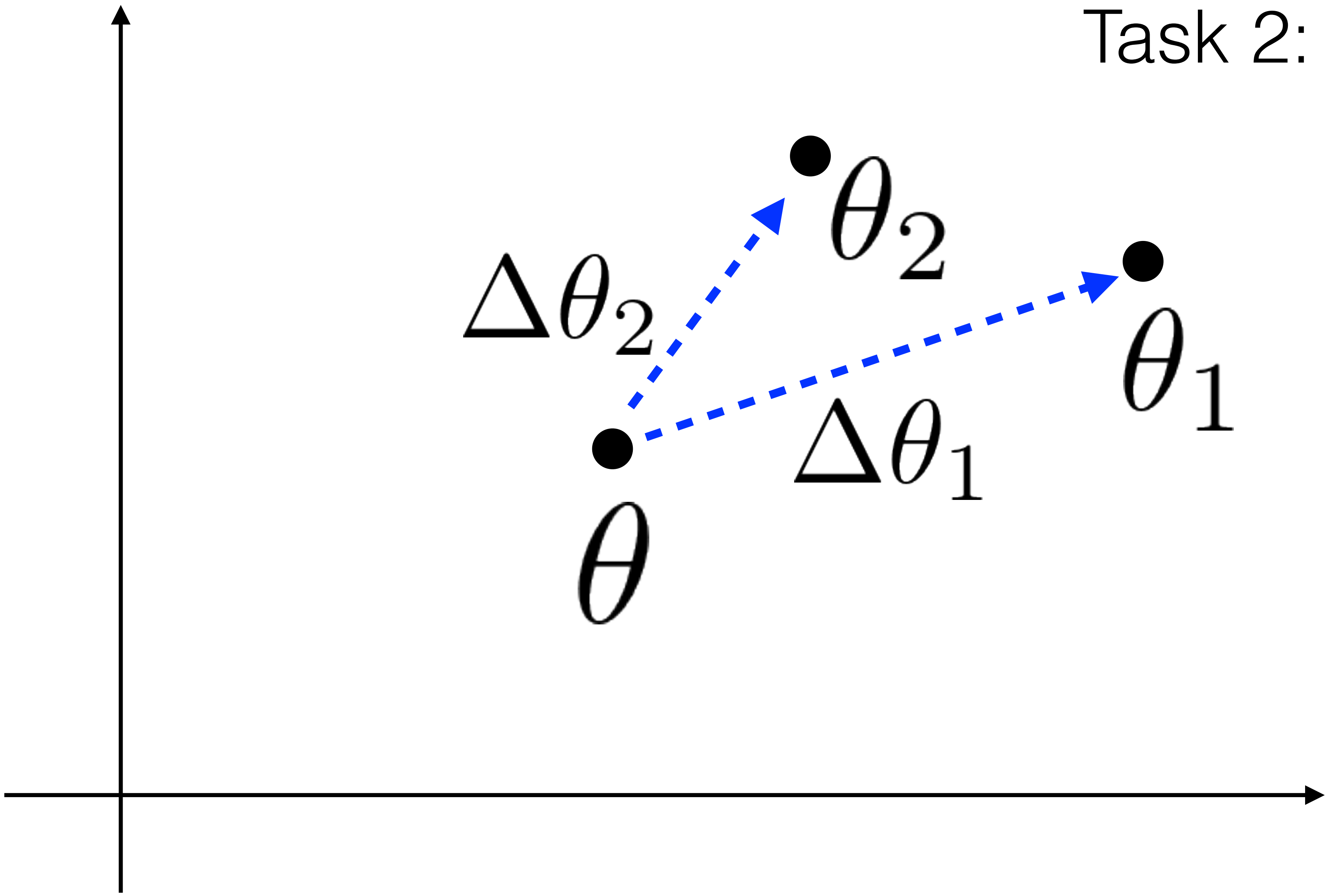


θ: parameters after training on say Imagenet

Another perspective

Task1: Apple v/s Orange

Task 2: Dog v/s Cat

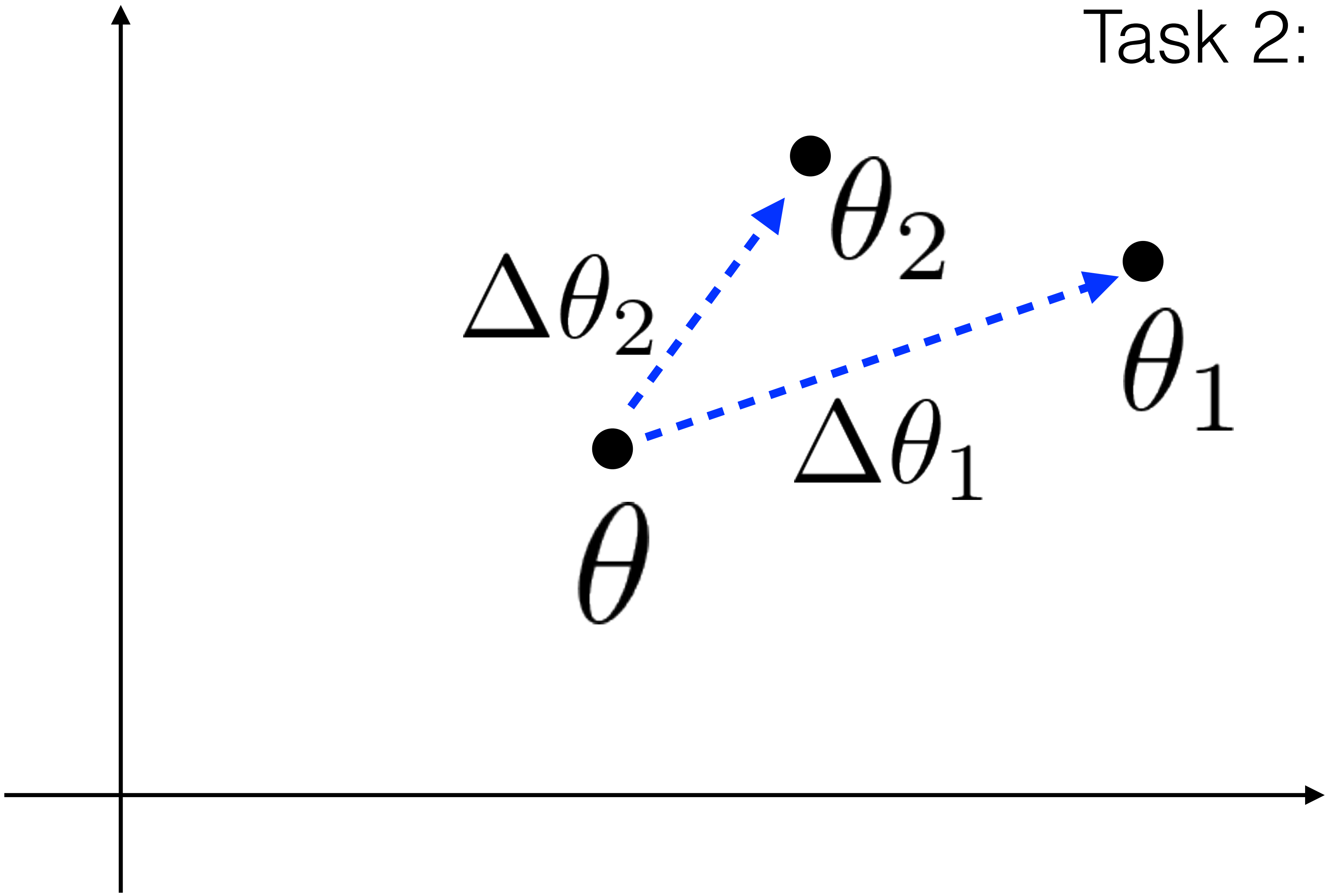


θ: parameters after training on say Imagenet

Another perspective

Task1: Apple v/s Orange

Task 2: Dog v/s Cat

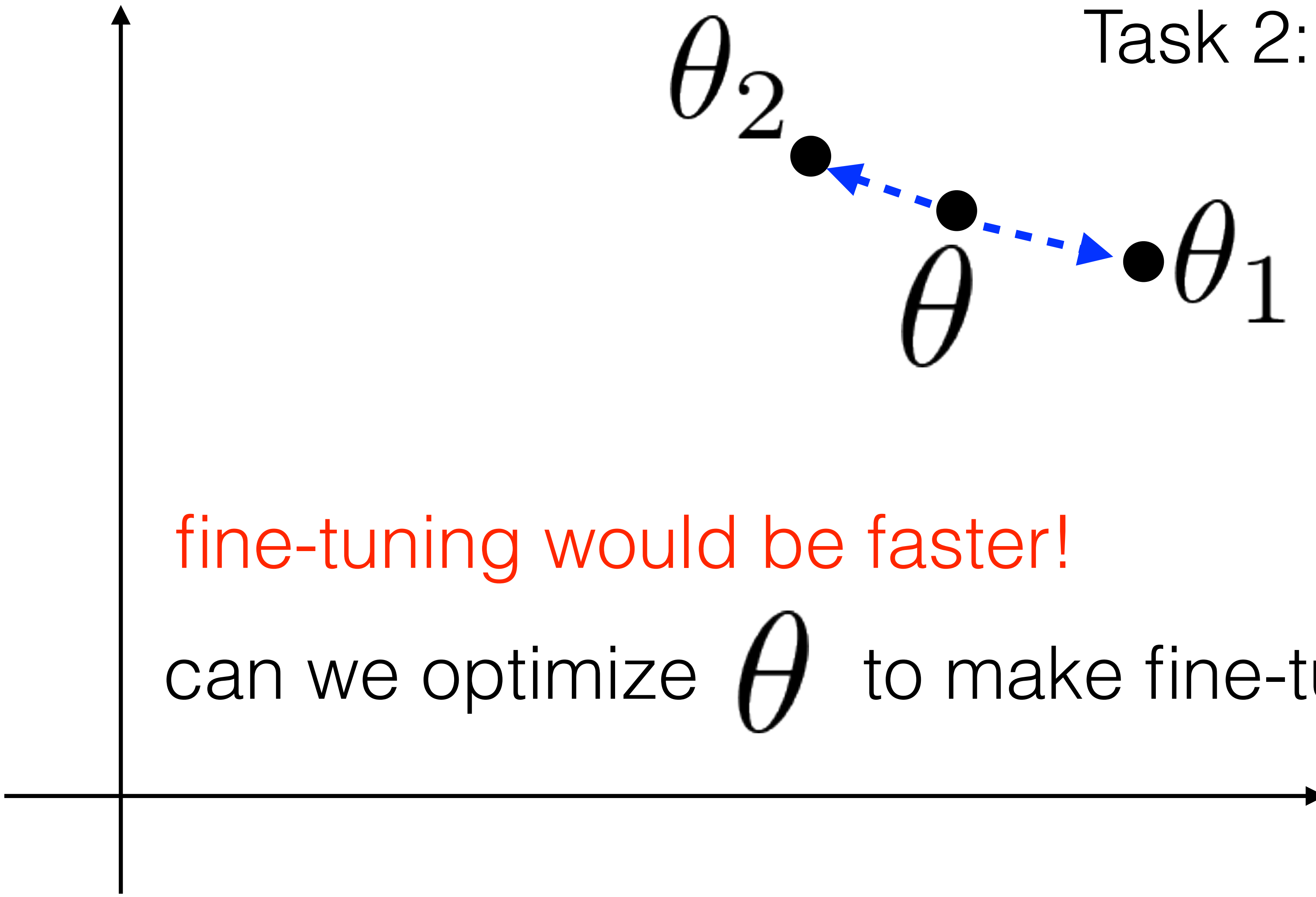


Amount of fine-tuning: $\approx (\Delta\theta_1 + \Delta\theta_2)$

What if?

Task1: Apple v/s Orange

Task 2: Dog v/s Cat



fine-tuning would be faster!

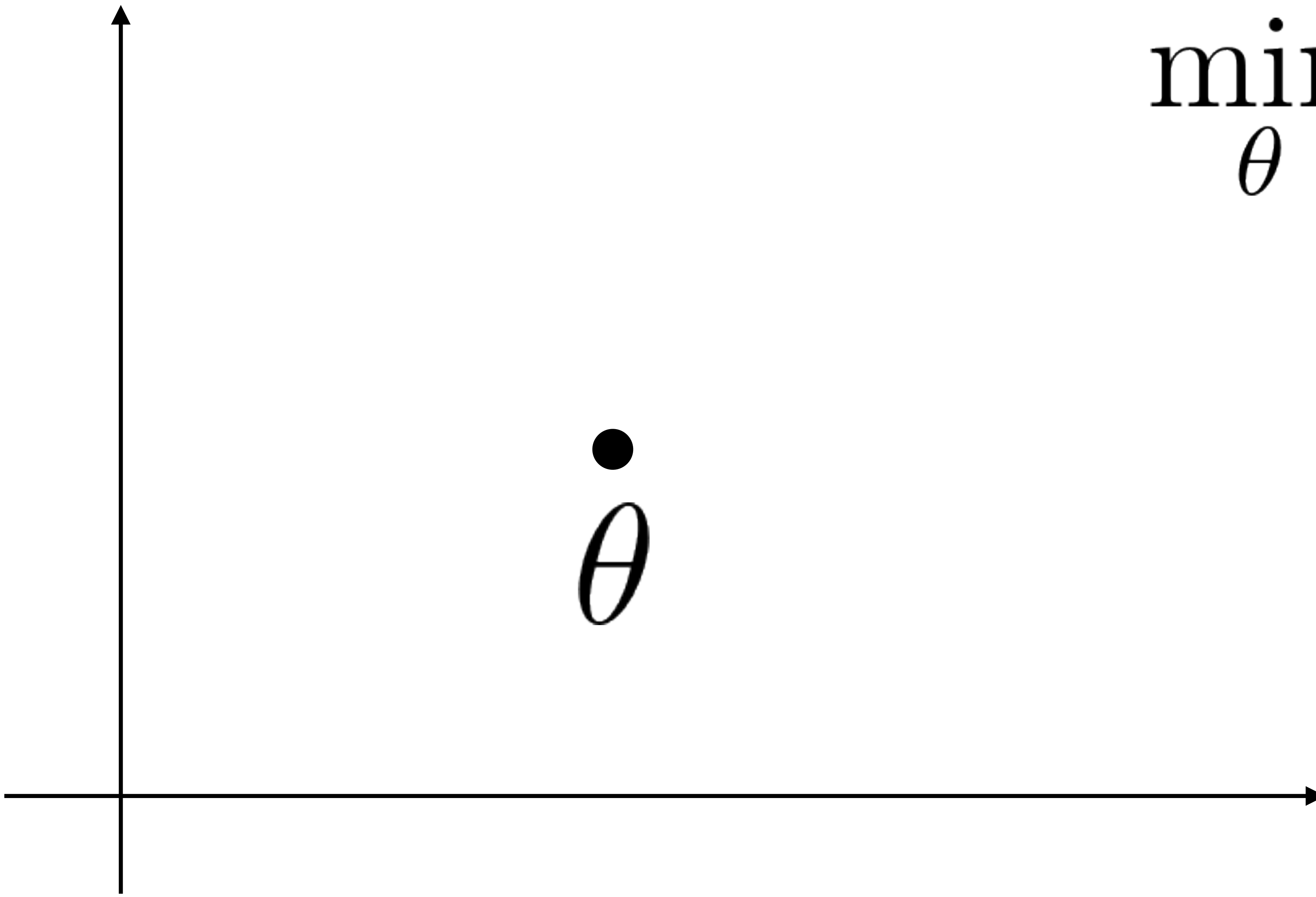
can we optimize θ to make fine-tuning easier?

Amount of fine-tuning: $\approx (\Delta\theta_1 + \Delta\theta_2)$

How to do it?

Task1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_\theta)$$

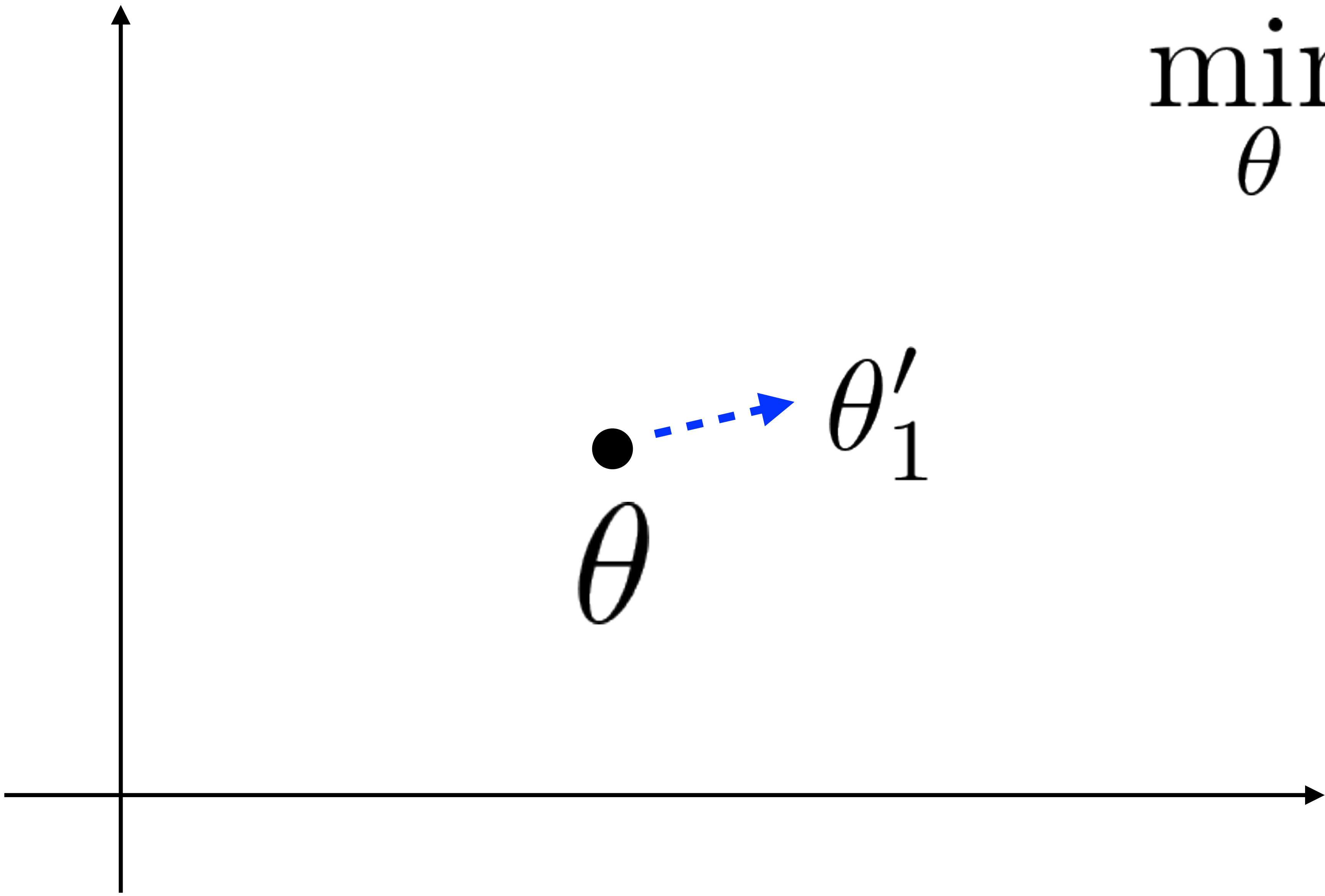


Hariharan et al. 2016, Finn et al. 2017

How to do it?

Task1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_\theta)$$



$$\theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_\theta)$$

Hariharan et al. 2016, Finn et al. 2017

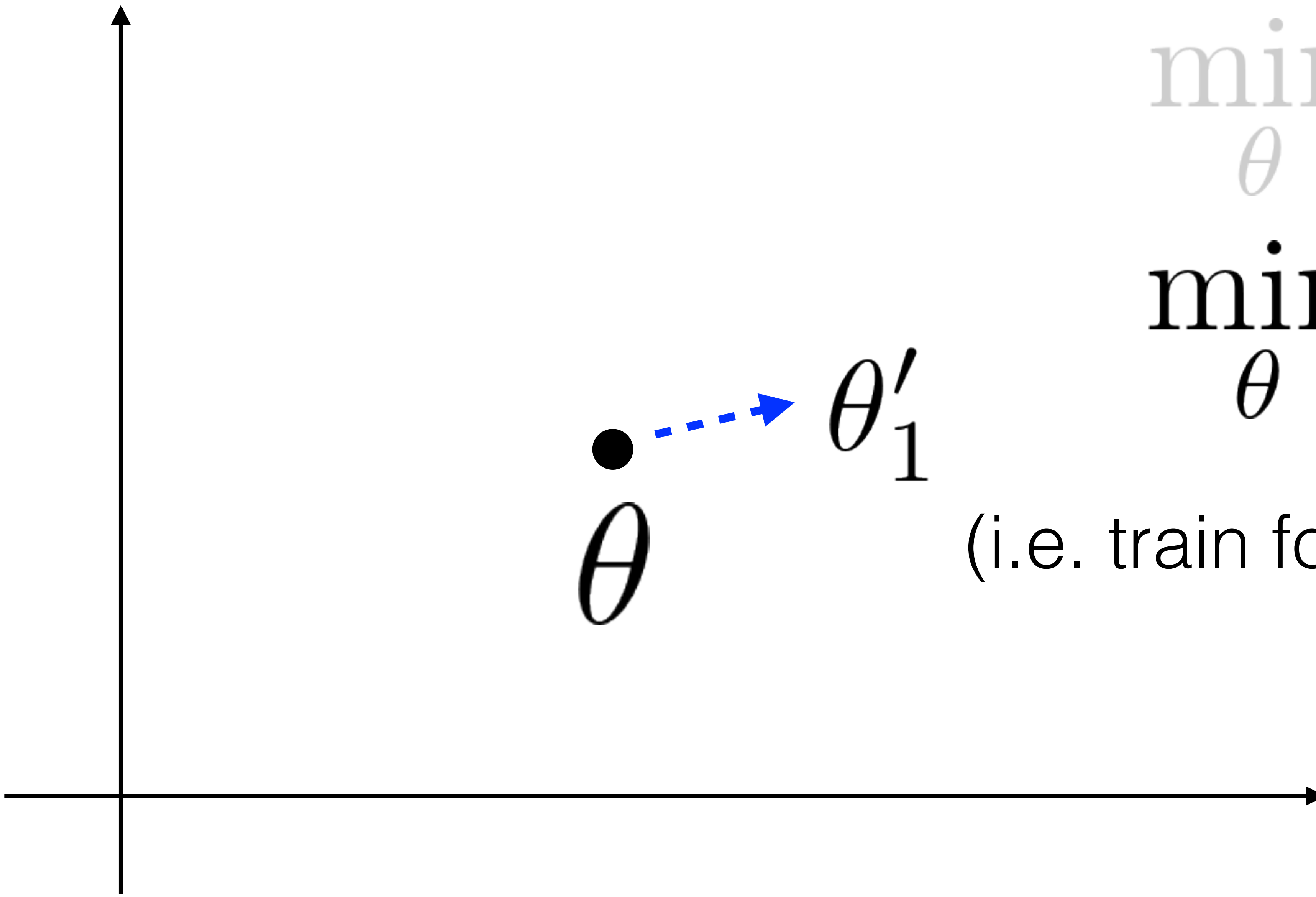
How to do it?

Task1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_\theta)$$

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta'_1})$$

(i.e. train for fast fine-tuning!)



$$\theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_\theta)$$

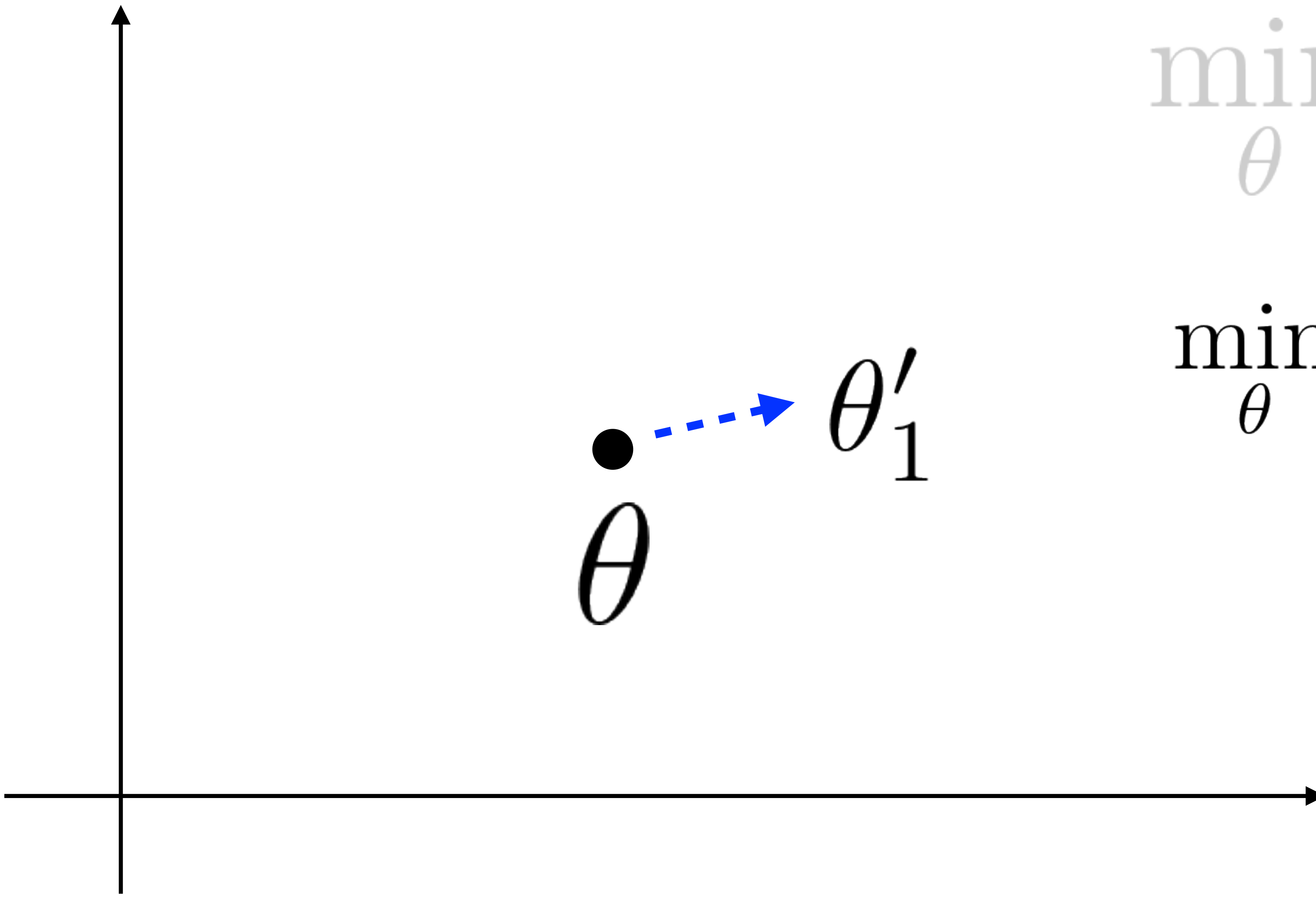
Hariharan et al. 2016, Finn et al. 2017

Generalizing to N tasks

Task1: Apple v/s Orange

$$\min_{\theta} \mathcal{L}_{\tau_1}(f_{\theta})$$

$$\min_{\theta} \sum_i \mathcal{L}_{\tau_i}(f_{\theta'_i})$$



$$\theta'_1 = \theta - \alpha \nabla \mathcal{L}_{\tau_1}(f_{\theta})$$

Hariharan et al. 2016, Finn et al. 2017

More Details

Low Shot Visual Recognition
Hariharan et al. 2016

Model Agnostic Meta-learning
Finn et al. 2017

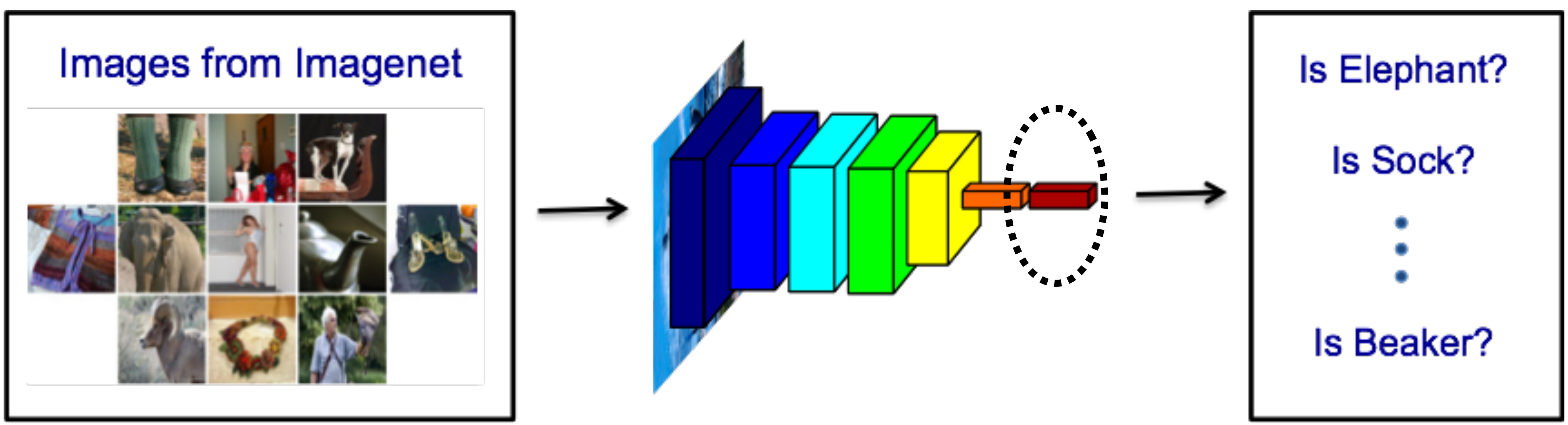
Until Now

Finetuning

Nearest Neighbor Matching

Siamese Network based Metric Learning

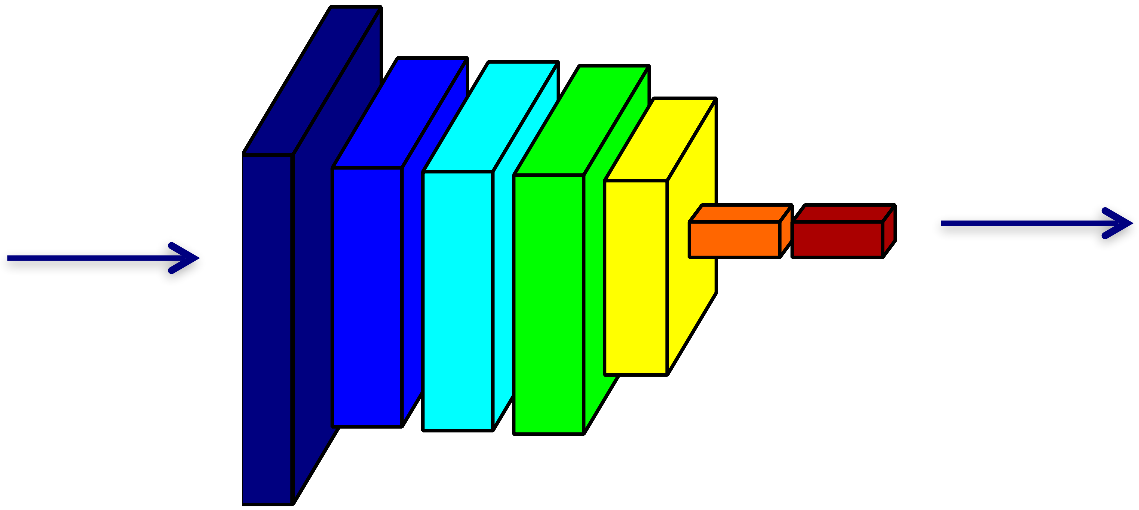
Meta-Learning: Training for fine-tuning



Better Features —> Better Transfer!

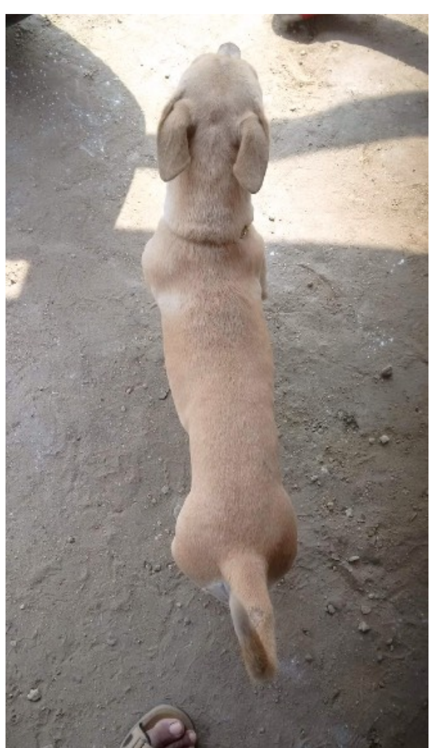
In practice, how good are these features?

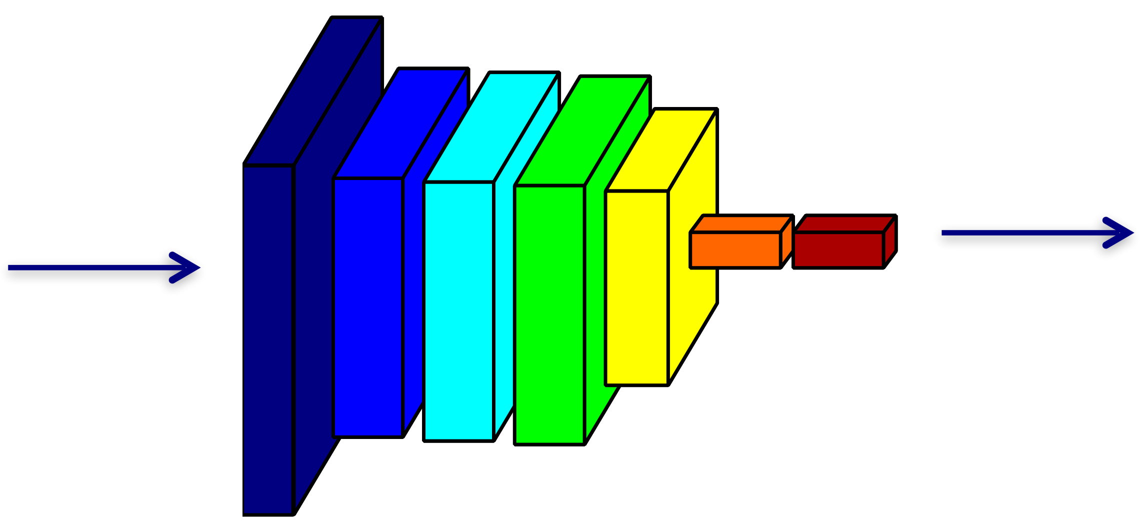
Dog from Imagenet



Accuracy ~80%

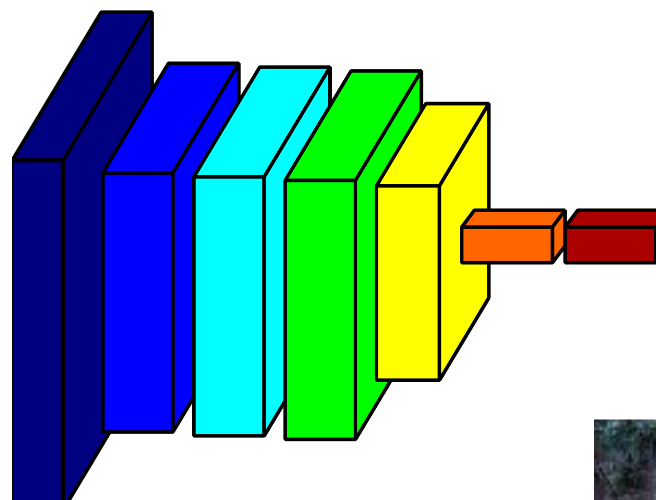
Dog





Accuracy ~20%

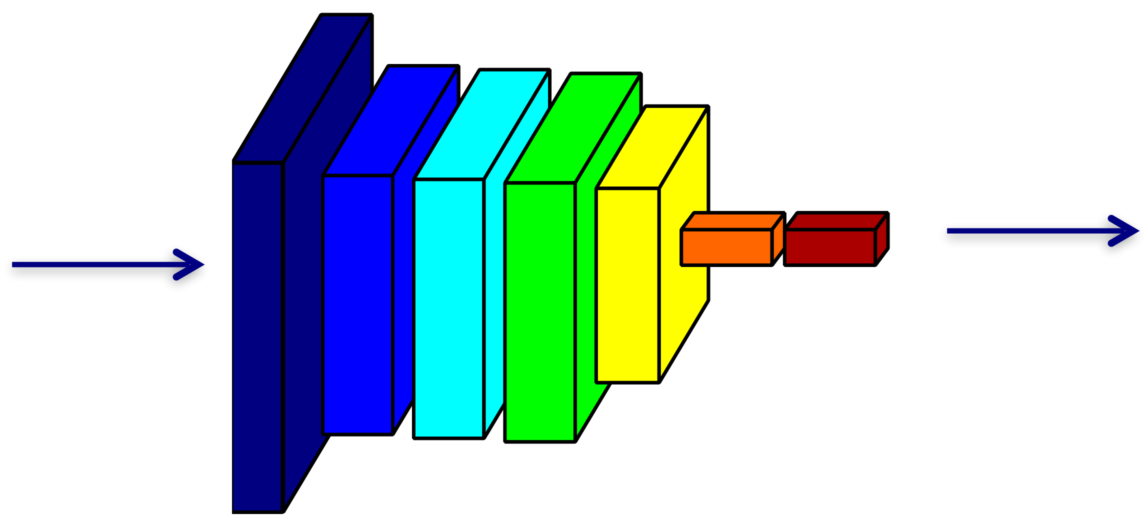
Consider the task of identifying cars ...



Positives

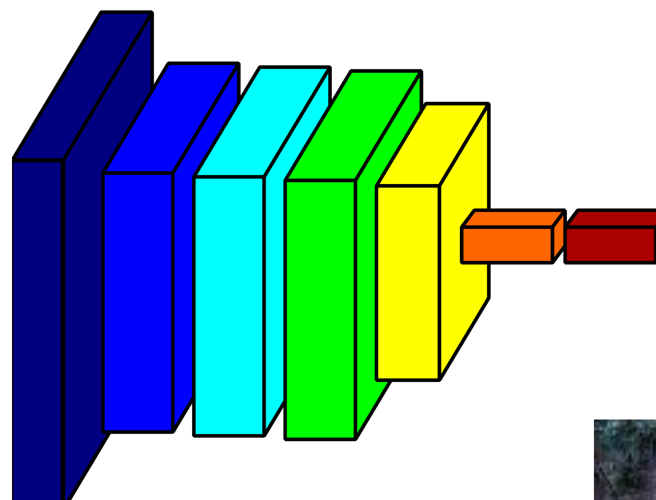
Negatives

Testing the model



???

Learning Spurious Correlations



Unbiased look at Dataset bias, Torralba et al. 2011

More parameters in the network

$\downarrow$

More chances of learning spurious correlations!!

Maybe this problem will be avoided if we first learn simple tasks and then more complex ones??

Sequential/Continual Task Learning

Fine-tuning

Poor performance on Task-1 !!!

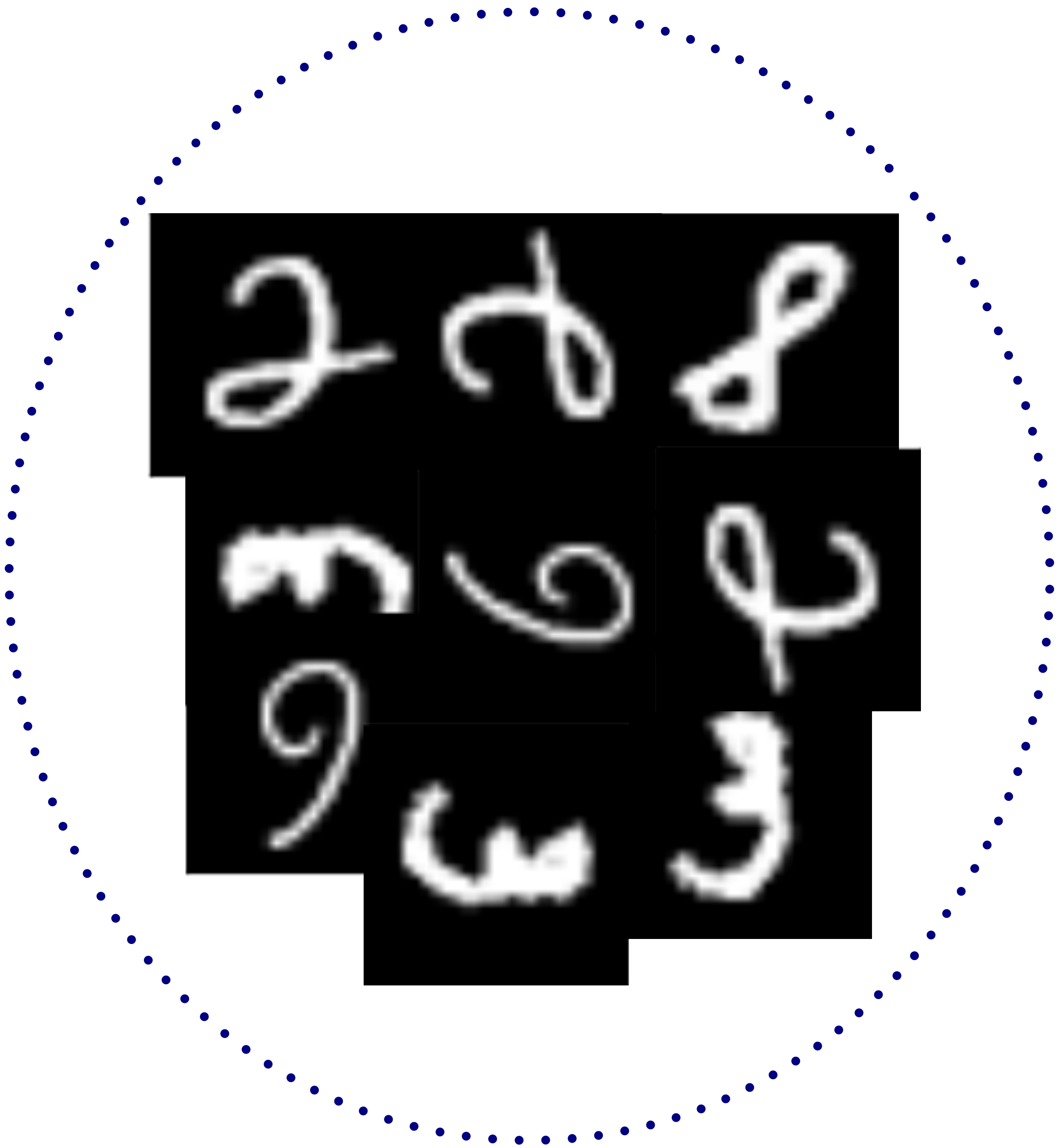
Catastrophic forgetting in closely related tasks

Training on rotating MNIST

Test

High
Low
Accuracy

In machine learning, we generally assume IID* data



Sample batches of data!

Each batch: uniform distribution of rotations

*IID: Independently and Identically Distributed

In machine learning, we generally assume IID* data

In real world, data is often not batched :)

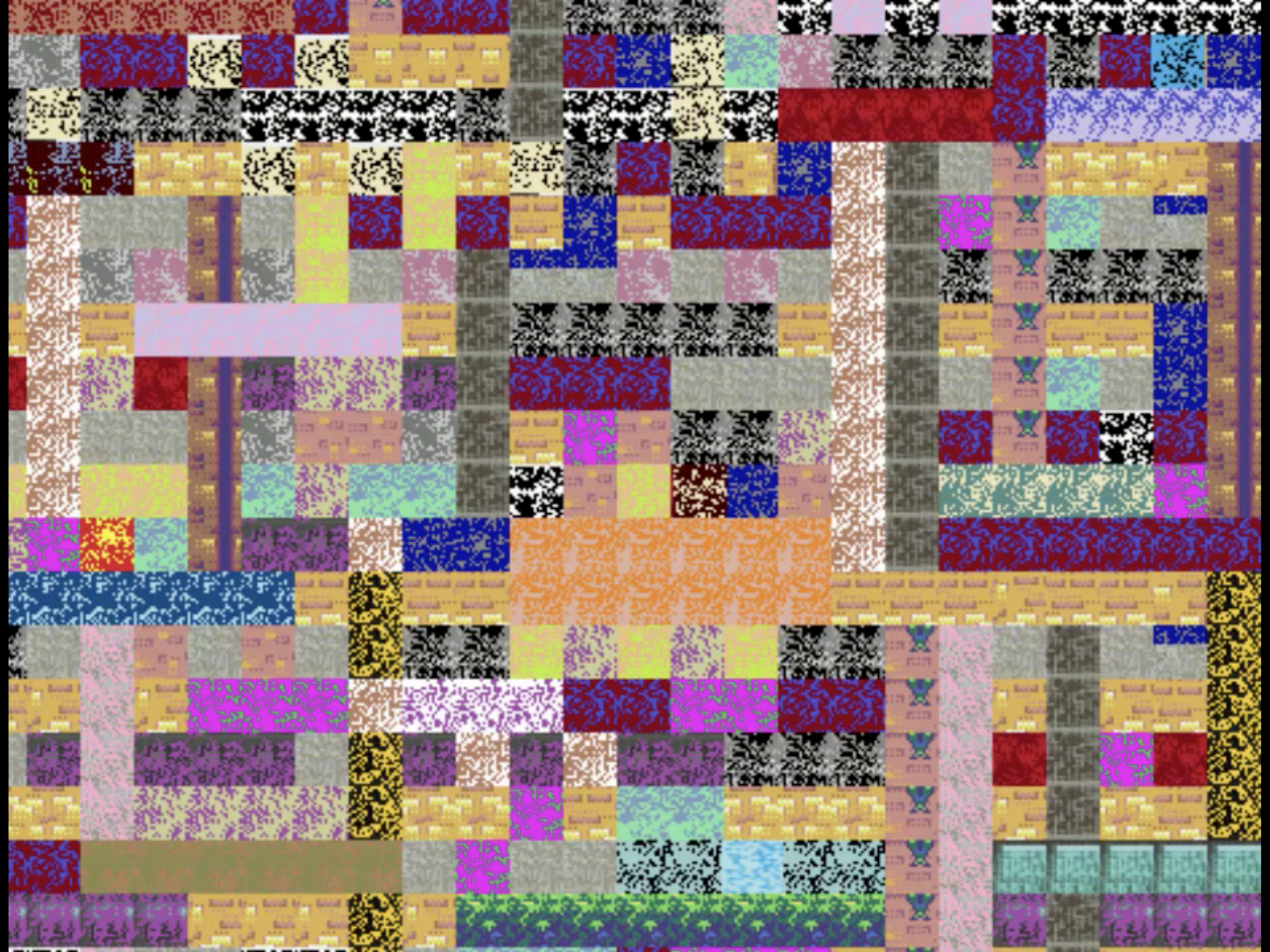
of data!

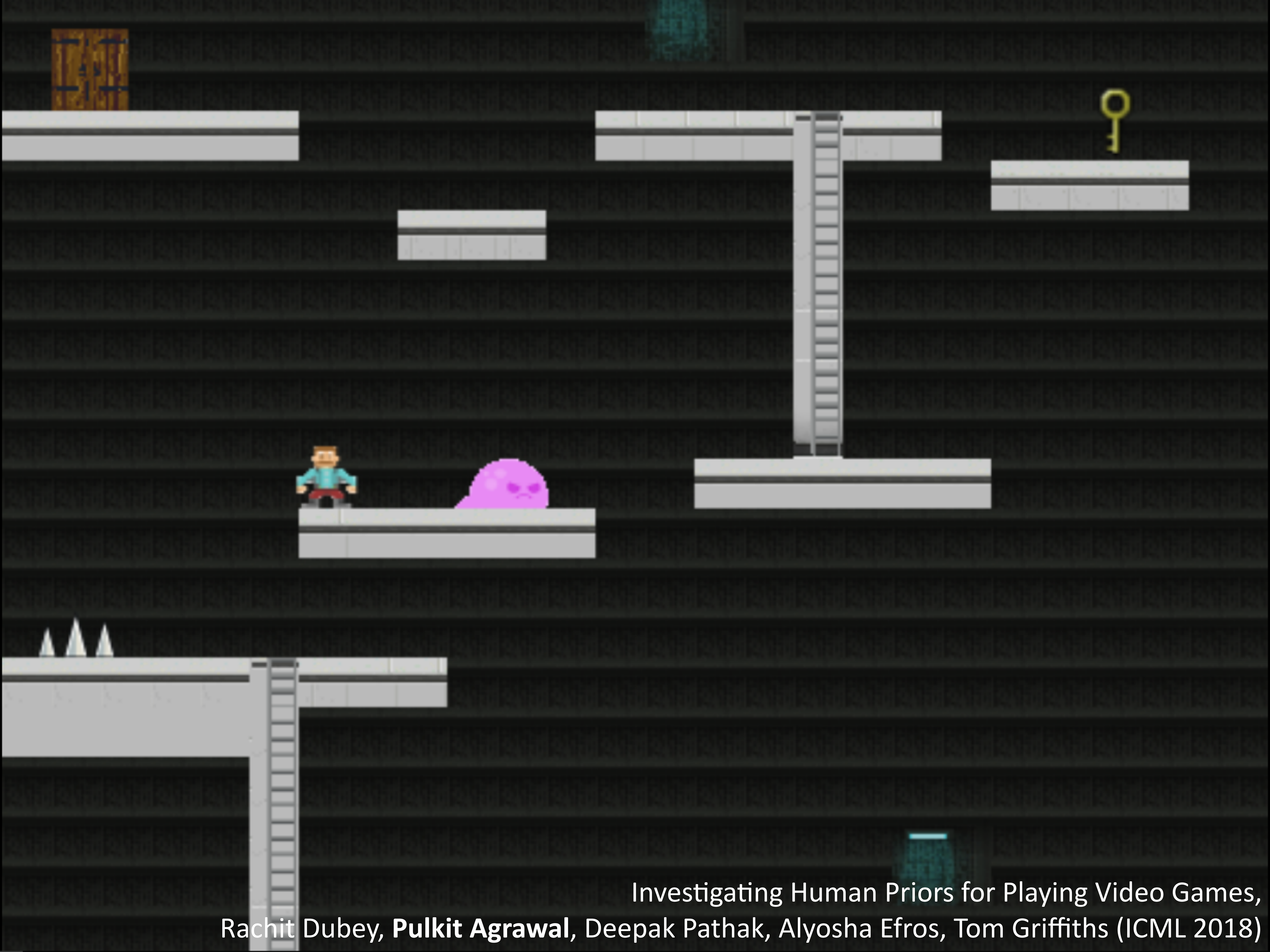
Each batch: uniform distribution of rotations

*IID: Independently and Identically Distributed

Continual learning is natural …

In the context of reinforcement learning



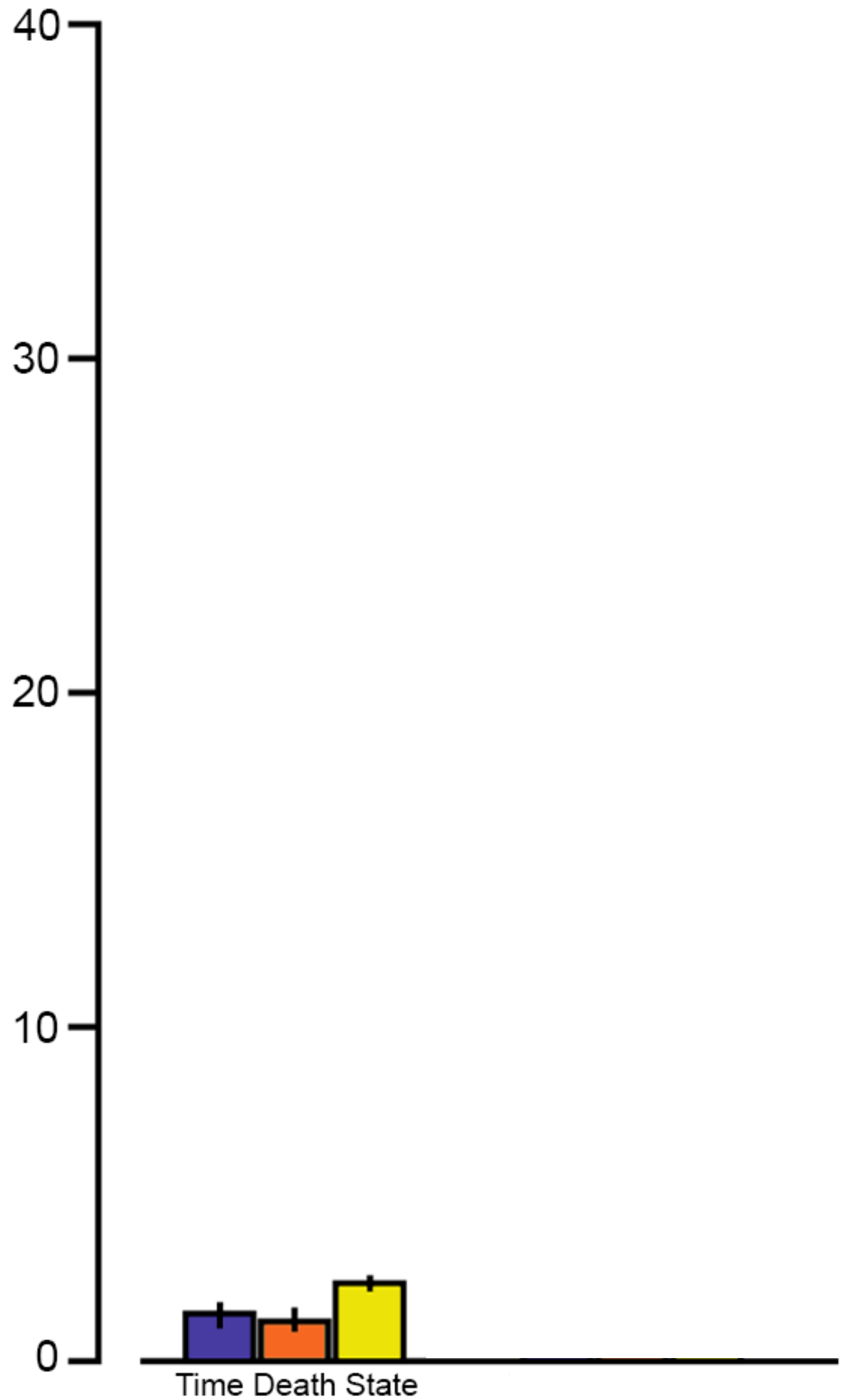


Investigating Human Priors for Playing Video Games,
Rachit Dubey, **Pulkit Agrawal**, Deepak Pathak, Alyosha Efros, Tom Griffiths (ICML 2018)

Humans make use of prior knowledge for exploration

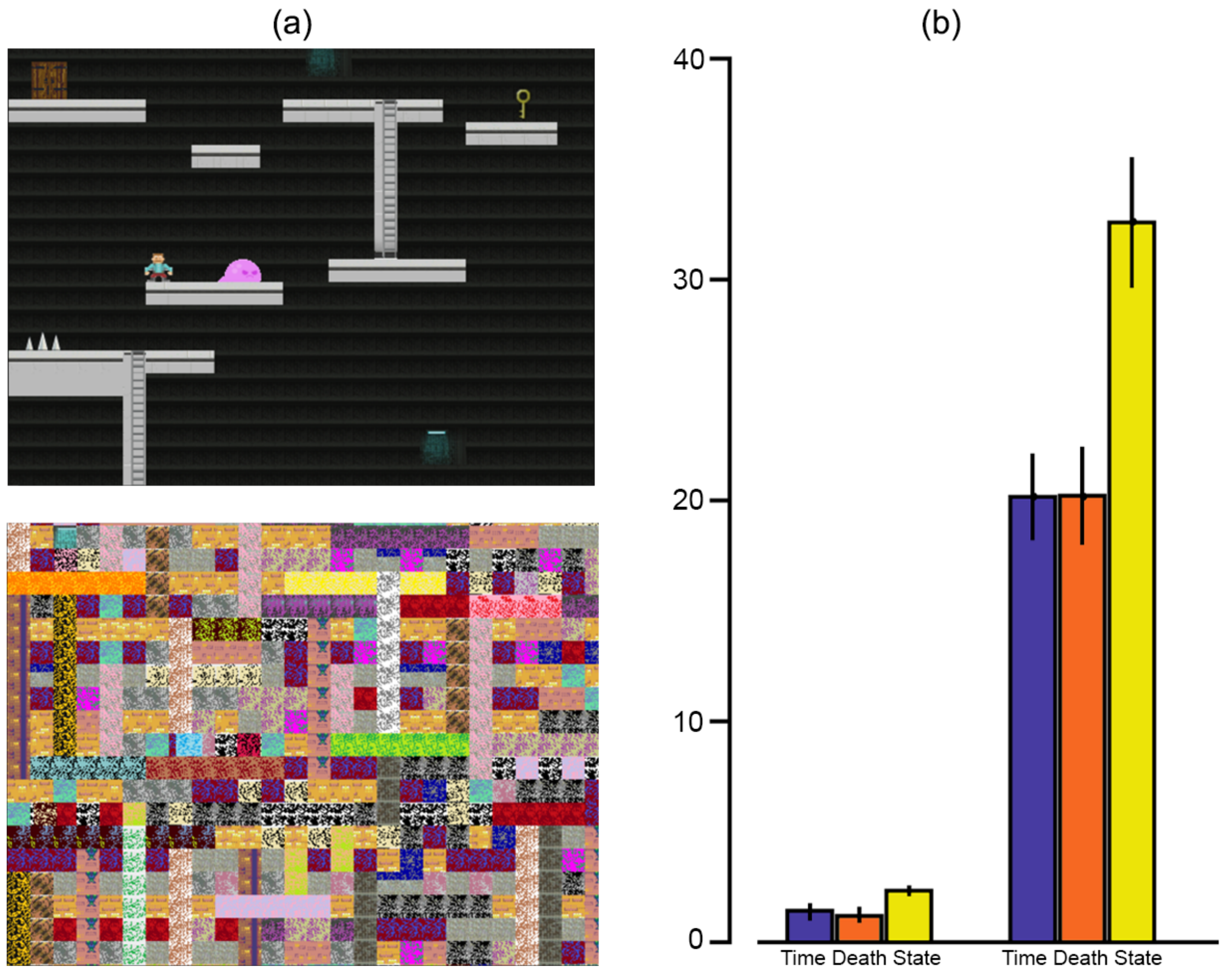
(a)

(b)



Investigating Human Priors for Playing Video Games,
Dubey R., **Agrawal P.**, Deepak P., Efros A., Griffiths T. (ICML 2018)

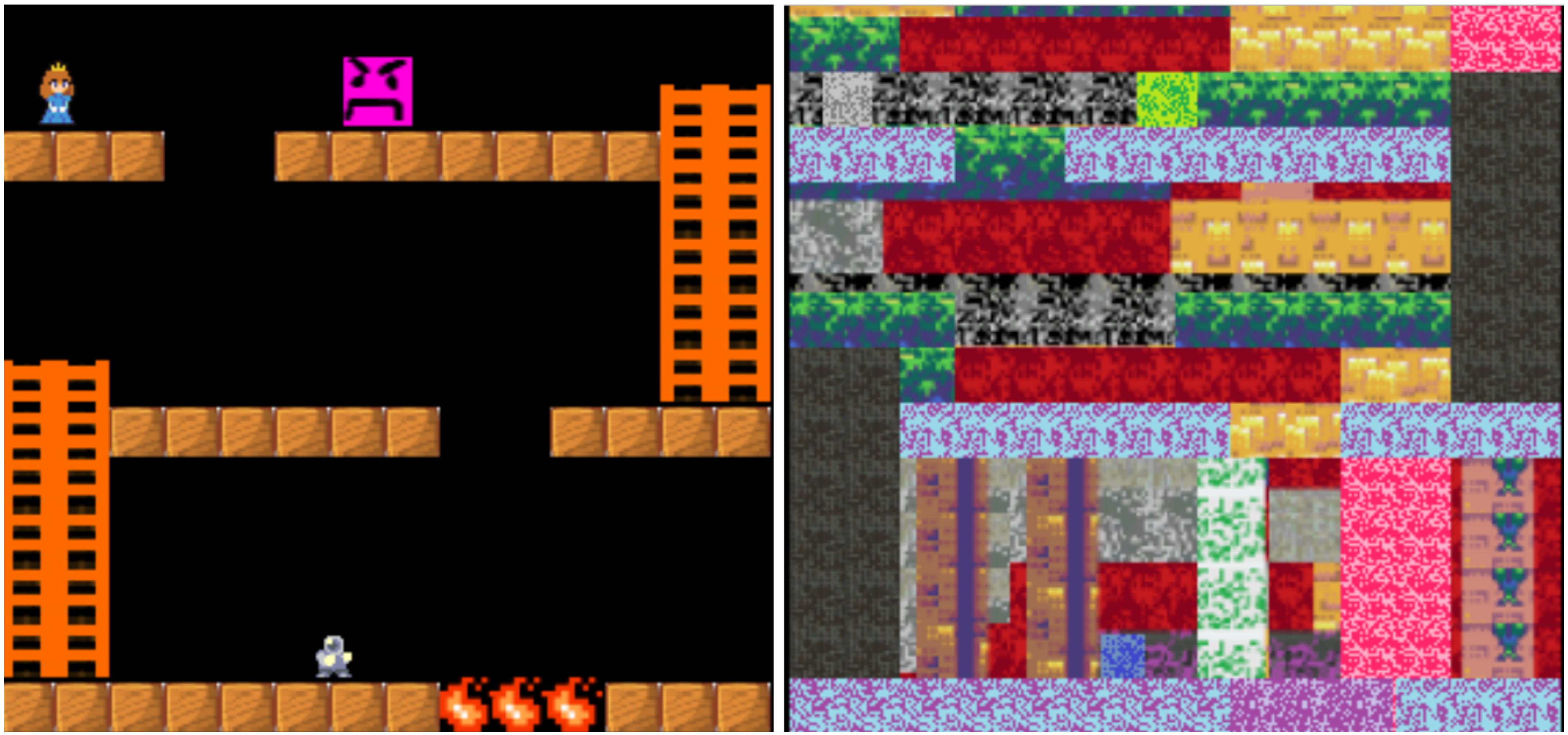
Humans make use of prior knowledge for exploration

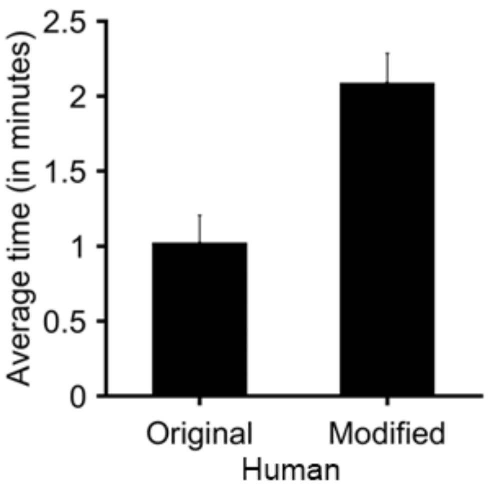


Investigating Human Priors for Playing Video Games,
Dubey R., **Agrawal P.**, Deepak P., Efros A., Griffiths T. (ICML 2018)

What about Reinforcement Learning Agents?

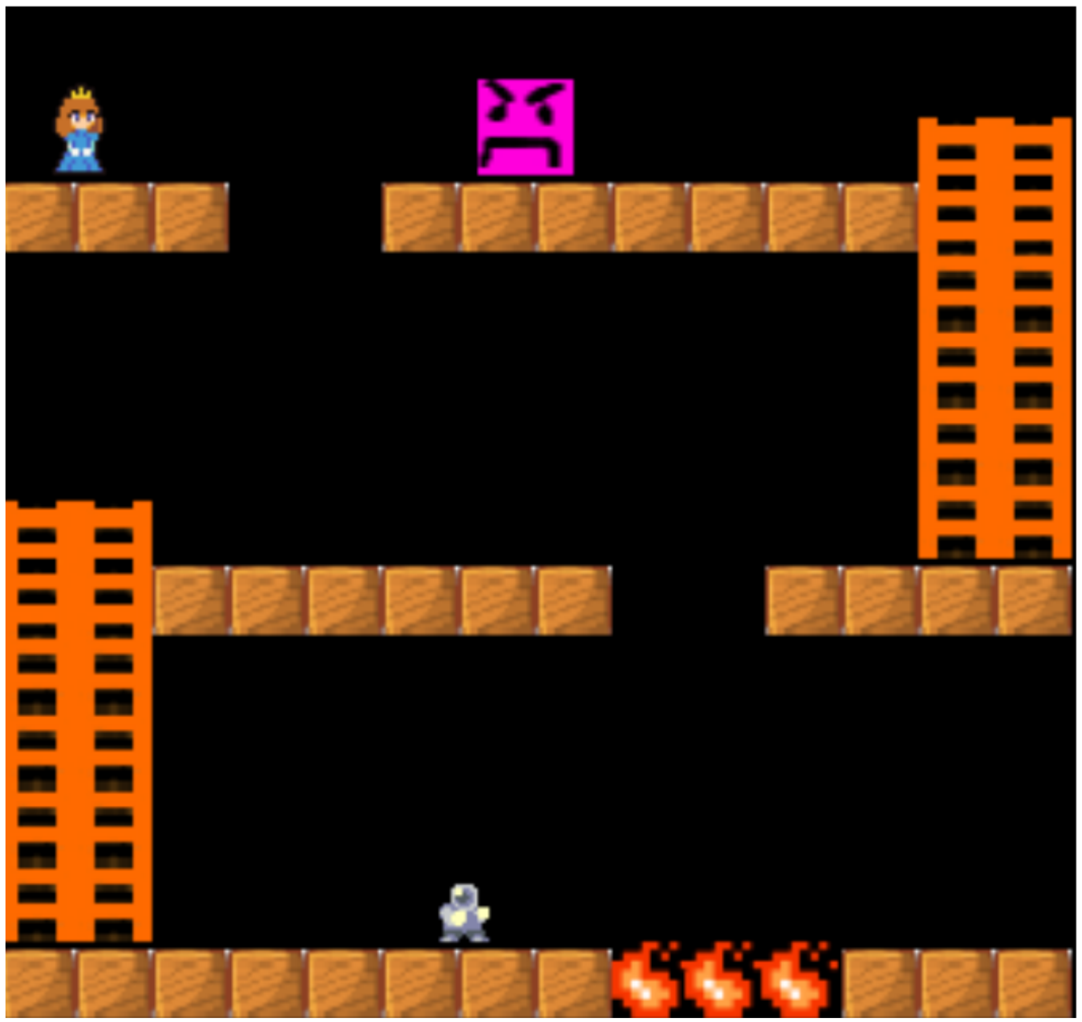
In a simpler version of the game ..

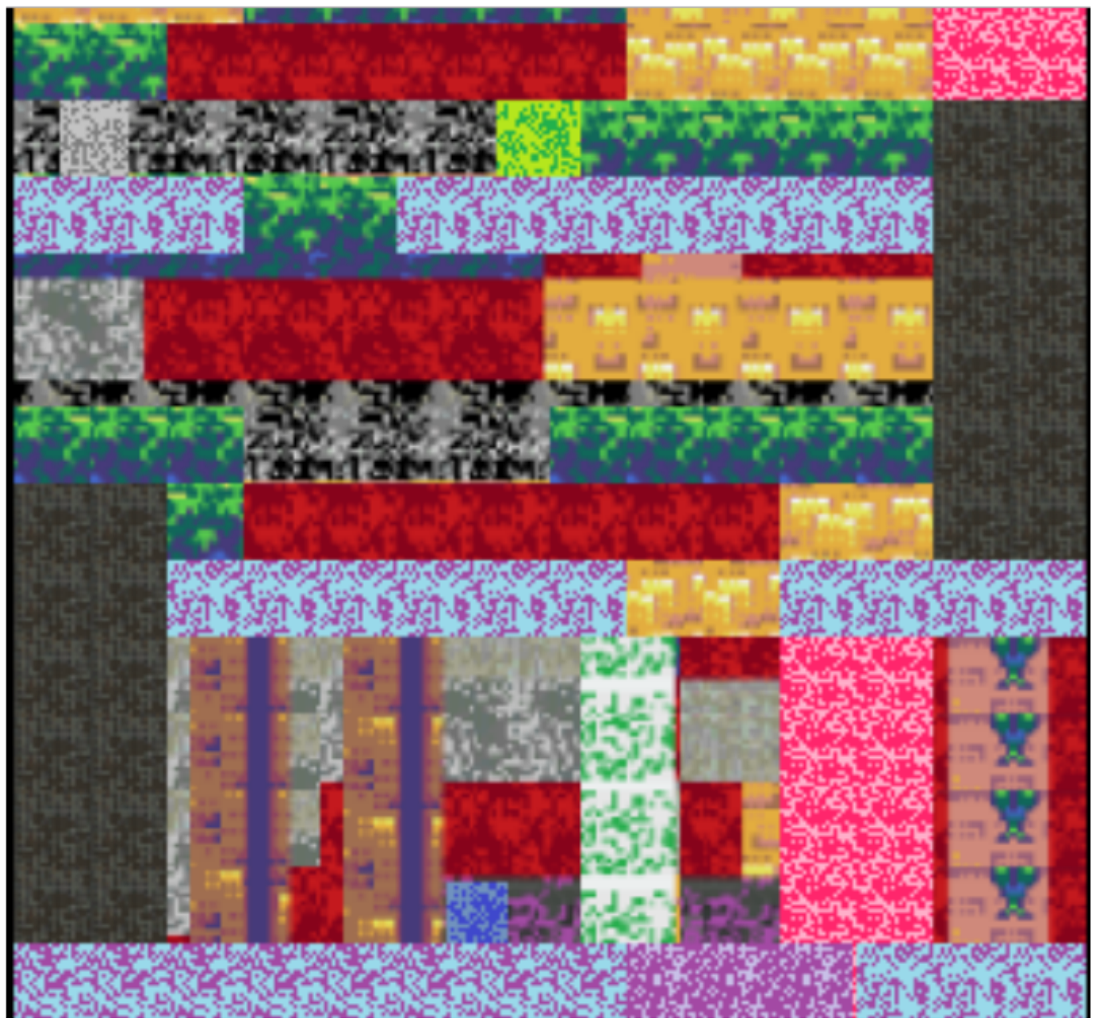


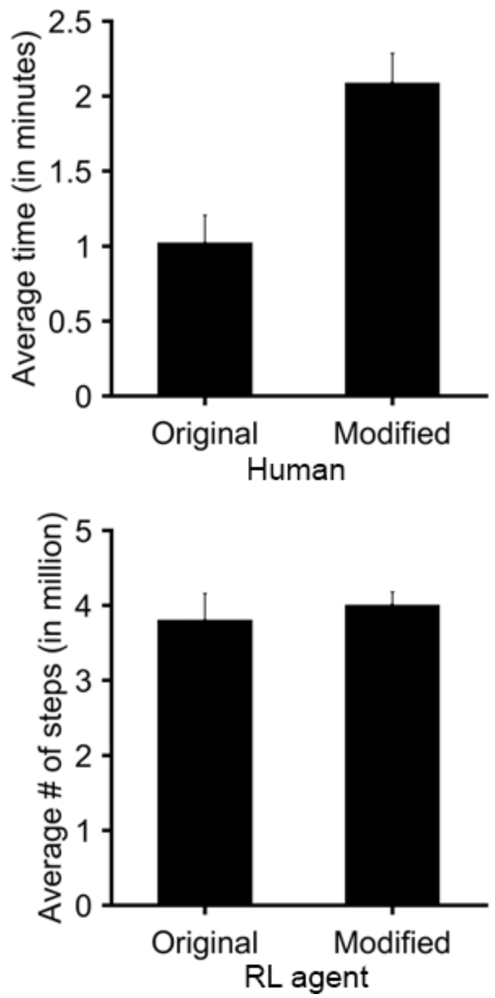


Investigating Human Priors for Playing Video Games,
Dubey R., **Agrawal P.**, Deepak P., Efros A., Griffiths T. (ICML 2018)

For RL agents, both games are the same!







Investigating Human Priors for Playing Video Games,
Dubey R., **Agrawal P.**, Deepak P., Efros A., Griffiths T. (ICML 2018)

Equip Reinforcement Learning Agents
with
prior knowledge?

Common-Sense/Prior Knowledge

Hand-design

Common-Sense/Prior Knowledge

Hand-design

Learn from Experience

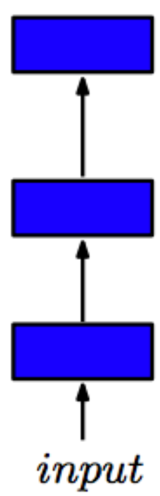
Transfer in Reinforcement Learning —> Very limited success

Good solution to continual learning required!

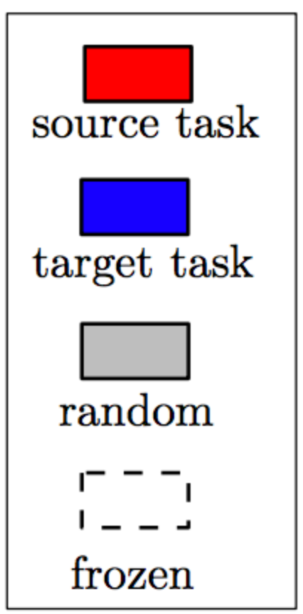
How to deal with catastrophic forgetting?

Just remember the weights for each task!

Progressive Networks (Rusu et al. 2016)

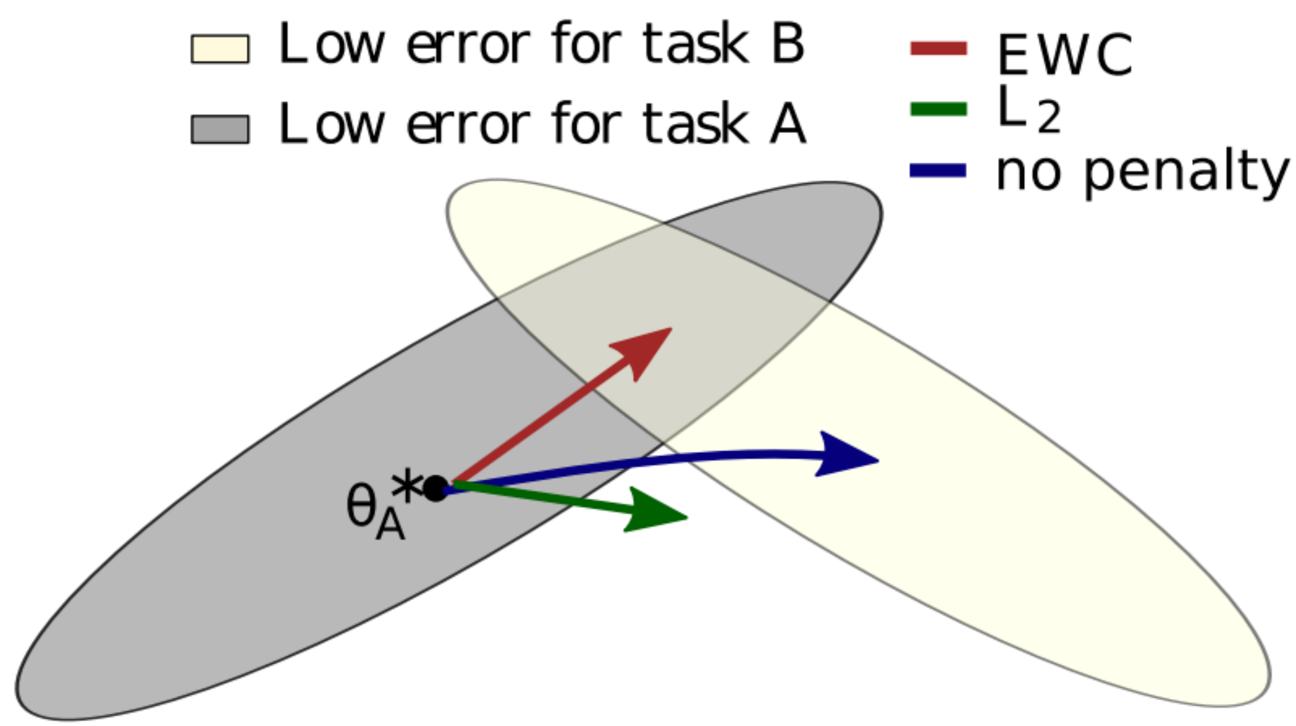


(1) **Baseline 1**



Can we do something smarter than storing all the weights?

Overcoming Catastrophic Forgetting (Kirkpatrick et al. 2017)



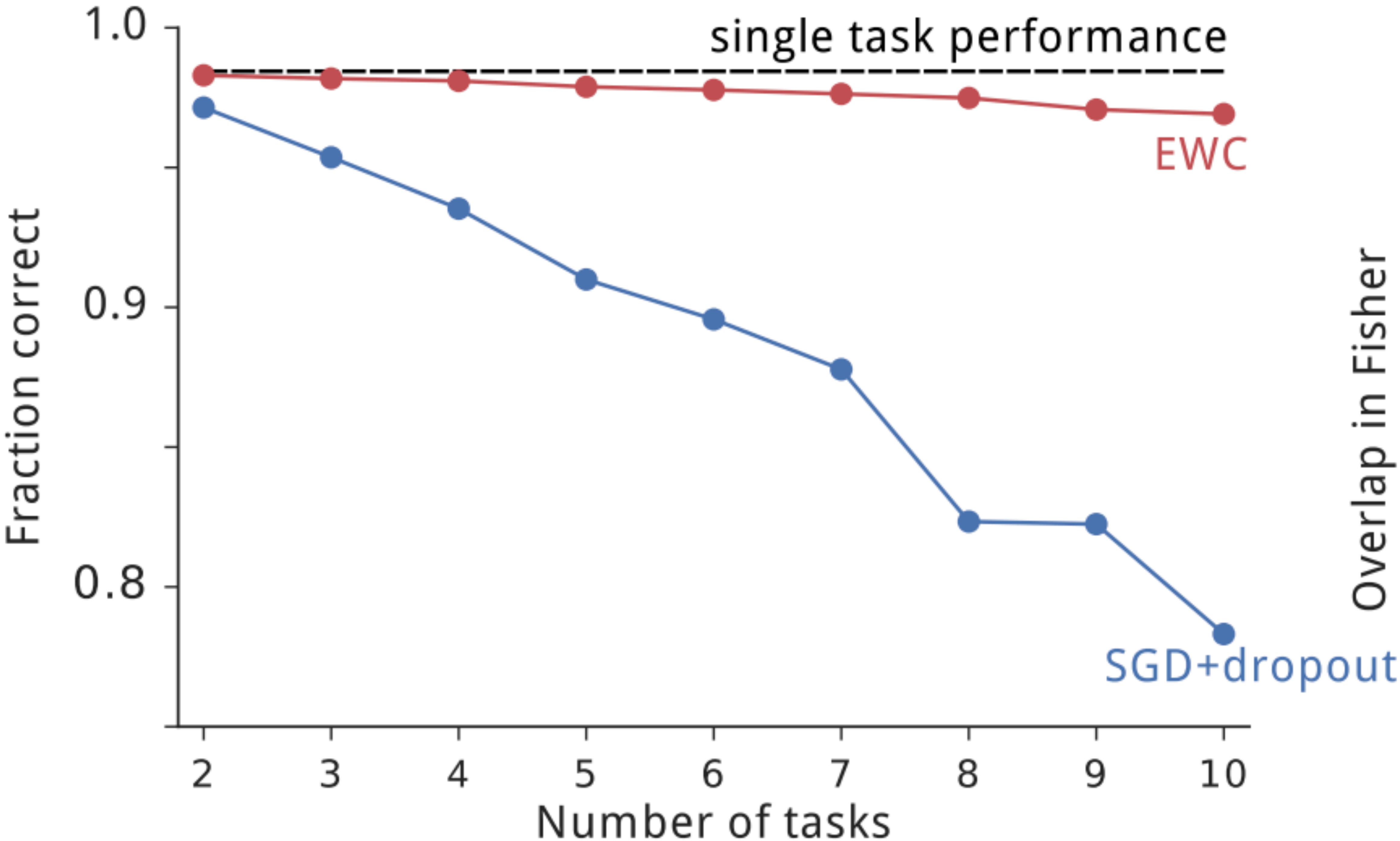
$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \sum_i \frac{\lambda}{2} F_i (\theta_i - \theta^*_{A,i})^2$$

Don't change weights that are informative of task A

Fisher Information

EWC: Elastic Weight Consolidation

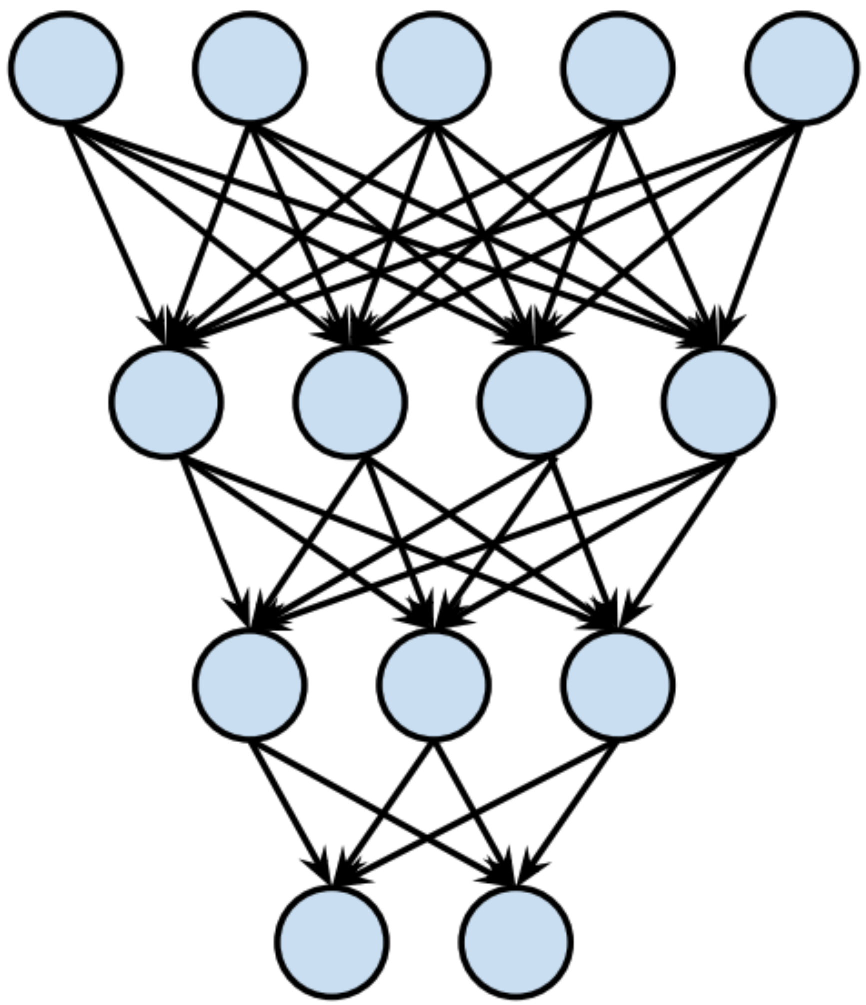
Overcoming Catastrophic Forgetting (Kirkpatrick et al. 2017)



Eventually we will run out of capacity!

Is there a better way to make use of the neural network capacity?

Neural Networks are compressible post-training

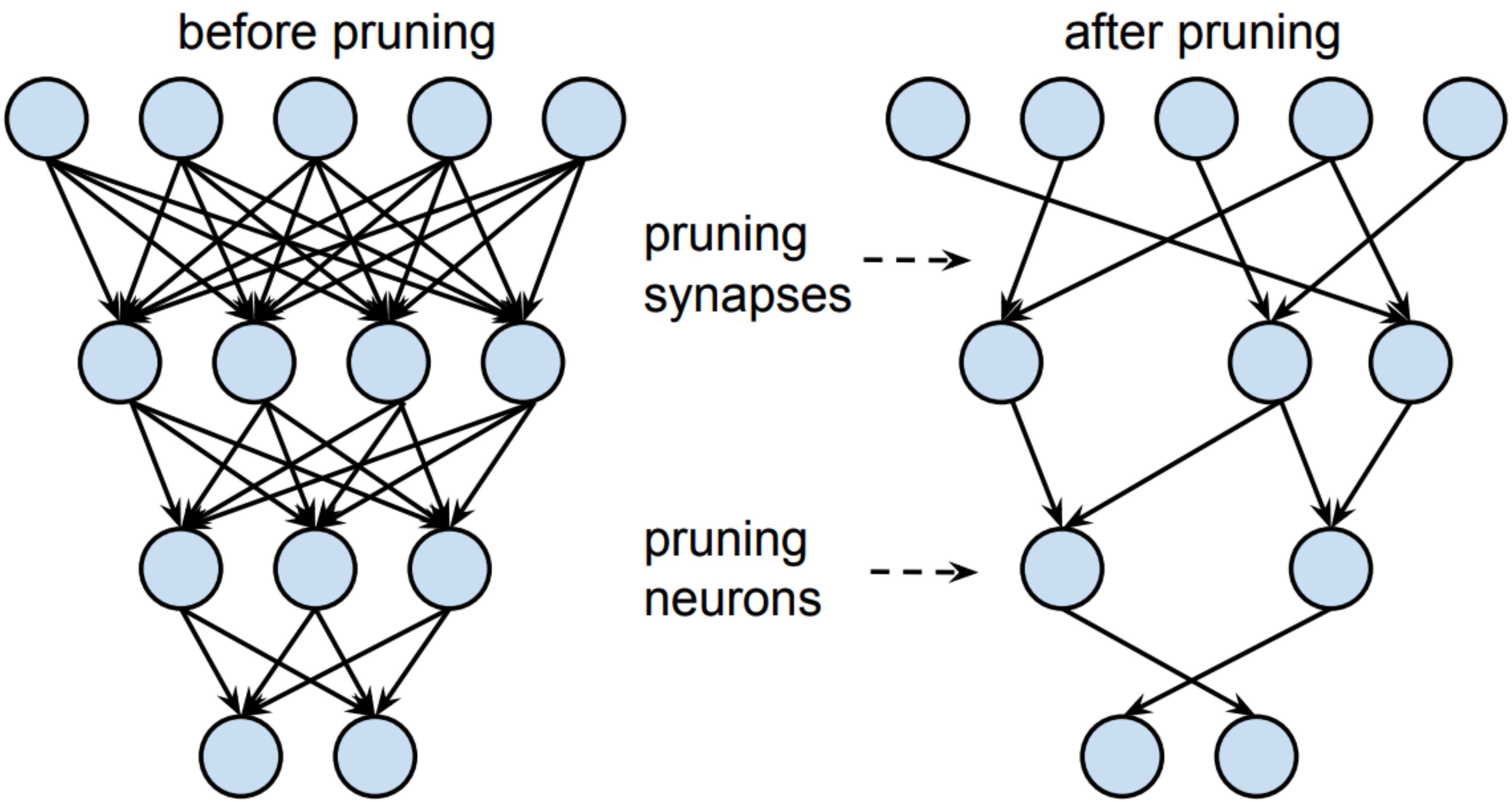


$X\%$

(Slide adapted from Brian Cheung)

(Han et. al. 2015)

Neural Networks are compressible post-training



$$X\%$$

$$(X - \epsilon)\%$$

(Slide adapted from Brian Cheung)

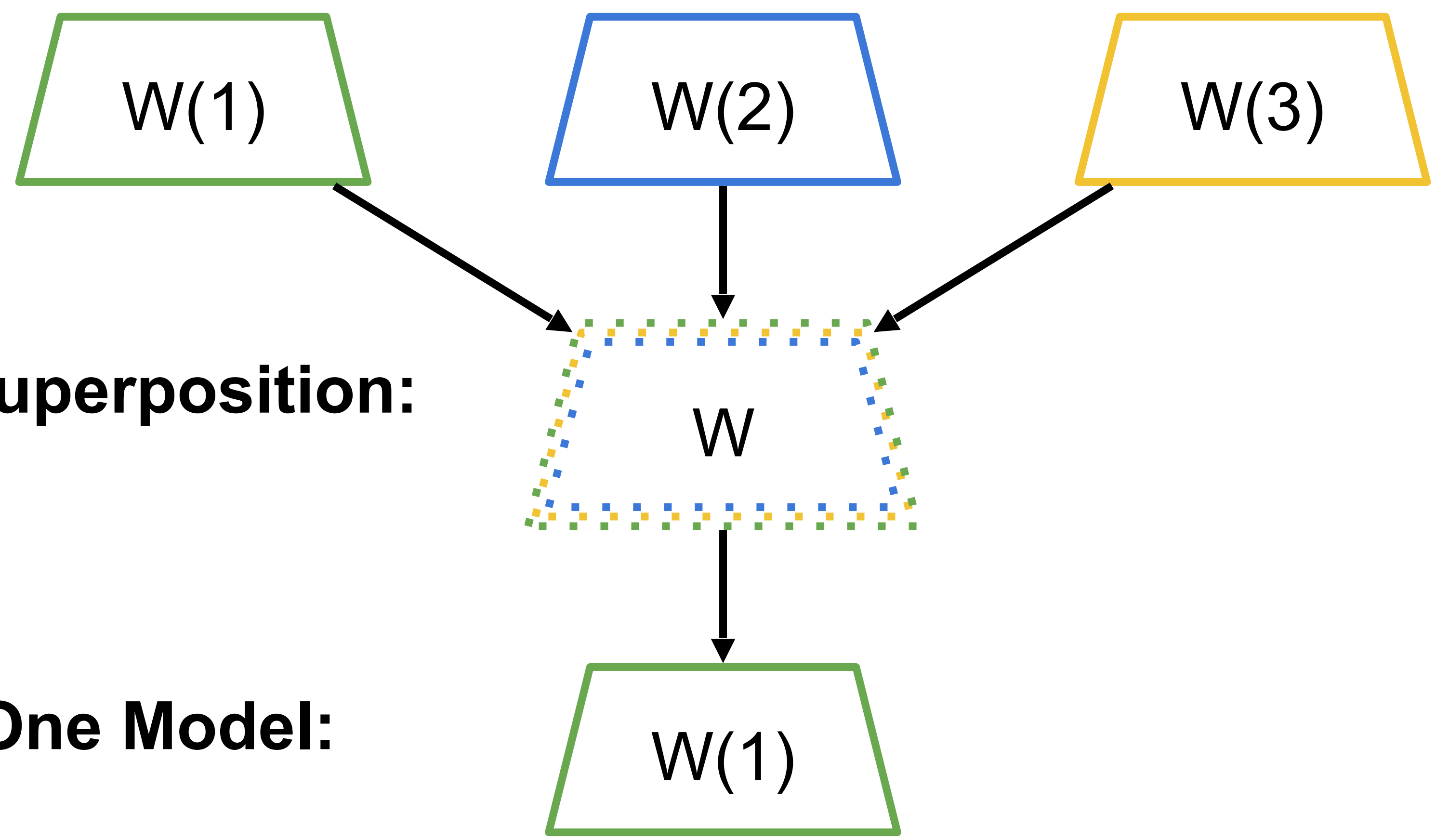
(Han et. al. 2015)

Negligible performance change after pruning —> Neural Networks are over-parameterized

Can we make use of over-parameterization?

We will have to make use of “excess” capacity during training

Superposition of many models into one (Cheung et al., 2019)



Superposition:

One Model:

Implementation: $y = W(c(1) \odot x)$

Refer to the paper for details