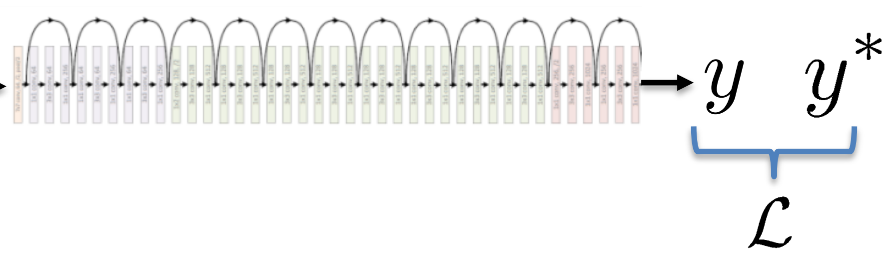
Lecture: Deep Learning and Differential Programming

3.3 Transfer Learning

https://liris.cnrs.fr/christian.wolf/teaching

INSA LYON Christian Wolf

Invariances / symmetries



Common deep networks are not invariant to standard image changes (rotations, illumination changes, noise, deformations).

But they can be trained to be.

Data augmentation

1. Randomly choose a batch of images+labels from the training set

y^*

T

2. Apply a random tranformation on images and labels

3. Train the model on this data

$\underbrace{y \quad y^*}_{\mathcal{L}}$

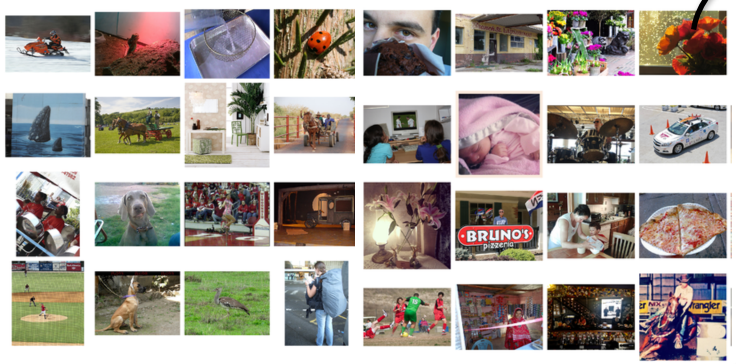
4. Goto 1

Knowledge transfer

Goal: transfer knowledge learned from a source domain (e.g. a large dataset of labeled images) to a (often smaller) target domain

Example:

Source:
ImageNet (public)
1 000 000 images + labels
1000 classes

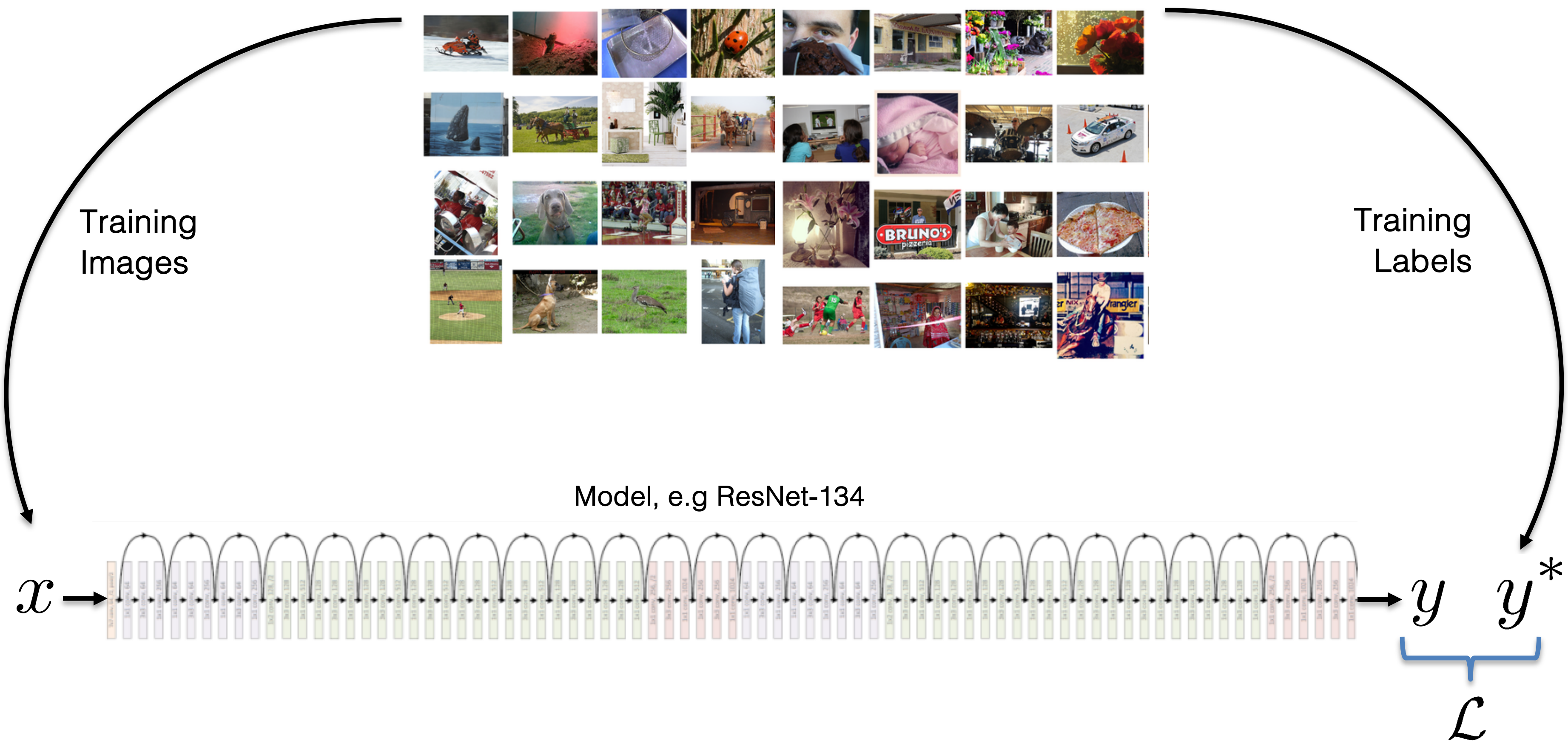


For each image:
Label
(e.g. which class among 1000)

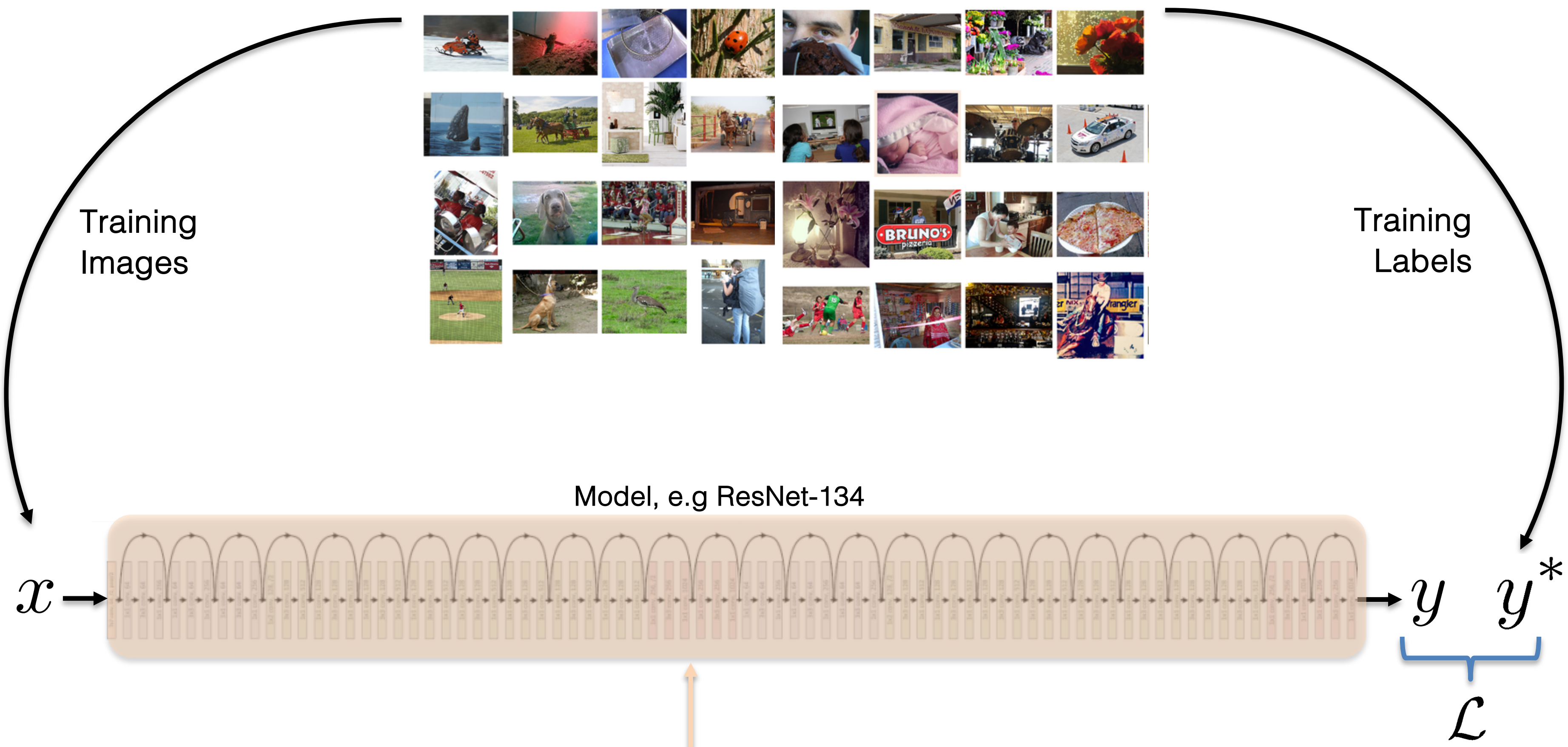
Target:
Industrial application: Crop growth stage
3000 images + labels
16 classes

For each image:
Label
(e.g. which class among 16)

Pre-training

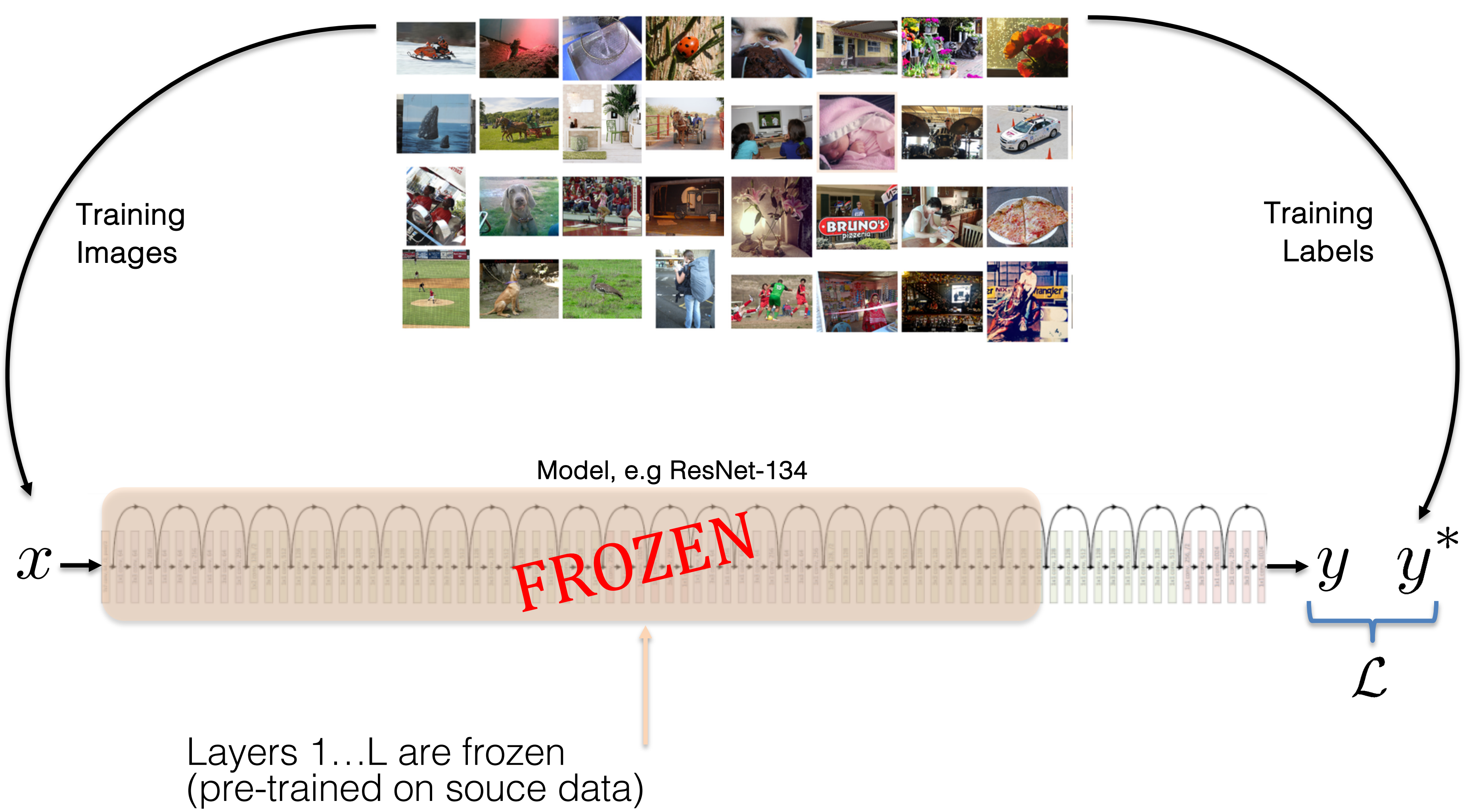


Pre-training

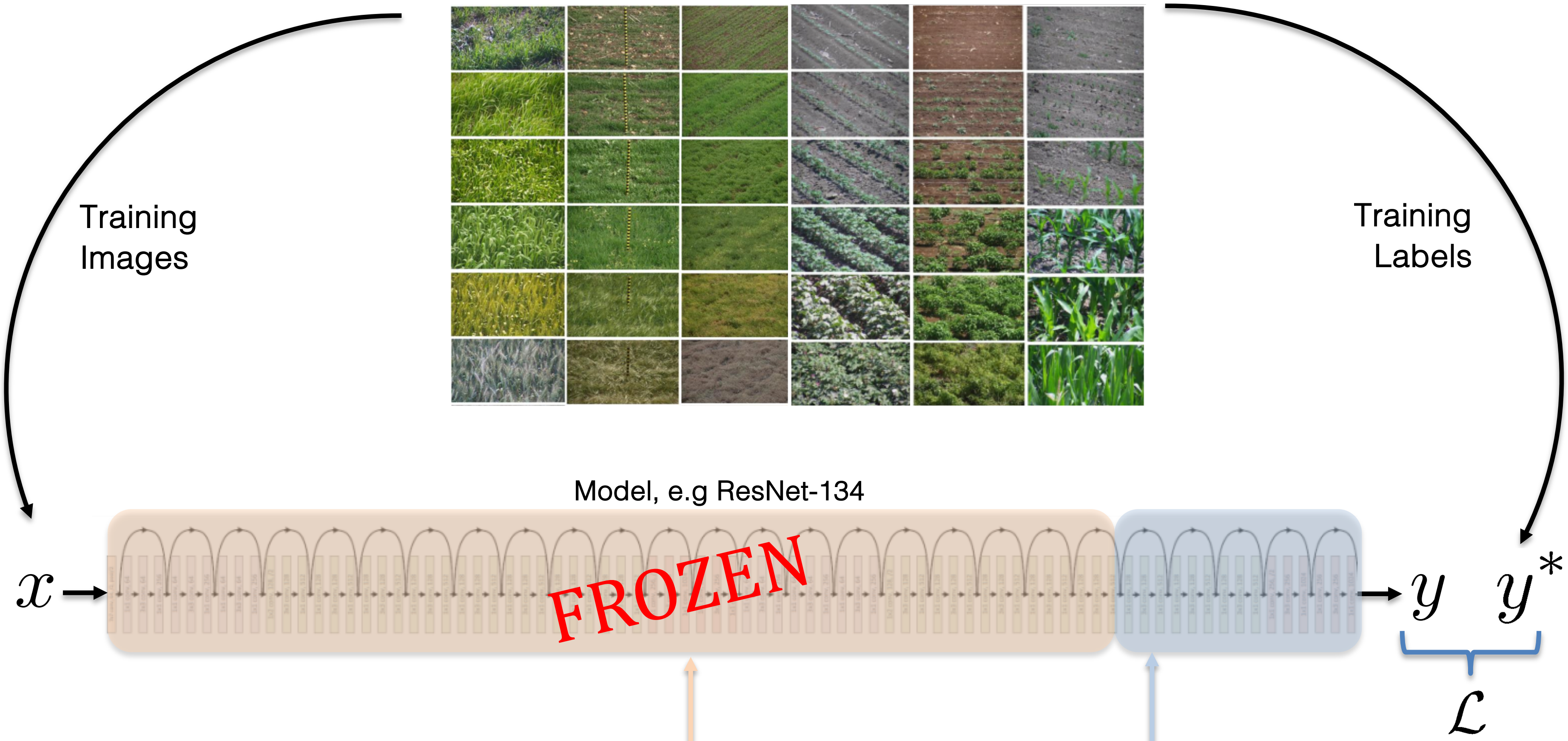


Pre-train all weights (all layers)

Pre-Training



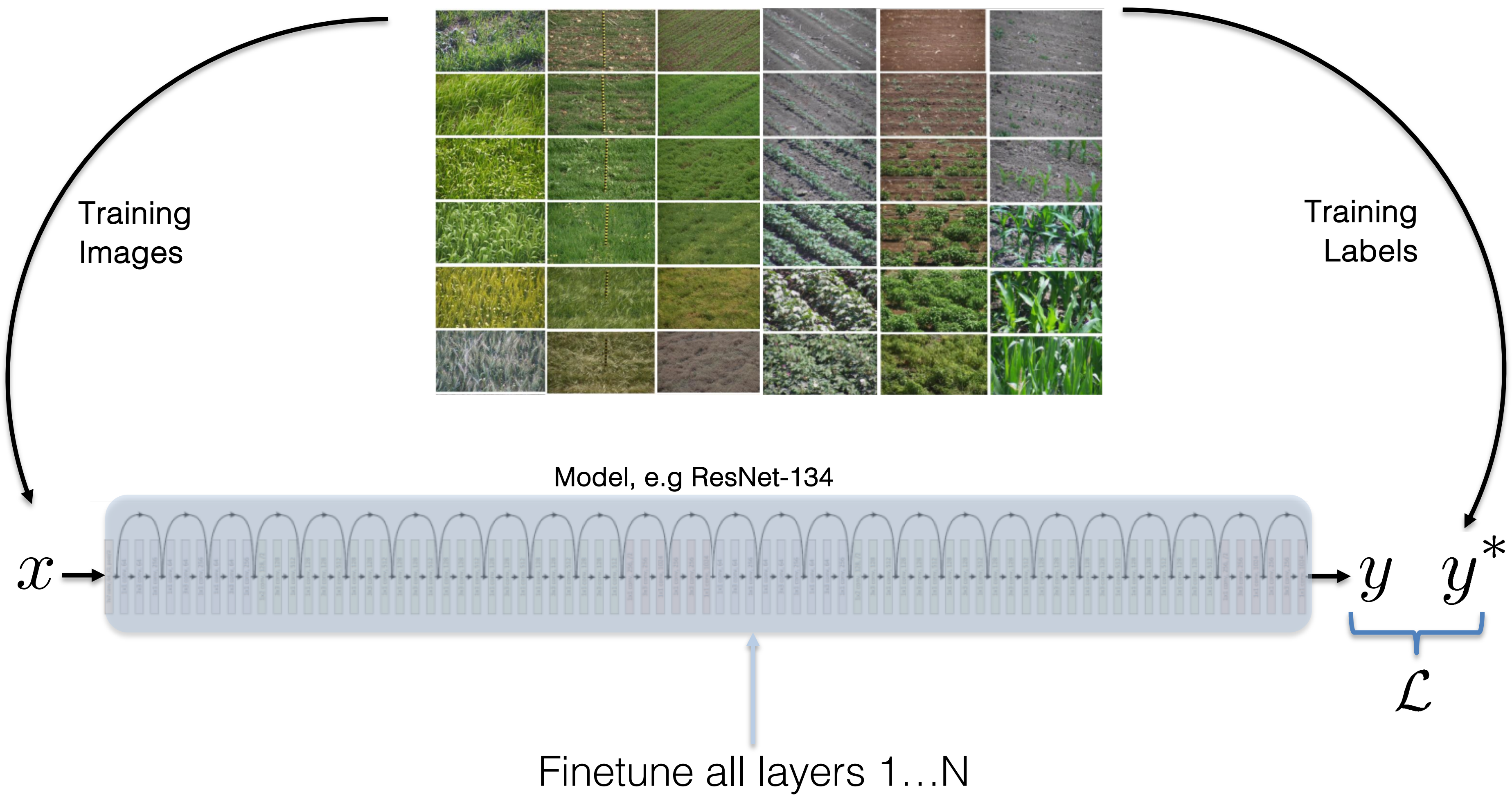
Training



Layers 1…L are frozen
(pre-trained on souce data)

Layers L…N are trained on the target dataset

Training



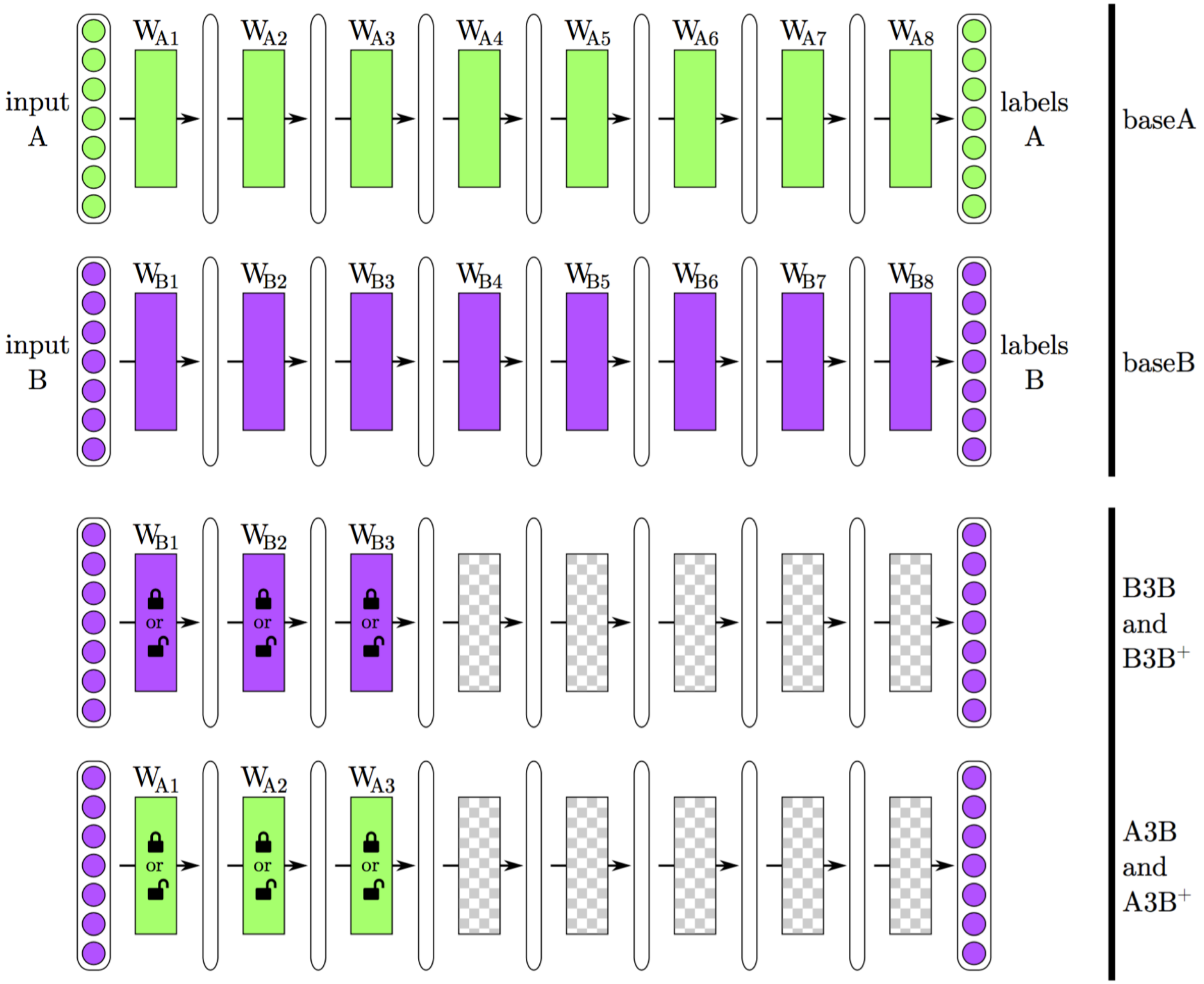
If we have enough data: unfreeze and finetune

How transferable are features in deep neural networks?

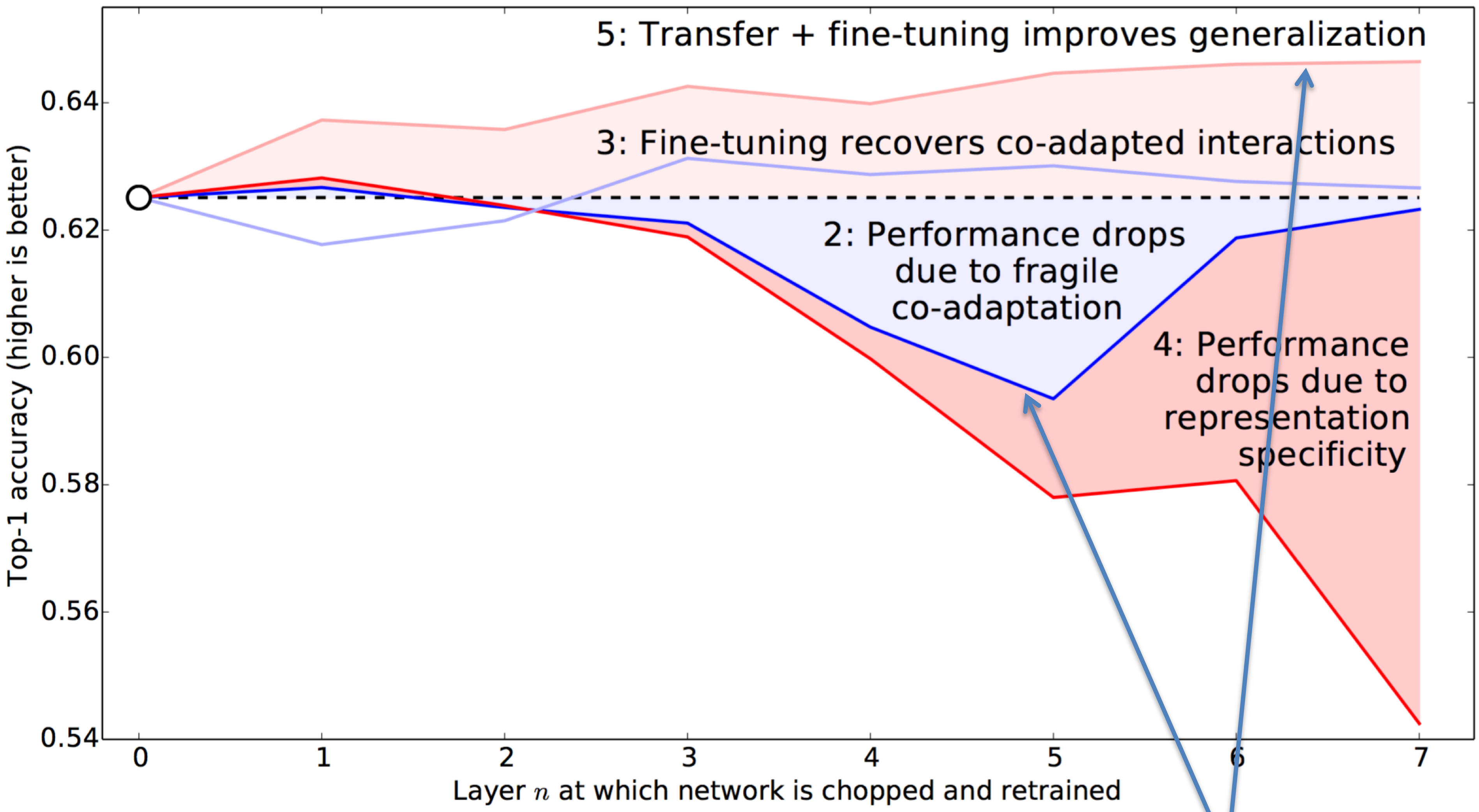
- Split of the ImageNet (ILSVRC) dataset into two subsets:
  - Subset A of images with man made content (cars et.)
  - Subset B of images with natural content (trees etc.)
- Train networks on both subsets.
- Choose a layer i, a randomly reinitialize parameters from layer i to the end.
- Run experiments retraining these layers:
  - on the same subset (AiA, BiB)
  - On the other subset (BiA)
  - Variant +: finetune layers up to I (AiA+,BiB+,BiA+)
- Compare performance to the baseline

[Yosinski, Clune, Bengio, Lipson, ICML 2014]

How transferable are features in deep neural networks?



[Yosinski, Clune, Bengio, Lipson, ICML 2014]



Numerical/optimization issues
(Positive or negative)

[Yosinski, Clune, Bengio, Lipson, "How transferable are features in deep neural networks?", 2014]

Domain adaptation

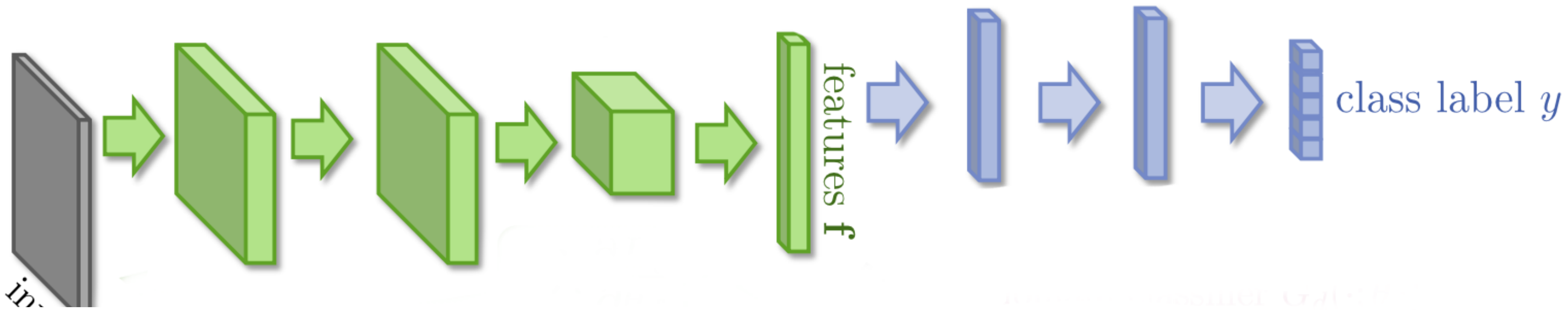
Goal: transfer knowledge learned from a source domain (e.g. a large dataset of labeled images) to a target domain a where often data is more sparse, and sometimes not labeled.

« Domain adaptation »

Knowledge transfer gone wrong from source domain « street » to target domain « railroad »

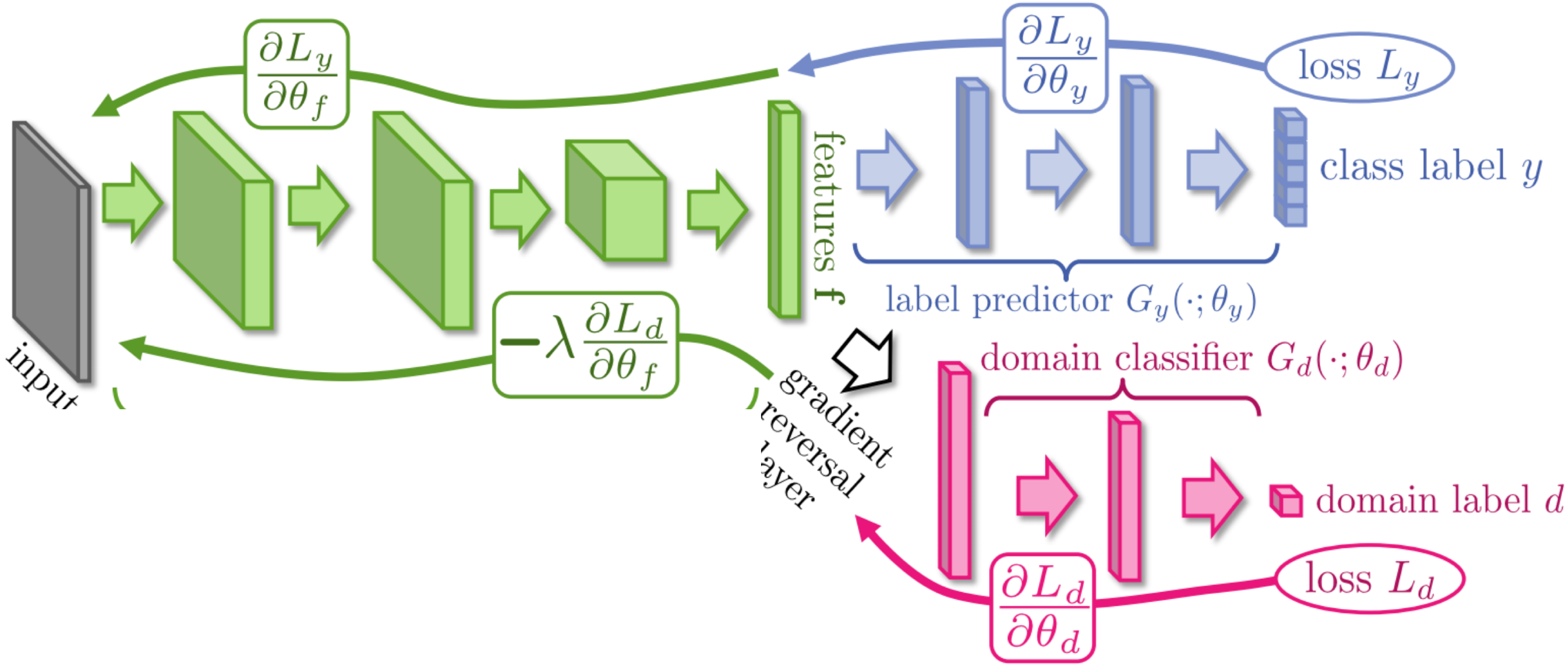
Adversarial domain adaptation

Select a feature layer and train it to be invariant to the shift in distribution between source and domain.



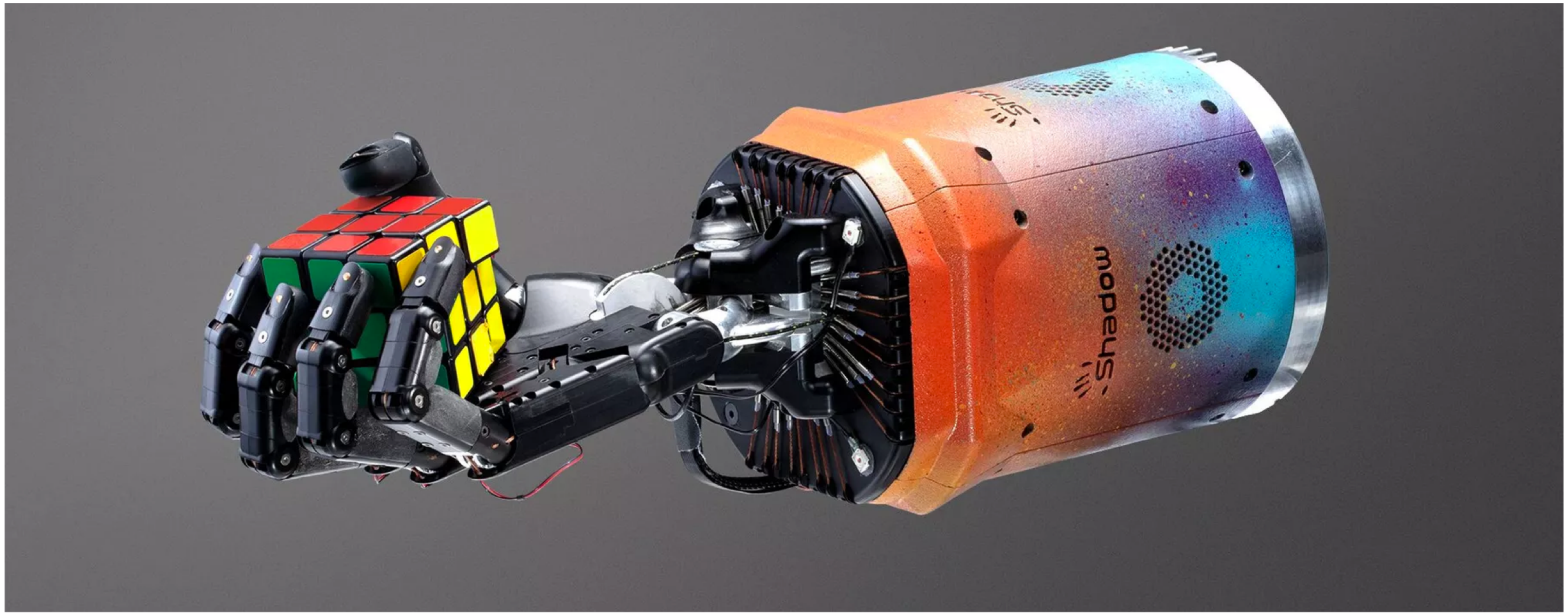
[Ganin and Lempitsky, ICML 2015]

Adversarial domain adaptation



[Ganin and Lempitsky, ICML 2015]

Learning dexterity and grasping



https://openai.com/blog/solving-rubiks-cube/

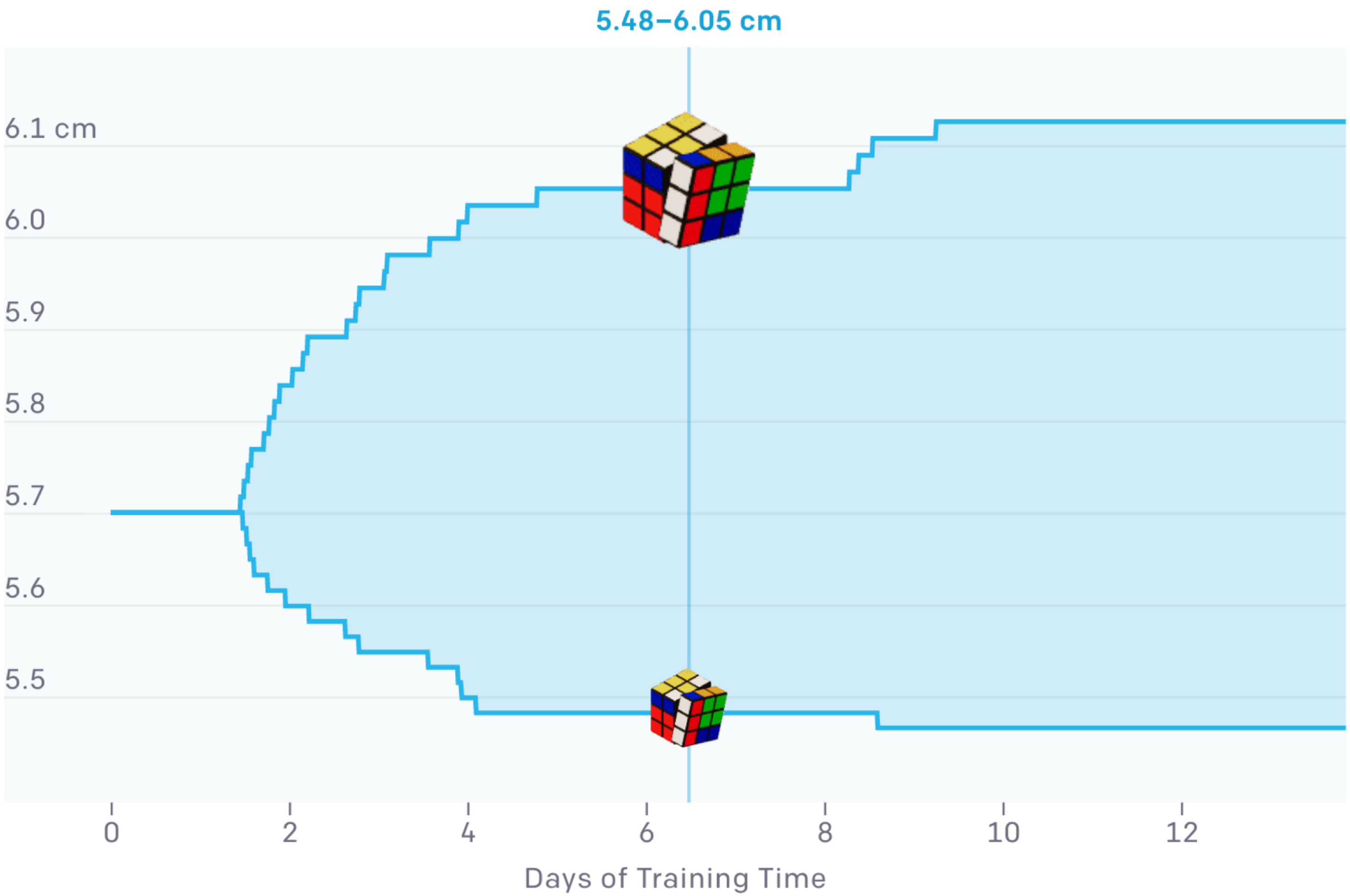
[Open-AI et al., October 2019]

Sim2real transfer

How can we transfer knowledge (eg policies) from simulations to real physical environments / robots?
Domain randomizations!

[Open-AI et al., October 2019]

Domain randomizations



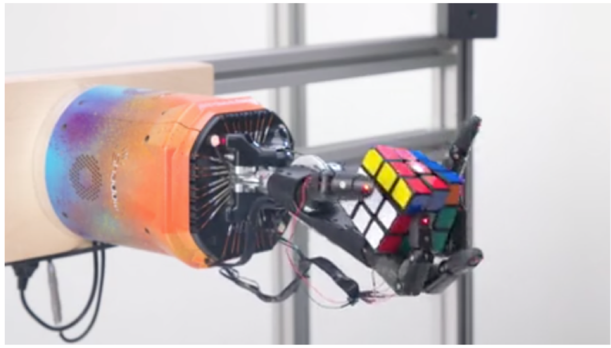
[Open-AI et al., October 2019]

Domain Randomizations

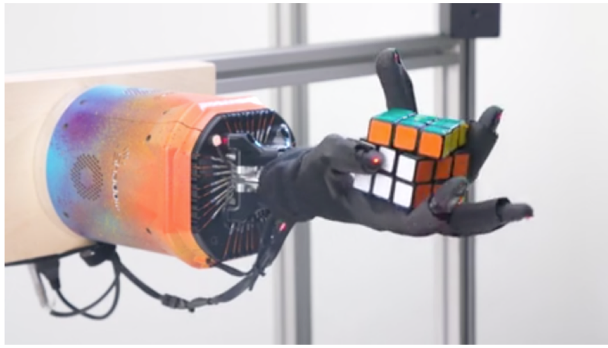
- Simulator physics
- Generique noise
- Custom randomization:
  - Cube and robot friction
  - Cube size
  - Joint and tendon limits, margins
  - Action delay, latency, noise, Motor backlash
  - Gravity
- Vision:
  - Camera position, Rotation, field of view
  - Lighting conditions: rig, intensity
  - Materials
  - Color post processing

[Open-AI et al., October 2019]

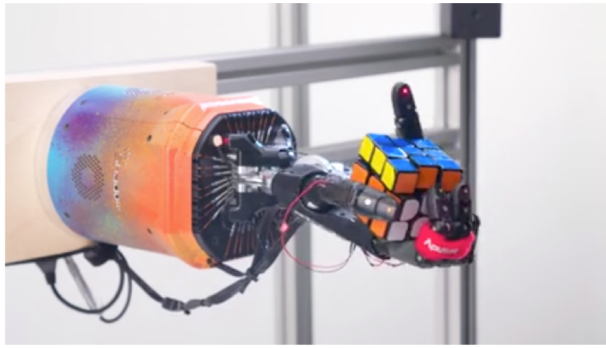
Robustness to unseen perturbations



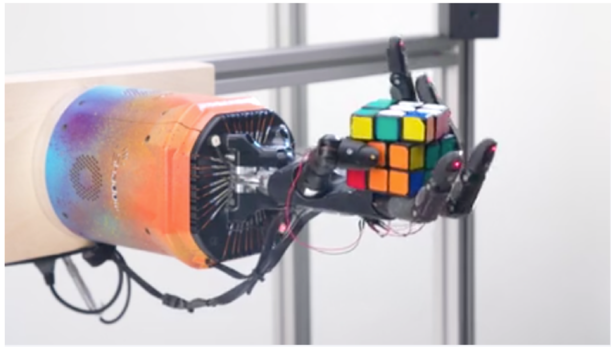
Unperturbed (for reference)



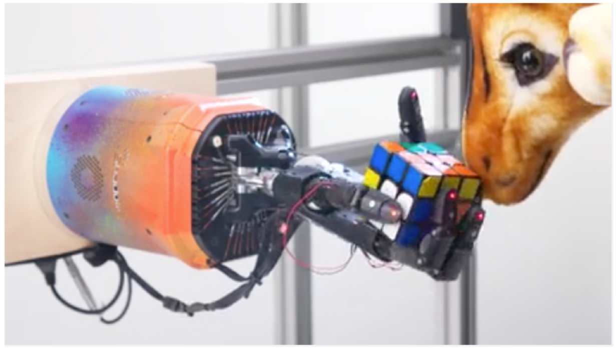
Rubber glove



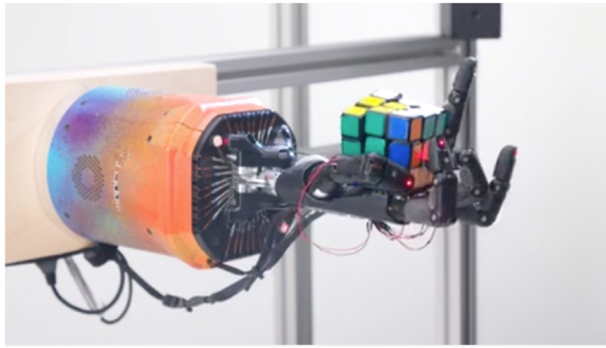
Tied fingers



Blanket occlusion and perturbation



Plush giraffe perturbation



Pen perturbation

[Open-AI et al., October 2019]