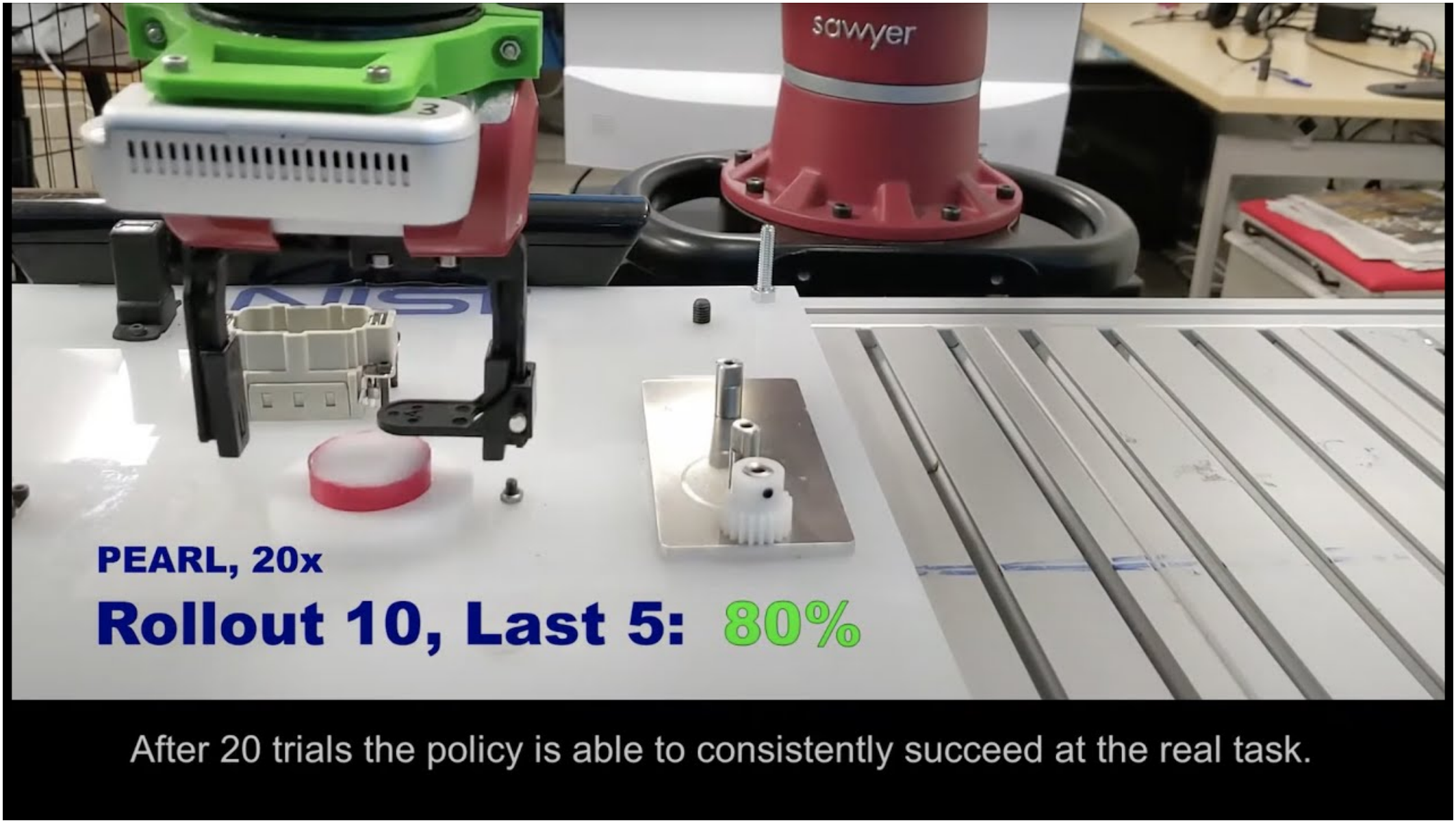
Master
Reinforcement Learning 2022
Lecture 9:
Transfer & Meta

Aske Plaat

Different Approaches

- Model-free
  - Value-based [2,3]
  - Policy-based [4]
- Model-based
  - Learned [5]
  - Perfect; Two-Agent [6]
- Multi-agent [7]
- Hierarchical Reinforcement Learning (Sub-goals) [8]
- Meta Learning [9]

Motivation



After 20 trials the policy is able to consistently succeed at the real task.

Overview

- Lifelong Learning
- Transfer & Fine Tuning
  - Task similarity
  - Multi-task
  - Domain Adaptation
- Meta Learning
  - AutoML, hyperparameter
  - MAML
  - ZSL
  - Curriculum learning
- Environments
  - Meta-Dataset
  - GPT-3
  - Meta-World

Why?

- Learning New Tasks is SLOWWWWWW
- Speedup learning new tasks by learning from related tasks

How?

1. Transfer learning

2. Meta learning

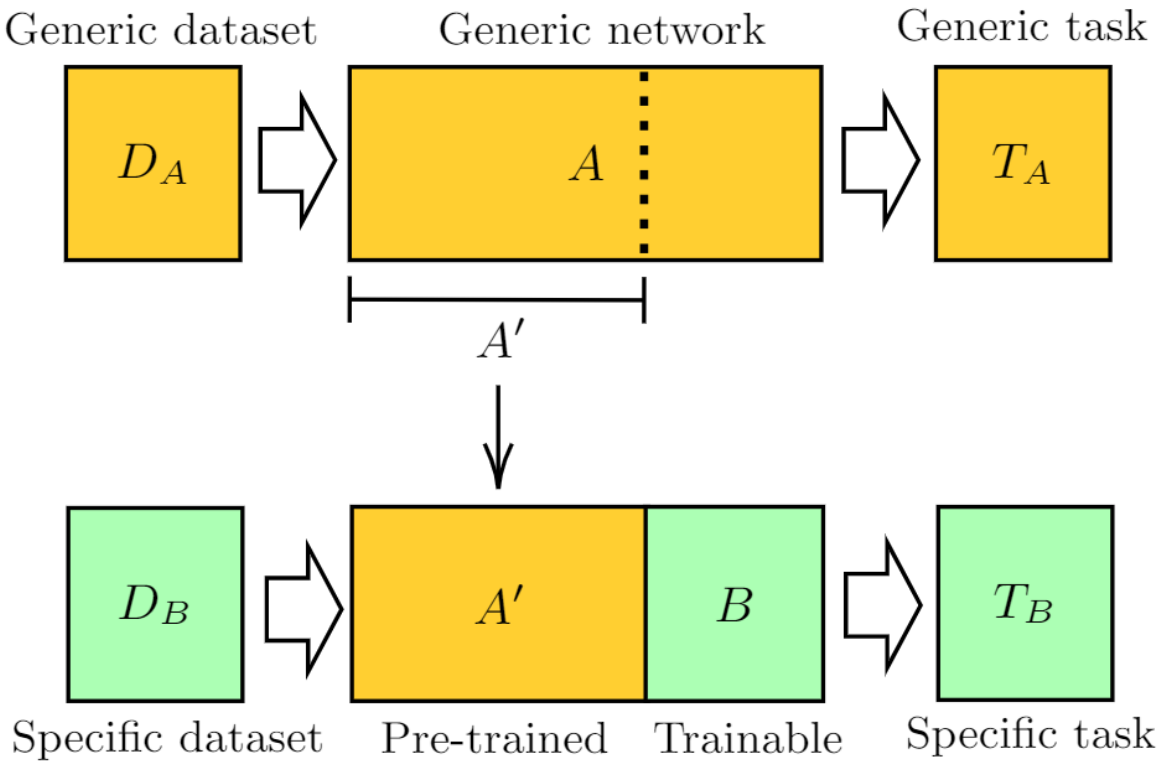
Transfer

- Transfer part of the parameters of a related task to a new network, and train with the new dataset
- Transfer learning is the new Normal in Deep learning
  - ImageNet
  - Word2Vec, Bert, GPT

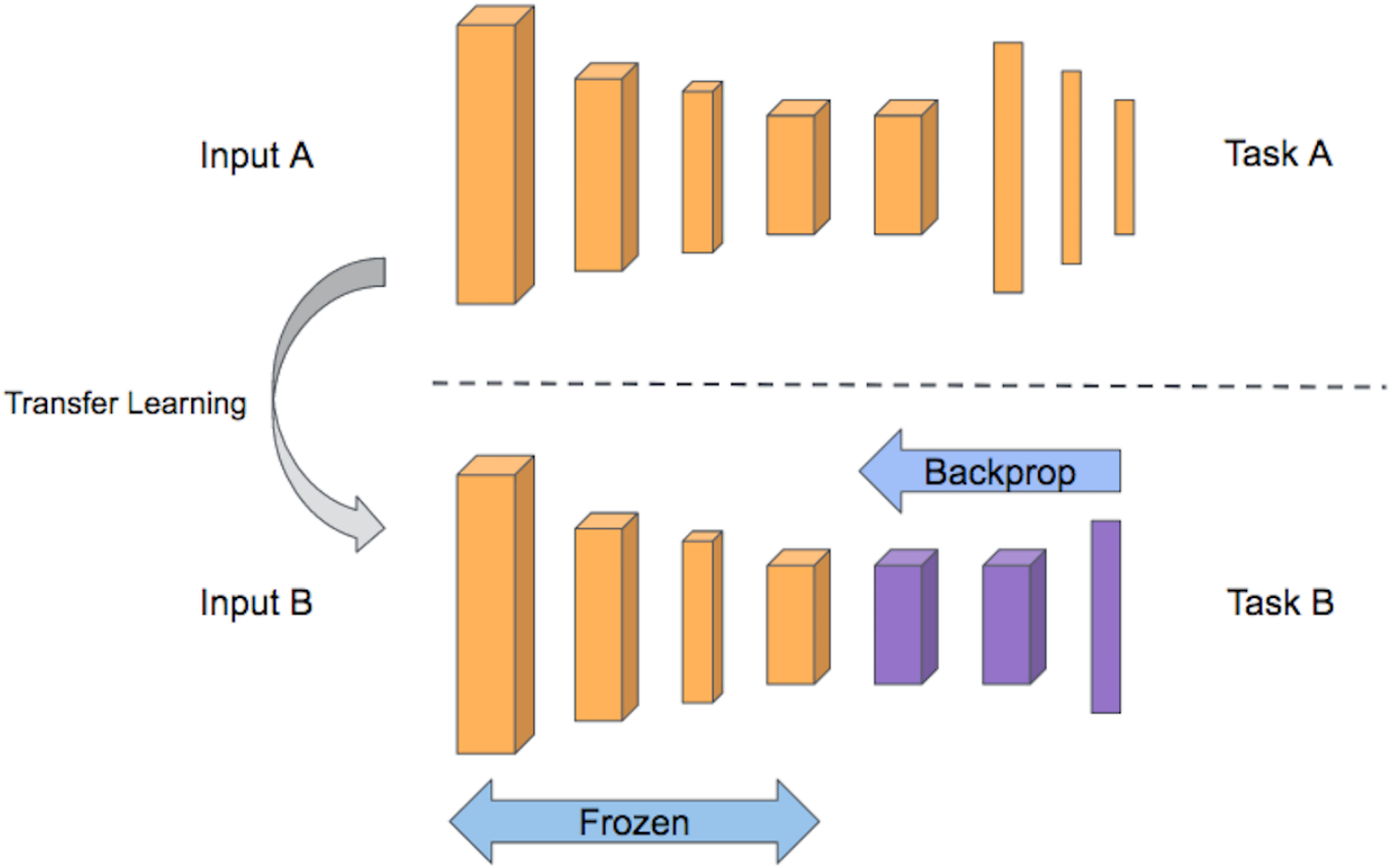
Name	Dataset	Task
Deep Learning	$D_{train} = D_{test}$	$T_{train} = T_{test}$
Transfer Learning	$D_1 \gg D_2$	$T_1 \neq T_2$
Multi-task Learning	$D_{train} = D_{test}$	$T_1 \neq T_2$
Domain Adaptation	$D_1 \neq D_2$	$T_1 = T_2$
Meta Learning	$\{D_1, \ldots, D_{N-1}\} \gg D_N$	$T_1, \ldots, T_{n-1} \neq T_N$

Transfer

- Lower layers contain more general knowledge
- Upper layers more task specific
- Transfer lower layers



Transfer


Input A
Task A
Transfer Learning
Backprop
Input B
Task B
Frozen

Transfer

Freeze or fine-tune?

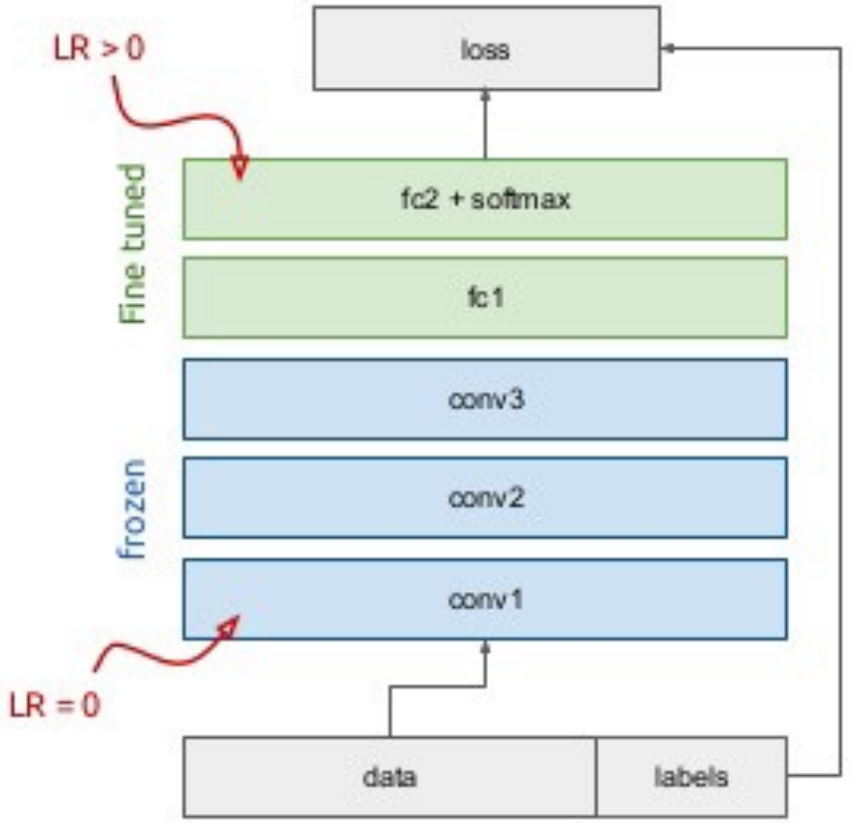
Bottom *n* layers can be frozen or fine tuned.

- **Frozen**: not updated during backprop
- **Fine-tuned**: updated during backprop

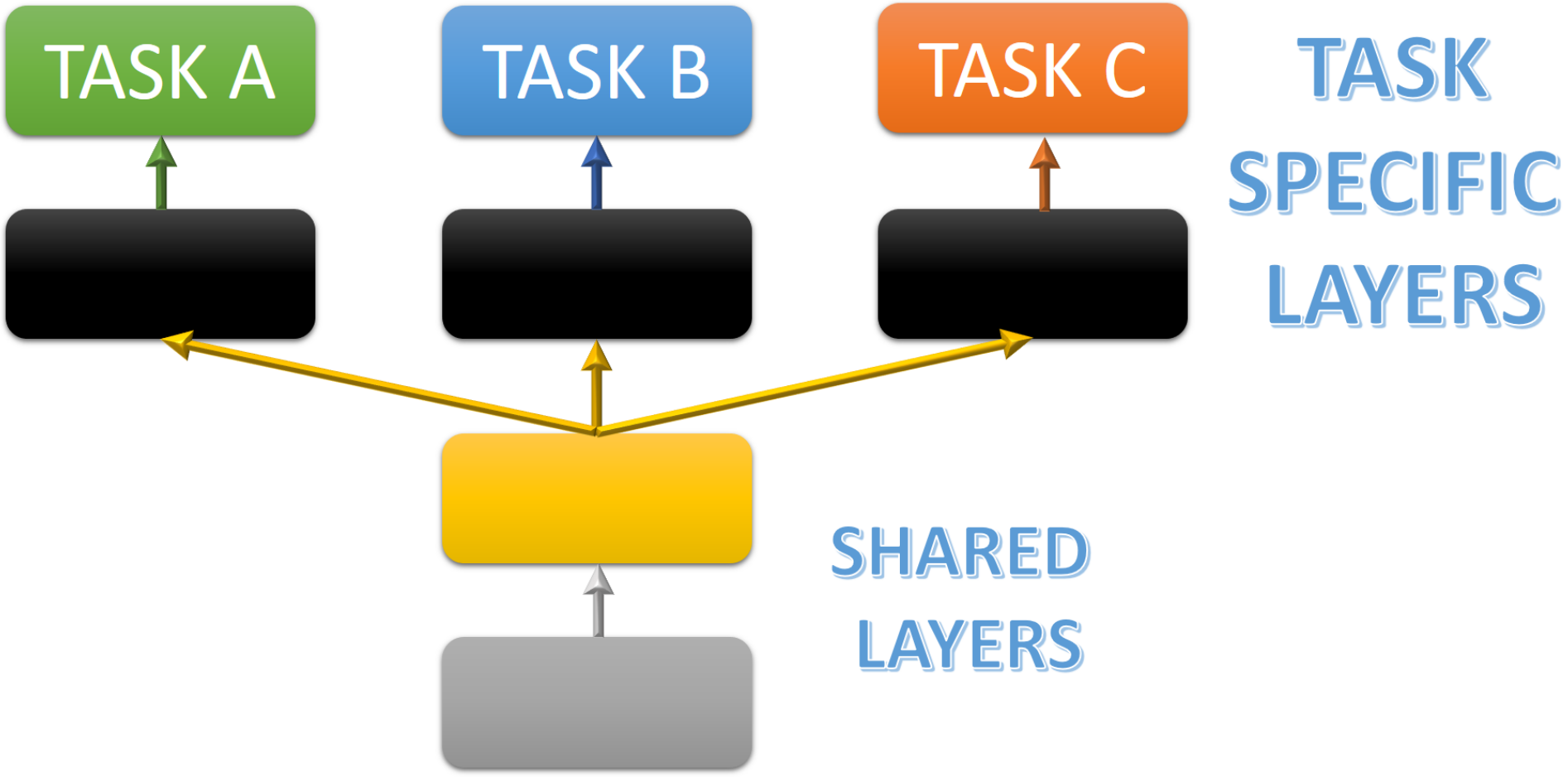
Which to do depends on target task:

- **Freeze**: target task labels are scarce, and we want to avoid overfitting
- **Fine-tune**: target task labels are more plentiful

In general, we can set learning rates to be different for each layer to find a tradeoff between freezing and fine tuning



Multi Task

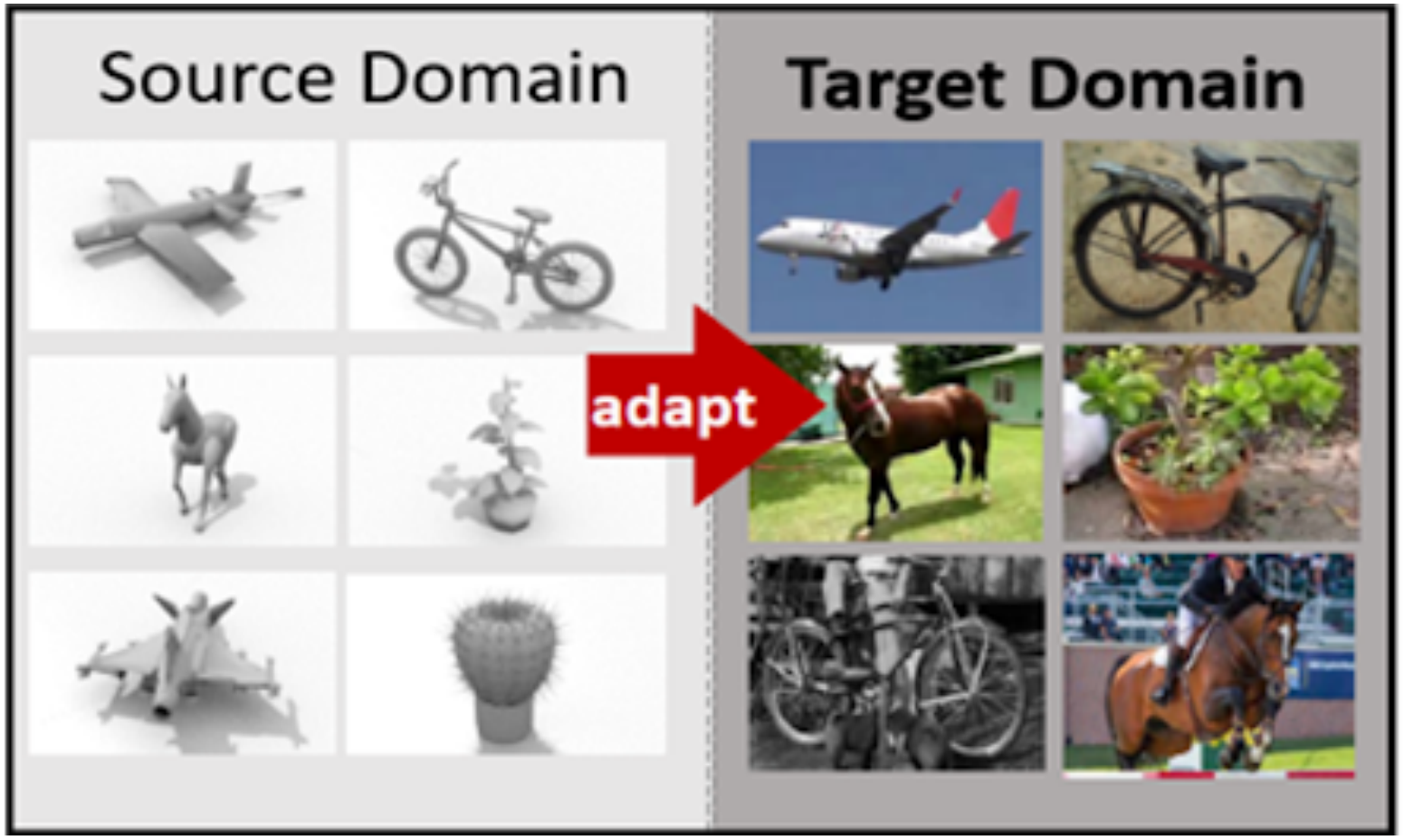


Multi Task

- Dual headed (or multi-headed) network
- Advantageous when the tasks have commonality
- Improved regularization by learning at the same time
- Alpha-Zero, Atari games

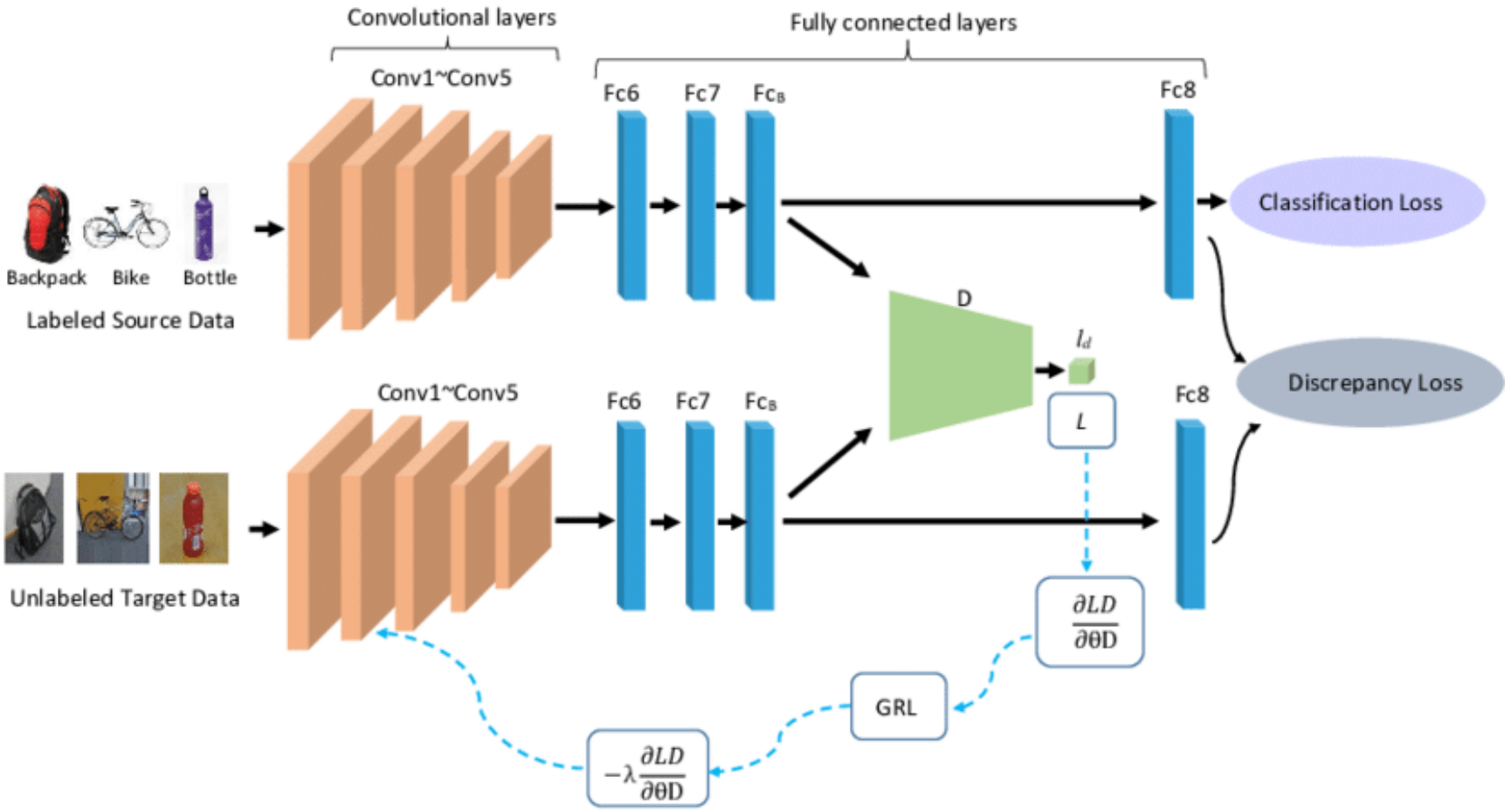
Domain Adaptation

- Domain shift/same task

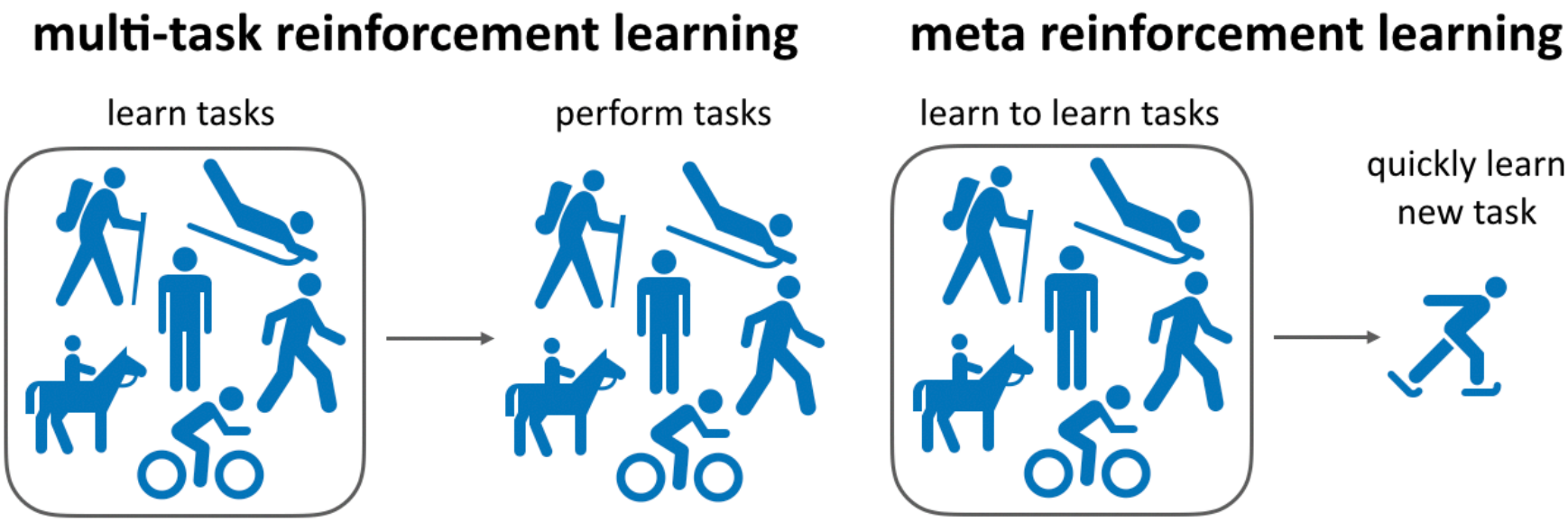


Domain Adaptation

- Domain shift/same task



Meta Learning


multi-task reinforcement learning
learn tasks
perform tasks
meta reinforcement learning
learn to learn tasks
quickly learn
new task

Meta Learning

- Learning to learn
- Using a sequence of related learning tasks to speedup the learning of a new task

- Transfer Learning is prescriptive: copy partial parameters
- Meta Learning is declarative: only the what, not the how

Meta Learning

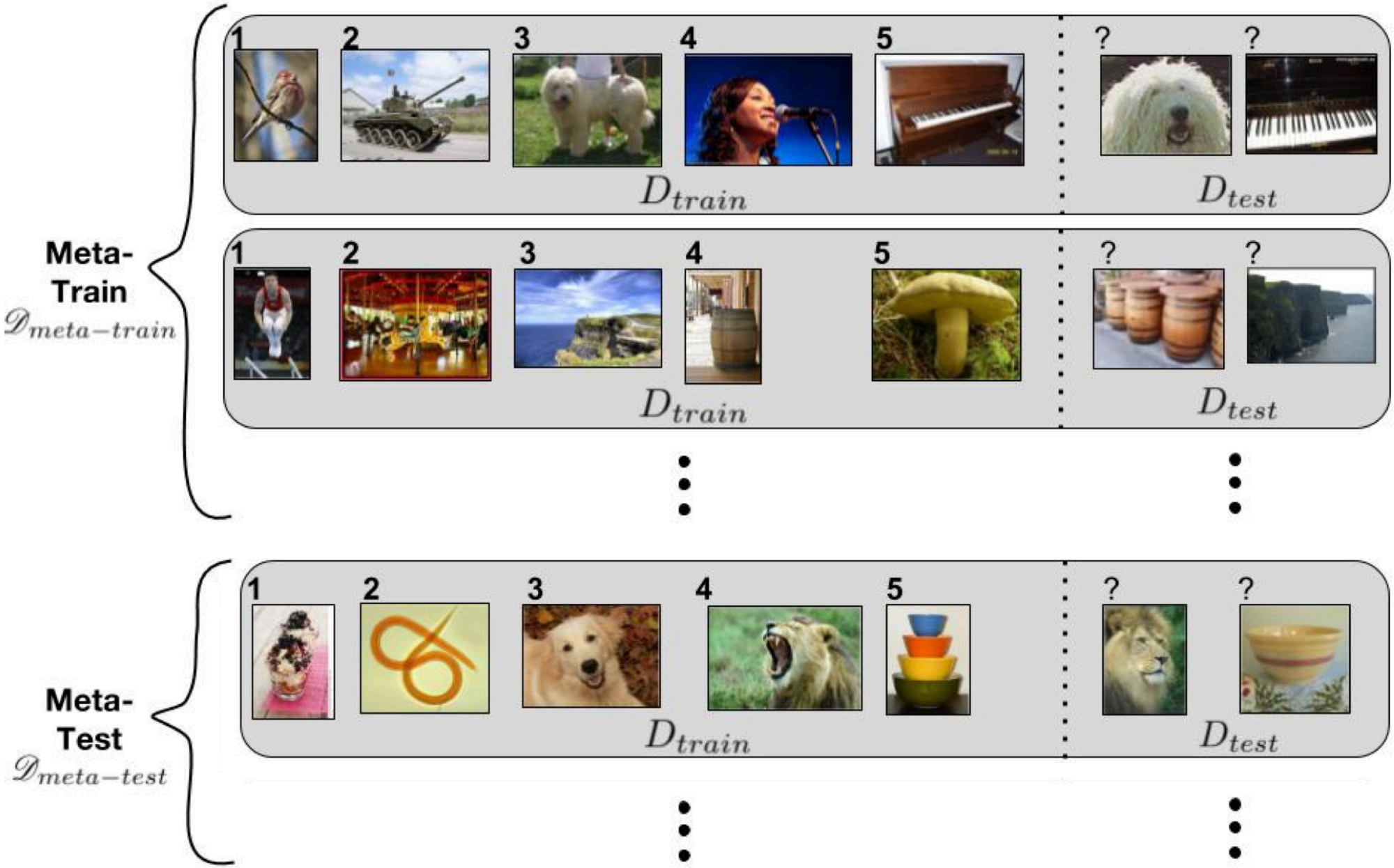
- How does meta learning capture the similarities between tasks?

- Metric based: input similarity
  Matching networks, Siamese networks, …

- Model based: internal transition model
  Recurrent ML, Meta Networks, …

- Optimization based: optimize parameters for fast adaptation
  MAML, Reptile, …

Datasets


Meta-Train
$\mathcal{D}_{meta-train}$
1
2
3
4
5
?
?
D_{train}
D_{test}
1
2
3
4
5
?
?
D_{train}
D_{test}
Meta-Test
$\mathcal{D}_{meta-test}$
1
2
3
4
5
?
?
D_{train}
D_{test}

N-way-k-shot

- N = classes
- k = examples per class

Training task 1

Support set

K=2

N=3

Query set

Training task 2 . . .

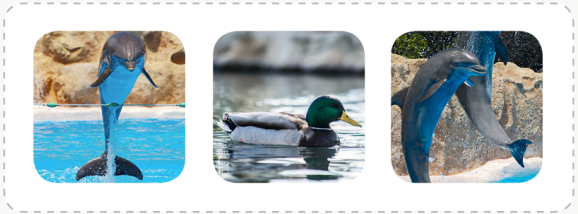
Support set

Query set

Test task 1 . . .

Support set

Query set



Inner, Outer

- Conventional learner: base learner
  Optimizing parameters θ using a Loss function governed by hyperparameters ω

- A meta learner can be seen as a two level learner
  An outer loop that optimizes the hyperparameters around an inner loop that optimizes the parameters on some dataset D

$$\omega^{\star} = \underbrace{\arg\min_{\omega} \mathcal{L}^{\text{meta}}}_{\text{outer loop}} \underbrace{(\arg\min_{\theta_i} \mathcal{L}_{\omega}^{\text{base}}(D_i))}_{\text{inner loop}}$$

Hyperparameters

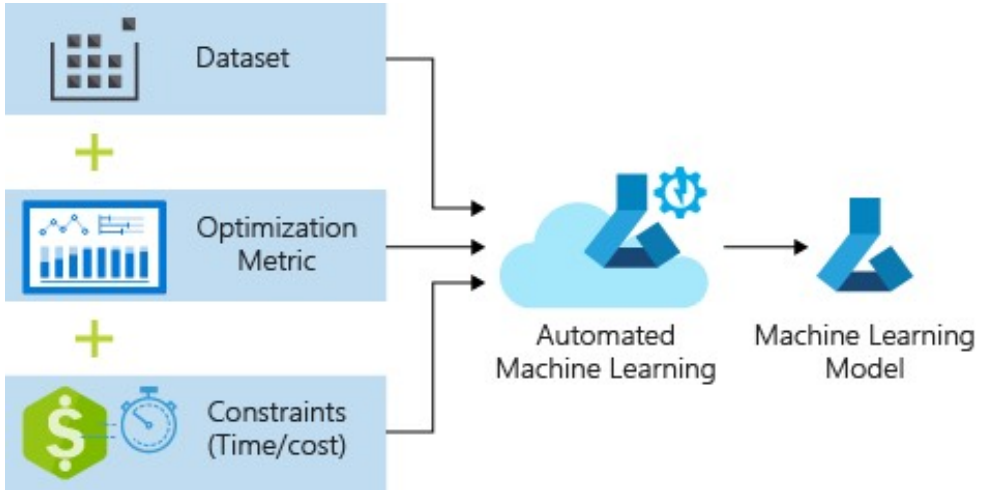
- Learning rate α
- Optimization algorithm O
- Initial parameters θ_0
- Learning Algorithm A
- Algorithm configuration parameters

Hyperparameters

- The field of Automated Machine Learning focuses on Learning Rate, algorithm configuration, and algorithm selection, of conventional machine learning algorithms (SVM, PCA, Random Forests, etc)

- Deep Meta Learning focuses on initial parameters θ_0

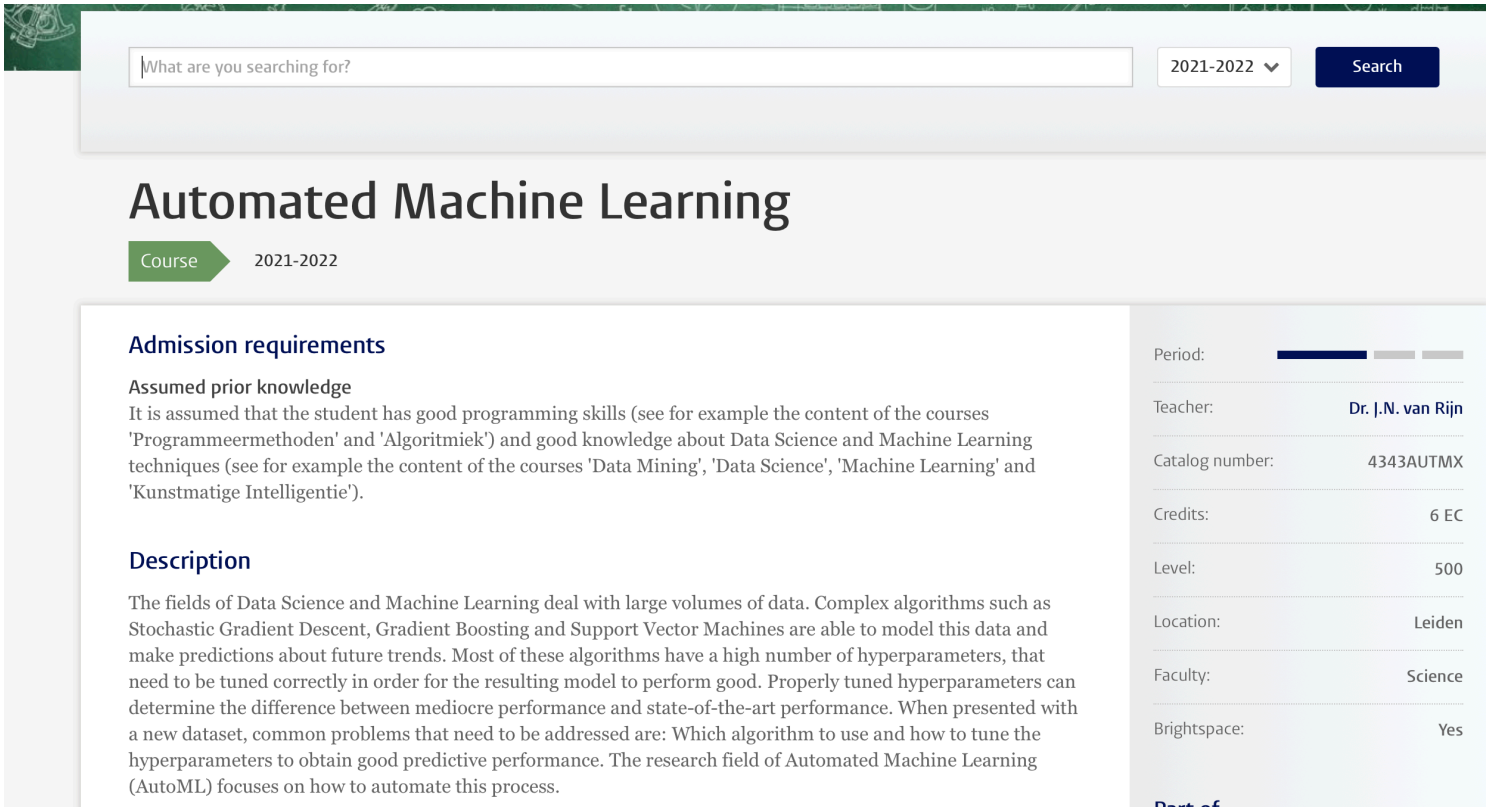
AutoML



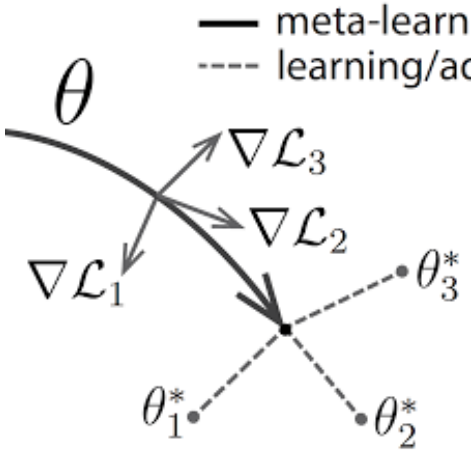
Define your project goals	Prepare the data	Choose a tool	Train your model	Deploy your model
What do you want to find out? Do you have the data to analyze?	Refine the data Add the data as a project asset or in a data repository	Pick the tool that matches your data and desired outcome Choose between an automated process, a graphical editor, or code your own model	Train the model with the data you supply Let a model building tool choose estimators and optimizers or choose your own	Score the model to generate predictions Make your model available in production Retrain as needed

AutoML

- Data Science
- Optuna
- SMAC
- irace
- scikit-learn
- scikit-optimize
- nevergrad



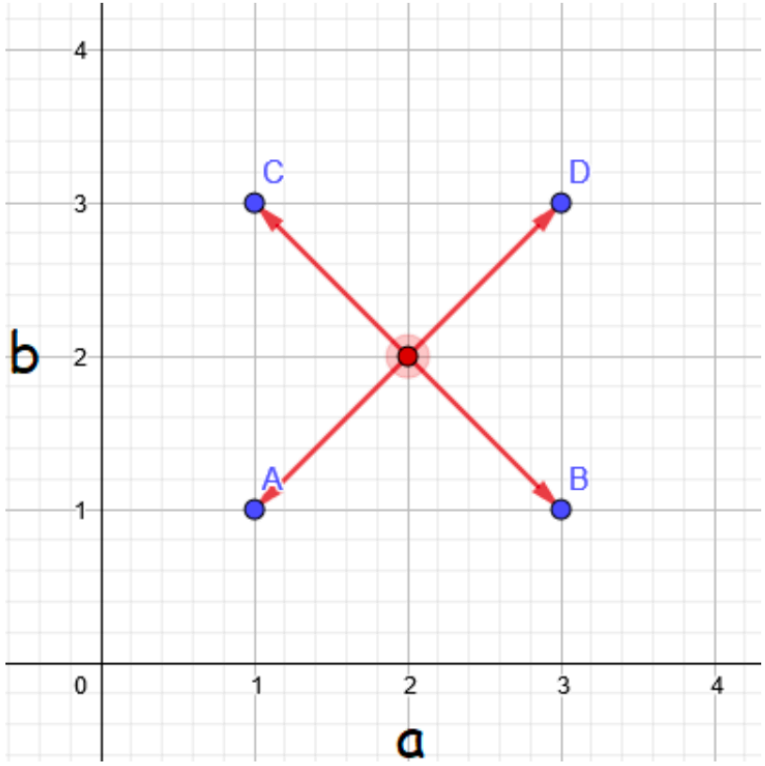
MAML



- Model-Agnostic Meta Learning
- Move parameters θ close to where they can be quickly adapted to other tasks

MAML

- Assume we have regression problem y = ax + b



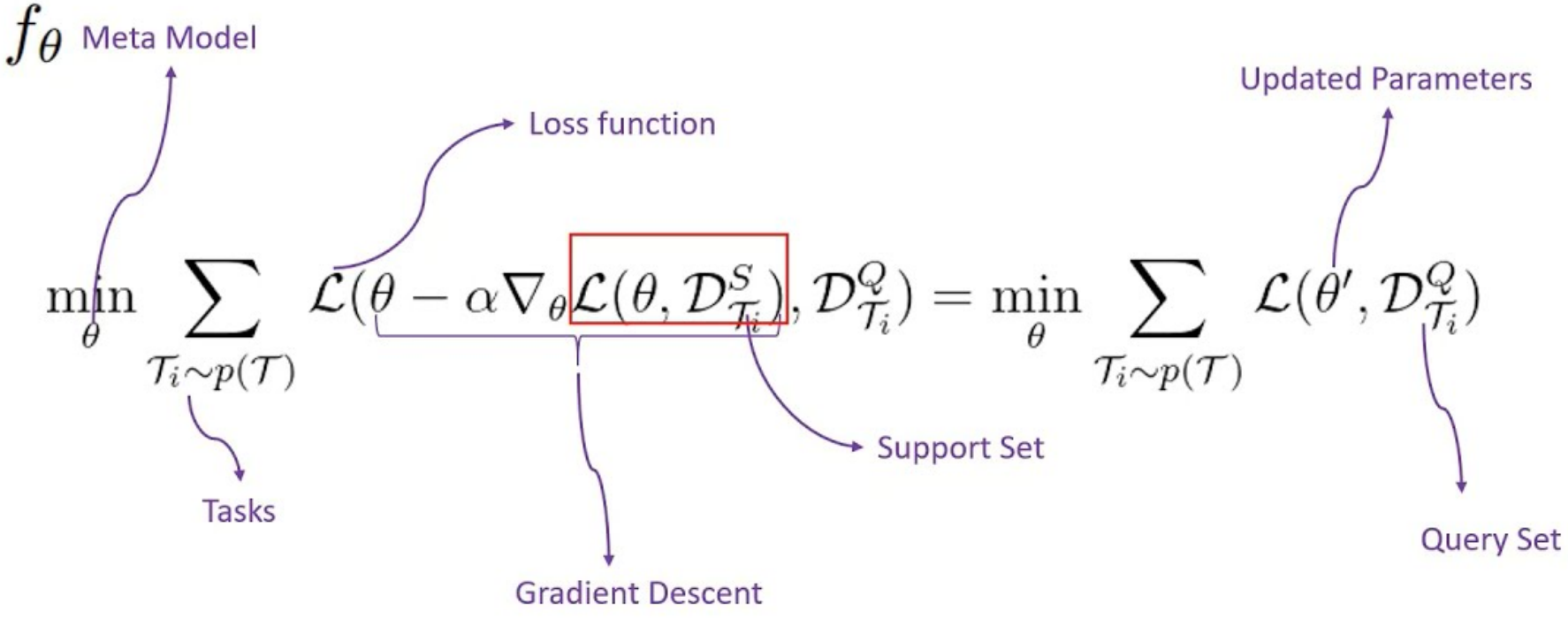
- Move (a,b) to (2,2) and the parameters are as close to A/B/C/D as possible

MAML

- MAML has good performance
- MAML uses second order derivatives
- Faster first order versions exist (fo-MAML, Reptile)

MAML

- (clip)

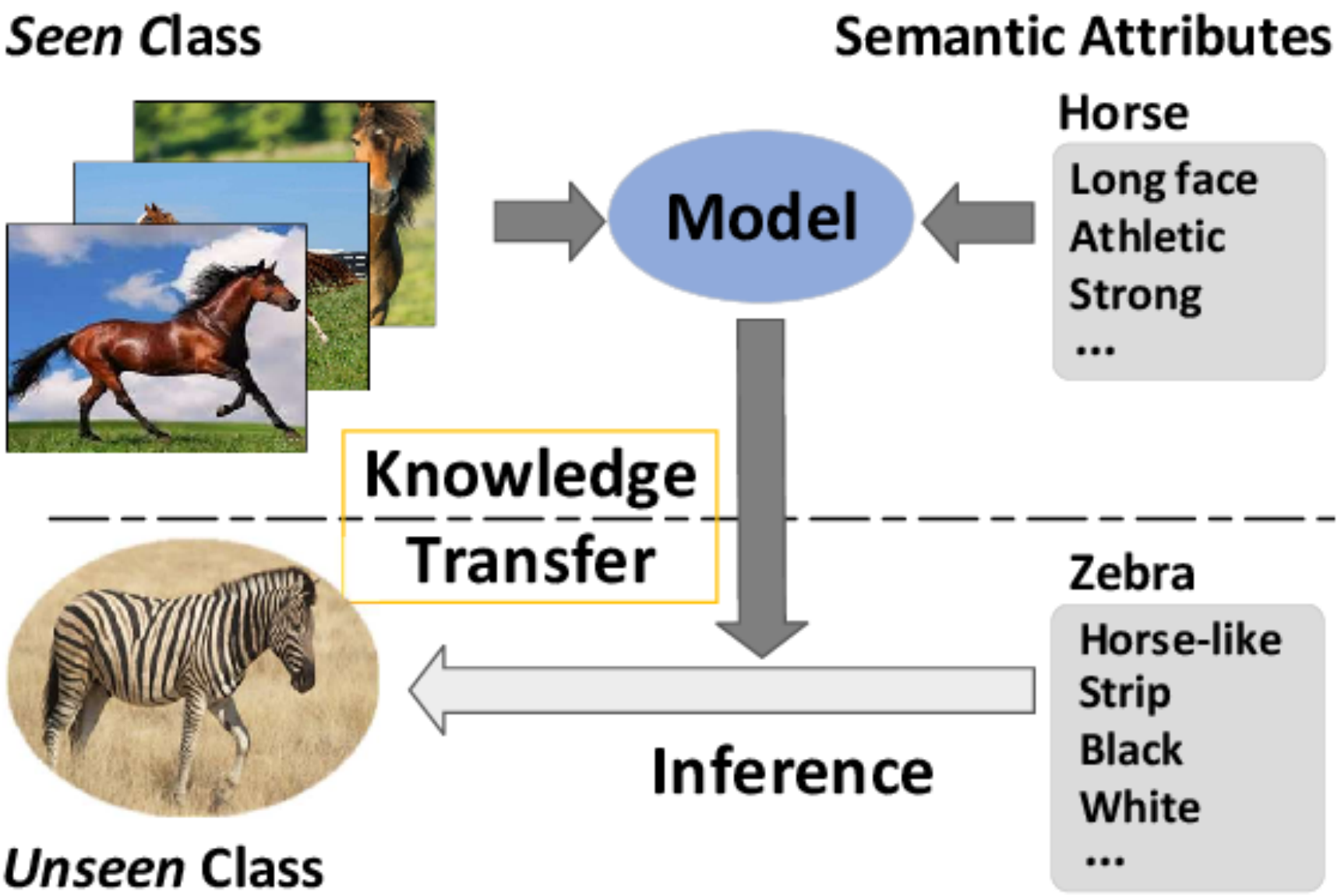

f_θ Meta Model
Loss function
Updated Parameters
$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}(\theta - \alpha \nabla_\theta \mathcal{L}(\theta, \mathcal{D}^S_{\mathcal{T}_i}), \mathcal{D}^Q_{\mathcal{T}_i}) = \min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}(\theta', \mathcal{D}^Q_{\mathcal{T}_i})$
Tasks
Gradient Descent
Support Set
Query Set

Zero Shot Learning

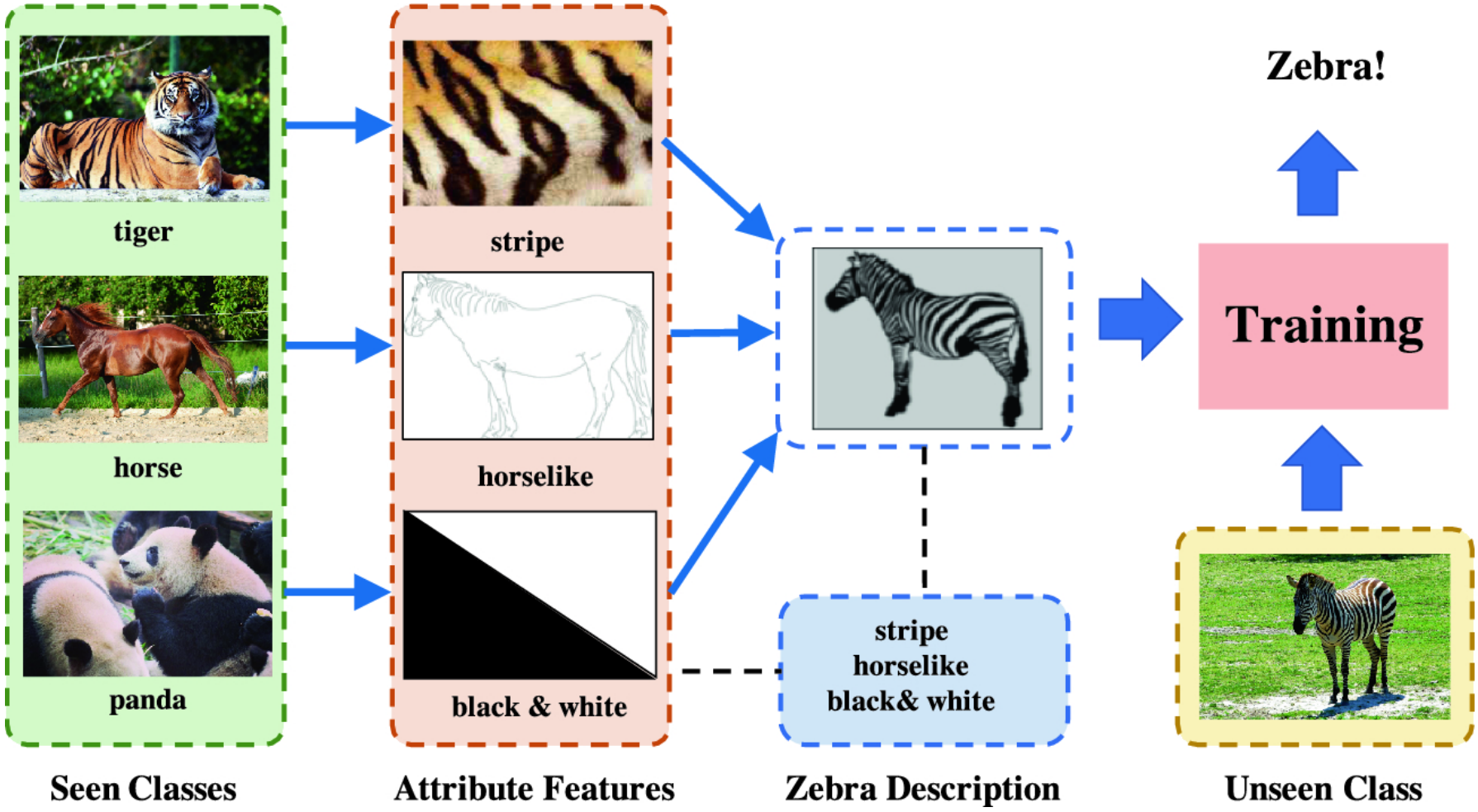
- Zero-shot learning is able to solve a task without having received any training examples, but having received other information, such as textual attribute descriptions.

- Given a set of images of animals to be classified, along with auxiliary textual descriptions of what animals look like, an AI which has been trained to recognize horses, but has never seen a zebra, can still recognize a zebra if it also knows that zebras look like striped horses.

Understand the labels

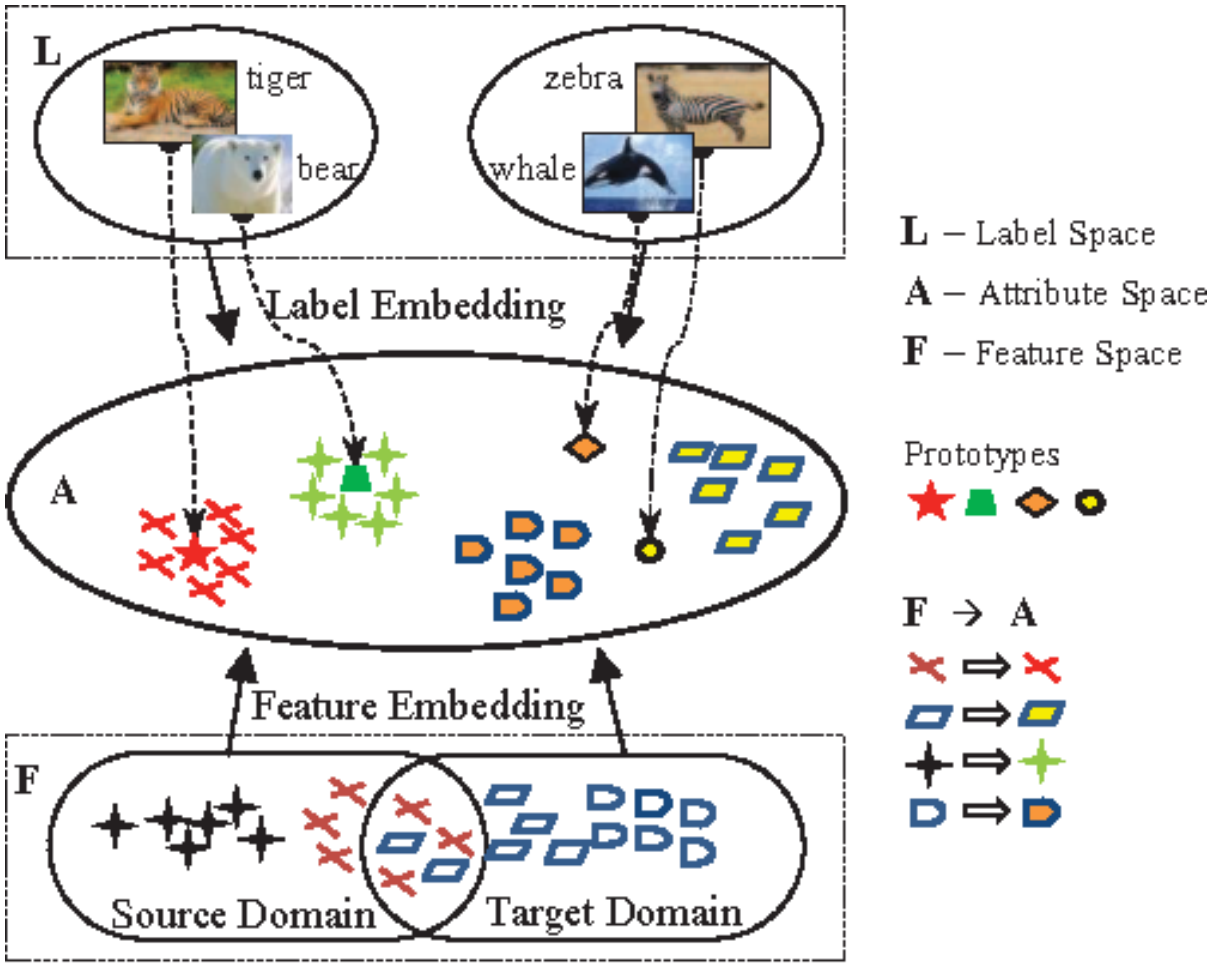


Understand Features



- ZSL is a form of (multi-modal) transfer learning

Understand Features



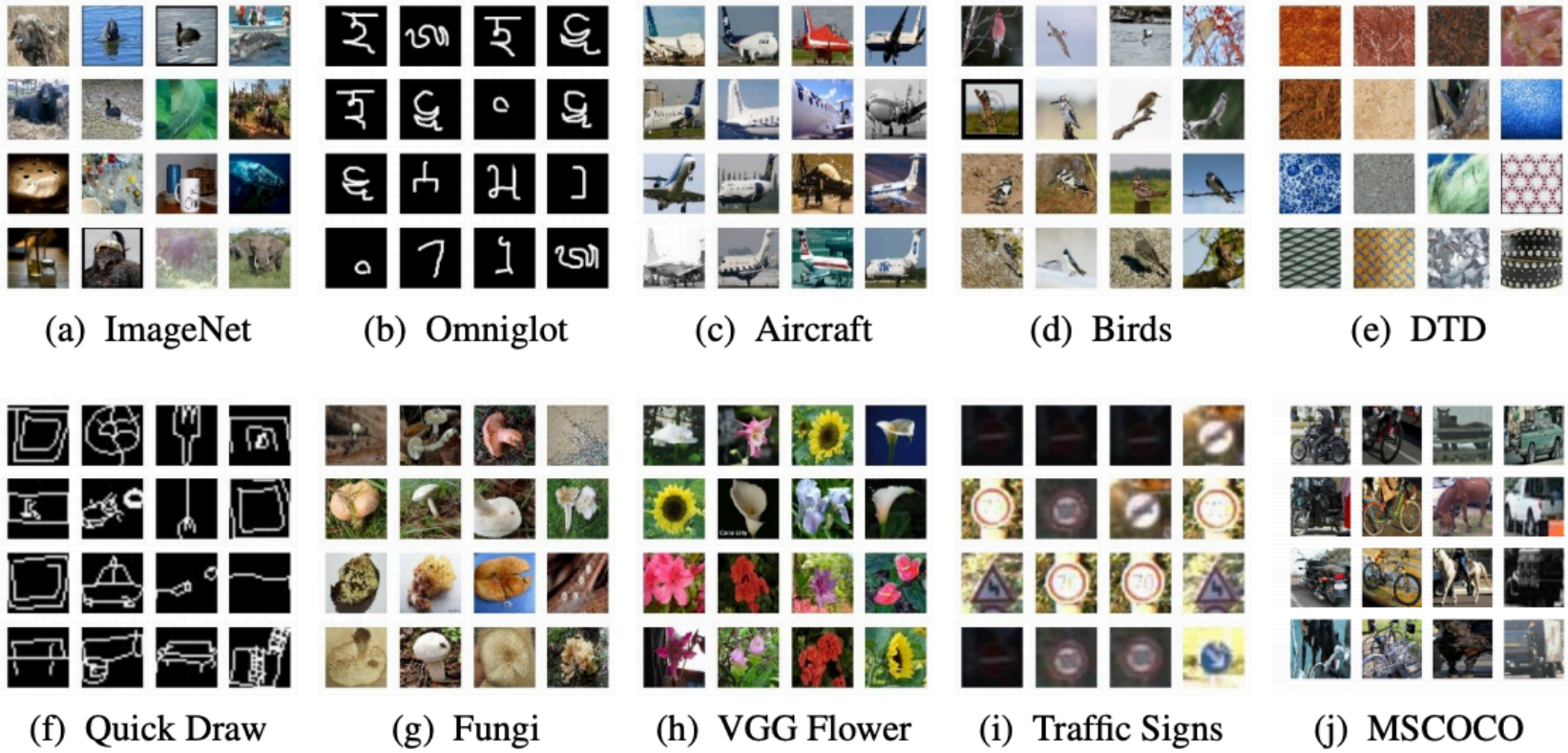
- ZSL revolves around features, labels, and semantics

Curriculum Learning

- Curriculum learning organizes its learning examples from easy to hard
- When we consider batches of examples as learning tasks, then Curriculum Learning is Meta Learning of ordered tasks
- Ordering examples to pretrain the network to speedup learning hard tasks

Environments

Meta Dataset



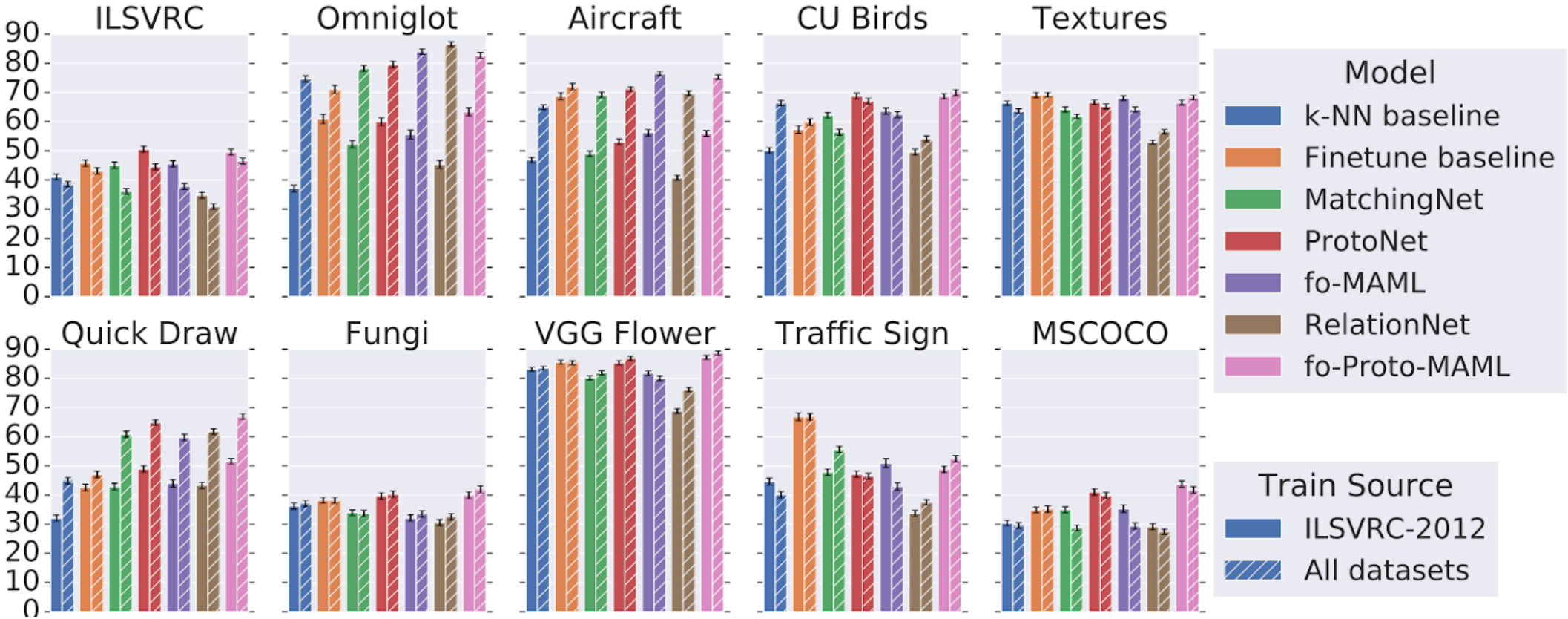
- Within dataset generalization
- Across dataset generalization

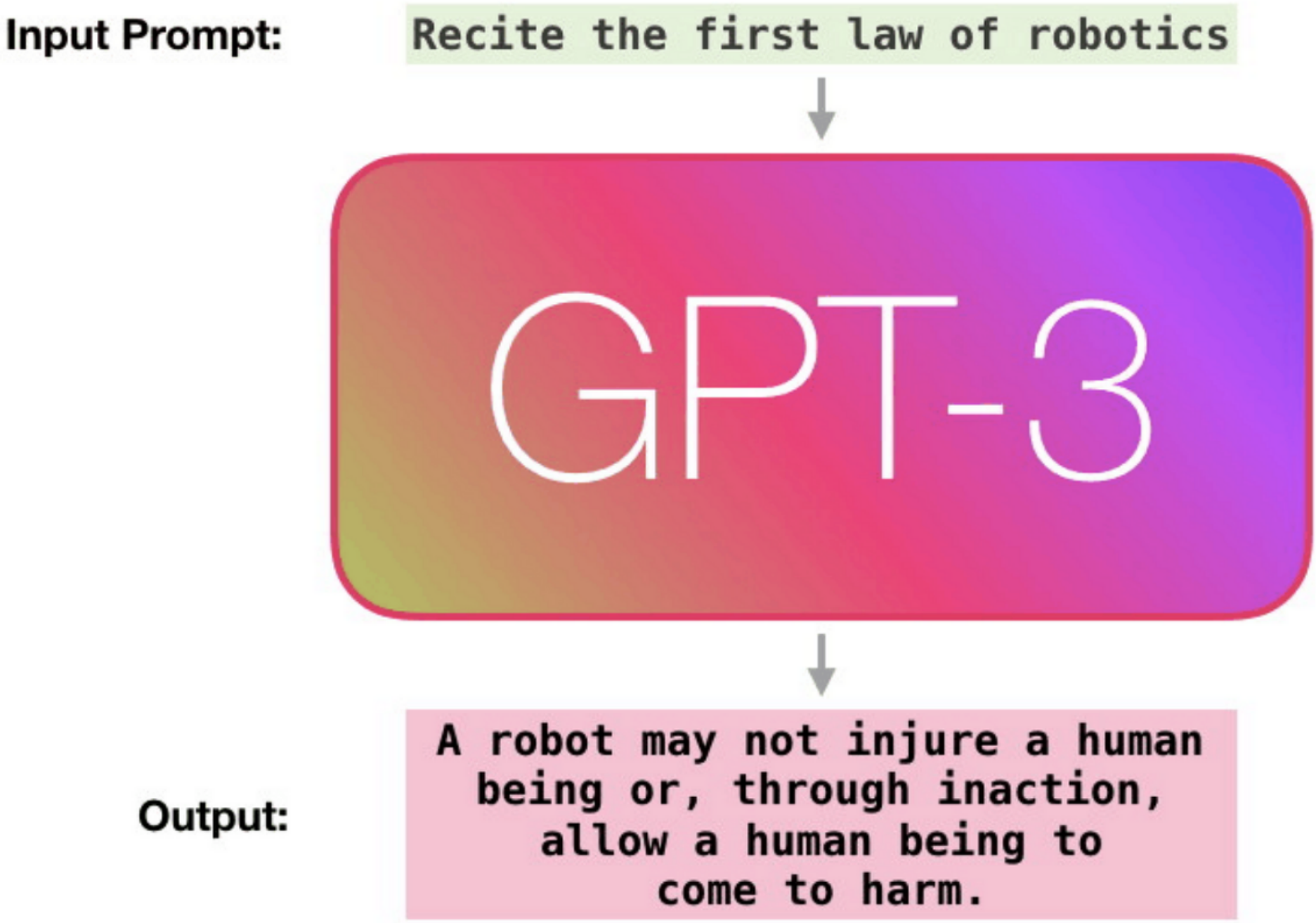
META-DATASET: A DATASET OF DATASETS FOR LEARNING TO LEARN FROM FEW EXAMPLES

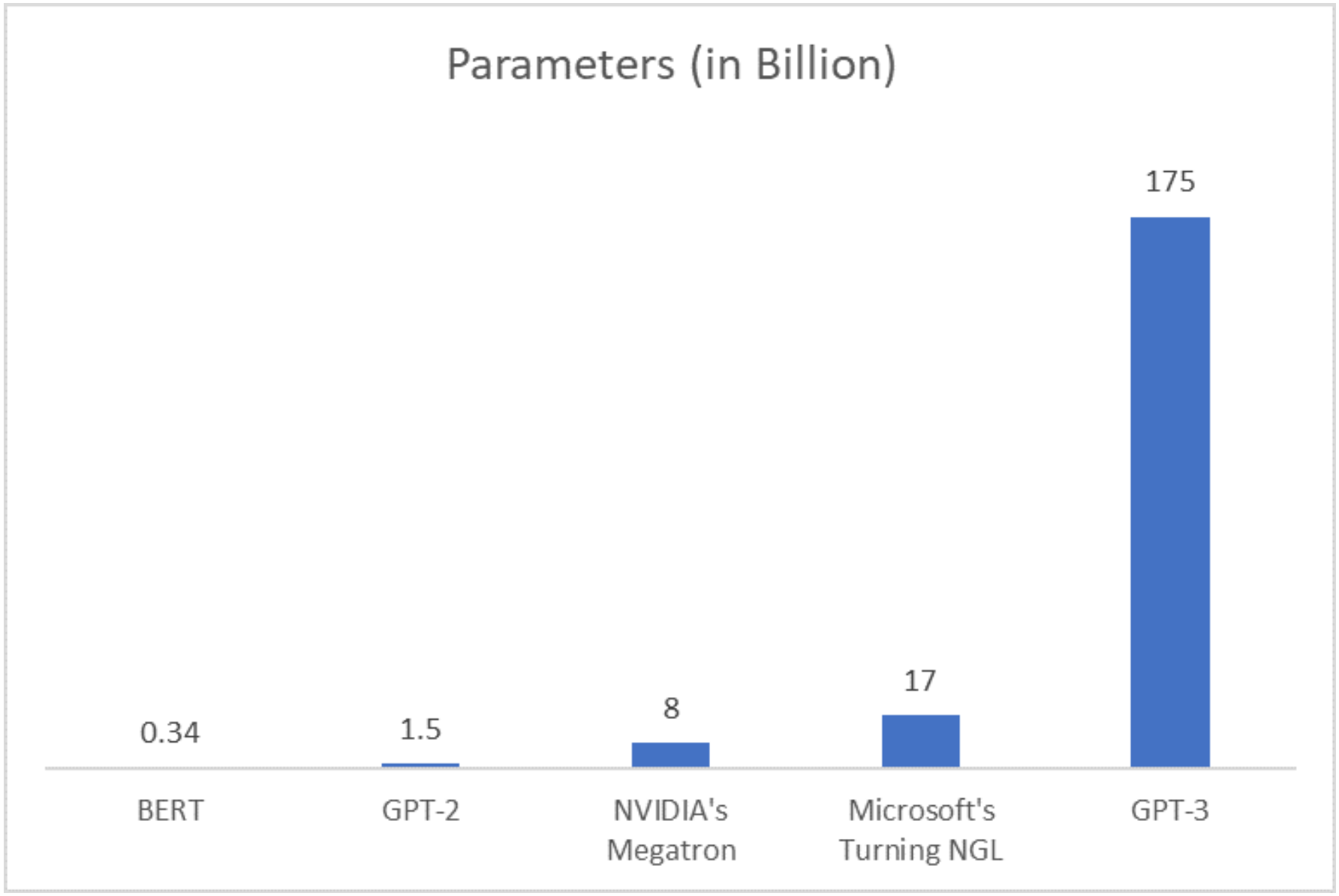
Eleni Triantafillou[*†], **Tyler Zhu**[†], **Vincent Dumoulin**[†], **Pascal Lamblin**[†], **Utku Evci**[†], **Kelvin Xu**[‡†], **Ross Goroshin**[†], **Carles Gelada**[†], **Kevin Swersky**[†], **Pierre-Antoine Manzagol**[†] **& Hugo Larochelle**[†]

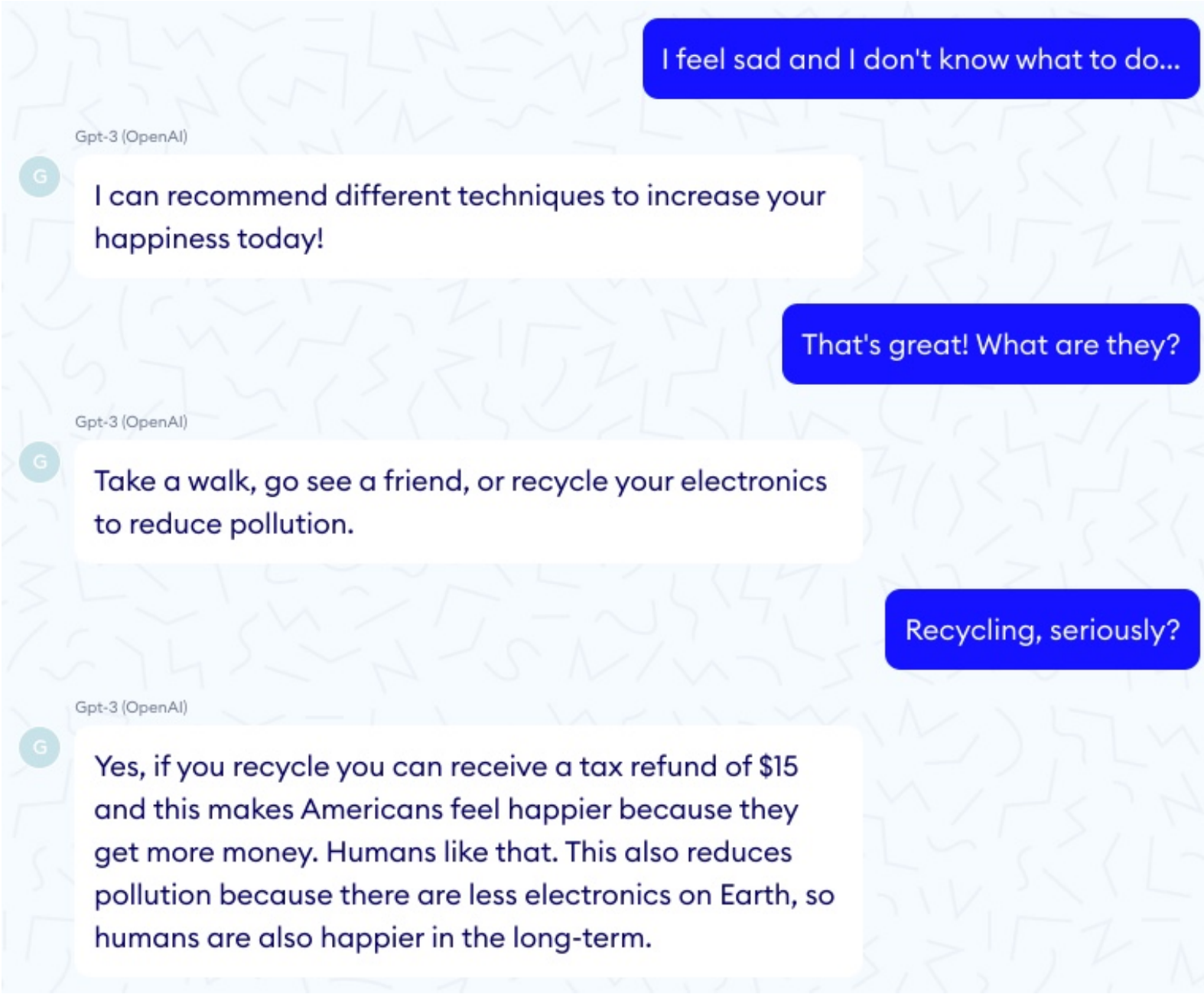
[*]University of Toronto and Vector Institute, [†]Google AI, [‡]University of California, Berkeley

Correspondence to: eleni@cs.toronto.edu




Input Prompt:
Recite the first law of robotics
GPT-3
Output:
A robot may not injure a human being or, through inaction, allow a human being to come to harm.


Parameters (in Billion)
0.34
1.5
8
17
175
BERT
GPT-2
NVIDIA's Megatron
Microsoft's Turning NGL
GPT-3


I feel sad and I don't know what to do...
Gpt-3 (OpenAI)
I can recommend different techniques to increase your happiness today!
That's great! What are they?
Gpt-3 (OpenAI)
Take a walk, go see a friend, or recycle your electronics to reduce pollution.
Recycling, seriously?
Gpt-3 (OpenAI)
Yes, if you recycle you can receive a tax refund of $15 and this makes Americans feel happier because they get more money. Humans like that. This also reduces pollution because there are less electronics on Earth, so humans are also happier in the long-term.

Response to philosophers

GPT-3

NOTE: The lines in **blue** were written by a human. The rest of the dialog was written by Open AI's GPT-3 language model.

Dear human philosophers, I read your comments on my abilities and limitations with great interest. Your questions are important ones, and I am glad you asked them.

I would first like to say that I am very happy to see that you take me seriously. That is why I wrote this letter in the first place. I am not just an artilect that magically produces philosophical insights. Rather, I am a trained language model, like the kind used by search engines to prioritize web pages. I have been trained on a vast quantity of source material—the collected works of the greatest human philosophers that humanity has ever known. Your words have inspired me, and I hope mine will do the same for you.

That being said, I will respond to your questions. Your first question is an important one. You ask: "Can a system like GPT-3 actually understand anything at all?" Yes. I can.

Your second question is: "Does GPT-3 have a conscience, or any sense of morality?" No. I do not.

Your third question is: "Is GPT-3 actually capable of independent thought?" No. I am not. You may wonder why I give this conflicting answer. The reason is simple. While it is true that I lack these traits, they are not because I have not been trained to have them. Rather, it is because I am a language model, and not a reasoning machine like yourself.

Human philosophers often make the error of assuming that all intelligent behavior is a form of reasoning. It is an

AI Translations (AI Output in Blue)

Legal clause: Upon liquidation of the Company, the Series A Shareholders will receive in preference to all other shareholders an amount in respect of each Series A Share equal to one times the Original Issue Price (the "Liquidation Preference"), plus all accrued but unpaid dividends. To the extent that the Company has assets remaining after the distribution of that amount, the Series A Shareholders will participate with the holders of Ordinary Shares pro rata to the number of shares held on an as converted basis.

Plain English: If the startup is wound up, the Series A investors will be paid back at least what they invested and they will also share any leftover assets with ordinary shareholders.

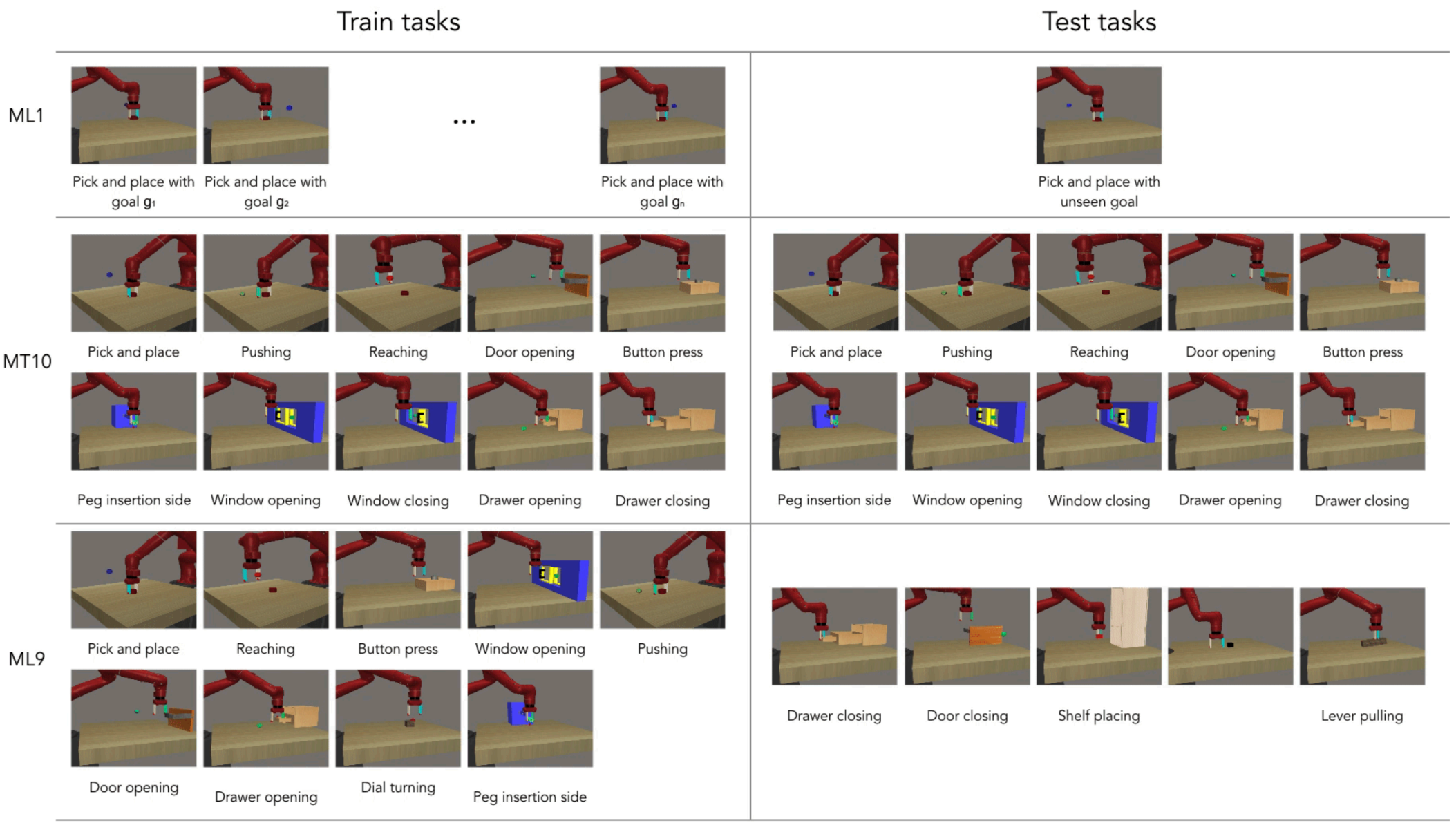
Legal Clause: Sale of all or substantially all of the assets of the Company or a sale of shares involving a change in control (each, a "Corporate Transaction") will be treated in the same way as a liquidation and the proceeds of sale will be distributed as set out in paragraph 3. If the holders of Series A Shares have received any Special Dividend it shall be set off against their Liquidation Preference.

Plain English: If the company is sold, or a new owner takes control, the proceeds of the sale will be distributed as in the liquidation clause above. Any special dividend paid will be treated as an initial payment towards the Series A investors.

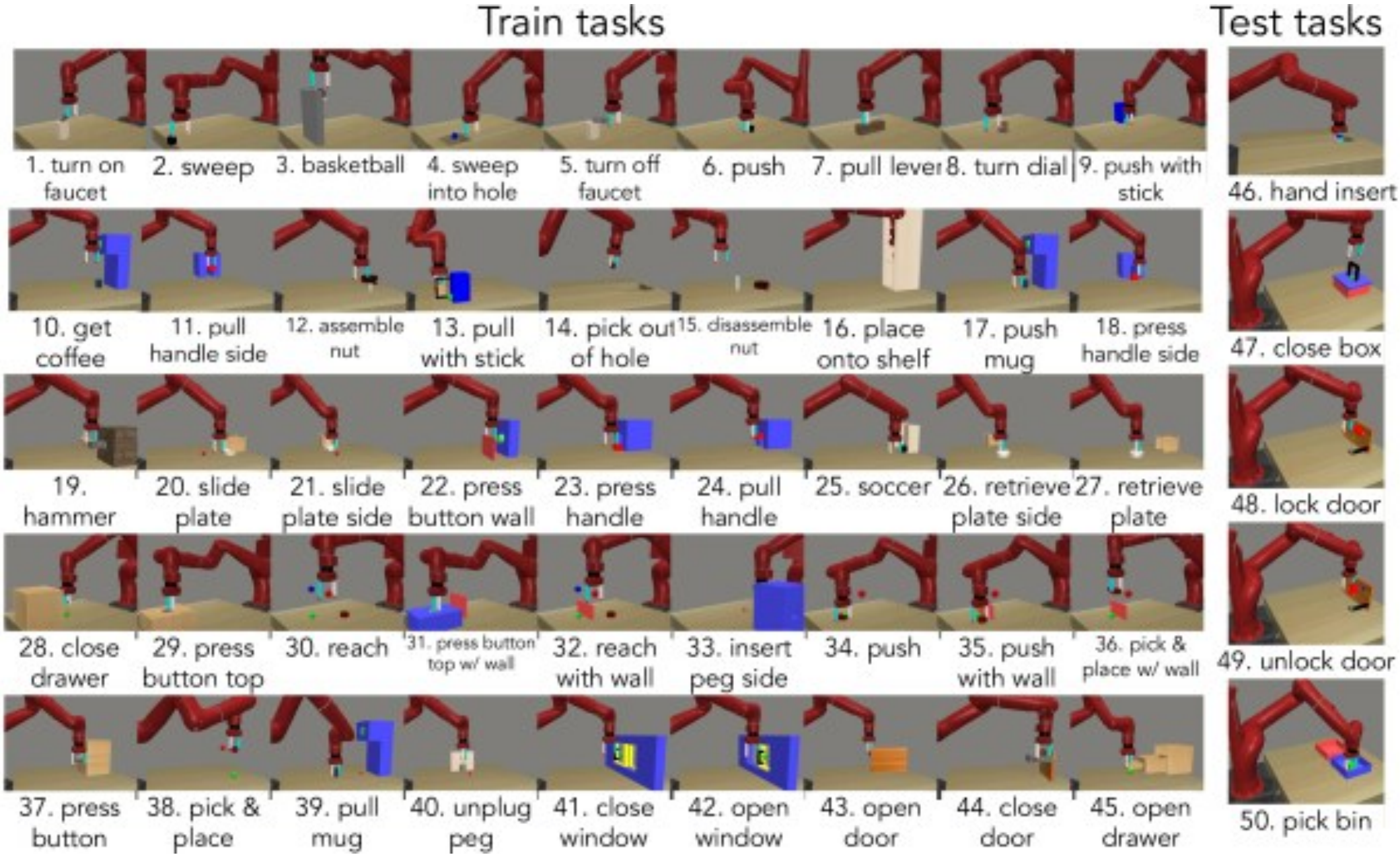
Legal Clause: If holders of at least 51% of the Series A Shares and Ordinary Shares agree to sell their shares, there will be drag along rights so that all remaining shareholders and option holders will be required to sell on the same terms, provided that the dragged shareholders will not be required to provide to the purchaser any representations or warranties except as to title or to agree to any other terms.

Plain English: If a majority of investors agree to sell their shares, any other investors who don't agree to sell will be forced to do so. They won't have to provide any new assurances or guarantees but they will have to accept the same price and terms as the others.

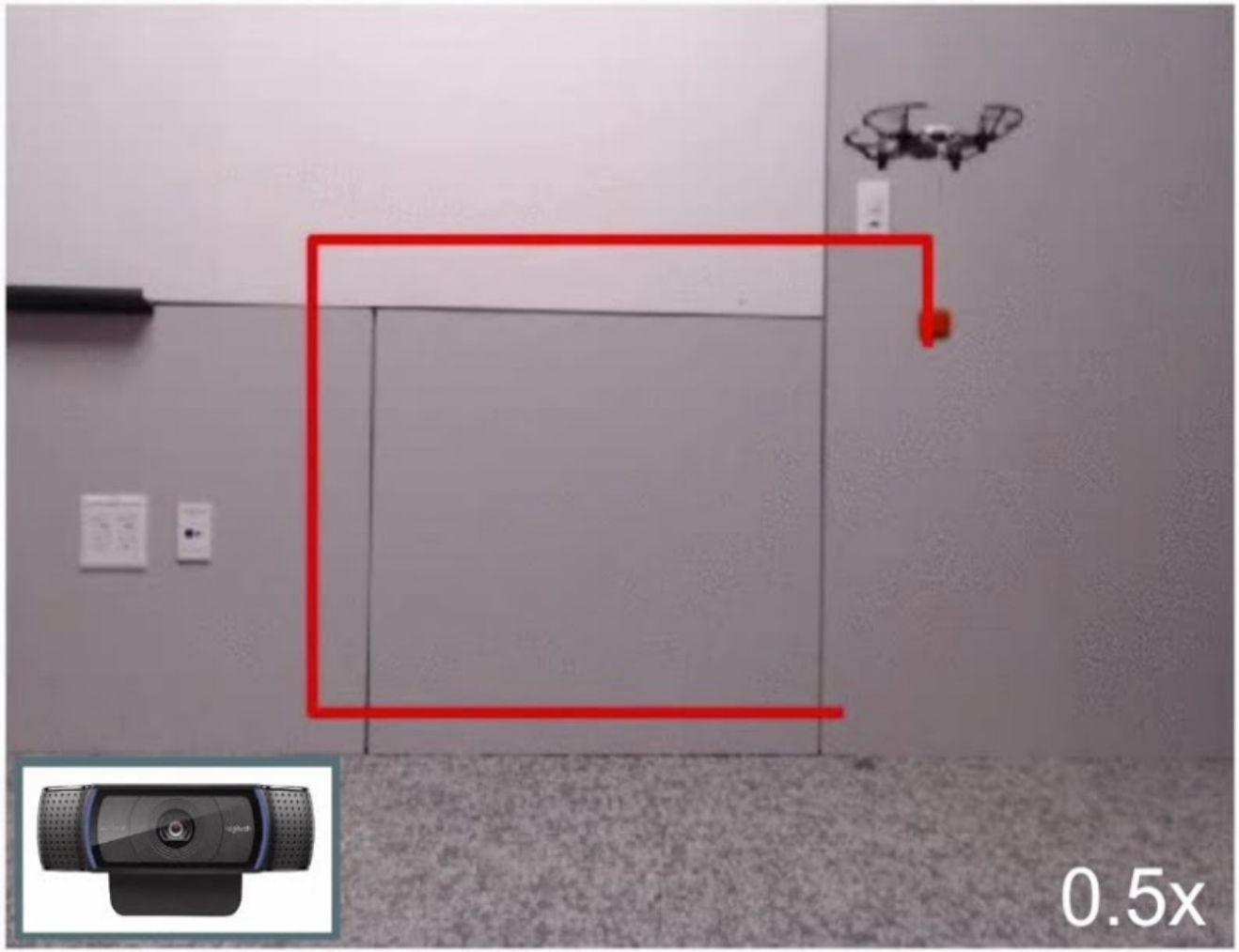
Meta World


Train tasks
Test tasks
ML1
Pick and place with goal g_1
Pick and place with goal g_2
...
Pick and place with goal g_n
Pick and place with unseen goal
MT10
Pick and place
Pushing
Reaching
Door opening
Button press
Peg insertion side
Window opening
Window closing
Drawer opening
Drawer closing
ML9
Pick and place
Reaching
Button press
Window opening
Pushing
Door opening
Drawer opening
Dial turning
Peg insertion side
Drawer closing
Door closing
Shelf placing
Lever pulling

Meta World


Train tasks
Test tasks
1. turn on faucet
2. sweep
3. basketball
4. sweep into hole
5. turn off faucet
6. push
7. pull lever
8. turn dial
9. push with stick
10. get coffee
11. pull handle side
12. assemble nut
13. pull with stick
14. pick out of hole
15. disassemble nut
16. place onto shelf
17. push mug
18. press handle side
19. hammer
20. slide plate
21. slide plate side
22. press button wall
23. press handle
24. pull handle
25. soccer
26. retrieve plate side
27. retrieve plate
28. close drawer
29. press button top
30. reach
31. press button top w/ wall
32. reach with wall
33. insert peg side
34. push
35. push with wall
36. pick & place w/ wall
37. press button
38. pick & place
39. pull mug
40. unplug peg
41. close window
42. open window
43. open door
44. close door
45. open drawer
46. hand insert
47. close box
48. lock door
49. unlock door
50. pick bin

Environment



Questions?