Thoughts on Progress Made and Challenges Ahead in Few-Shot Learning

Hugo Larochelle
Google Brain

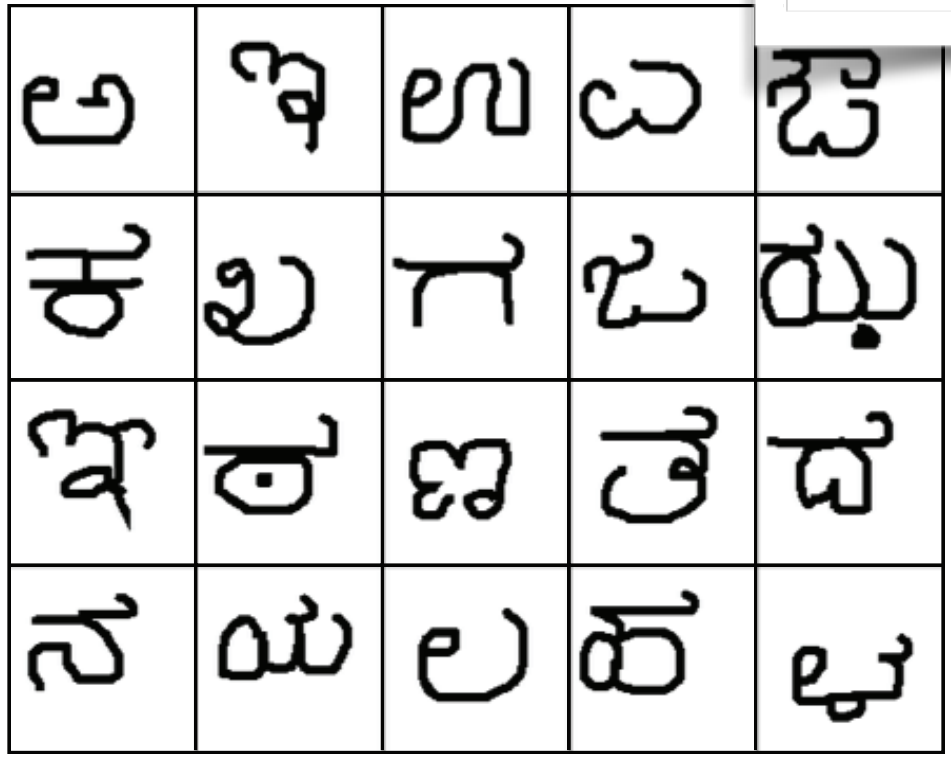

1
2
3
4
5
?
?
D_{train}
D_{test}

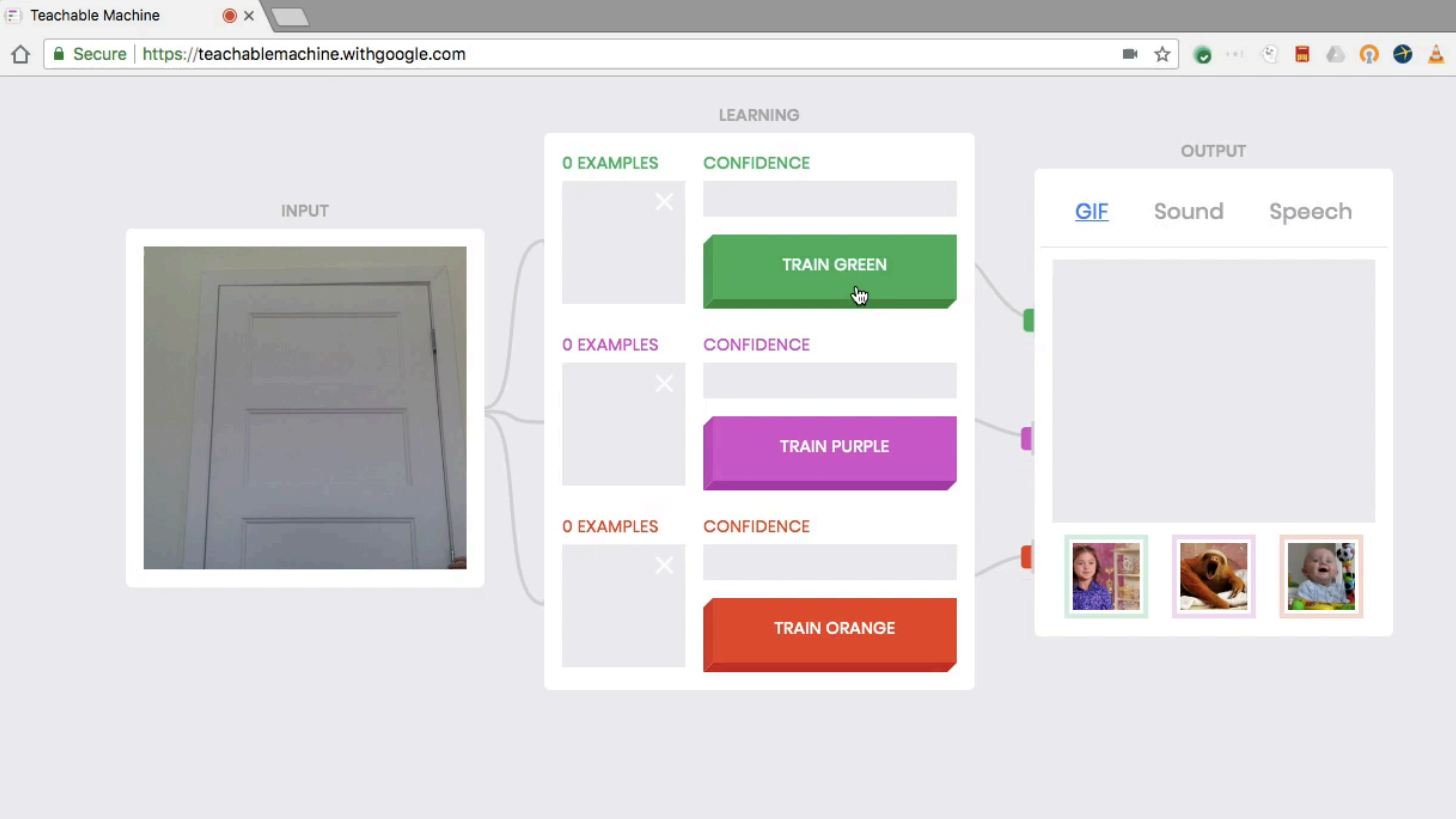
People are good at it

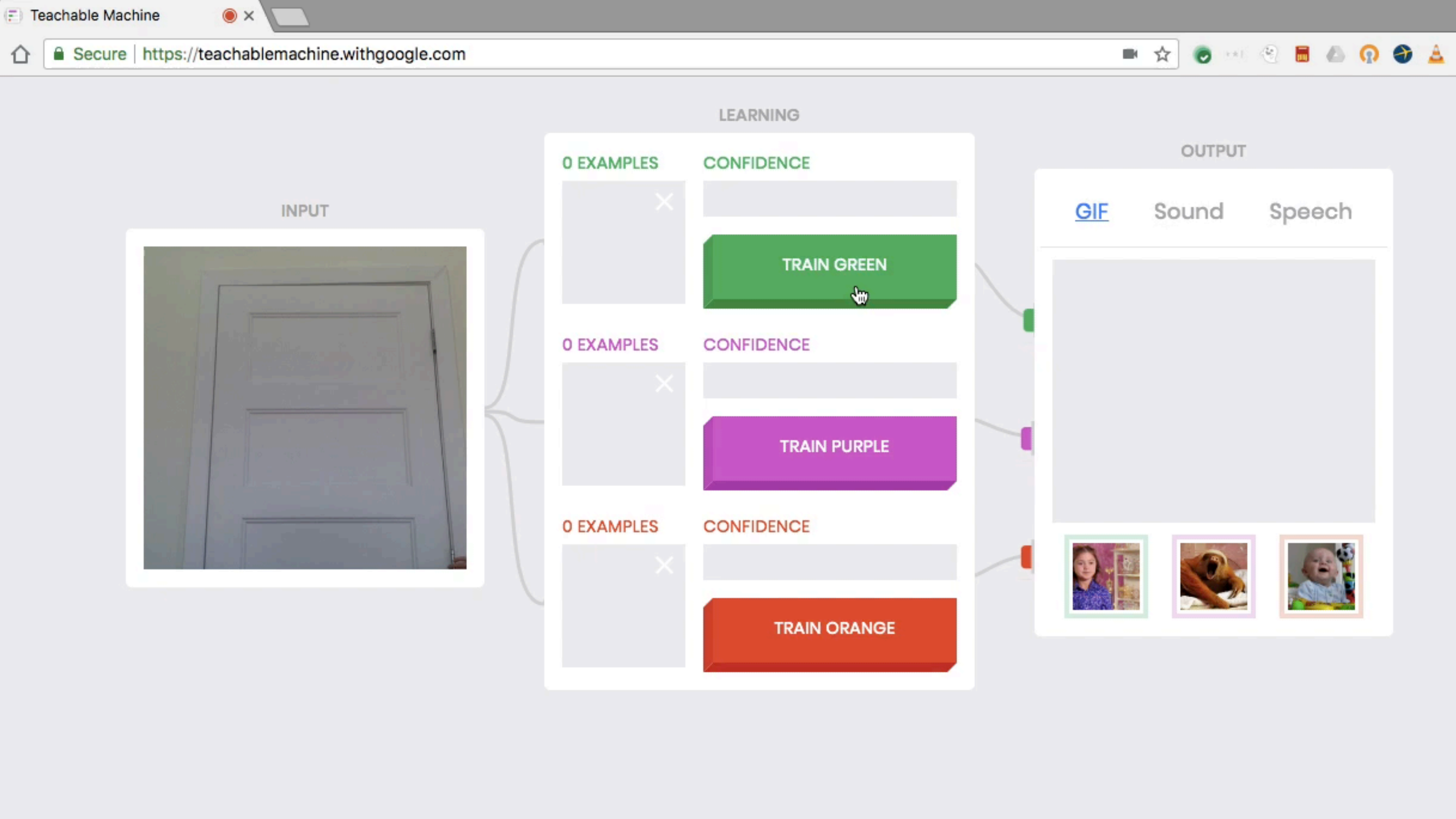
Human-level concept learning through probabilistic program induction

Brenden M. Lake,[1*] Ruslan Salakhutdinov,[2] Joshua B. Tenenbaum[3]

Machines are getting better at it




Teachable Machine
Secure
https://teachablemachine.withgoogle.com
LEARNING
INPUT
OUTPUT
0 EXAMPLES
CONFIDENCE
TRAIN GREEN
0 EXAMPLES
CONFIDENCE
TRAIN PURPLE
0 EXAMPLES
CONFIDENCE
TRAIN ORANGE
GIF
Sound
Speech


Teachable Machine
Secure
https://teachablemachine.withgoogle.com
LEARNING
INPUT
0 EXAMPLES
CONFIDENCE
TRAIN GREEN
0 EXAMPLES
CONFIDENCE
TRAIN PURPLE
0 EXAMPLES
CONFIDENCE
TRAIN ORANGE
OUTPUT
GIF
Sound
Speech

RELATED WORK: ONE-SHOT LEARNING

- One-shot learning has been studied before
  - One-Shot learning of object categories (2006)
    Fei-Fei Li, Rob Fergus and Pietro Perona
  - Knowledge transfer in learning to recognize visual objects classes (2004)
    Fei-Fei Li
  - Object classification from a single example utilizing class relevance pseudo-metrics (2004)
    Michael Fink
  - Cross-generalization: learning novel classes from a single example by feature replacement (2005)
    Evgeniy Bart and Shimon Ullman
- These largely relied on hand-engineered features and algorithms
  - with recent progress in **end-to-end** deep learning, we hope to jointly learn a **representation** and **algorithm** better suited for few-shot learning

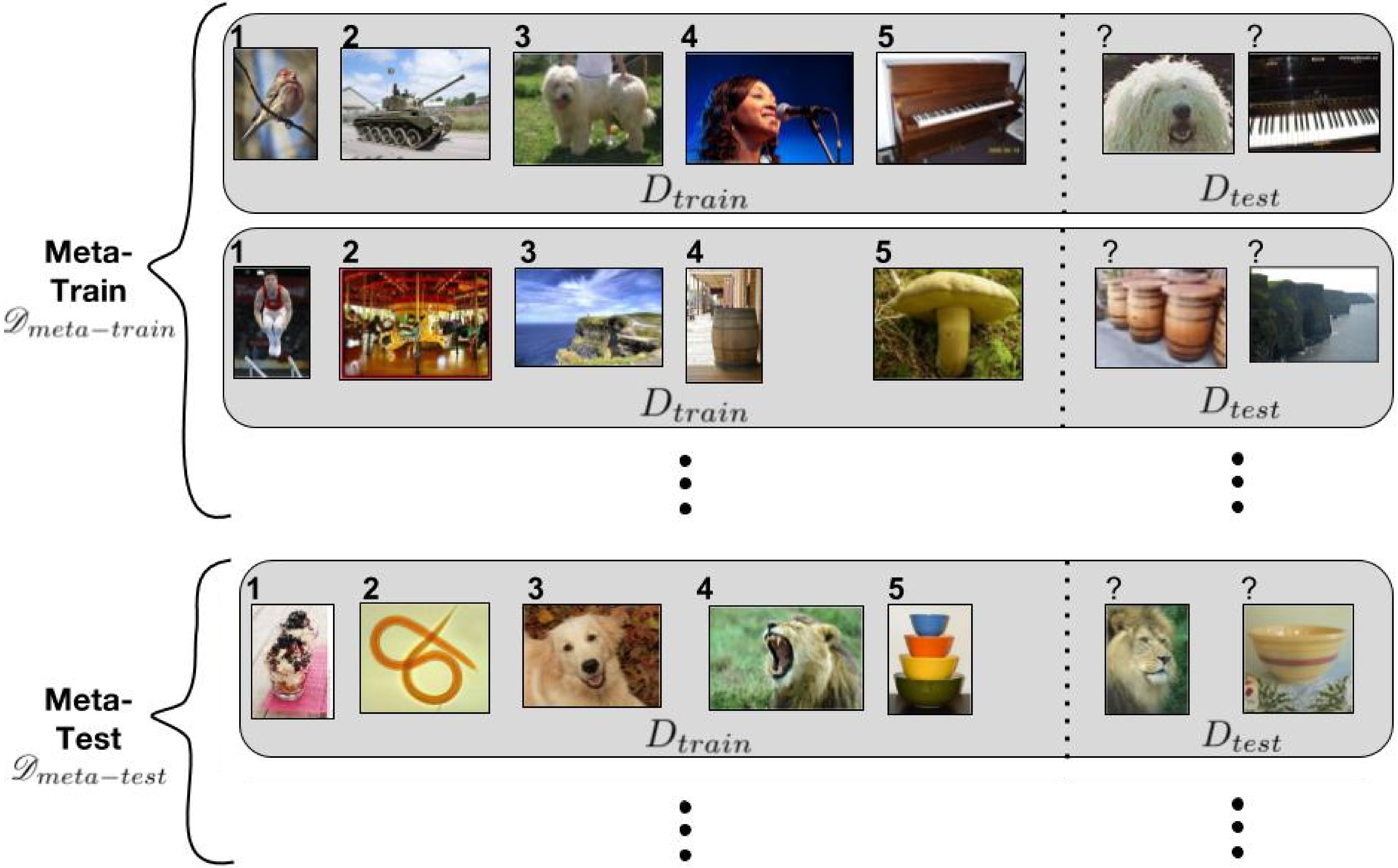
META-LEARNING

META-LEARNING

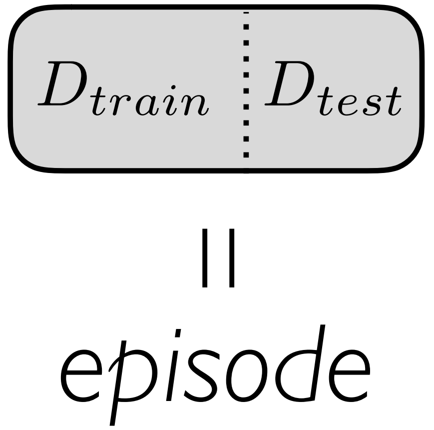
D_{train} D_{test}

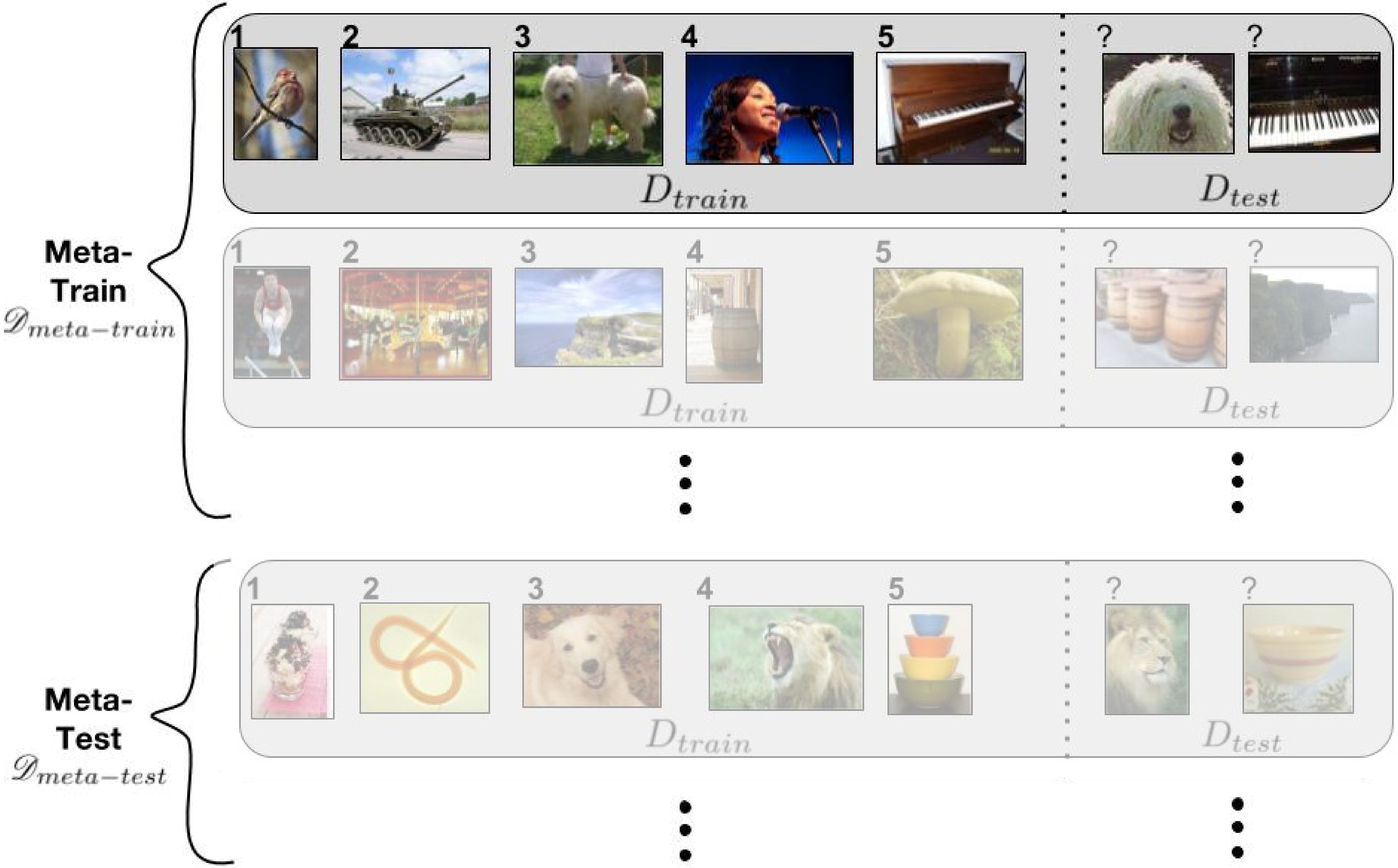
=

episode



META-LEARNING

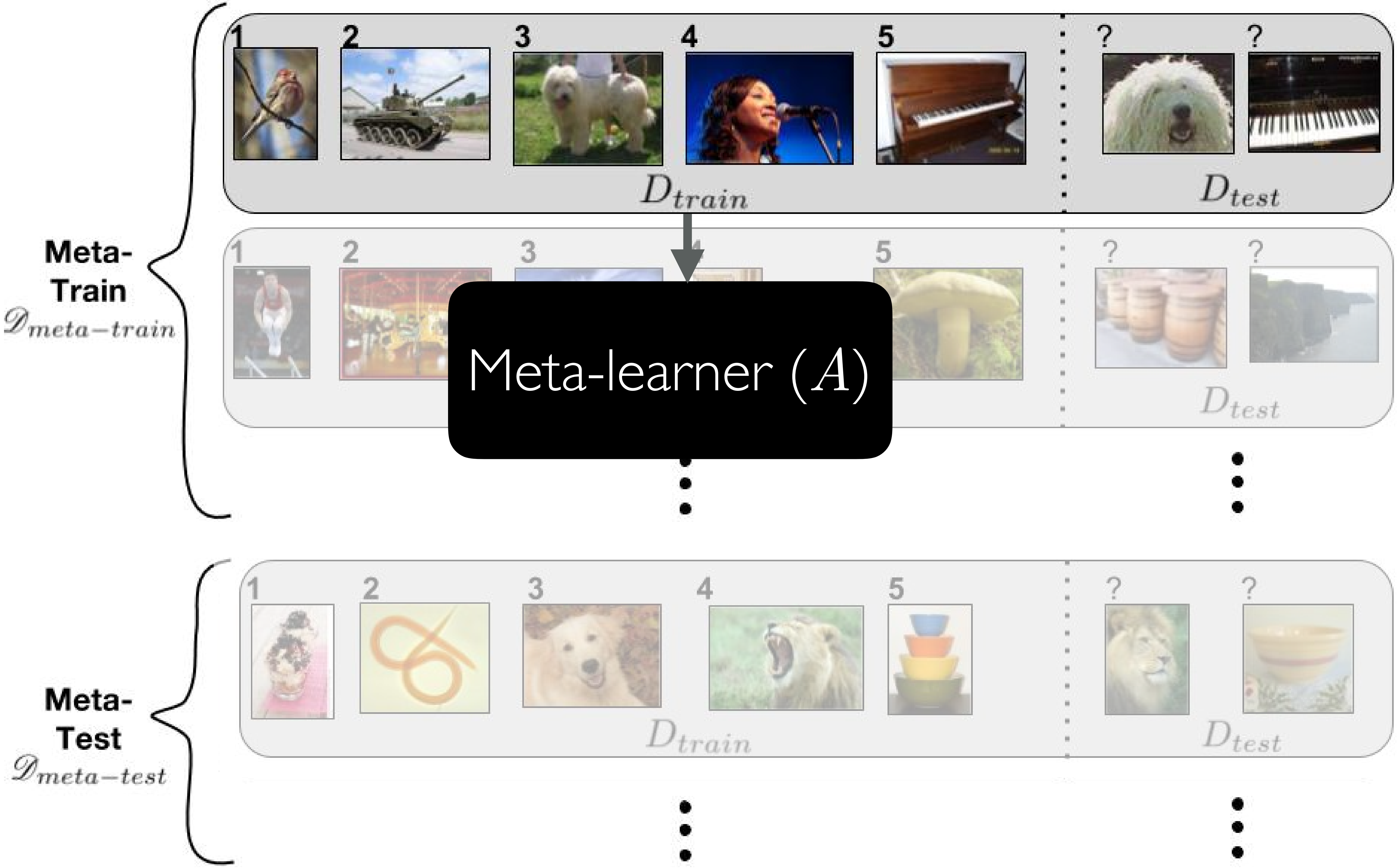




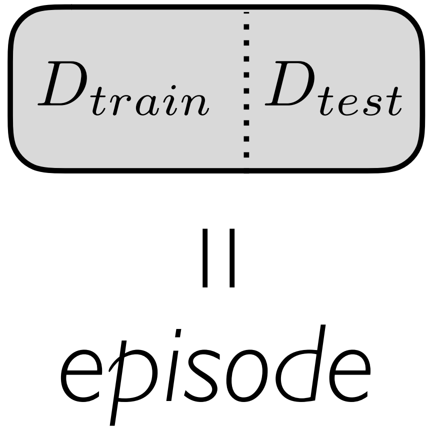
META-LEARNING

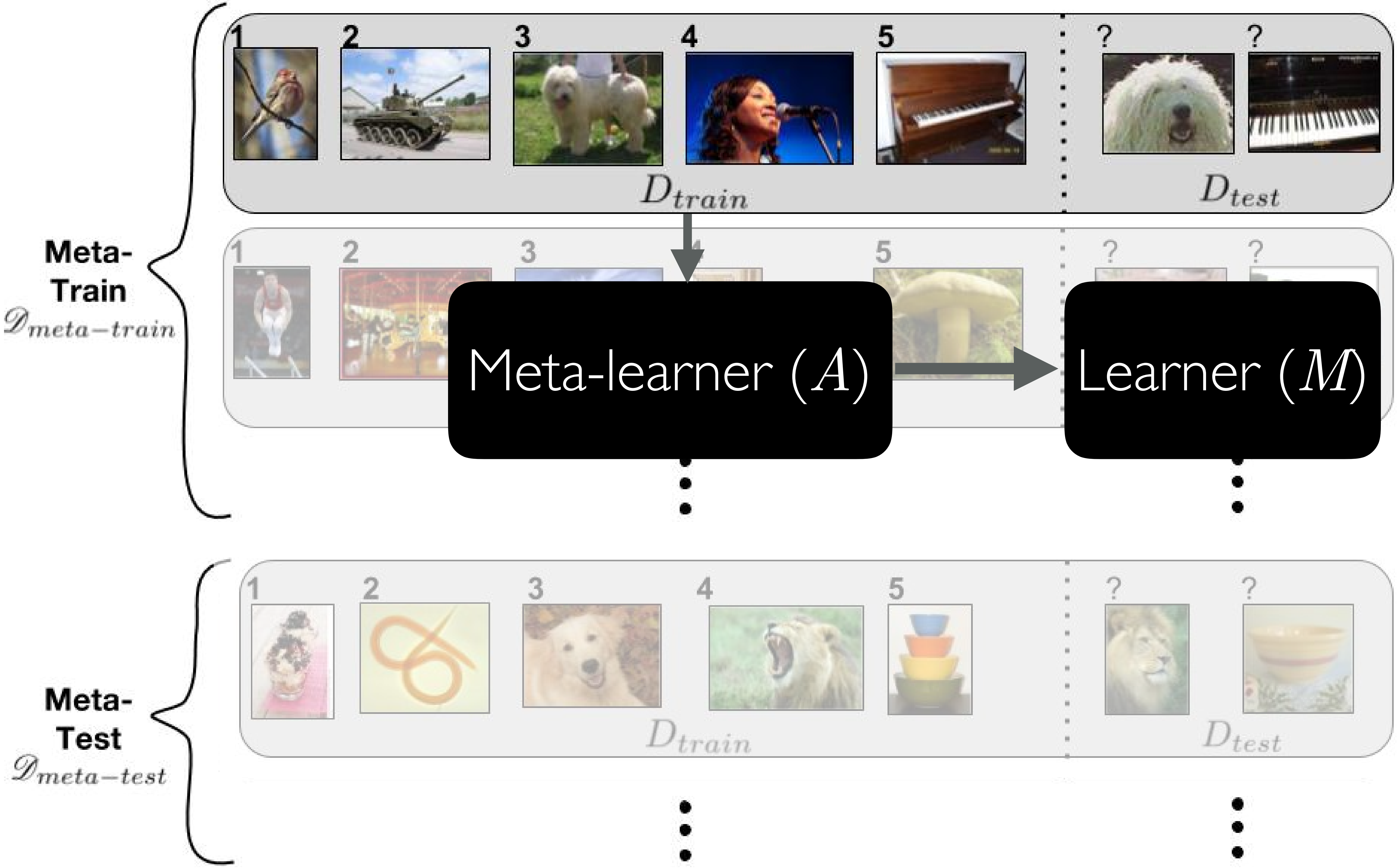
$=$

episode

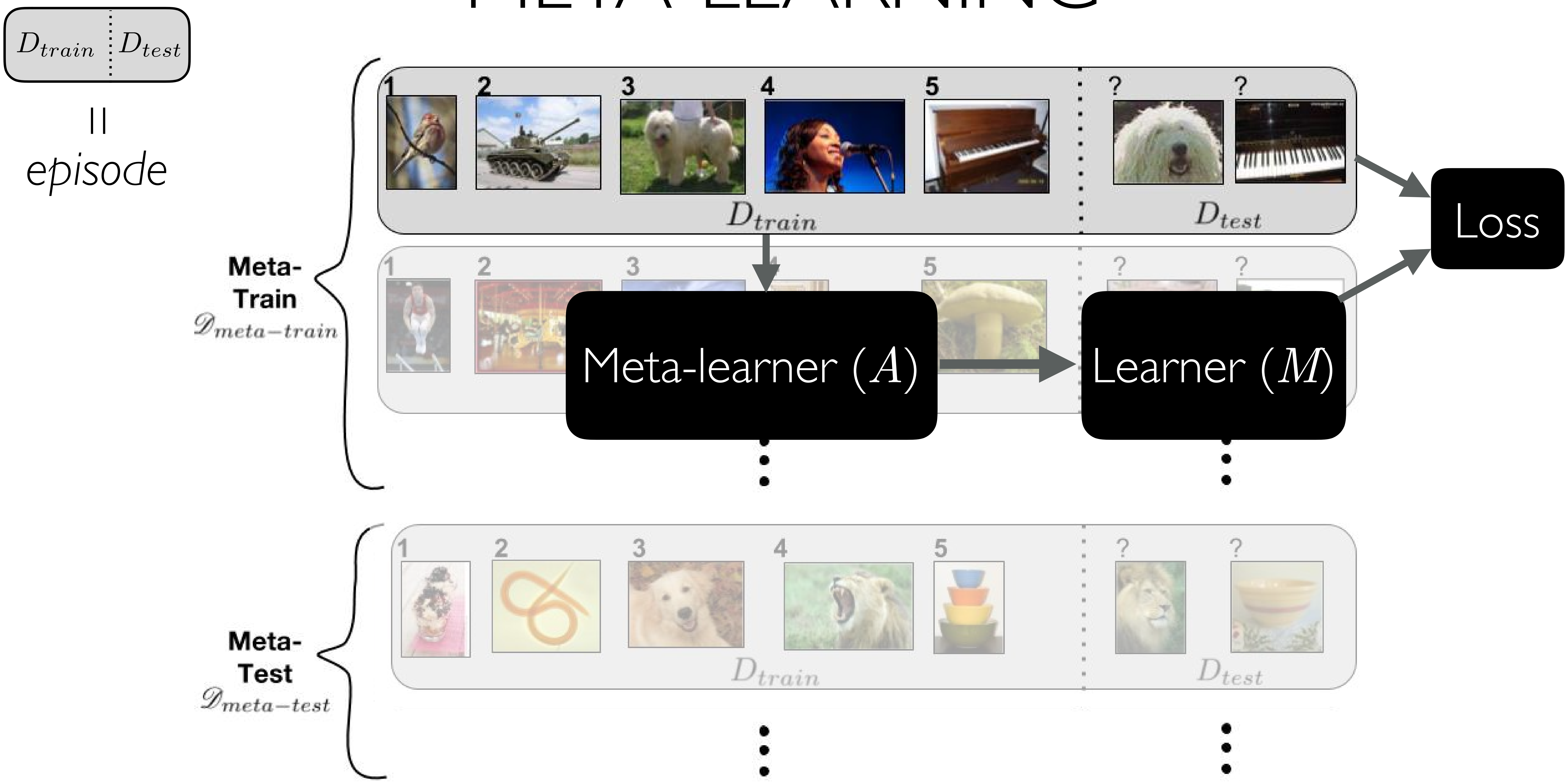


META-LEARNING

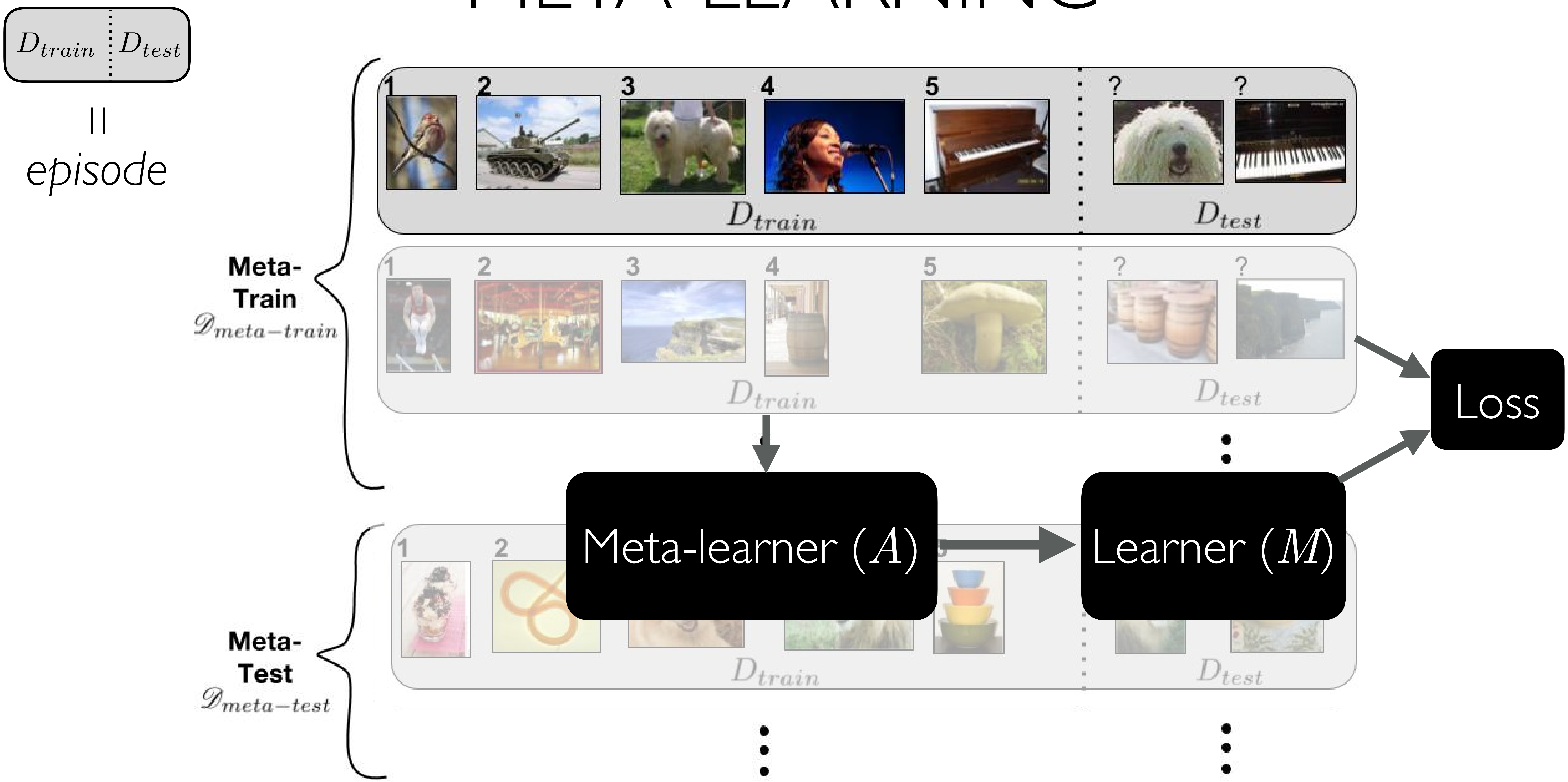




META-LEARNING



META-LEARNING



META-LEARNING

If you don’t evaluate on never-seen problems/datasets…

If you don’t evaluate on never-seen problems/datasets…

… it’s not meta-learning!

LEARNING PROBLEM STATEMENT

- Assuming a probabilistic model M over labels, the cost per episode can written as

$$C(D_{train}, D_{test}) = \frac{1}{|D_{test}|} \sum_{\substack{(\mathbf{x}_t, y_t) \\ \in D_{test}}} -\log p(y_t|\mathbf{x}_t, D_{train})$$

- Here $p(y|\mathbf{x}, D_{train})$ jointly represents the meta-learner A (which processes D_{train}) and the learner M (which processes $\mathbf{x}$)

CHOOSING A META-LEARNER

- How to parametrize learning algorithms (meta-learners $p(y|\mathbf{x}, D_{train})$)?

- Two approaches to defining a meta-learner
  - Take inspiration from a known learning algorithm
    - kNN/kernel machine: Matching networks (Vinyals et al. 2016)
    - Gaussian classifier: Prototypical Networks (Snell et al. 2017)
    - Gradient Descent: Meta-Learner LSTM (Ravi & Larochelle, 2017) , MAML (Finn et al. 2017)
  - Derive it from a black box neural network
    - SNAIL (Mishra et al. 2018)

CHOOSING A META-LEARNER

- How to parametrize learning algorithms (meta-learners $p(y|\mathbf{x}, D_{train})$)?

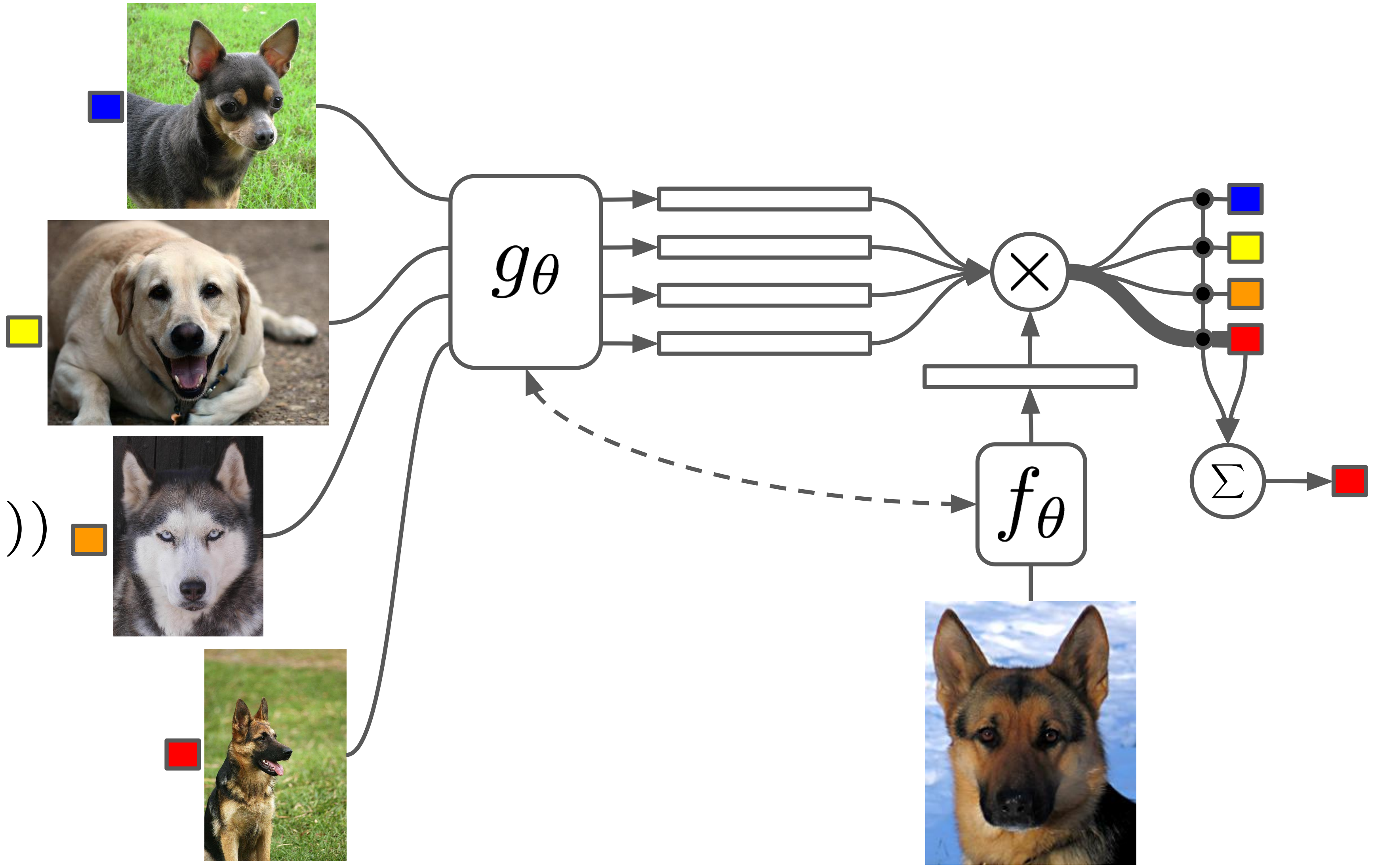
- Two approaches to defining a meta-learner
  - **Take inspiration from a known learning algorithm**
    - kNN/kernel machine: Matching networks (Vinyals et al. 2016)
    - Gaussian classifier: Prototypical Networks (Snell et al. 2017)
    - Gradient Descent: Meta-Learner LSTM (Ravi & Larochelle, 2017) , MAML (Finn et al. 2017)
  - Derive it from a black box neural network
    - SNAIL (Mishra et al. 2018)

MATCHING NETWORKS

- Training a "**pattern matcher**" (kNN/kernel machine)

$$\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$$

$$a(\hat{x}, x_i) = e^{c(f(\hat{x}), g(x_i))} / \sum_{j=1}^{k} e^{c(f(\hat{x}), g(x_j))}$$



- Matching networks for one shot learning (2016)
  Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra

PROTOTYPICAL NETWORKS

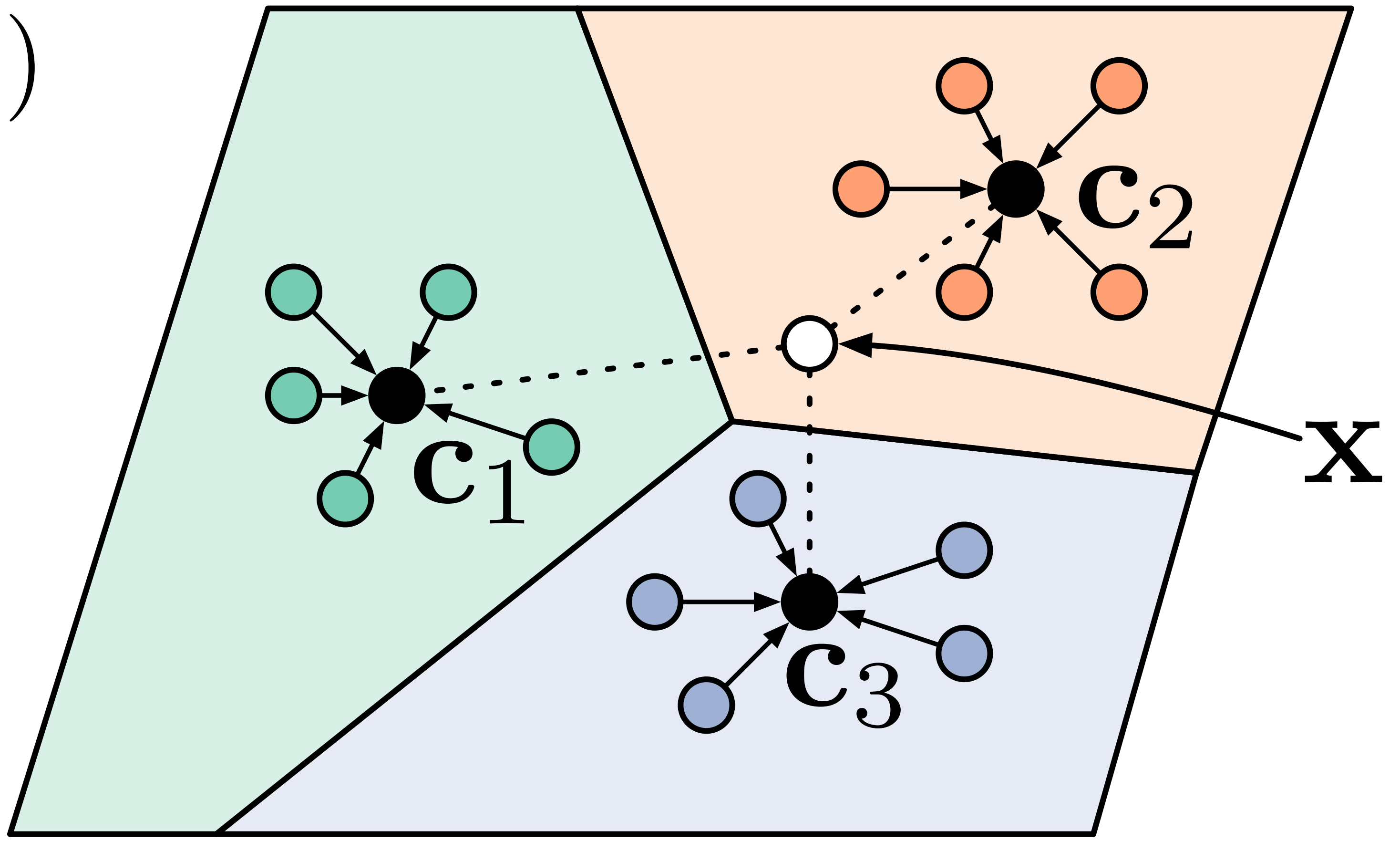
- Training a "**prototype extractor**" (Gaussian classifier)

$$p_{\boldsymbol{\phi}}(y = k \,|\, \mathbf{x}) = \frac{\exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_k))}{\sum_{k'} \exp(-d(f_{\boldsymbol{\phi}}(\mathbf{x}), \mathbf{c}_{k'}))}$$

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\boldsymbol{\phi}}(\mathbf{x}_i)$$

$$S_k = \{(\mathbf{x}_i, y_i) | y_i = k, (\mathbf{x}_i, y_i) \in D_{train}\}$$

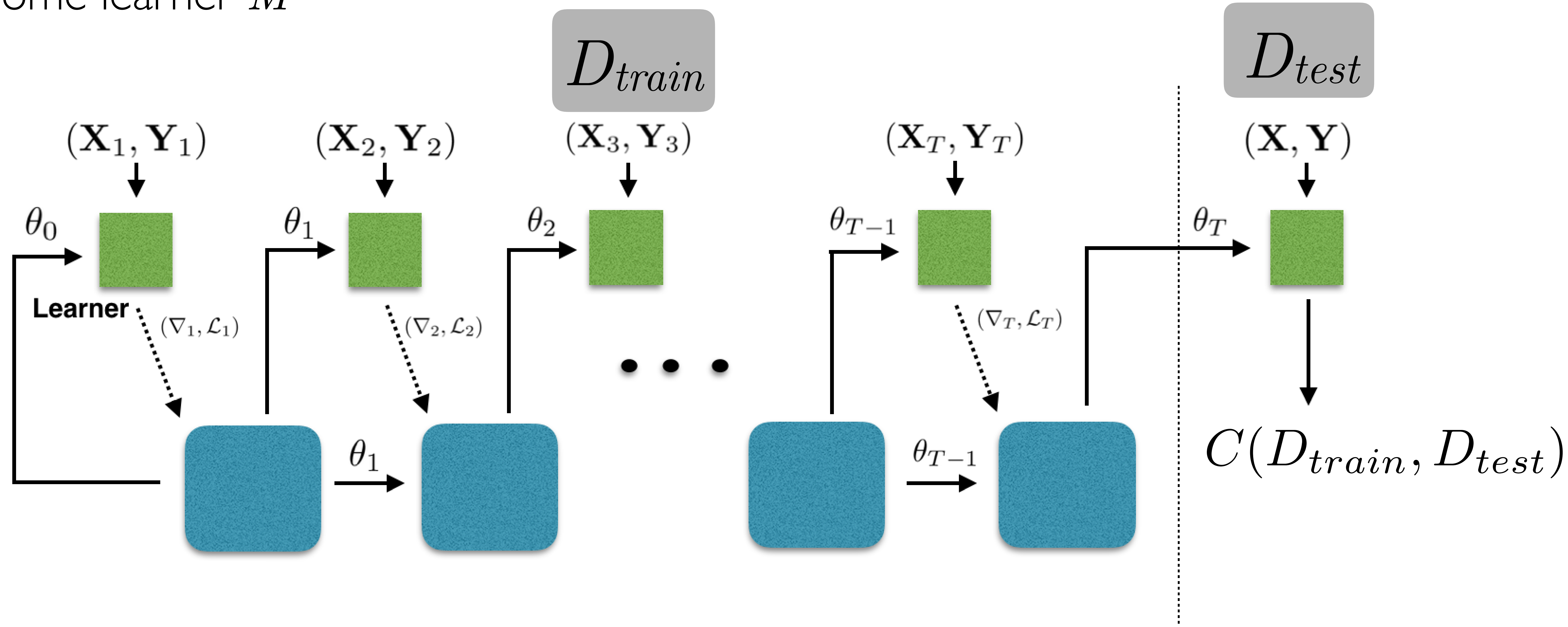
$$\phi \equiv \boldsymbol{\Theta}$$



- Prototypical Networks for Few-shot Learning (2017)
  Jake Snell, Kevin Swersky and Richard Zemel

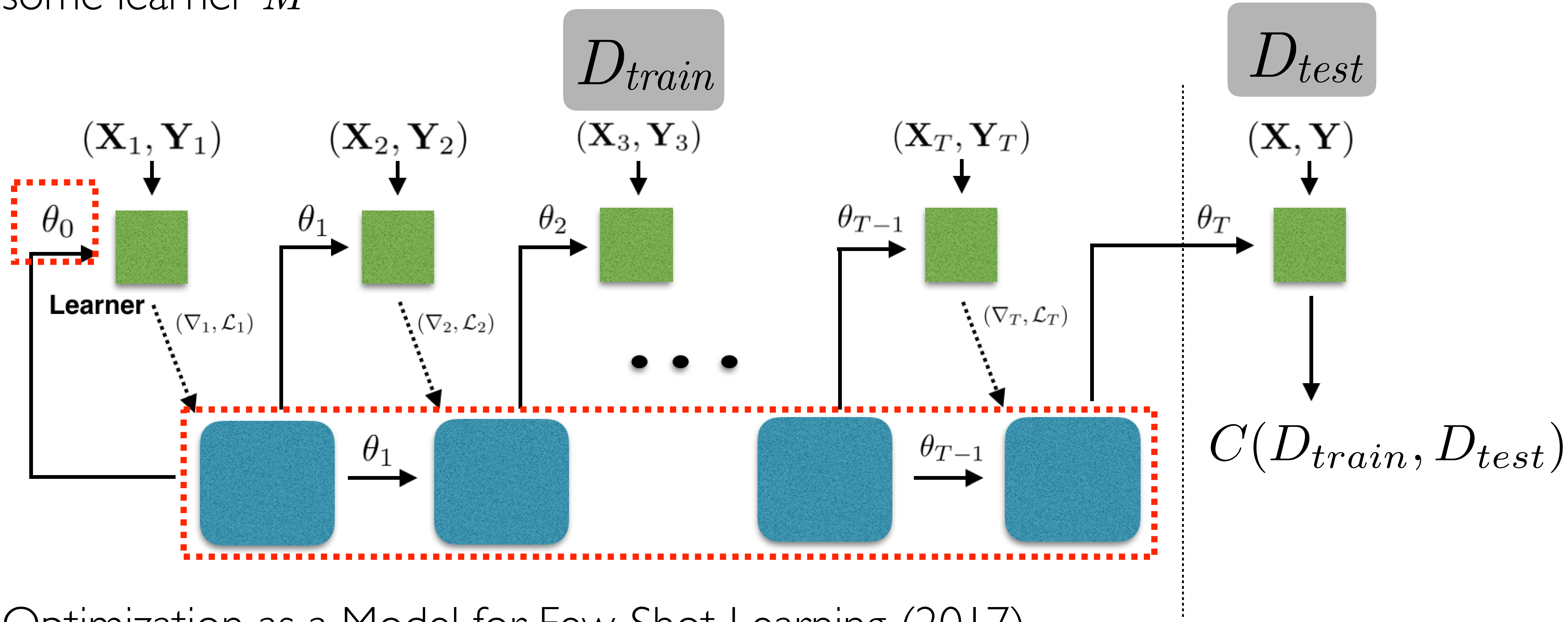
META-LEARNER LSTM

- Training an "**initialize and gradient descent procedure**" applied on some learner M



META-LEARNER LSTM

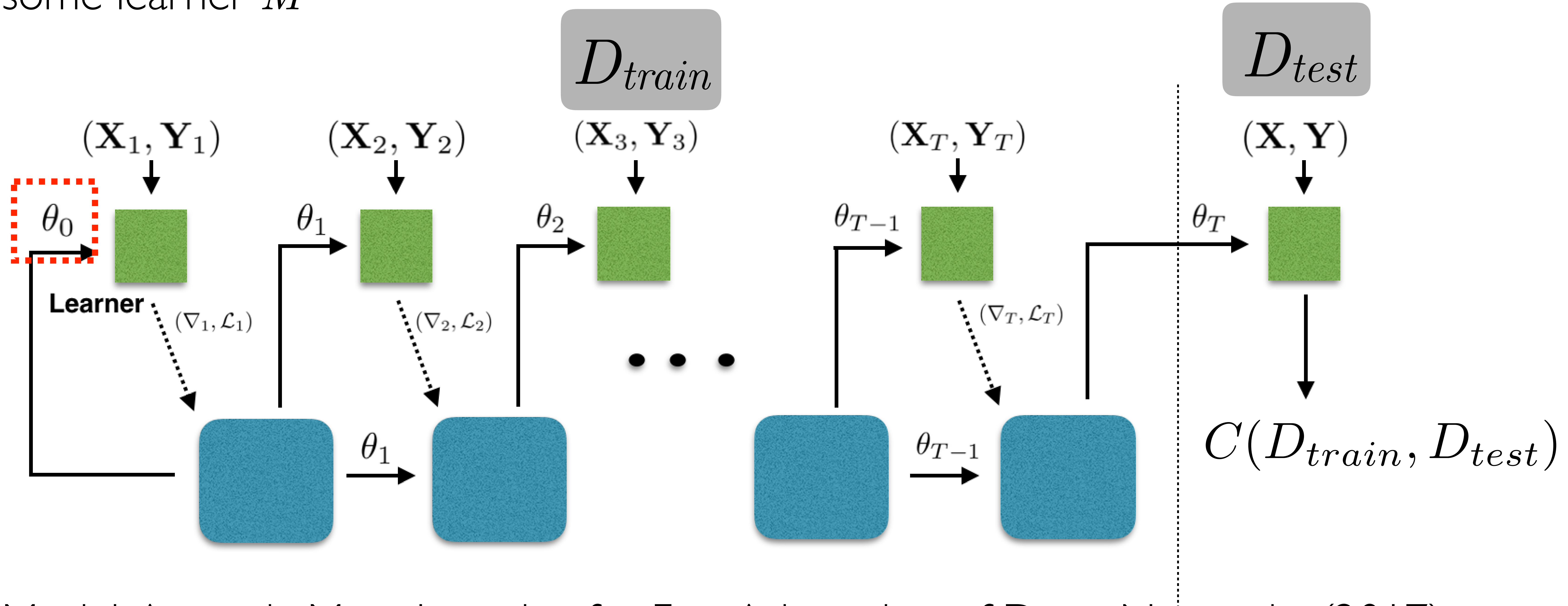
- Training an "**initialize and gradient descent procedure**" applied on some learner M



- Optimization as a Model for Few-Shot Learning (2017)
Sachin Ravi and Hugo Larochelle

META-LEARNER LSTM

- Training an "**initialize and gradient descent procedure**" applied on some learner M



- Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks (2017)
  Chelsea Finn, Pieter Abbeel and Sergey Levine

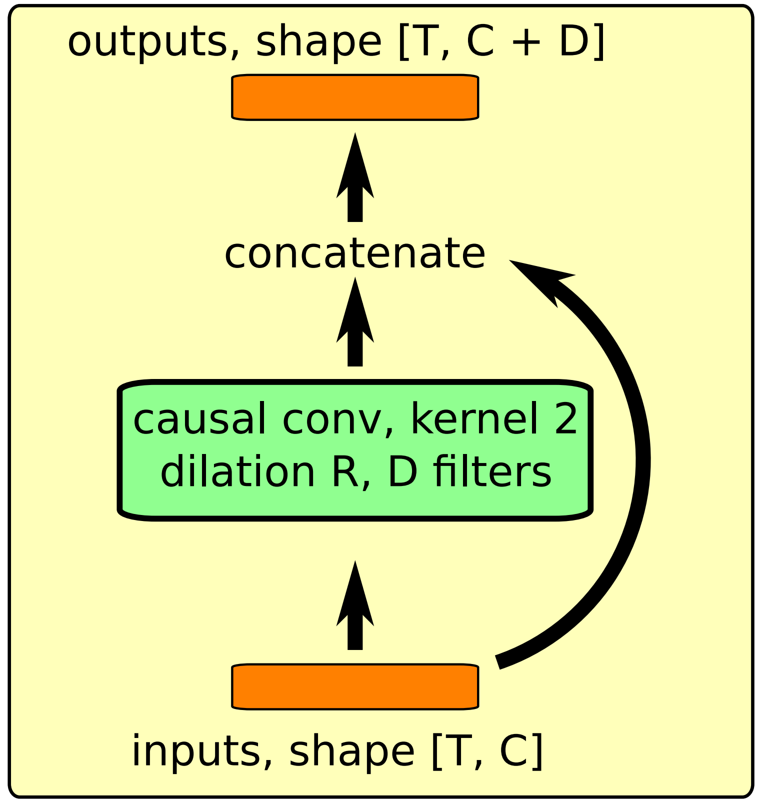
CHOOSING A META-LEARNER

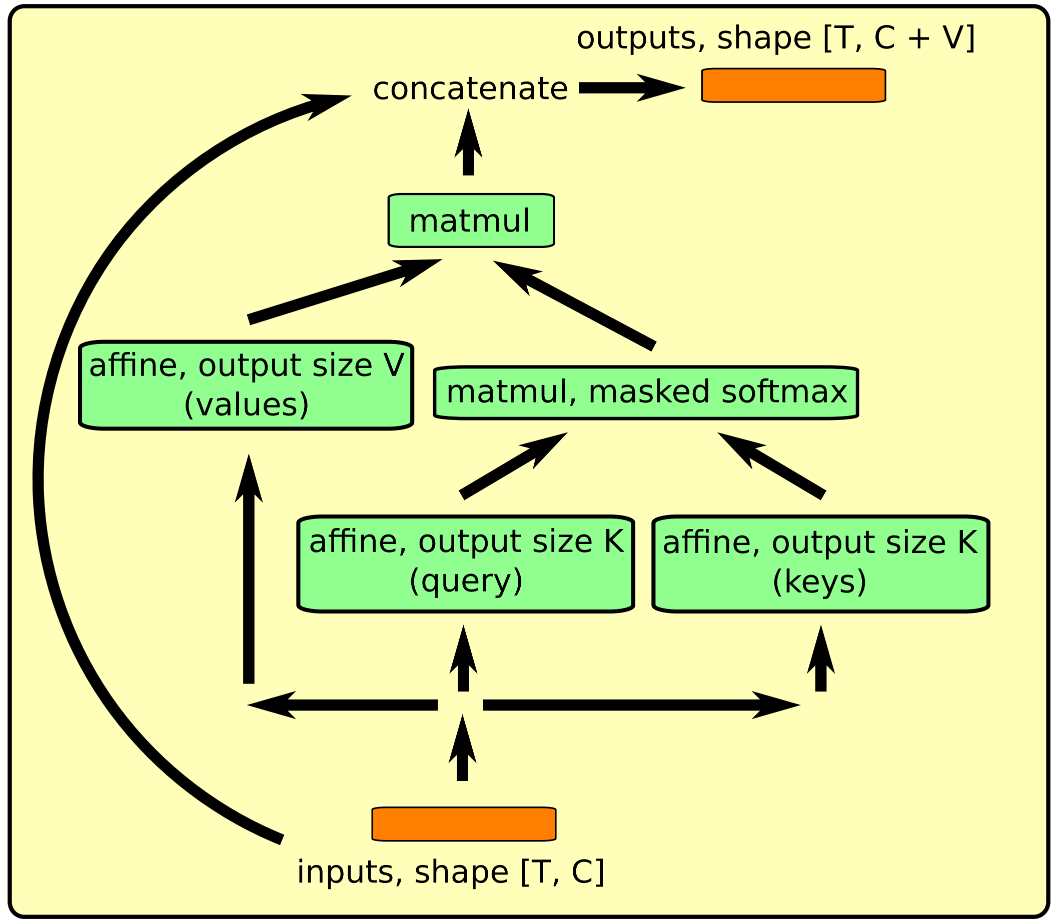
- How to parametrize learning algorithms (meta-learners $p(y|\mathbf{x}, D_{train})$)?

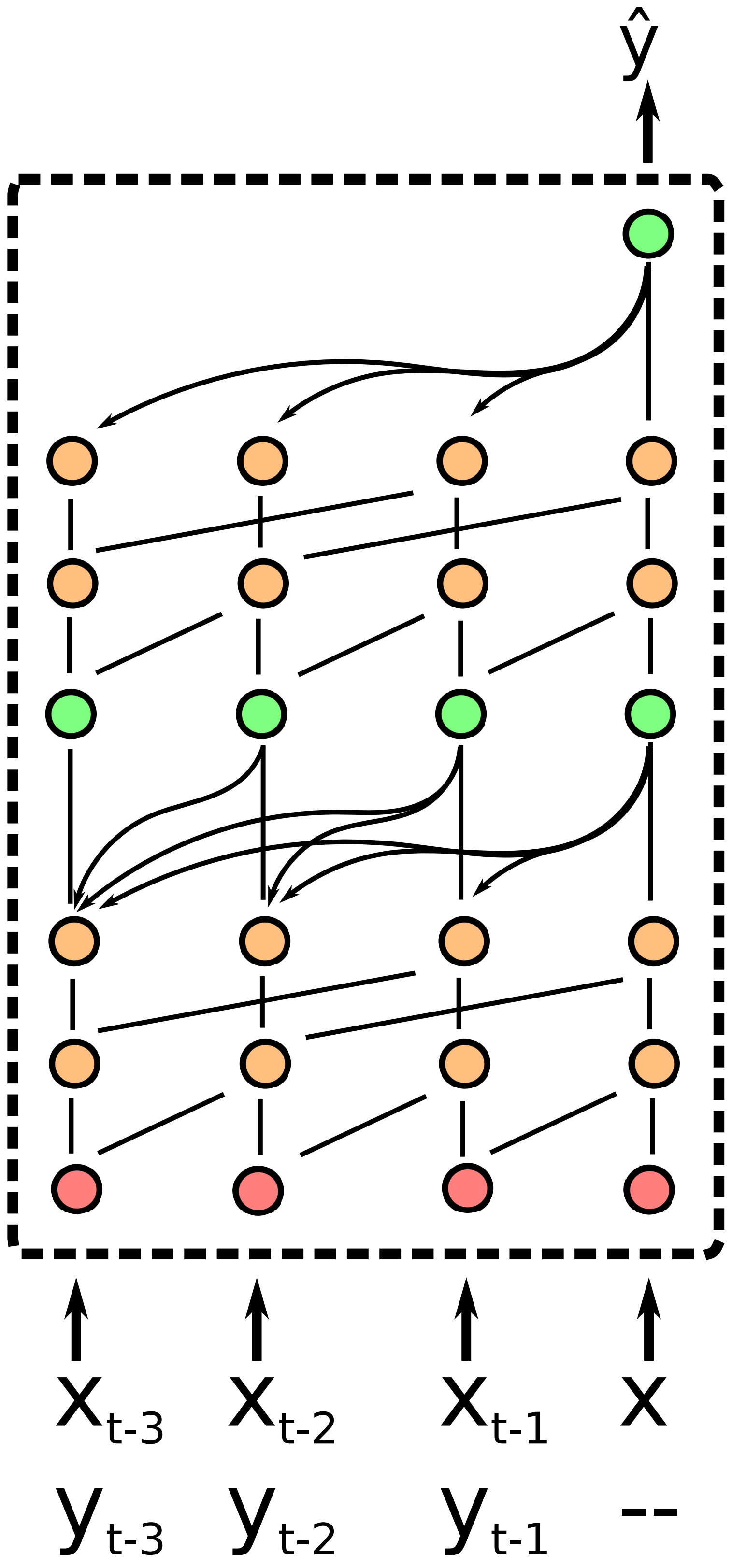
- Two approaches to defining a meta-learner
  - Take inspiration from a known learning algorithm
    - kNN/kernel machine: Matching networks (Vinyals et al. 2016)
    - Gaussian classifier: Prototypical Networks (Snell et al. 2017)
    - Gradient Descent: Meta-Learner LSTM (Ravi & Larochelle, 2017) , MAML (Finn et al. 2017)
  - **Derive it from a black box neural network**
    - SNAIL (Mishra et al. 2018)

SIMPLE NEURAL ATTENTIVE LEARNER

- Using a **convolutional/attentional network** to represent $p(y|\mathbf{x}, D_{train})$
  - alternates between dilated convolutional layers and attentional layers
  - when inputs are images, an convolutional embedding network is used to map to a vector space







- A Simple Neural Attentive Meta-Learner (2018)
  Nikhil Mishra, Mostafa Rohaninejad, Xi Chen and Pieter Abbeel

AND SO MUCH MORE!!!

Hugo Larochelle - Few-shot Learning with Meta-Learning: Progress Made and Challenges Ahead

bit.ly/2PikS82

EXPERIMENT

- Mini-ImageNet (split used in Ravi & Larochelle, 2017)
  - random subset of 100 classes (64 training, 16 validation, 20 testing)
  - random sets D_{train} are generated by randomly picking 5 classes from class subset

Model	5-class	
	1-shot	5-shot
Baseline-finetune	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
Baseline-nearest-neighbor	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
Matching Network	$\mathbf{43.40 \pm 0.78\%}$	$51.09 \pm 0.71\%$
Matching Network FCE	$\mathbf{43.56\% \pm 0.84\%}$	$55.31\% \pm 0.73\%$
Meta-Learner LSTM (OURS)	$\mathbf{43.44\% \pm 0.77\%}$	$\mathbf{60.60\% \pm 0.71\%}$

EXPERIMENT

- Mini-ImageNet (split used in Ravi & Larochelle, 2017)
  - random subset of 100 classes (64 training, 16 validation, 20 testing)
  - random sets D_{train} are generated by randomly picking 5 classes from class subset

Model	**5-class**	
	1-shot	5-shot
Prototypical Nets (Snell et al.)	49.42% ± 0.78%	**68.20% ± 0.66%**
MAML (Finn et al.)	48.70% ± 1.84%	63.10% ± 0.92%
SNAIL (Mishra et al.)	**55.71% ± 0.99%**	**68.88% ± 0.98%**
Matching Network FCE	43.56% ± 0.84%	55.31% ± 0.73%
Meta-Learner LSTM (OURS)	43.44% ± 0.77%	60.60% ± 0.71%

REMAINING CHALLENGES

- Going beyond supervised classification
  - unsupervised learning, structured output, interactive learning

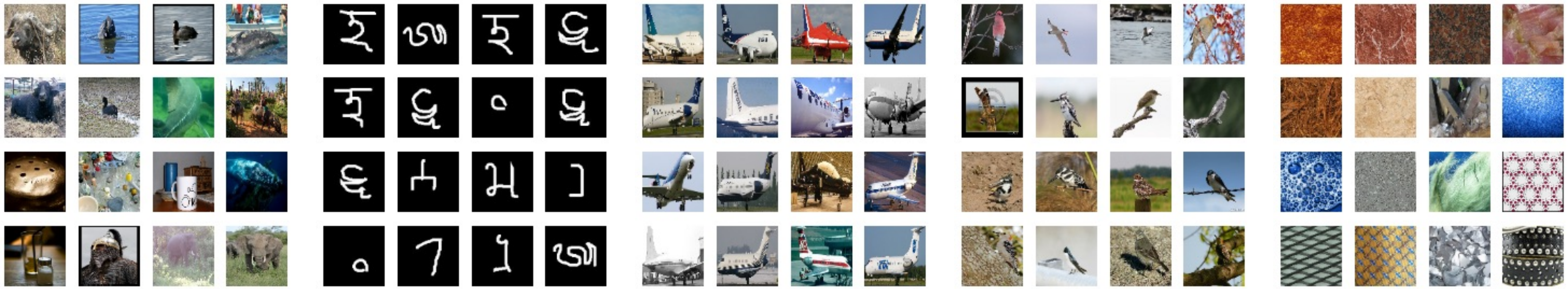
- Going beyond Mini-ImageNet
  - coming up with a realistic definition of distributions over problems/datasets

Meta-Dataset: A Dataset of Datasets for Learning to Learn from Few Examples

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, Hugo Larochelle

META-DATASET

- To learn across many tasks requires learning over **many datasets**

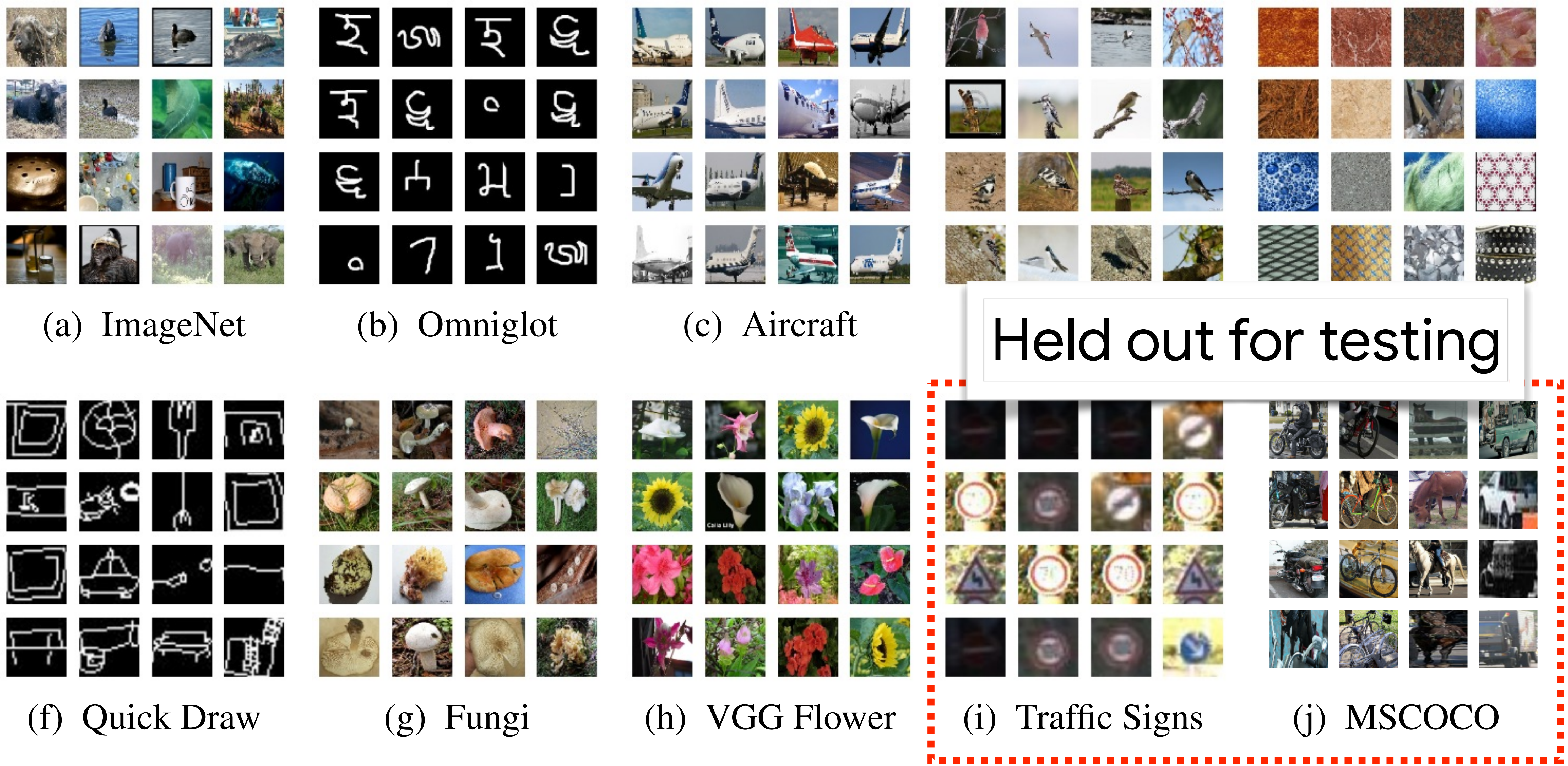


(a) ImageNet (b) Omniglot (c) Aircraft (d) Birds (e) DTD

(f) Quick Draw (g) Fungi (h) VGG Flower (i) Traffic Signs (j) MSCOCO

META-DATASET

- To learn across many tasks requires learning over **many datasets**



META-DATASET

- Meta-training only on ImageNet

Test Source	Method: Accuracy ± confidence				
	k-NN	Finetune	MatchingNet	ProtoNet	MAML
ILSVRC	34.70±0.95	38.34±1.12	40.89±1.08	**43.37±1.17**	38.10±1.13
Omniglot	59.84±0.96	59.19±1.18	61.85±1.00	**66.18±1.12**	54.00±1.47
Aircraft	36.47±0.93	**41.18±1.07**	**41.91±0.96**	**42.14±0.97**	**42.52±1.16**
Birds	40.38±1.09	45.82±1.25	54.26±1.16	**57.85±1.23**	50.78±1.32
Textures	56.45±0.78	58.06±0.88	**61.70±0.84**	**60.95±0.80**	**61.26±0.93**
Quick Draw	36.09±1.19	38.43±1.39	38.52±1.12	**44.02±1.35**	30.71±1.51
Fungi	23.70±0.97	22.20±0.92	27.21±0.97	**31.18±1.15**	20.35±0.87
VGG Flower	66.16±0.99	69.32±1.13	75.05±0.91	**79.89±0.90**	65.12±1.15
Traffic Signs	**44.81±1.47**	39.36±1.28	**45.36±1.31**	**44.04±1.24**	31.10±1.20
MSCOCO	29.69±1.00	30.25±1.17	32.32±1.08	**36.44±1.23**	25.17±1.15
Avg. rank	4	3.4	2.2	**1.35**	4.05

META-DATASET

- Meta-training on all training datasets

Test Source	Method: Accuracy ± confidence				
	k-NN	Finetune	MatchingNet	ProtoNet	MAML
ILSVRC	25.88±0.83	25.84±0.83	35.88±0.98	**38.51±1.01**	30.56±1.00
Omniglot	**92.45±0.41**	85.20±0.73	90.21±0.46	91.32±0.50	78.05±0.98
Aircraft	54.60±0.97	58.22±1.02	**70.71±0.78**	**71.54±0.84**	68.62±0.90
Birds	36.74±1.01	38.56±1.08	59.28±1.06	**61.81±1.13**	54.59±1.24
Textures	50.06±0.77	48.37±0.82	**60.61±0.82**	**59.31±0.75**	**59.25±0.80**
Quick Draw	**59.54±1.08**	54.05±1.30	57.44±1.17	**60.99±1.21**	44.48±1.41
Fungi	24.60±0.95	22.90±0.95	31.10±1.04	**35.96±1.25**	21.12±0.88
VGG Flower	62.49±0.91	59.72±1.17	76.72±0.83	**81.06±0.87**	66.05±1.09
Traffic Signs	**41.68±1.46**	30.02±1.13	**43.20±1.33**	39.95±1.18	30.23±1.24
MSCOCO	23.55±0.99	23.01±0.96	26.87±1.00	**30.81±1.13**	21.13±1.06
Avg. rank	3.4	4.3	2.15	**1.4**	3.75

META-DATASET

- Difference in performance when meta-training on all datasets

Test Source	Method: Accuracy $\pm$ confidence				
	k-NN	Finetune	MatchingNet	ProtoNet	MAML
ILSVRC	-8.82±1.26	-12.5±1.39	-5.01±1.46	-4.86±1.55	-7.54±1.51
Omniglot	32.61±1.04	26.01±1.39	28.36±1.1	25.14±1.23	24.05±1.77
Aircraft	18.13±1.34	17.04±1.48	28.8±1.24	29.4±1.28	26.1±1.47
Birds	-3.64±1.49	-7.26±1.65	5.02±1.57	3.96±1.67	3.81±1.81
Textures	-6.39±1.1	-9.69±1.2	-1.09±1.17	-1.64±1.1	-2.01±1.23
Quick Draw	23.45±1.61	15.62±1.9	18.92±1.62	16.97±1.81	13.77±2.07
Fungi	0.9±1.36	0.7±1.32	3.89±1.42	4.78±1.7	0.77±1.24
VGG Flower	-3.67±1.34	-9.6±1.63	1.67±1.23	1.17±1.25	0.93±1.58
Traffic Signs	-3.13±2.07	-9.34±1.71	-2.16±1.87	-4.09±1.71	-0.87±1.73
MSCOCO	-6.14±1.41	-7.24±1.51	-5.45±1.47	-5.63±1.67	-4.04±1.56

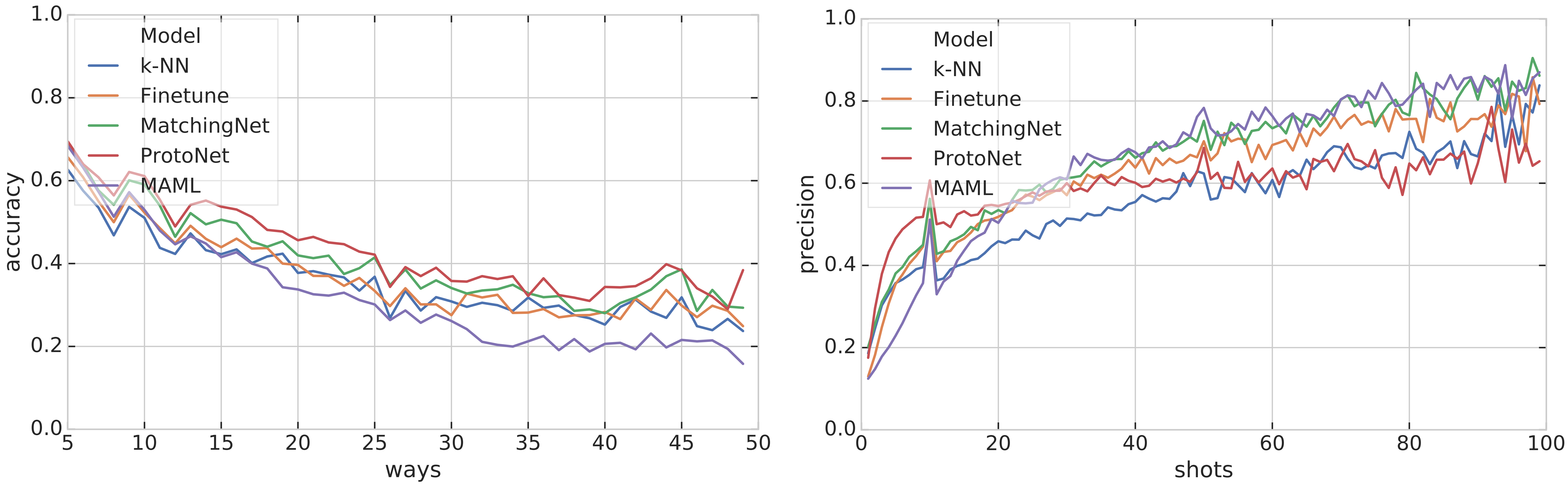
META-DATASET

- Difference in performance when meta-training on all datasets

Test Source	Method: Accuracy ± confidence				
	k-NN	Finetune	MatchingNet	ProtoNet	MAML
ILSVRC	-8.82±1.26	-12.5±1.39	-5.01±1.46	-4.86±1.55	-7.54±1.51
Omniglot	32.61±1.04	26.01±1.39	28.36±1.1	25.14±1.23	24.05±1.77
Aircraft	18.13±1.34	17.04±1.48	28.8±1.24	29.4±1.28	26.1±1.47
Birds	-3.64±1.49	-7.26±1.65	5.02±1.57	3.96±1.67	3.81±1.81
Textures	-6.39±1.1	-9.69±1.2	-1.09±1.17	-1.64±1.1	-2.01±1.23
Quick Draw	23.45±1.61	15.62±1.9	18.92±1.62	16.97±1.81	13.77±2.07
Fungi	0.9±1.36	0.7±1.32	3.89±1.42	4.78±1.7	0.77±1.24
VGG Flower	-3.67±1.34	-9.6±1.63	1.67±1.23	1.17±1.25	0.93±1.58
Traffic Signs	-3.13±2.07	-9.34±1.71	-2.16±1.87	-4.09±1.71	-0.87±1.73
MSCOCO	-6.14±1.41	-7.24±1.51	-5.45±1.47	-5.63±1.67	-4.04±1.56

META-DATASET

- Varying the number of shots and ways



TAKE AWAYS (SO FAR)

- Meta-training distribution of episodes can make a big difference (at least for current methods)

- Using "regular training" as initialization makes a big difference

- MAML needs to be adjusted to be more robust

DISCUSSION

- Now is time to move beyond our current simple benchmarks
- What is the "right" meta-training distribution?
- How should we be increasing the size of the benchmark (what should be V2)?
- What are the properties of the optimization landscape of the episodic framework?
- What fairness-relate questions does meta-learning pose?

MERCI !