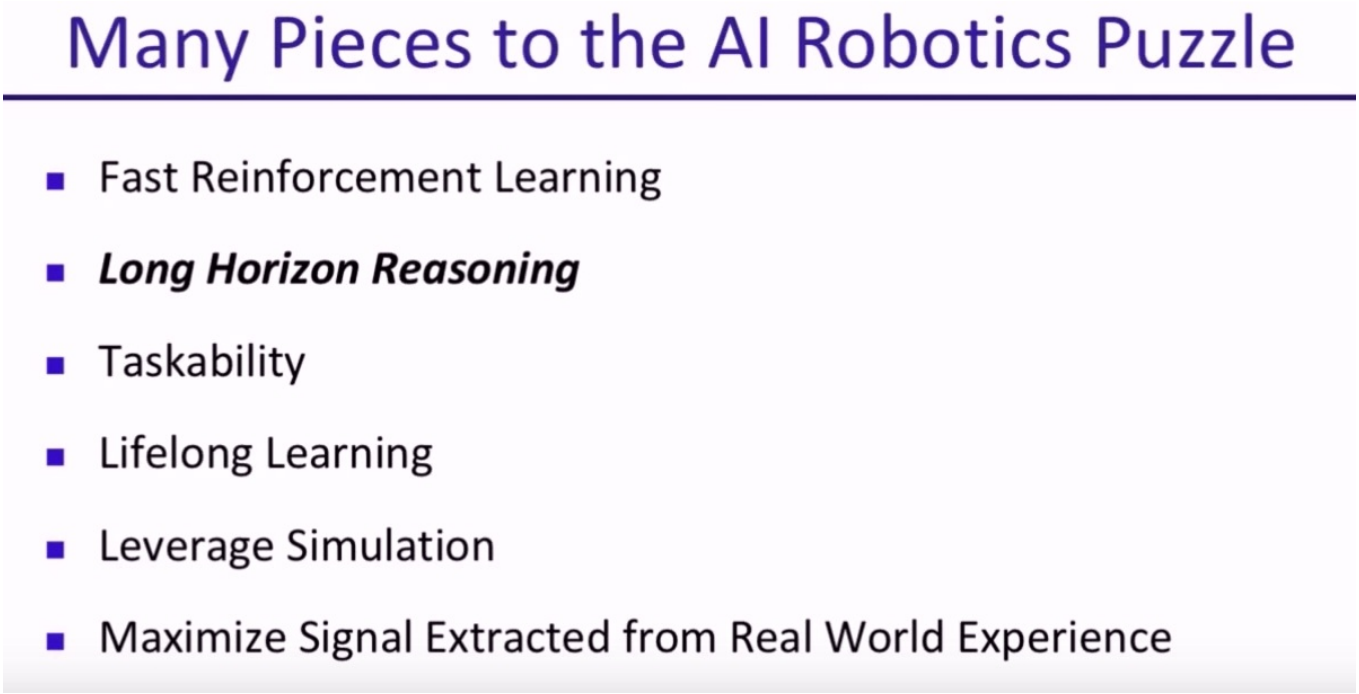
Deep Learning for Robotics - Pieter Abbeel - NIPS 2017 Keynotes

- **Yunqiu Xu**
- Video: https://www.youtube.com/watch?v=MBQTu8P1N4M

---

1. Current challenges for AI



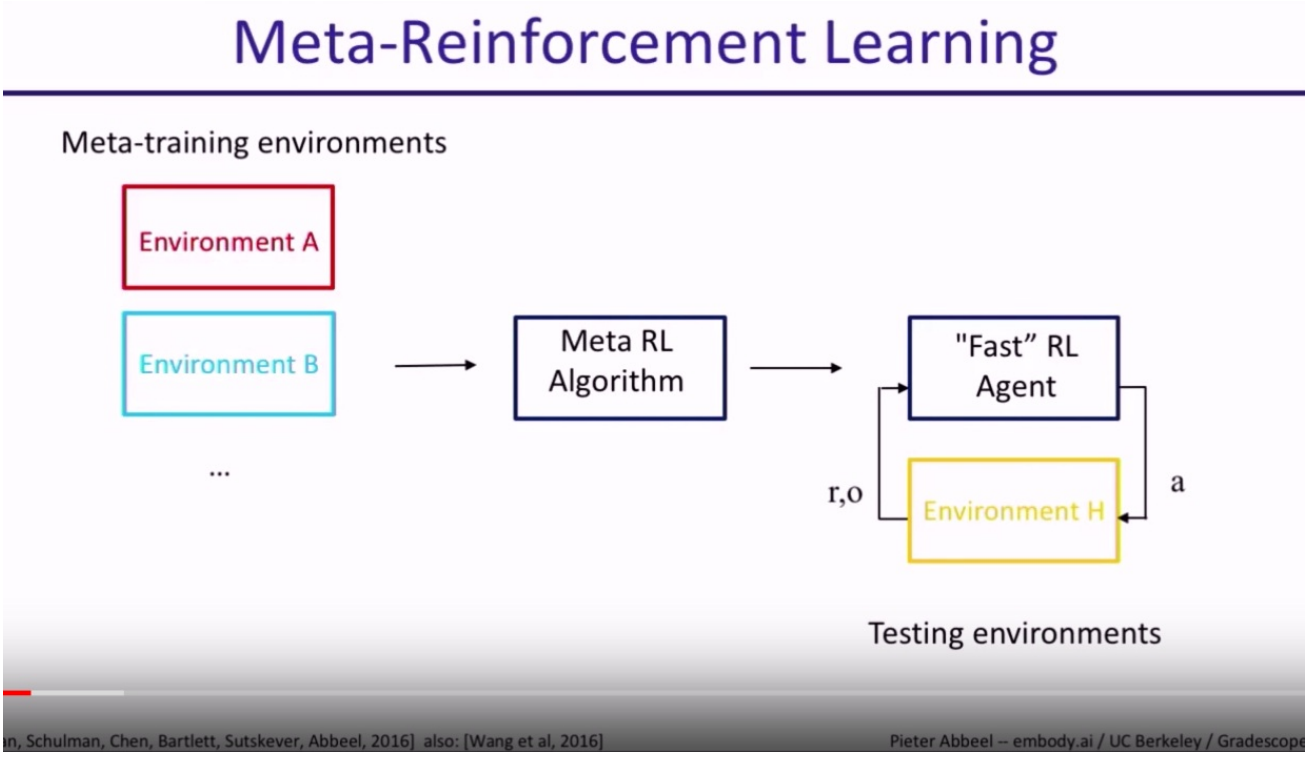
2. RL vs real environment

- TRPO, DQN, A3C, DDPG, PPO, Rainbow, … are fully general RL algorithms
  - i.e., for any environment that can be mathematically defined, these algorithms are equally applicable
- Environments encountered in real world
  = tiny, tiny subset of all environments that could be defined
  (e.g. they all satisfy our universe's physics)

Can we develop "fast" RL algorithms that take advantage of this?

3. Fast RL : Meta learning

- Meta learning: learning to learn



- Meta learning formalize

$$\max_\theta \mathbb{E}_M \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_\theta \right]$$

M : sample MDP

$\tau_M^{(k)}$: k'th trajectory in MDP M

Meta-train:

$$\max_\theta \sum_{M \in M_{\text{train}}} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_\theta \right]$$

$$\max_\theta \sum_{M \in M_{\text{train}}} \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_\theta \right]$$

- RLagent = RNN = generic computation architecture
  - different weights in the RNN means different RL algorithm and prior
  - different activations in the RNN means different current policy
  - meta-train objective can be optimized with an existing (slow) RL algorithm

43

- Simple neural attentive meta learner

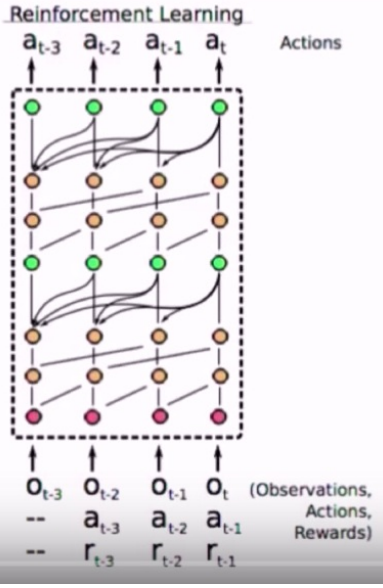
Simple Neural Attentive Meta-Learner

- Like RL2 but:

replace the LSTM with dilated temporal convolution (like wavenet) + attention

[Wavenet: van den Oord et al, 2016]

[Attention-is-all-you-need: Vaswani et al, 2017]



- MAML

Model-Agnostic Meta-Learning (MAML)

- Starting observation:
  - Computer vision practice:
    - Train on ImageNet [Deng et al. '09]
    - Fine-tune on actual task

    works really well! [Decaf: Donahue et al. '14; ...]

- Questions:
  - How to generalize this to behavior learning?
  - And can we explicitly train end-to-end for being maximally ready for efficient fine-tuning?

Model-Agnostic Meta-Learning (MAML)

Key idea: End-to-end learning of parameter vector θ that is good init for fine-tuning for many tasks

finetuned parameters

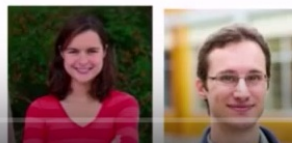
pretrained parameters

MAML test time: fine-tuning: $\theta' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{tr}}(\theta)$

train data for new task

MAML training: $\min_\theta \sum_{\text{tasks}} \mathcal{L}_{\text{val}} \left(\theta - \alpha \nabla_\theta \mathcal{L}_{\text{train}}(\theta)\right)$

finetuned parameters



- Related work
  - Classification:

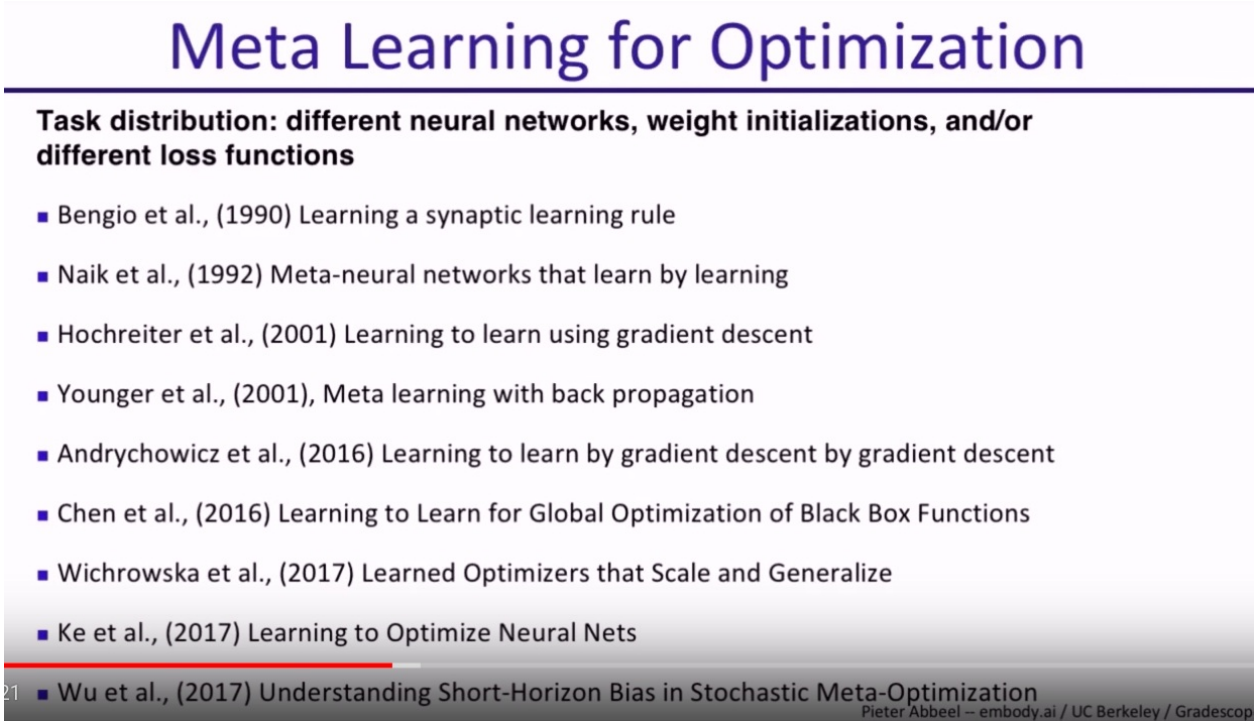
Meta Learning for Classification

Task distribution: different classification datasets (input: images, output: class labels)

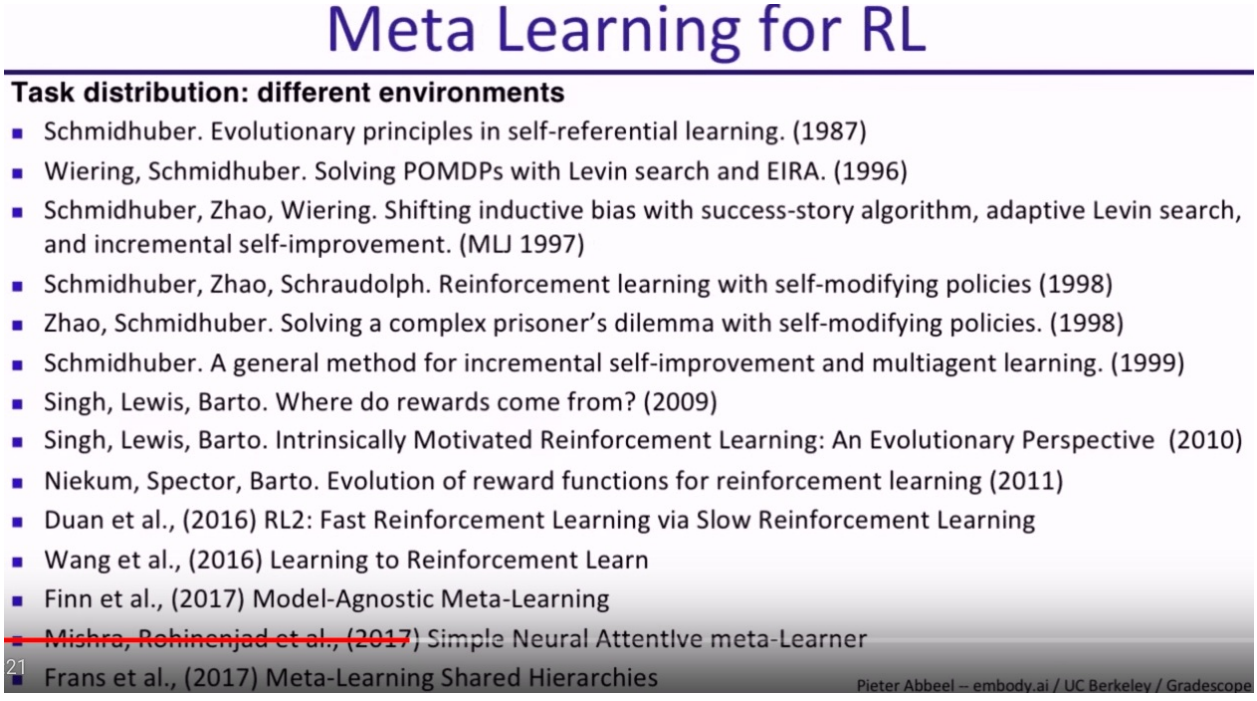
- Hochreiter et al., (2001) Learning to learn using gradient descent
- Younger et al., (2001), Meta learning with back propagation
- Koch et al., (2015) Siamese neural networks for one-shot image recognition
- Santoro et al., (2016) Meta-learning with memory-augmented neural networks
- Vinyals et al., (2016) Matching networks for one shot learning
- Edwards et al., (2016) Towards a Neural Statistician
- Ravi et al., (2017) Optimization as a model for few-shot learning
- Munkhdalai et al., (2017) Meta Networks
- Snell et al., (2017) Prototypical Networks for Few-shot Learning
- Shyam et al., (2017) Attentive Recurrent Comparators
- Finn et al., (2017) Model-Agnostic Meta-Learning for Fast Adaptation of Deep Netwc
- Mehrotra et al., (2017) Generative Adversarial Residual Pairwise Networks for One S
- Mishra et al., (2017) Meta-Learning with Temporal Convolutions
- Li et al., (2017) Meta-SGD: Learning to Learn Quickly for Few Shot Learning
- Finn and Levine, (2017) Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm
- Anon@OpenReview, (2017) Recasting Gradient-Based Meta-Learning as Hierarchical Bayes

Pieter Abbeel – embody.ai / UC Berkeley / Gradescope

  - Optimization:



- RL:



4. Long Horizon Reasoning : HRL

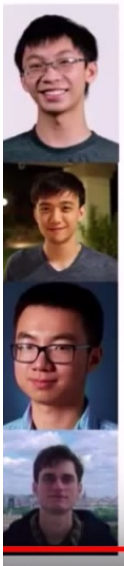
- Related work

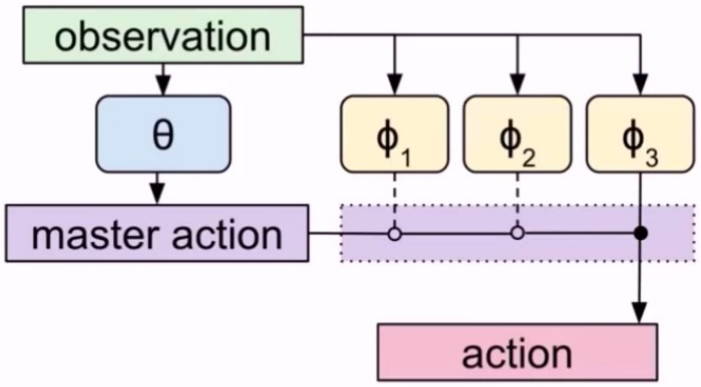
Hierarchical Reinforcement Learning

- Dayan and Hinton, 1993 Feudal RL
- Parr and Russell, 1998 RL with Hierarchies of Machines
- Sutton, Precup, Singh, 1999 Options
- Precup, 2000 Temporal abstraction in reinforcement learning
- Dietterich et al, 2000 MaxQ
- Fox, Moshkovitz, Tishby, 2016 Principled Option Learning
- Heess et al, 2016 Learning and transfer of modulated locomotor controllers
- Vezhnevets et al 2017 Feudal Networks for HRL
- Bacon, Harb, Precup, 2017 Option-Critic
- Florensa, Duan, Abbeel, 2017 SNNs for HRL
- Andreas, Klein, Levine, 2017 Policy Sketches

- MLSH

Meta-Learning Shared Hierarchies





Goal: find subpolicies that enable fast learning of master policy θ

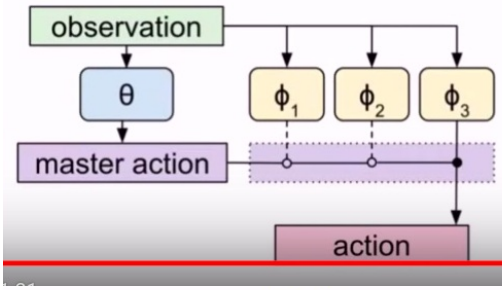
[Frans, Ho, Chen, Abbeel, Schulman, 2017]

RL2 Meta-Learning Objective:

$$\max_{\theta} \mathbb{E}_M \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \text{RLagent}_\theta \right]$$

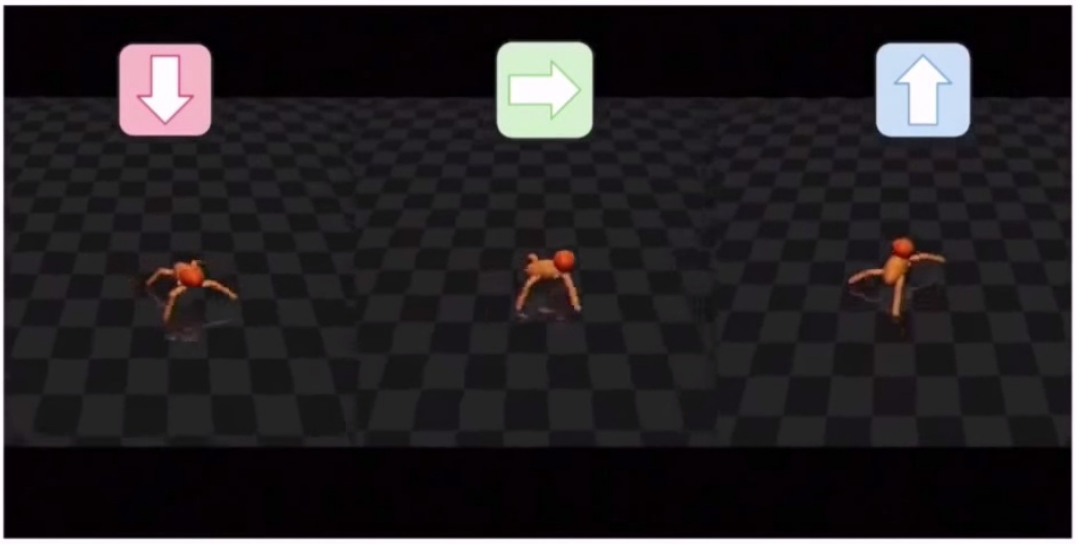
MLSH Meta-Learning Objective:

$$\max_{\phi} \mathbb{E}_{\theta_0} \mathbb{E}_M \mathbb{E}_{\tau_M^{(k)}} \left[\sum_{k=1}^{K} R(\tau_M^{(k)}) \mid \phi, \text{RLagent}_{\theta_0} \right]$$



= find a set of subpolicies that enable fast learning of the master policy

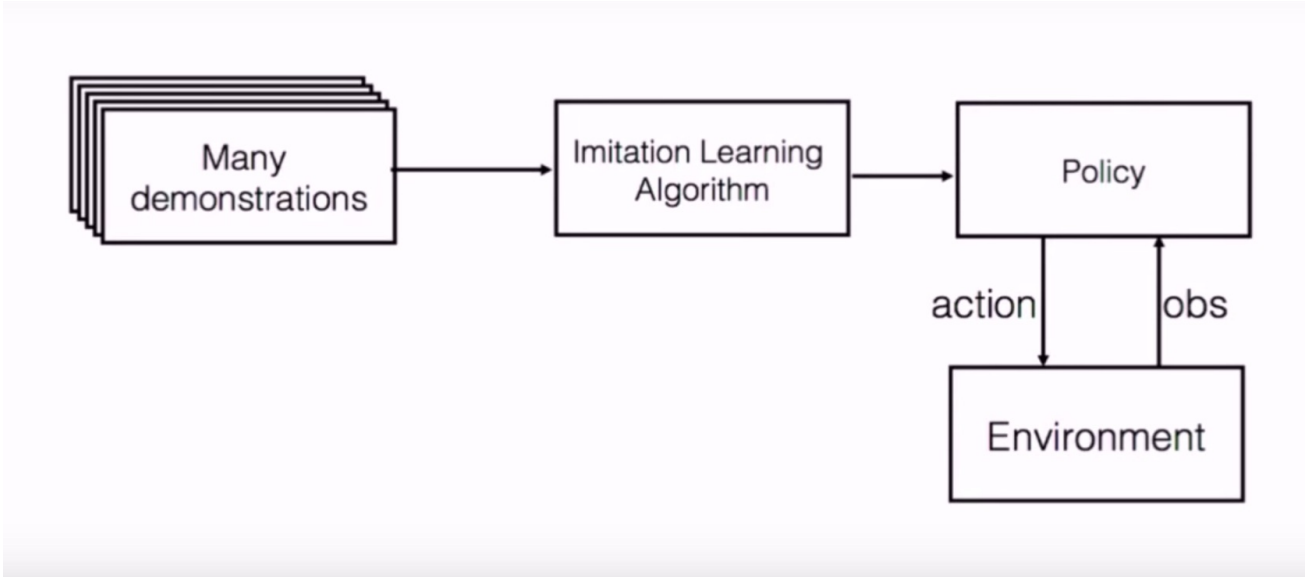
Discovered Three Gaits



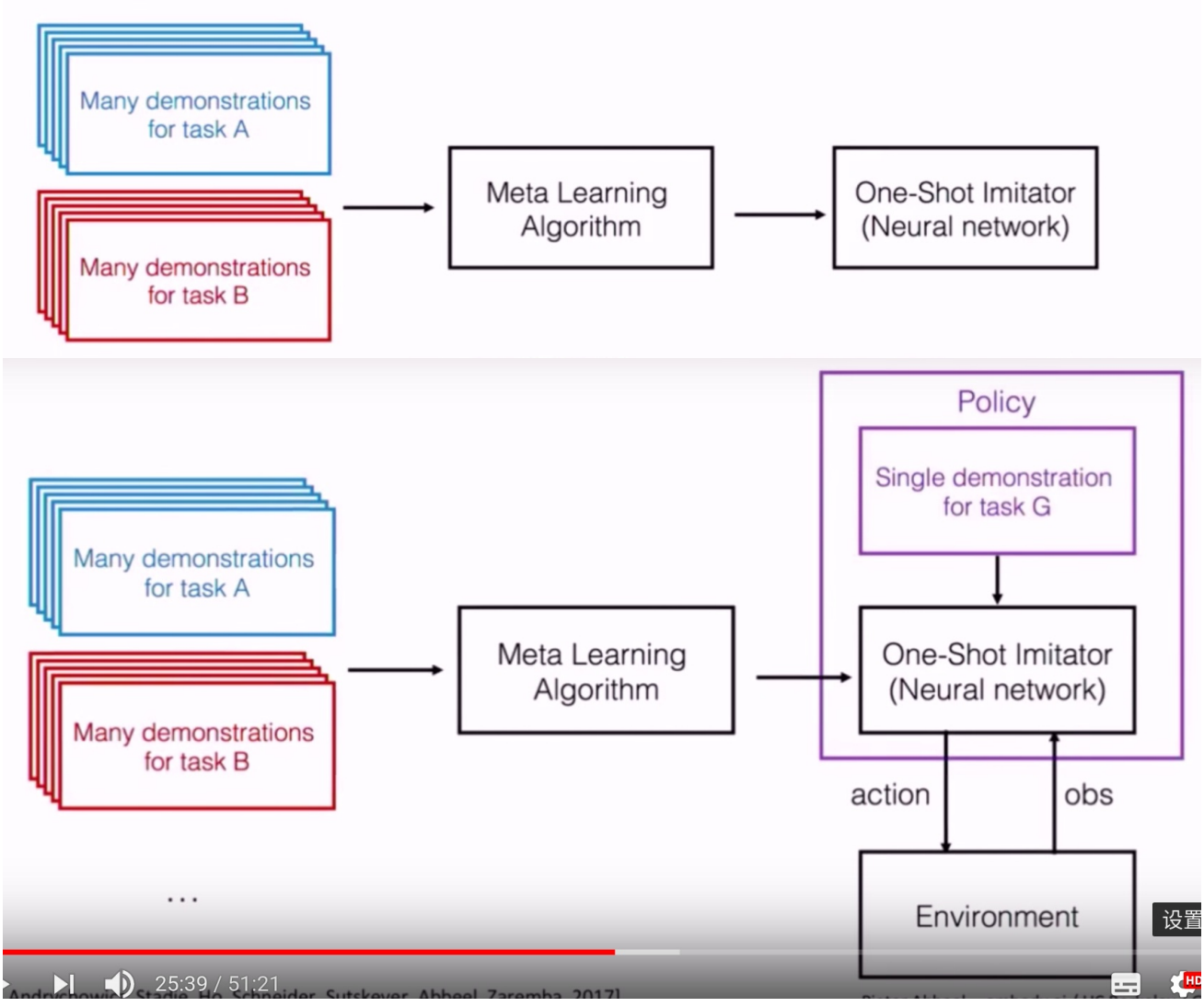
MLSH agent was trained on nine separate mazes.
It discovered sub-policies for upwards, rightwards, and downwards movement.

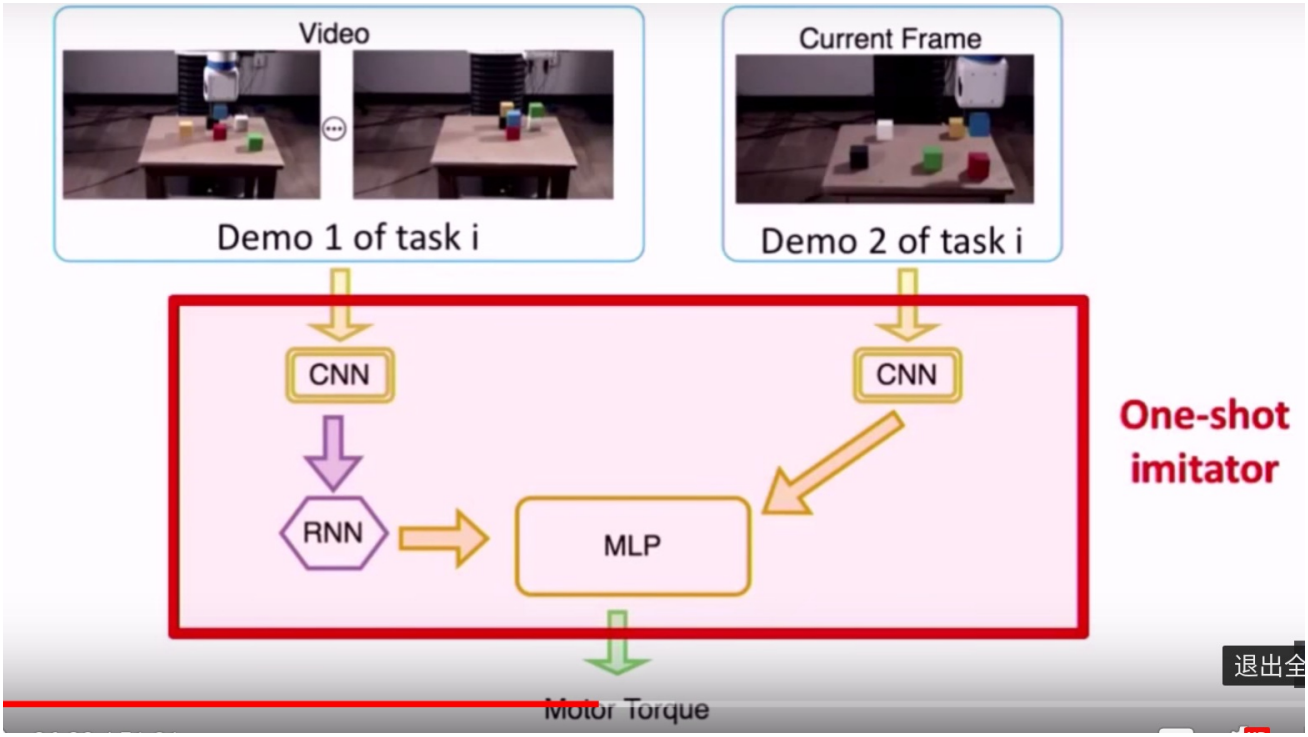
5. Imitation Learning

- Typical IL

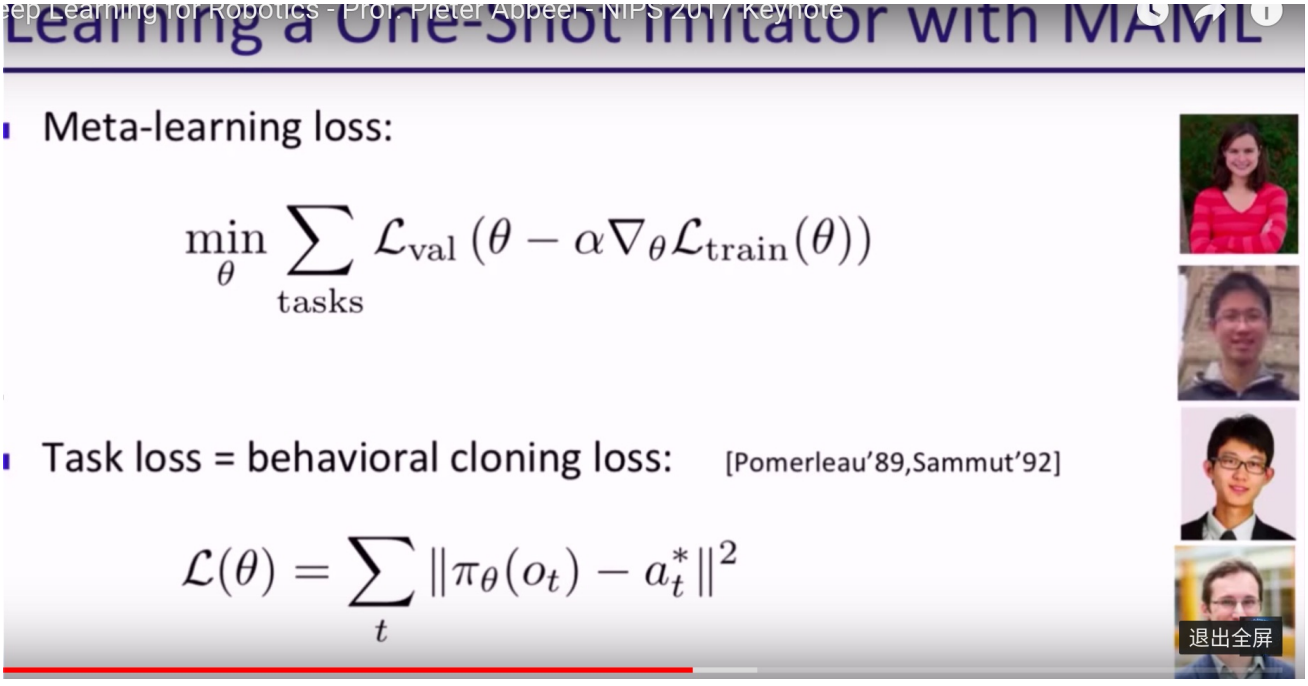


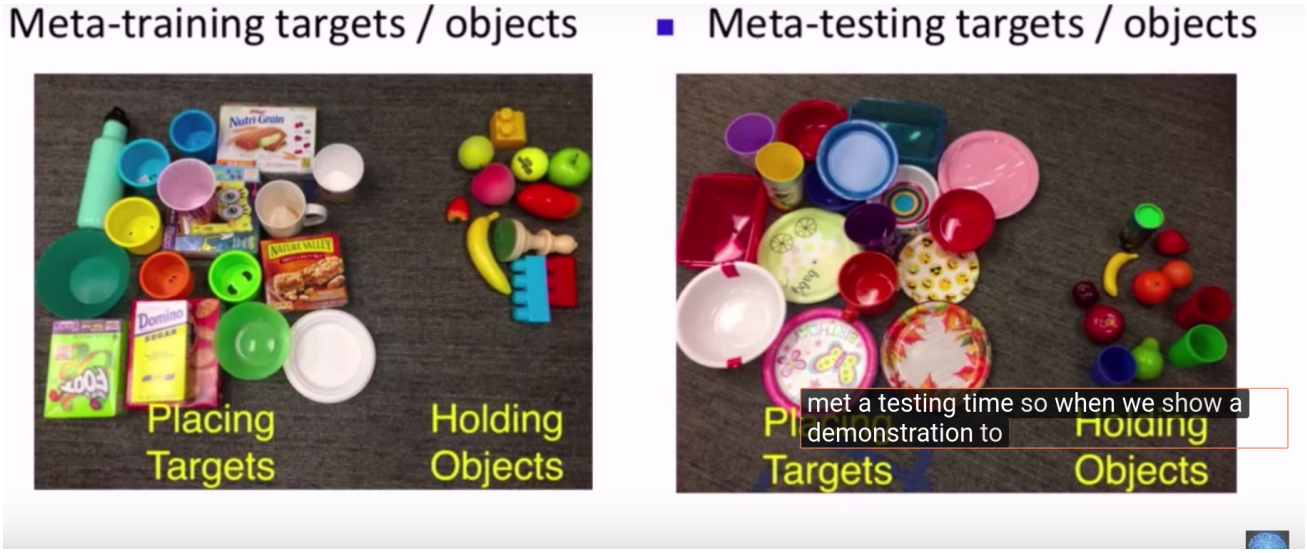
- One shot imitation learning




Video
Demo 1 of task i
Current Frame
Demo 2 of task i
CNN
CNN
RNN
MLP
Motor Torque
One-shot imitator
退出全

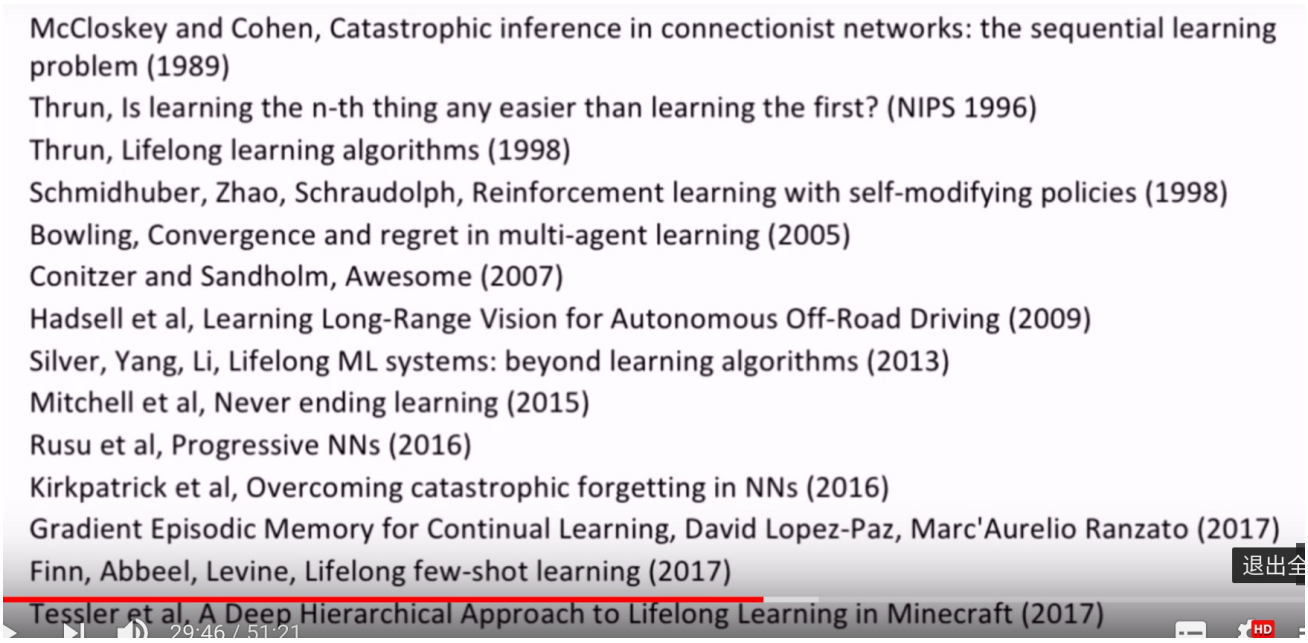
- MIL


Learning a One-Shot Imitator with MAML
Meta-learning loss:
$$\min_{\theta} \sum_{\text{tasks}} \mathcal{L}_{\text{val}} \left(\theta - \alpha \nabla_{\theta} \mathcal{L}_{\text{train}}(\theta)\right)$$
Task loss = behavioral cloning loss: [Pomerleau'89,Sammut'92]
$$\mathcal{L}(\theta) = \sum_{t} \|\pi_{\theta}(o_t) - a_t^*\|^2$$
退出全屏



6. Lifelong Learning

- Related work



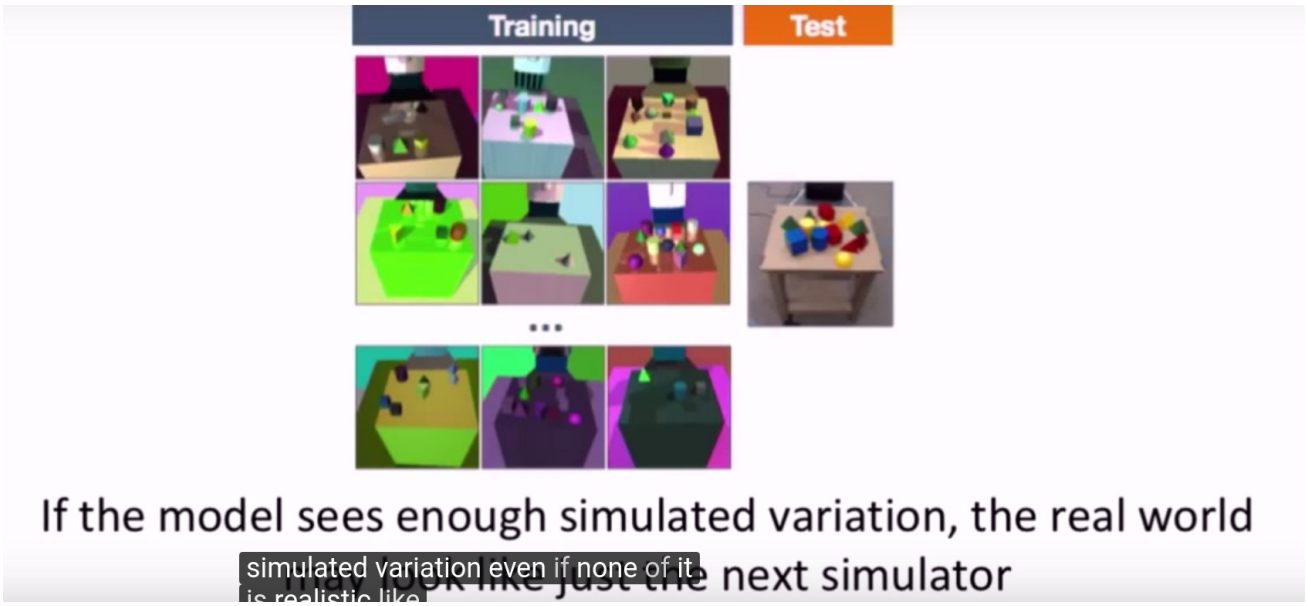
- What is lifelong learning

- Current machine learning paradigm:
  - Step 1: Run machine learning
  - Step 2: Deploy
  - All learning happens ahead of time
- BUT: real-world deployment will face ever changing situations

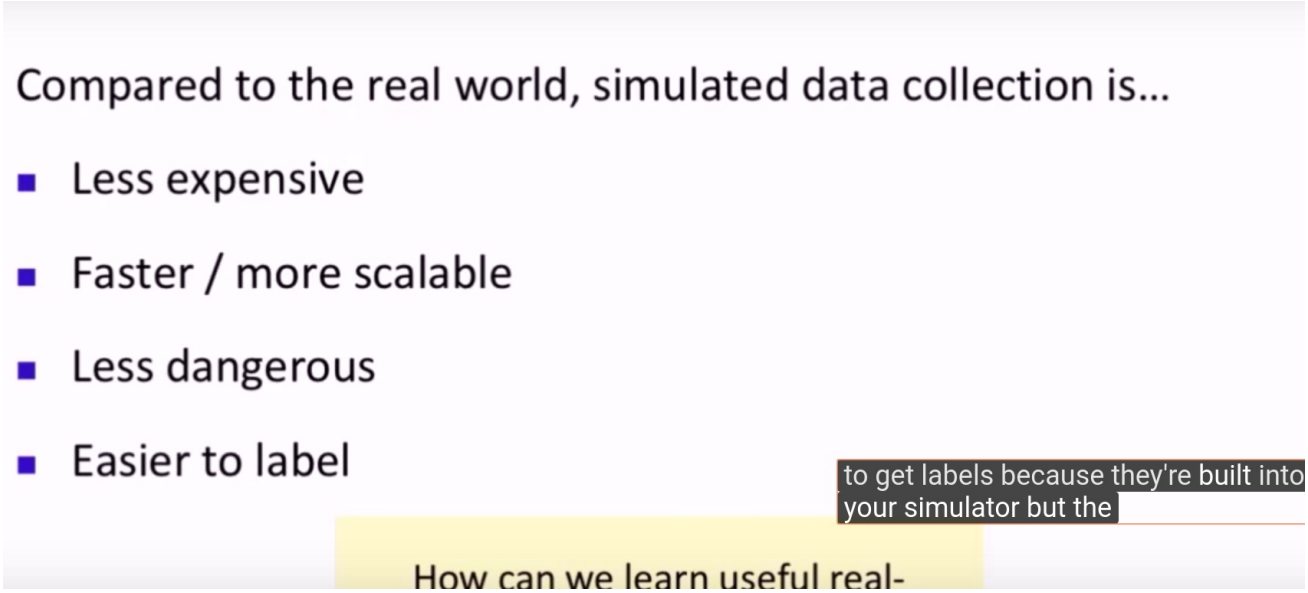
  → requires learning during deployment

7. Leverage Simulation

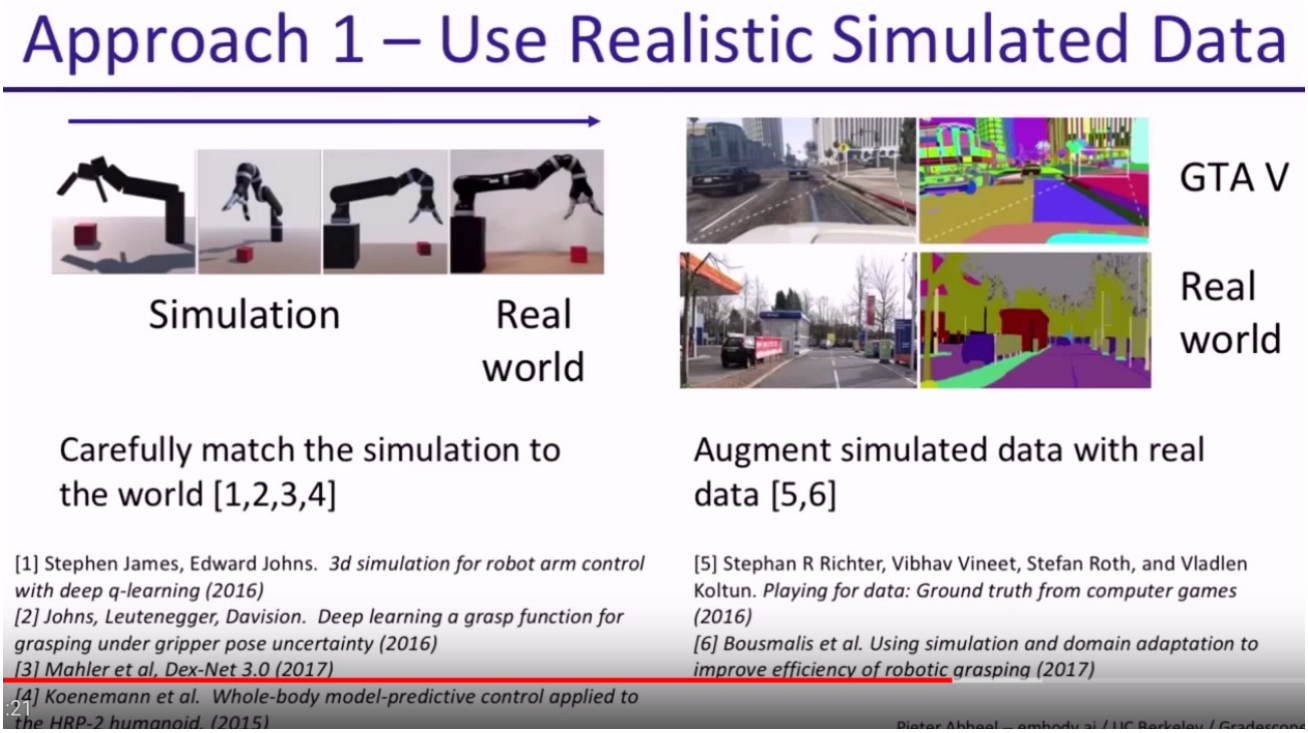
- Domain randomization



- Sim-to-real



- Related work



8. Review

- Meta-learning can be useful for other methods

- Long Horizon R… …ignment
- Taskability … + Learning
- Lifelong Learning …
- Leve…
- Maximi…

Recurring Theme:

Meta-Learning

Enables discovering algorithms that are powered by data/experience (vs. just human ingenuity)

Requires more compute
Which is something we are continuing to get