

IASD M2 at Paris Dauphine

Deep Reinforcement Learning

23: Meta-Learning

Eric Benhamou David Saliel



Acknowledgement

These materials are based on the seminal course of Sergey Levine CS285

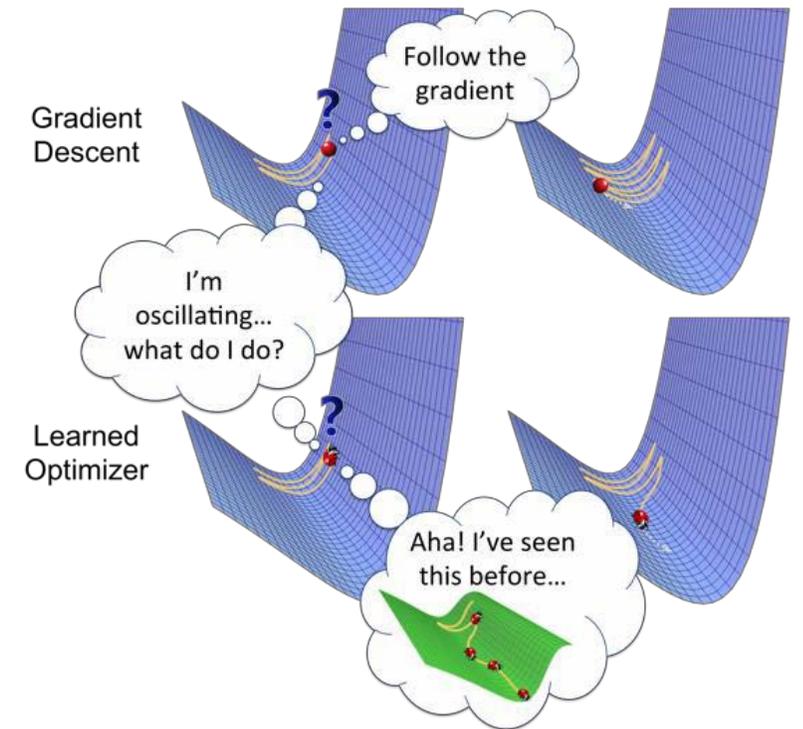


So far...

- Forward transfer: source domain to target domain
 - Diversity is good! The more varied the training, the more likely transfer is to succeed
- Multi-task learning: even more variety
 - No longer training on the same kind of task
 - But more variety = more likely to succeed at transfer
- How do we represent transfer knowledge?
 - Model (as in model-based RL): rules of physics are conserved across tasks
 - Policies – requires finetuning, but closer to what we want to accomplish
 - What about *learning methods*?

What is meta-learning?

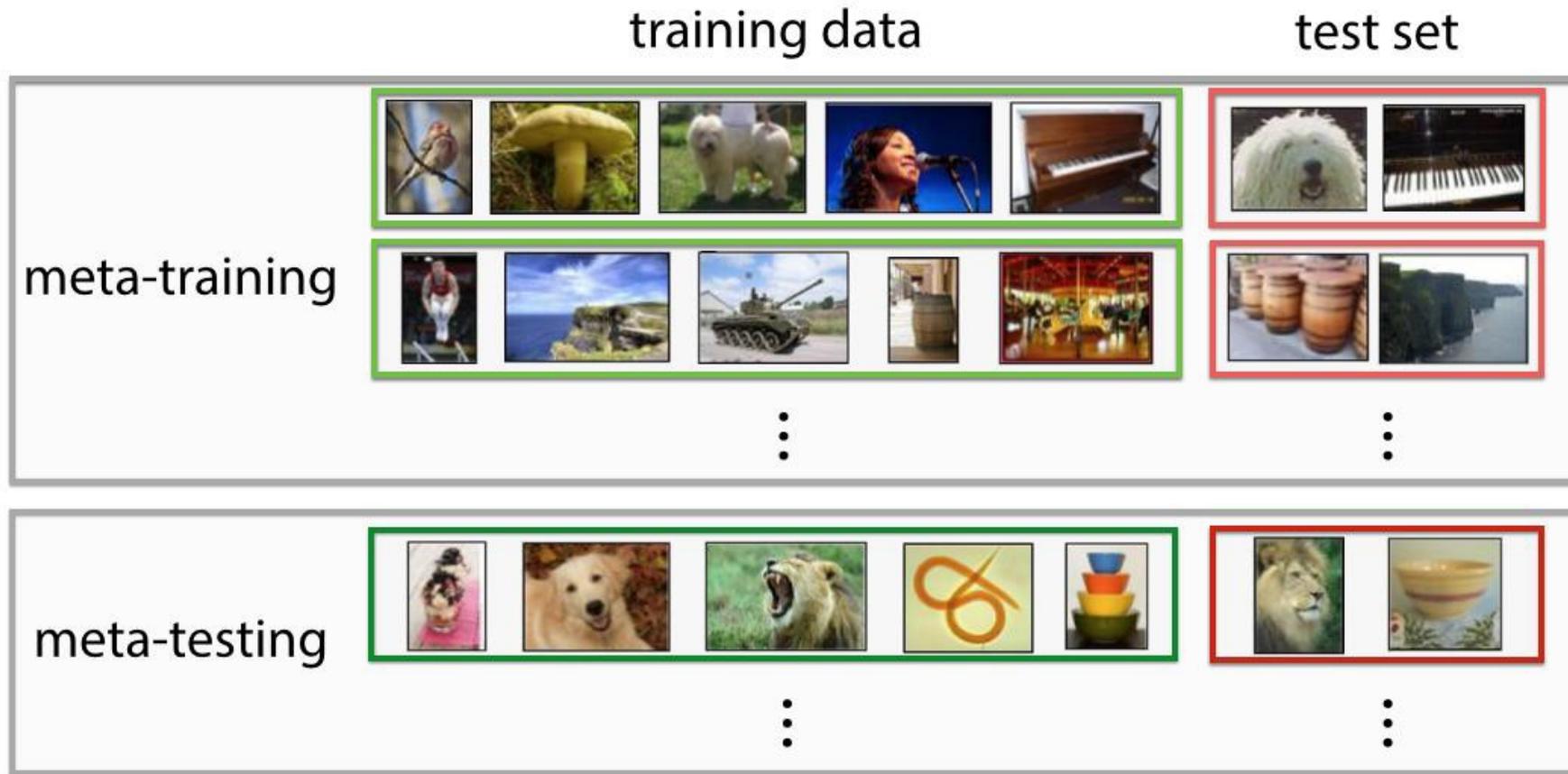
- If you've learned 100 tasks already, can you figure out how to *learn* more efficiently?
 - Now having multiple tasks is a huge advantage!
- Meta-learning = *learning to learn*
- In practice, very closely related to multi-task learning
- Many formulations
 - Learning an optimizer
 - Learning an RNN that ingests experience
 - Learning a representation



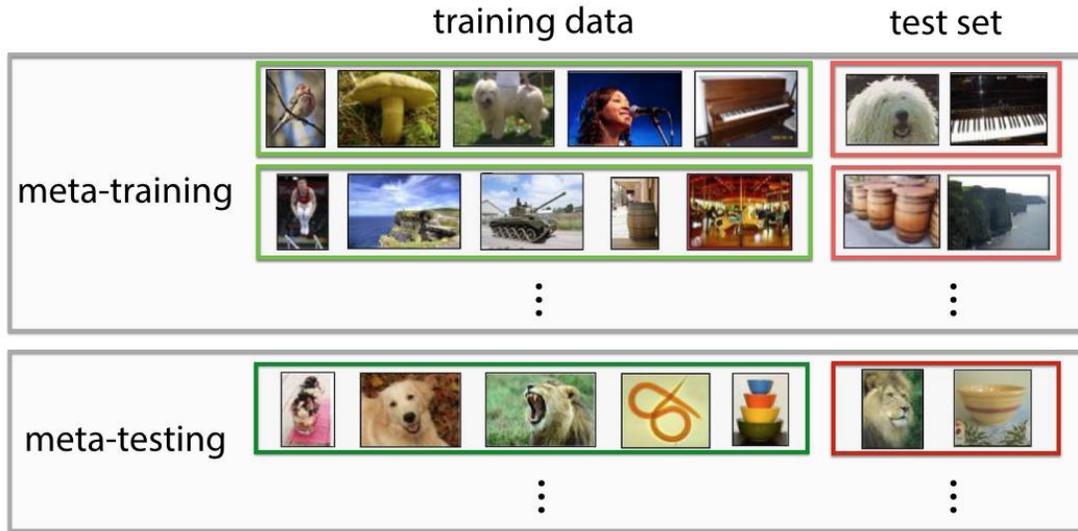
Why is meta-learning a good idea?

- Deep reinforcement learning, especially model-free, requires a huge number of samples
- If we can *meta-learn* a faster reinforcement learner, we can learn new tasks efficiently!
- What can a *meta-learned* learner do differently?
 - Explore more intelligently
 - Avoid trying actions that are known to be useless
 - Acquire the right features more quickly

Meta-learning with supervised learning



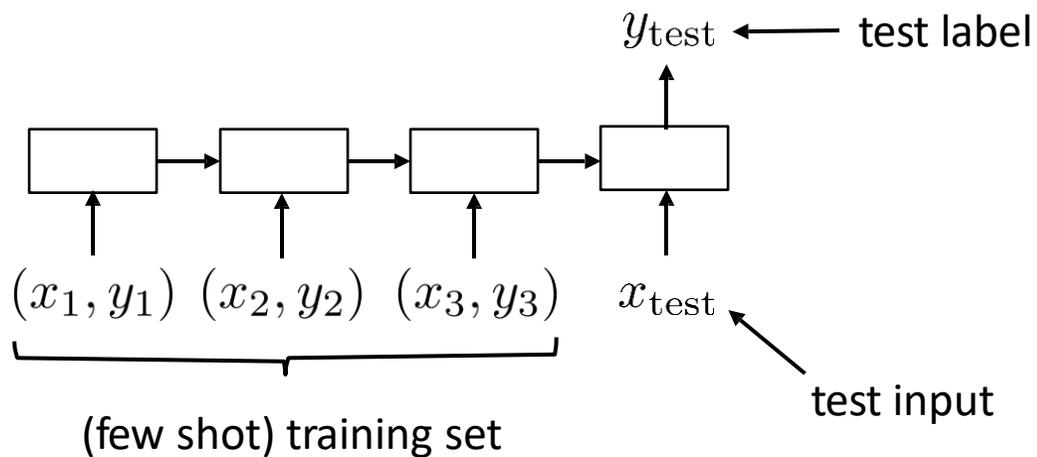
Meta-learning with supervised learning



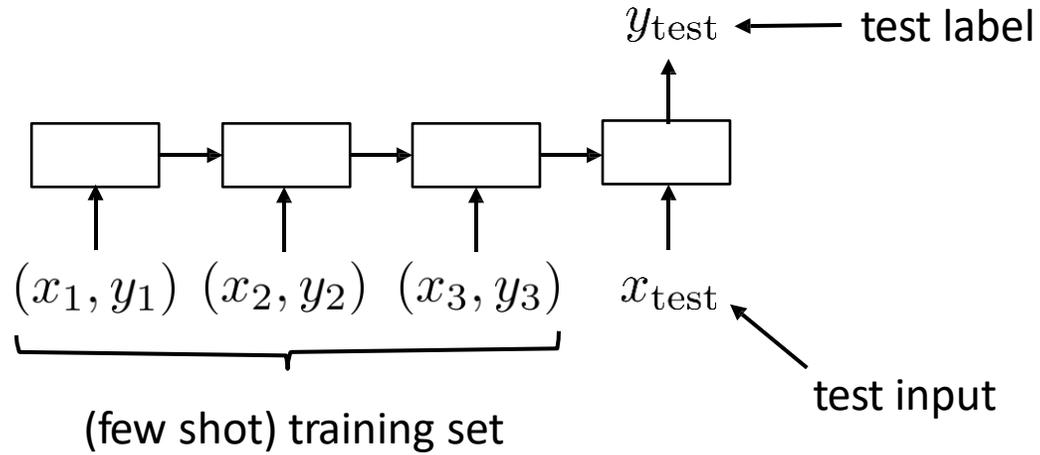
supervised learning: $f(x) \rightarrow y$
input (e.g., image) output (e.g., label)

supervised meta-learning: $f(\mathcal{D}^{\text{tr}}, x) \rightarrow y$
training set

- How to read in training set?
 - Many options, RNNs can work
 - More on this later



What is being “learned”?



supervised meta-learning: $f(\mathcal{D}^{\text{tr}}, x) \rightarrow y$

“Generic” learning:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

$$= f_{\text{learn}}(\mathcal{D}^{\text{tr}})$$

“Generic” meta-learning:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}})$$

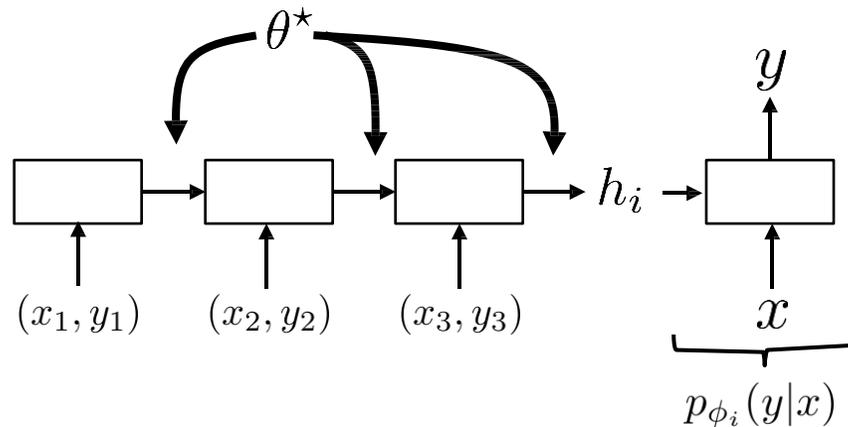
$$\text{where } \phi_i = f_{\theta}(\mathcal{D}_i^{\text{tr}})$$

What is being “learned”?

“Generic” learning:

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta, \mathcal{D}^{\text{tr}})$$

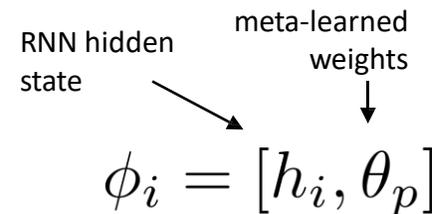
$$= f_{\text{learn}}(\mathcal{D}^{\text{tr}})$$



“Generic” meta-learning:

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}})$$

$$\text{where } \phi_i = f_{\theta}(\mathcal{D}_i^{\text{tr}})$$



Meta Reinforcement Learning

The meta reinforcement learning problem

“Generic” learning:

$$\begin{aligned}\theta^* &= \arg \min_{\theta} \mathcal{L}(\theta, \mathcal{D}^{\text{tr}}) \\ &= f_{\text{learn}}(\mathcal{D}^{\text{tr}})\end{aligned}$$

Reinforcement learning:

$$\begin{aligned}\theta^* &= \arg \max_{\theta} E_{\pi_{\theta}(\tau)}[R(\tau)] \\ &= f_{\text{RL}}(\mathcal{M}) \quad \mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}\end{aligned}$$

↑
MDP

“Generic” meta-learning:

$$\begin{aligned}\theta^* &= \arg \min_{\theta} \sum_{i=1}^n \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}}) \\ &\text{where } \phi_i = f_{\theta}(\mathcal{D}_i^{\text{tr}})\end{aligned}$$

Meta-reinforcement learning:

$$\begin{aligned}\theta^* &= \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)}[R(\tau)] \\ &\text{where } \phi_i = f_{\theta}(\mathcal{M}_i)\end{aligned}$$

↑
MDP for task i

The meta reinforcement learning problem

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

where $\phi_i = f_{\theta}(\mathcal{M}_i)$

assumption: $\mathcal{M}_i \sim p(\mathcal{M})$

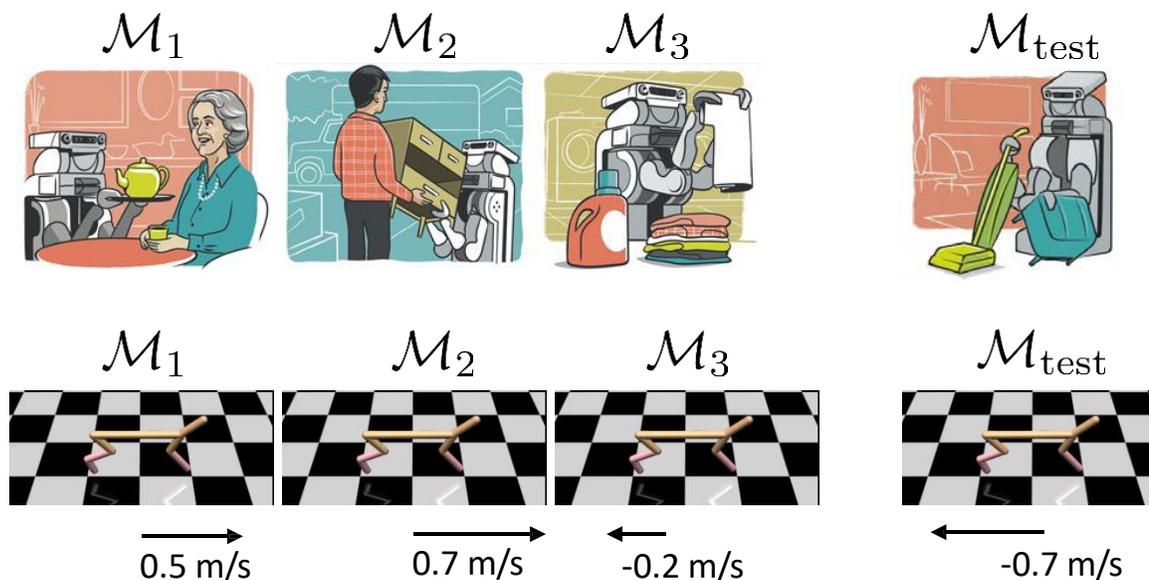
meta test-time:

sample $\mathcal{M}_{\text{test}} \sim p(\mathcal{M})$, get $\phi_i = f_{\theta}(\mathcal{M}_{\text{test}})$

$\{\mathcal{M}_1, \dots, \mathcal{M}_n\}$

\nwarrow
meta-training MDPs

Some examples:



Contextual policies and meta-learning

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

where $\phi_i = f_{\theta}(\mathcal{M}_i)$



$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\theta}} [R(\tau)]$$

$$\pi_{\theta}(a_t | s_t, \underbrace{s_1, a_1, r_1, \dots, s_{t-1}, a_{t-1}, r_{t-1}}_{\text{context}})$$

context used to infer whatever we need to solve \mathcal{M}_i
i.e., z_t or ϕ_i (which are really the same thing)

in meta-RL, the *context* is inferred from experience from \mathcal{M}_i

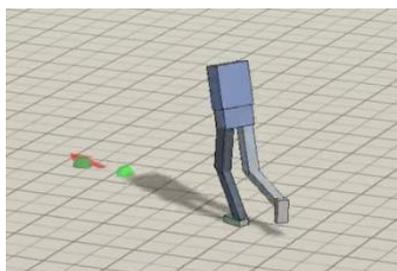
in multi-task RL, the context is typically given

$$\pi_{\theta}(a_t | s_t, \phi_i)$$

↑
“context”



ϕ : stack location



ϕ : walking direction



ϕ : where to hit puck

Meta-RL with recurrent policies

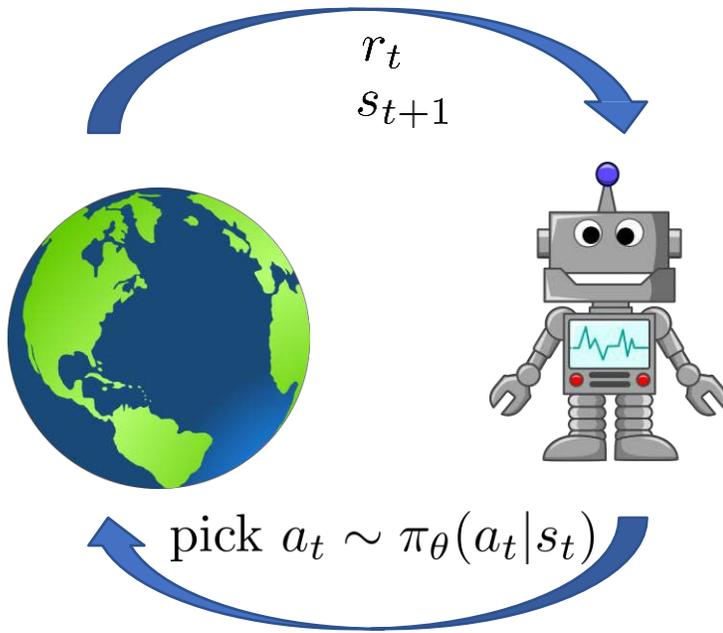
$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

where $\phi_i = f_{\theta}(\mathcal{M}_i)$

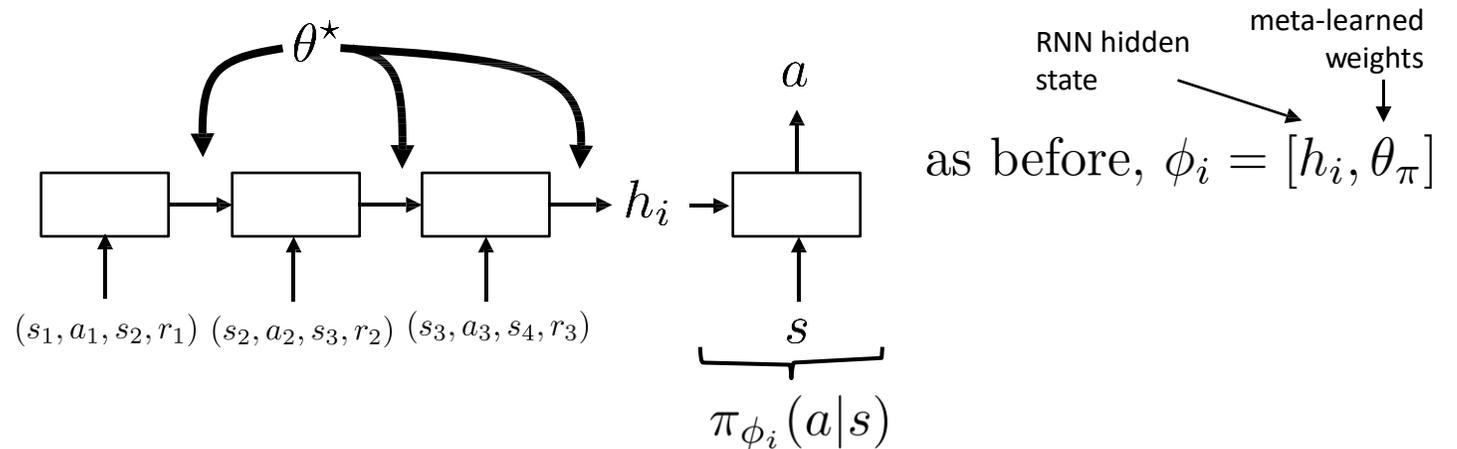
main question: how to implement $f_{\theta}(\mathcal{M}_i)$?

what should $f_{\theta}(\mathcal{M}_i)$ do?

1. improve policy with experience from \mathcal{M}_i
 $\{(s_1, a_1, s_2, r_1), \dots, (s_T, a_T, s_{T+1}, r_T)\}$
2. (new in RL): choose how to interact, i.e. choose a_t
 meta-RL must also *choose* how to *explore*!



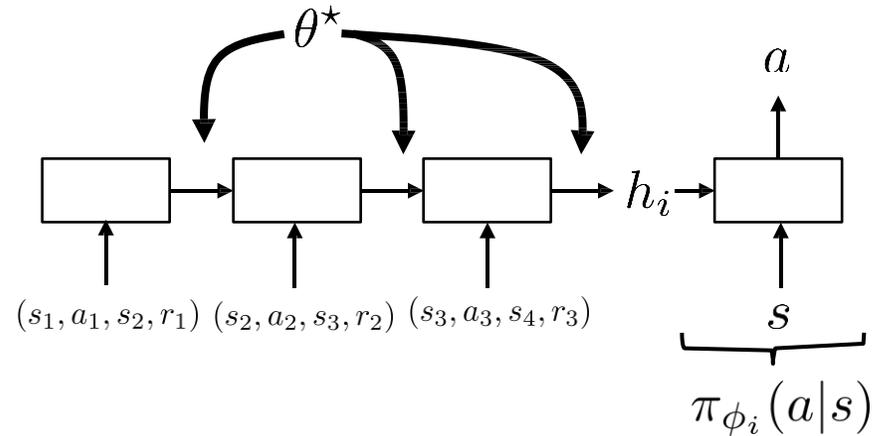
use (s_t, a_t, s_{t+1}, r_t) to improve π_{θ}



Meta-RL with recurrent policies

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

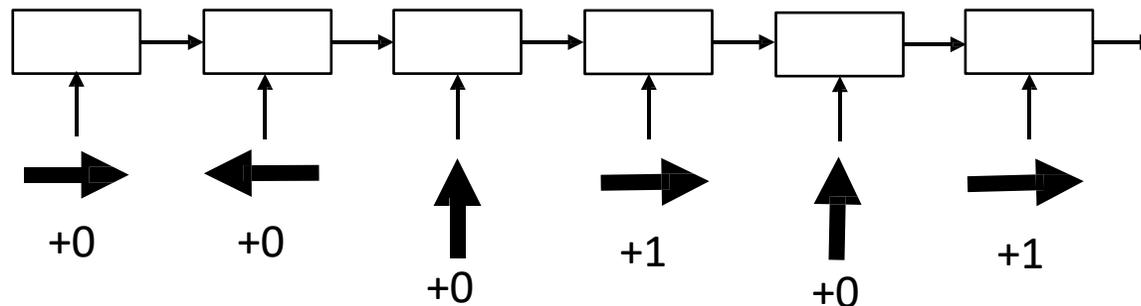
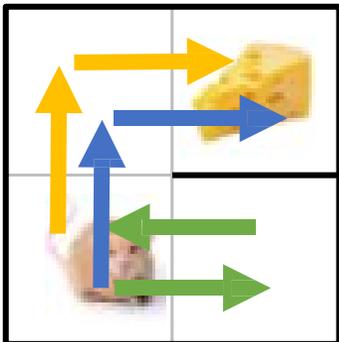
where $\phi_i = f_{\theta}(\mathcal{M}_i)$



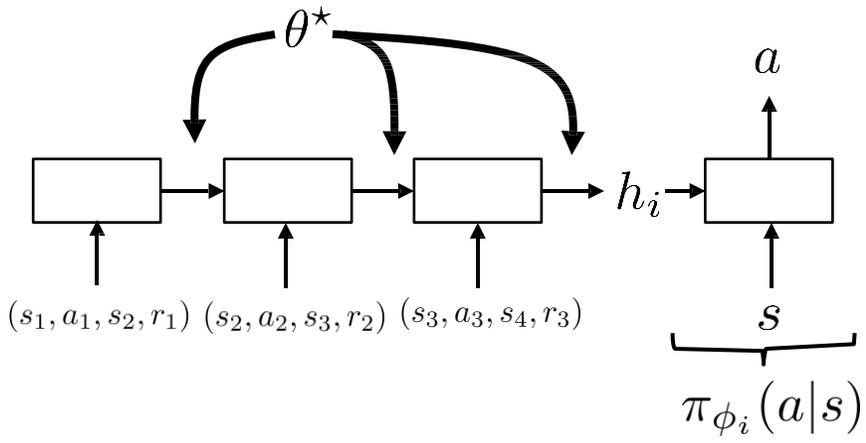
so... we just train an RNN policy?

yes!

crucially, RNN hidden state is **not** reset between episodes!

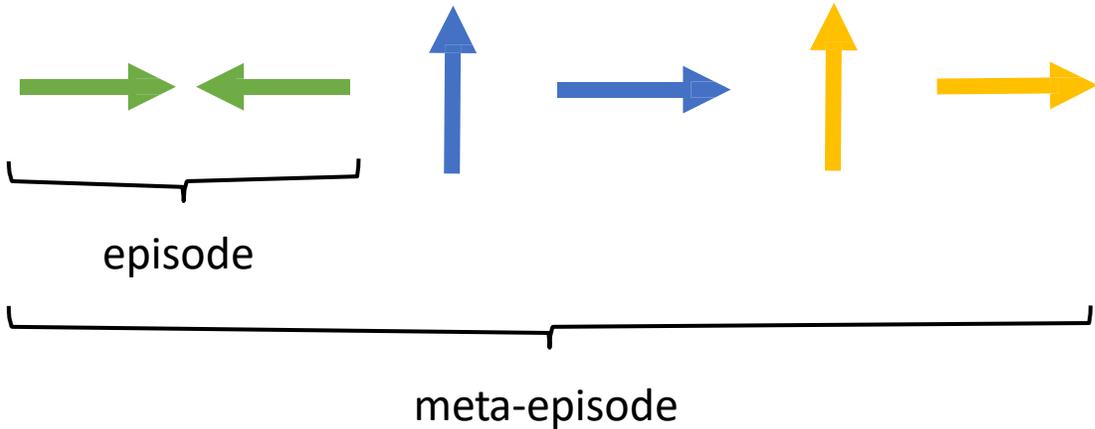
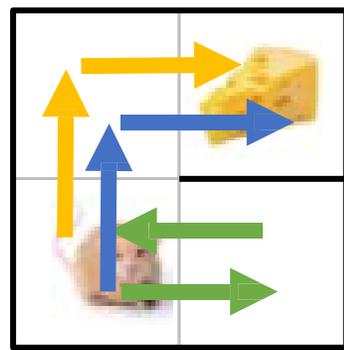


Why recurrent policies *learn to explore*



1. improve policy with experience from \mathcal{M}_i
 $\{(s_1, a_1, s_2, r_1), \dots, (s_T, a_T, s_{T+1}, r_T)\}$
2. (new in RL): choose how to interact, i.e. choose a_t
 meta-RL must also *choose* how to *explore*!

$$\theta^* = \arg \max_{\theta} E_{\pi_{\theta}} \left[\sum_{t=0}^T r(s_t, a_t) \right]$$

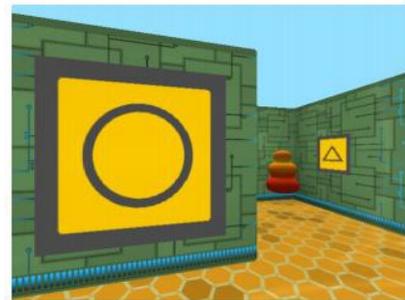
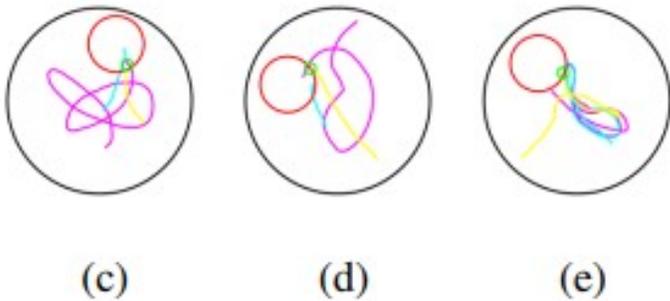
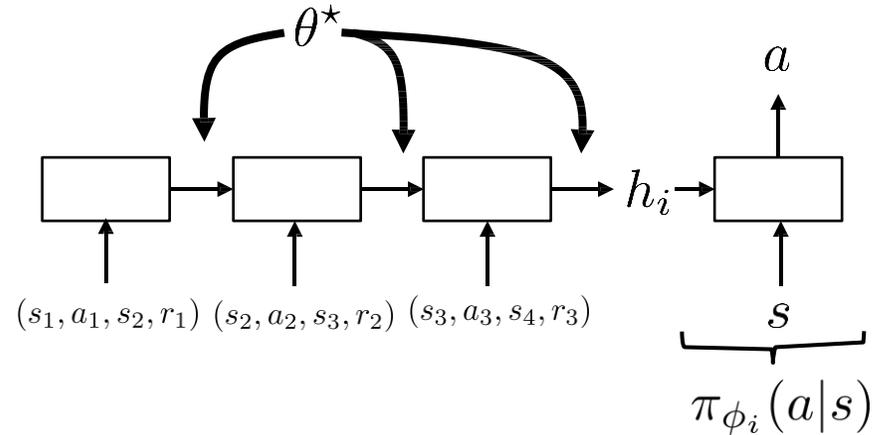


optimizing total reward over the entire **meta-episode** with RNN policy **automatically** learns to explore!

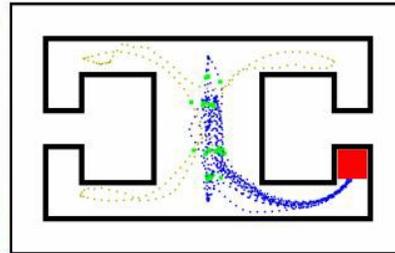
Meta-RL with recurrent policies

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

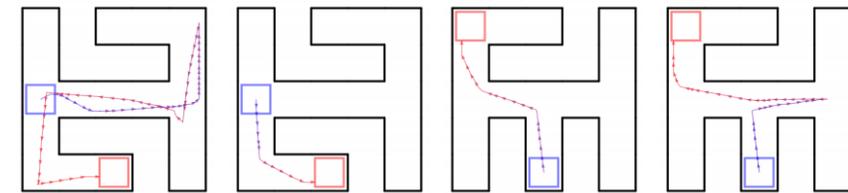
where $\phi_i = f_{\theta}(\mathcal{M}_i)$



(a) Labryinth I-maze



(b) Illustrative Episode



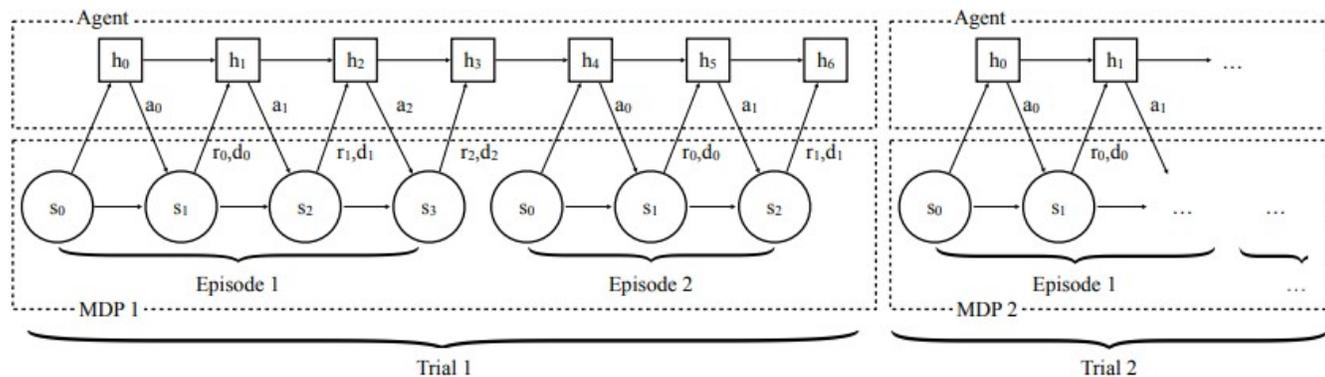
(a) Good behavior, 1st episode (b) Good behavior, 2nd episode (c) Bad behavior, 1st episode (d) Bad behavior, 2nd episode

Heess, Hunt, Lillicrap, Silver. **Memory-based control with recurrent neural networks**. 2015.

Wang, Kurth-Nelson, Tirumala, Soyer, Leibo, Munos, Blundell, Kumaran, Botvinick. **Learning to Reinforcement Learning**. 2016.

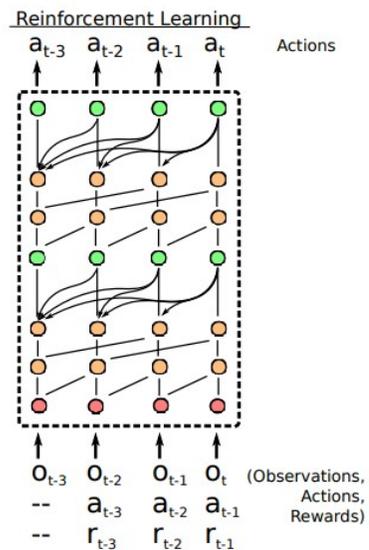
Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel. **RL2: Fast Reinforcement Learning via Slow Reinforcement Learning**. 2016.

Architectures for meta-RL



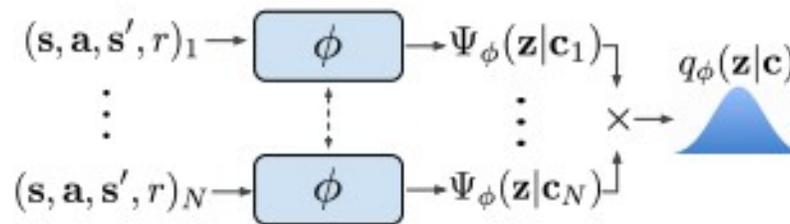
standard RNN (LSTM) architecture

Duan, Schulman, Chen, Bartlett, Sutskever, Abbeel. **RL2: Fast Reinforcement Learning via Slow Reinforcement Learning**. 2016.



attention + temporal convolution

Mishra, Rohaninejad, Chen, Abbeel. **A Simple Neural Attentive Meta-Learner**.

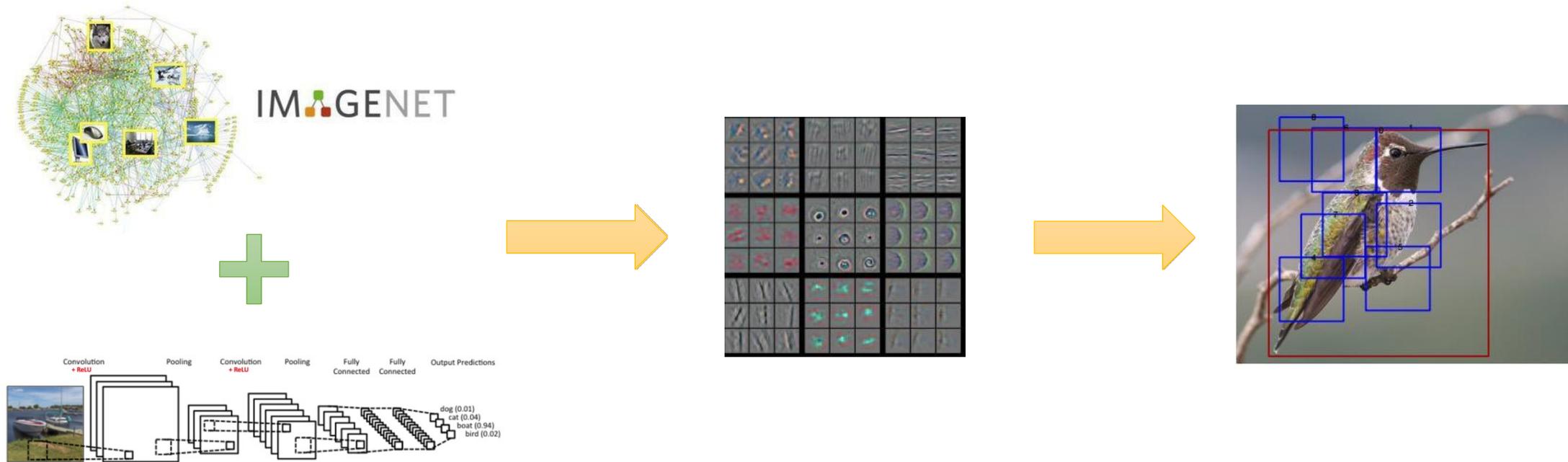


parallel permutation-invariant context encoder

Rakelly*, Zhou*, Quillen, Finn, Levine. **Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables**.

Gradient-Based Meta-Learning

Back to representations...



is pretraining a *type* of meta-learning? better features = faster learning of new task!

Meta-RL as an optimization problem

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

where $\phi_i = f_{\theta}(\mathcal{M}_i)$

1. improve policy with experience from \mathcal{M}_i
 $\{(s_1, a_1, s_2, r_1), \dots, (s_T, a_T, s_{T+1}, r_T)\}$

what if $f_{\theta}(\mathcal{M}_i)$ is *itself* an RL algorithm?

$$f_{\theta}(\mathcal{M}_i) = \theta + \alpha \nabla_{\theta} \underbrace{J_i(\theta)}$$

requires interacting with \mathcal{M}_i

to estimate $\nabla_{\theta} E_{\pi_{\theta}} [R(\tau)]$

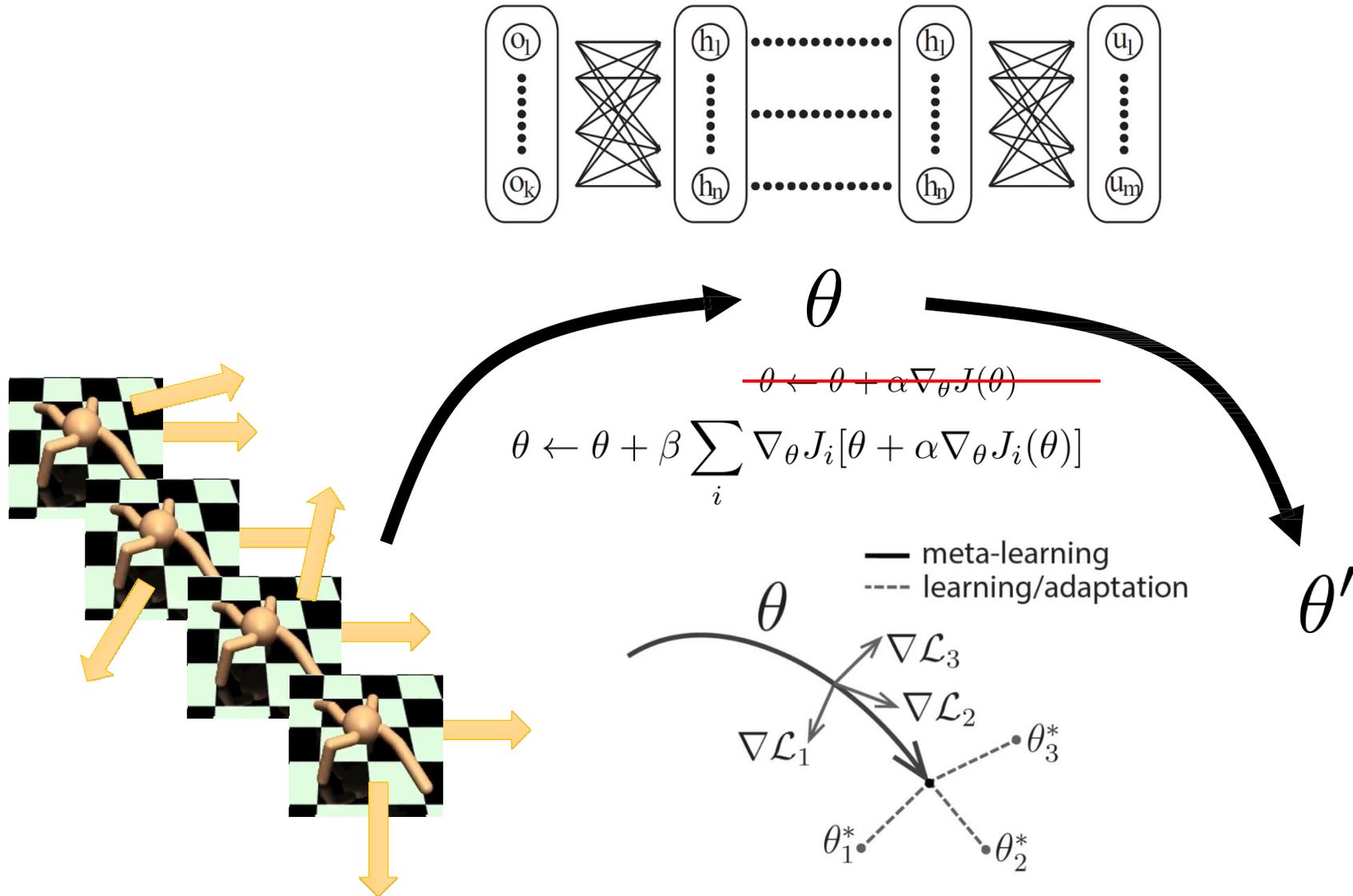
standard RL:

$$\theta^* = \arg \max_{\theta} \underbrace{E_{\pi_{\theta}(\tau)} [R(\tau)]}_{J(\theta)}$$

$$\theta^{k+1} \leftarrow \theta_k + \alpha \nabla_{\theta^k} J(\theta^k)$$

this is model-agnostic meta-learning (MAML) for RL!

MAML for RL in pictures



What did we just do??

supervised learning: $f(x) \rightarrow y$

supervised meta-learning: $f(\mathcal{D}^{\text{tr}}, x) \rightarrow y$

model-agnostic meta-learning: $f_{\text{MAML}}(\mathcal{D}^{\text{tr}}, x) \rightarrow y$

$$f_{\text{MAML}}(\mathcal{D}^{\text{tr}}, x) = f_{\theta'}(x)$$

$$\theta' = \theta - \alpha \sum_{(x,y) \in \mathcal{D}^{\text{tr}}} \nabla_{\theta} \mathcal{L}(f_{\theta}(x), y)$$

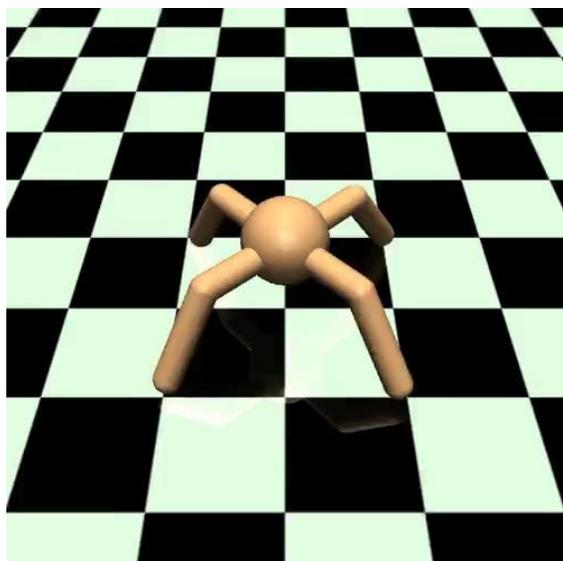
Just another computation graph...

Can implement with any autodiff package (e.g., TensorFlow)

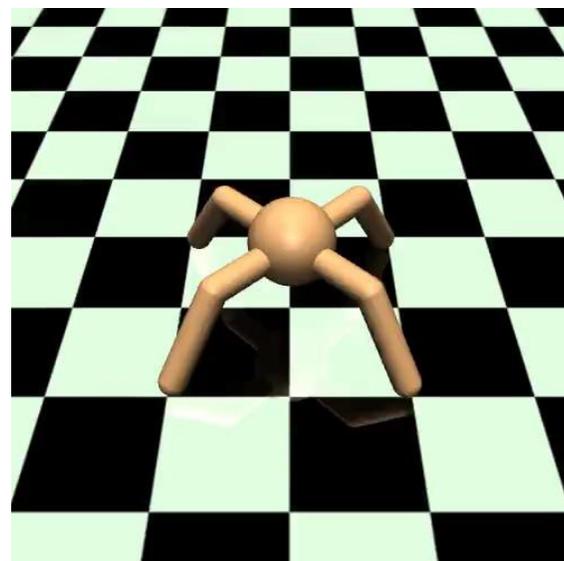
But has favorable inductive bias...

MAML for RL in videos

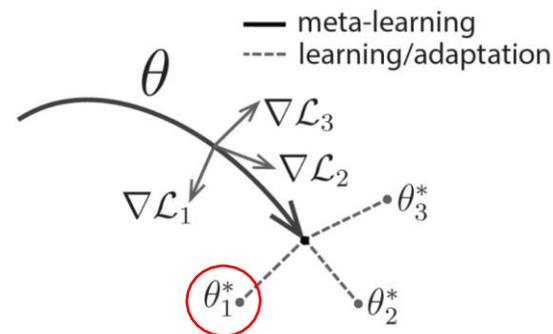
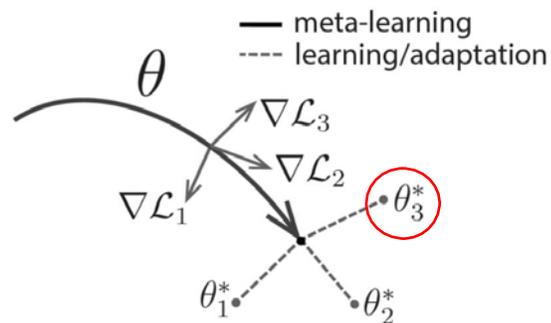
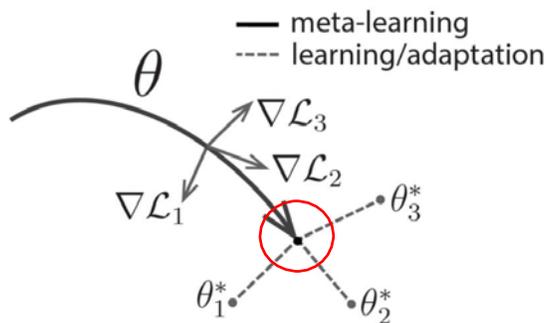
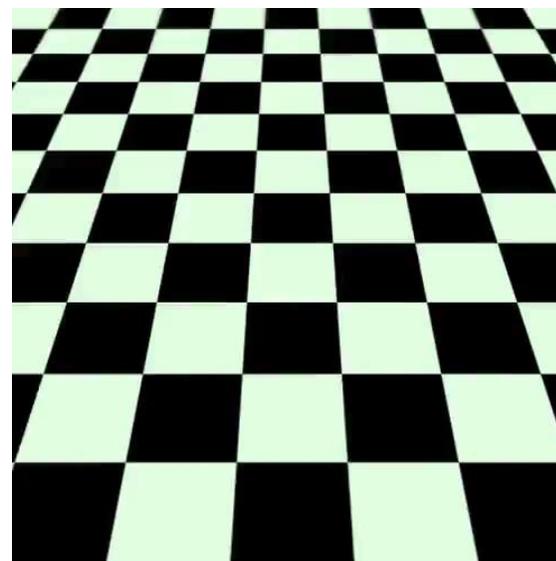
after MAML training



after 1 gradient step
(forward reward)



after 1 gradient step
(backward reward)



More on MAML/gradient-based meta-learning for RL

MAML meta-policy gradient estimators:

- Finn, Abbeel, Levine. **Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.**
- Foerster, Farquhar, Al-Shedivat, Rocktaschel, Xing, Whiteson. **DiCE: The Infinitely Differentiable Monte Carlo Estimator.**
- Rothfuss, Lee, Clavera, Asfour, Abbeel. **ProMP: Proximal Meta-Policy Search.**

Improving exploration:

- Gupta, Mendonca, Liu, Abbeel, Levine. **Meta-Reinforcement Learning of Structured Exploration Strategies.**
- Stadie*, Yang*, Houthoof, Chen, Duan, Wu, Abbeel, Sutskever. **Some Considerations on Learning to Explore via Meta-Reinforcement Learning.**

Hybrid algorithms (not necessarily gradient-based):

- Houthoof, Chen, Isola, Stadie, Wolski, Ho, Abbeel. **Evolved Policy Gradients.**
- Fernando, Sygnowski, Osindero, Wang, Schaul, Teplyashin, Sprechmann, Pirtzel, Rusu. **Meta-Learning by the Baldwin Effect.**

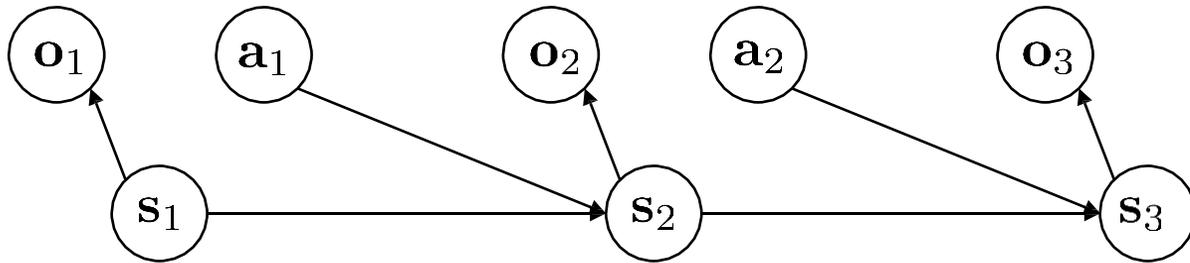
Meta-RL as a POMDP

Meta-RL as... partially observed RL?

$$\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{E}, r\}$$

\mathcal{O} – observation space observations $o \in \mathcal{O}$ (discrete or continuous)

\mathcal{E} – emission probability $p(o_t|s_t)$



policy must act on observations o_t !

$$\pi_{\theta}(a|o)$$

typically requires *either*:

explicit state estimation, i.e. to estimate $p(s_t|o_{1:t})$

policies with memory

Meta-RL as... partially observed RL?

$$\pi_{\theta}(a | \overbrace{s, z}^{\tilde{s}})$$

encapsulates information policy
needs to solve current task

learning a task = inferring z

from *context* $(s_1, a_1, s_2, r_1), (s_2, a_2, s_3, r_2), \dots$

this is just a POMDP!

before: $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$

now: $\tilde{\mathcal{M}} = \{\tilde{\mathcal{S}}, \mathcal{A}, \tilde{\mathcal{O}}, \tilde{\mathcal{P}}, \mathcal{E}, r\}$

$$\tilde{\mathcal{S}} = \mathcal{S} \times \mathcal{Z} \quad \tilde{s} = (s, z)$$

$$\tilde{\mathcal{O}} = \mathcal{S} \quad \tilde{o} = s$$

key idea: solving the POMDP $\tilde{\mathcal{M}}$ is equivalent to meta-learning!

Meta-RL as... partially observed RL?

$$\pi_{\theta}(a|s, z)$$

encapsulates information policy
needs to solve current task

learning a task = inferring z

from *context* $(s_1, a_1, s_2, r_1), (s_2, a_2, s_3, r_2), \dots$

exploring via posterior sampling with latent context



1. sample $z \sim \hat{p}(z_t | s_{1:t}, a_{1:t}, r_{1:t})$

← some approximate posterior
(e.g., variational)

2. act according to $\pi_{\theta}(a|s, z)$ to collect more data

← *act as though z was correct!*

this is just a POMDP!

typically requires *either*:

explicit state estimation, i.e. to estimate $p(s_t | o_{1:t})$

policies with memory

need to estimate $p(z_t | s_{1:t}, a_{1:t}, r_{1:t})$

this is *not* optimal!
why?

but it's pretty good,
both in theory and in
practice!

Variational inference for meta-RL

policy: $\pi_\theta(a_t|s_t, z_t)$

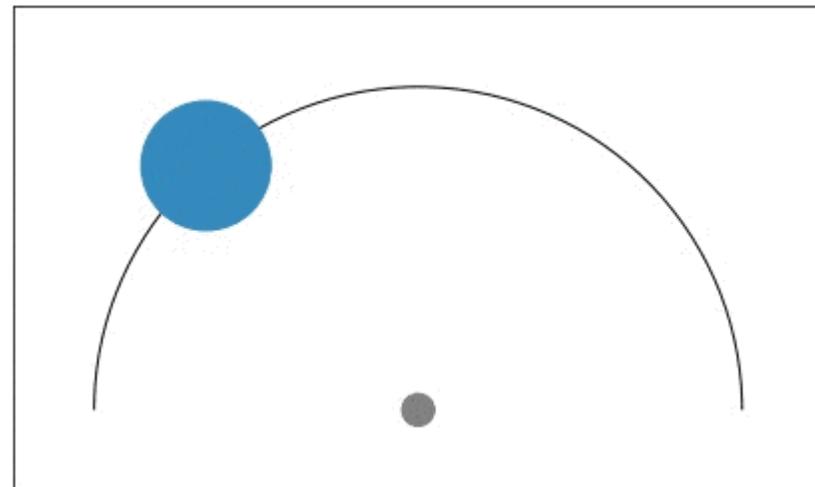
inference network: $q_\phi(z_t|s_1, a_1, r_1, \dots, s_t, a_t, r_t)$

$$(\theta, \phi) = \arg \max_{\theta, \phi} \frac{1}{N} \sum_{i=1}^n E_{z \sim q_\phi, \tau \sim \pi_\theta} [R_i(\tau) - D_{\text{KL}}(q(z|\dots) \| p(z))]$$

maximize *post-update* reward
(same as standard meta-RL)

stay close to prior

$$z_t \sim q_\phi(z_t|s_1, a_1, r_1, \dots, s_t, a_t, r_t)$$



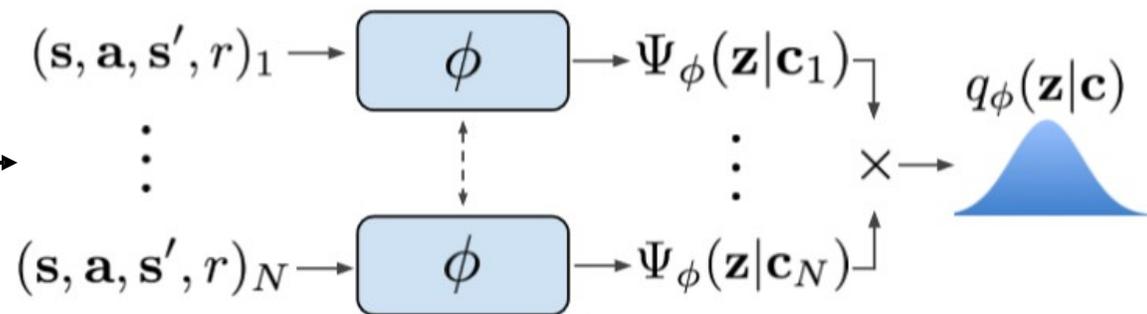
conceptually *very* similar to RNN meta-RL, but with stochastic z

stochastic z enables exploration via *posterior sampling*

Specific instantiation: PEARL

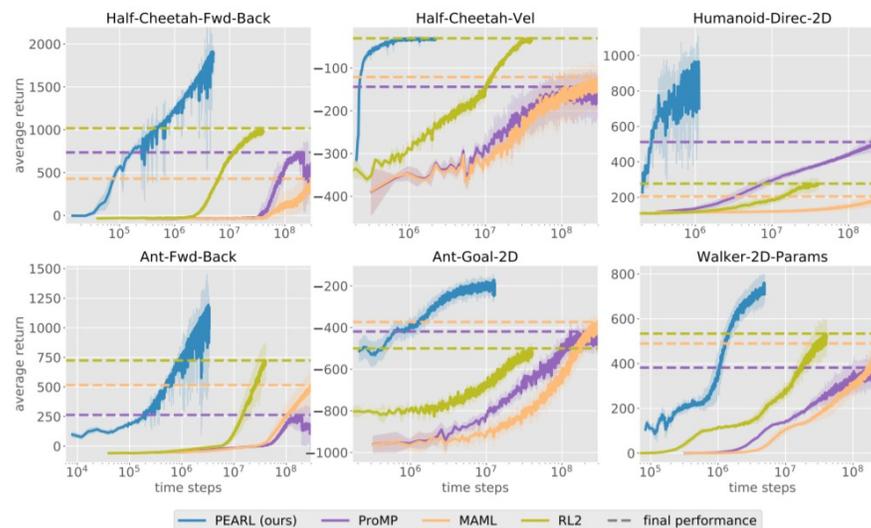
policy: $\pi_\theta(a_t|s_t, z_t)$

inference network: $q_\phi(z_t|s_1, a_1, r_1, \dots, s_t, a_t, r_t)$



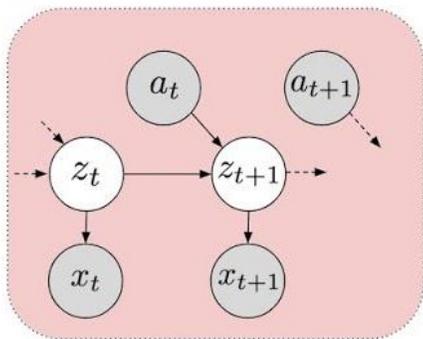
$$(\theta, \phi) = \arg \max_{\theta, \phi} \frac{1}{N} \sum_{i=1}^n E_{z \sim q_\phi, \tau \sim \pi_\theta} [R_i(\tau) - D_{\text{KL}}(q(z|\dots) || p(z))]$$

perform maximization using soft actor-critic (SAC),
state-of-the-art off-policy RL algorithm

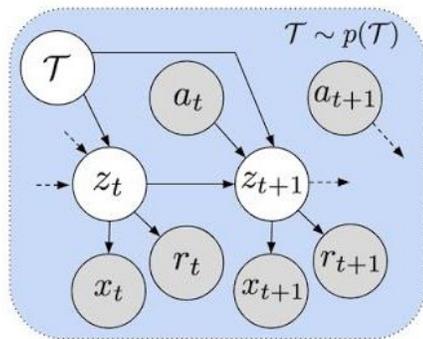


MELD: Model-Based Meta-RL with Images

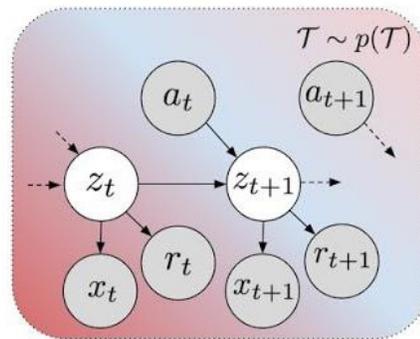
meta-learning can be viewed as a (kind of) POMDP



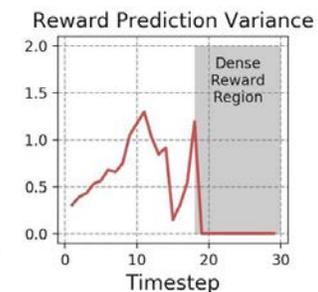
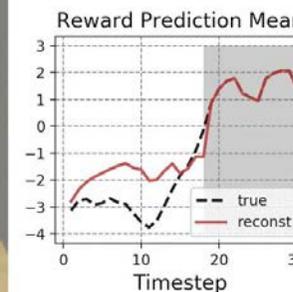
regular POMDP



meta-RL



MELD



Using this latent variable model generalizes meta-learning **and** POMDPs
Turns out to work very well as a meta-learning algorithm!

Task: right hole



Reward given when inserted into correct hole

Task: left hole



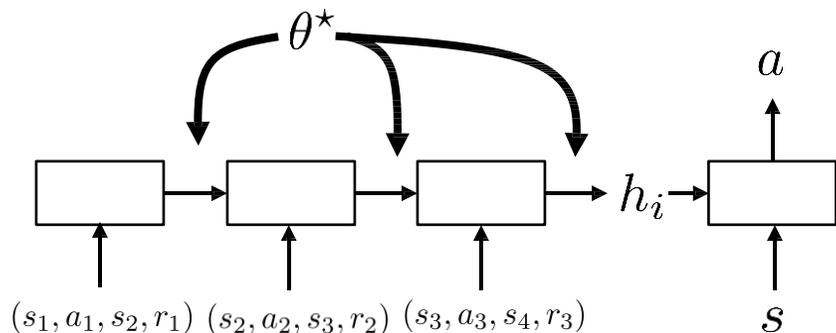
4x normal speed

References on meta-RL, inference, and POMDPs

- Rakelly*, Zhou*, Quillen, Finn, Levine. **Efficient Off-Policy Meta-Reinforcement learning via Probabilistic Context Variables.** ICML 2019.
- Zintgraf, Igl, Shiarlis, Mahajan, Hofmann, Whiteson. **Variational Task Embeddings for Fast Adaptation in Deep Reinforcement Learning.**
- Humplik, Galashov, Hasenclever, Ortega, Teh, Heess. **Meta reinforcement learning as task inference.**

The three perspectives on meta-RL

Perspective 1: just RNN it



Perspective 2: bi-level optimization

$$f_{\theta}(\mathcal{M}_i) = \theta + \alpha \nabla_{\theta} J_i(\theta)$$

MAML for RL

Perspective 3: it's an inference problem!

$$\pi_{\theta}(a|s, z) \quad z_t \sim p(z_t|s_{1:t}, a_{1:t}, r_{1:t})$$

everything needed to solve task

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

$$\text{where } \phi_i = f_{\theta}(\mathcal{M}_i)$$

what should $f_{\theta}(\mathcal{M}_i)$ do?

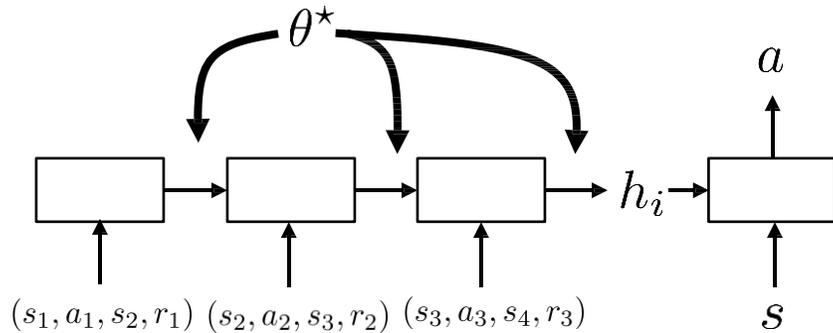
1. improve policy with experience from \mathcal{M}_i

$$\{(s_1, a_1, s_2, r_1), \dots, (s_T, a_T, s_{T+1}, r_T)\}$$

2. (new in RL): choose how to interact, i.e. choose a_t
meta-RL must also *choose* how to *explore*!

The three perspectives on meta-RL

Perspective 1: just RNN it



- + conceptually simple
- + relatively easy to apply
- vulnerable to *meta-overfitting*
- challenging to optimize in practice

Perspective 2: bi-level optimization

$$f_{\theta}(\mathcal{M}_i) = \theta + \alpha \nabla_{\theta} J_i(\theta)$$

MAML for RL

- + good extrapolation (“consistent”)
- + conceptually elegant
- complex, requires many samples

Perspective 3: it’s an inference problem!

$$\pi_{\theta}(a|s, z) \quad z_t \sim p(z_t|s_{1:t}, a_{1:t}, r_{1:t})$$

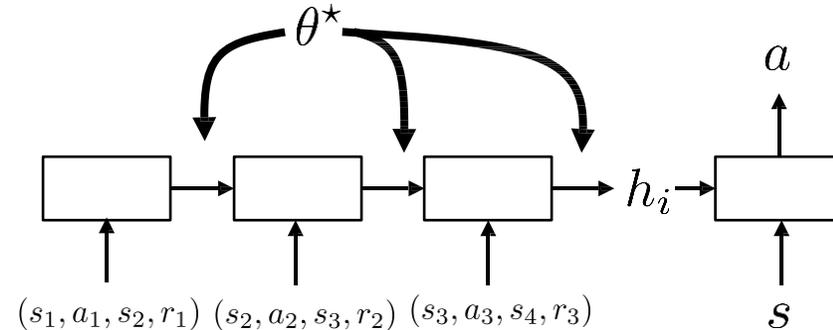
↑
everything needed to solve task

- + simple, effective exploration via posterior sampling
- + elegant reduction to solving a special POMDP
- vulnerable to *meta-overfitting*
- challenging to optimize in practice

But they're not that different!

just perspective 1,
but with stochastic
hidden variables!
i.e., $\phi = \mathbf{z}$

Perspective 1: just RNN it



Perspective 2: bi-level optimization

$$f_{\theta}(\mathcal{M}_i) = \theta + \alpha \nabla_{\theta} J_i(\theta)$$

MAML for RL

Perspective 3: it's an inference problem!

$$\pi_{\theta}(a|s, \mathbf{z}) \quad \mathbf{z}_t \sim p(\mathbf{z}_t | s_{1:t}, a_{1:t}, r_{1:t})$$

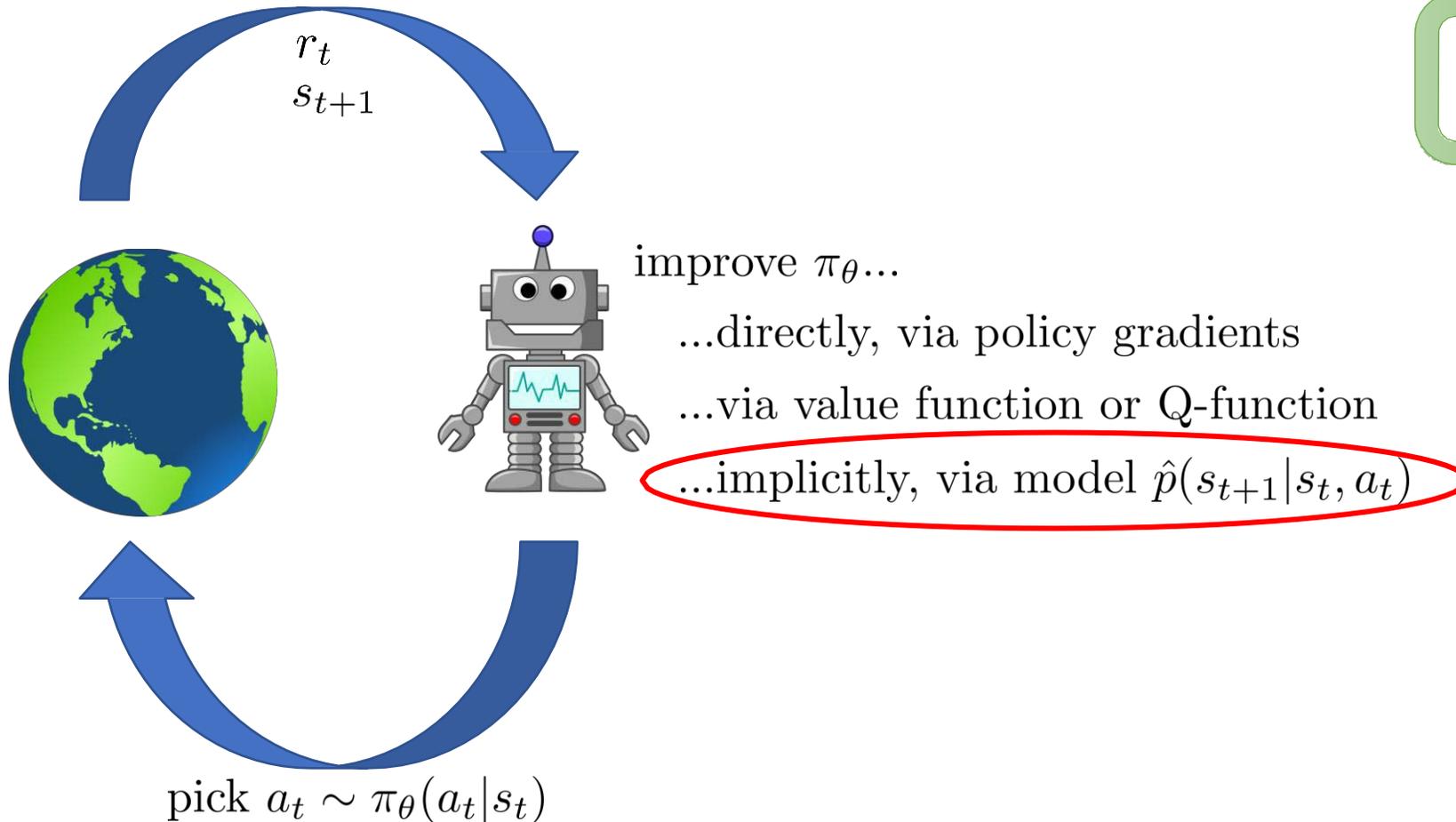
everything needed to solve task

just a particular
architecture choice
for these

Model-Based Meta-RL

Model-based meta-RL

$$\theta^* = \arg \max_{\theta} E_{\pi_{\theta}(\tau)} [R(\tau)]$$



short sketch of model-based RL:

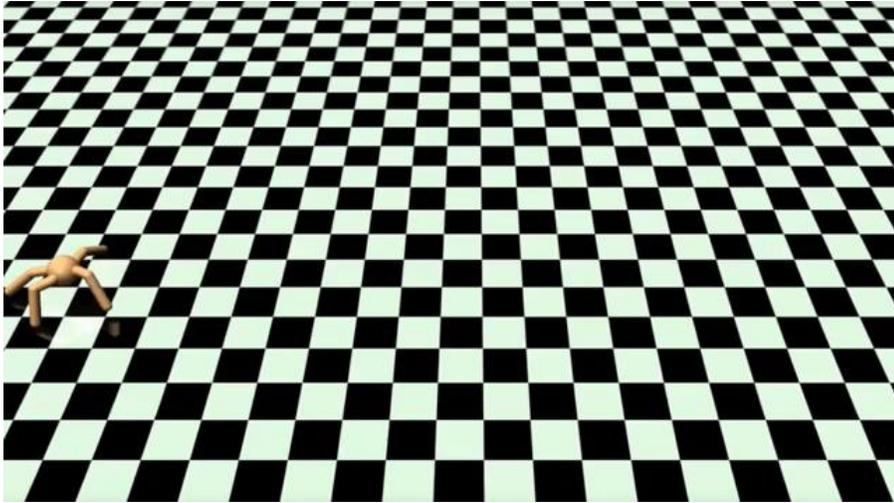
1. collect data \mathcal{B}
2. use \mathcal{B} to get $\hat{p}(s_{t+1}|s_t, a_t)$
3. use $\hat{p}(s_{t+1}|s_t, a_t)$ to plan a

why?

- + requires much less data vs model-free
- + a bit different due to model
- + can adapt extremely quickly!

Model-based meta-RL

example task: ant with broken leg



non-adaptive method:

a few episodes

1. collect data $\mathcal{B} = \{s_i, a_i, s'_i\}$
2. train $d_\theta(s, a) \rightarrow s'$ on \mathcal{B}
3. use d_θ to optimize actions

$$a_t, \dots, a_{t+k} = \arg \max_{a_t, \dots, a_{t+k}} \sum_{\tau=t}^{t+k} r(s_\tau, a_\tau)$$
$$\text{s.t. } s_{t+1} = d_\theta(s_t, a_t)$$

adaptive method:

nice idea, but how much
can we really adapt in just
one (or a few) step(s)?

1. take *one* step, get $\{s, a, s'\}$
2. $\theta \leftarrow \theta - \alpha \nabla_\theta \|d_\theta(s, a) - s'\|^2$
3. use d_θ to optimize a_t, \dots, a_{t+k} , take a_t

Model-based meta-RL

meta-training time

$$\mathcal{D}_{\text{meta-train}} = \{(\mathcal{D}_1^{\text{tr}}, \mathcal{D}_1^{\text{ts}}), \dots, (\mathcal{D}_n^{\text{tr}}, \mathcal{D}_n^{\text{ts}})\}$$

$$\mathcal{D}_i^{\text{tr}} = \{(x_1^i, y_1^i), \dots, (x_k^i, y_k^i)\}$$

$$\mathcal{D}_i^{\text{ts}} = \{(x_1^i, y_1^i), \dots, (x_l^i, y_l^i)\}$$

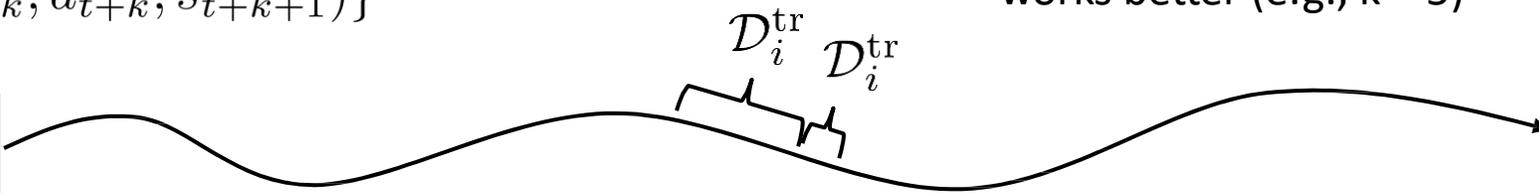
$$x \leftarrow (s, a) \quad y \leftarrow s'$$

generate each $\mathcal{D}_i^{\text{tr}}, \mathcal{D}_i^{\text{ts}}$:

sample subsequence $s_t, a_t, \dots, s_{t+k}, a_{t+k}, s_{t+k+1}$ from past experience

$$\mathcal{D}_i^{\text{tr}} \leftarrow \{(s_t, a_t, s_{t+1}), \dots, (s_{t+k-1}, a_{t+k-1}, s_{t+k})\}$$

$$\mathcal{D}_i^{\text{ts}} \leftarrow \{(s_{t+k}, a_{t+k}, s_{t+k+1})\}$$



meta-test time

adaptive method:

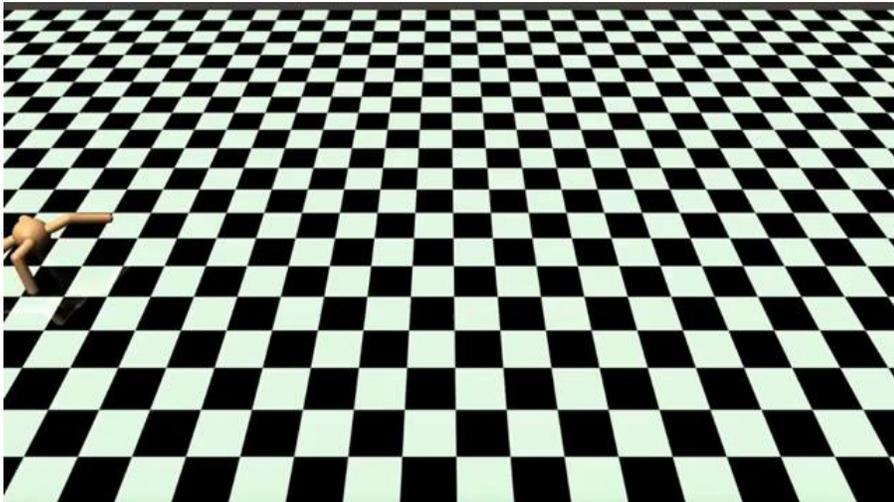
1. take *one* step, get $\{s, a, s'\}$
2. $\theta \leftarrow \theta - \alpha \nabla_{\theta} \|d_{\theta}(s, a) - s'\|^2$
3. use d_{θ} to optimize a_t, \dots, a_{t+k} , take a_t

assumes past experience has many different dynamics

could choose $k = 1$, but $k > 1$ works better (e.g., $k = 5$)

Model-based meta-RL

example task: ant with broken leg



See also:

Saemundsson, Hofmann, Deisenroth. **Meta-Reinforcement Learning with Latent Variable Gaussian Processes.**

Nagabandi, Finn, Levine. **Deep Online Learning via Meta-Learning: Continual Adaptation for Model-Based RL.**

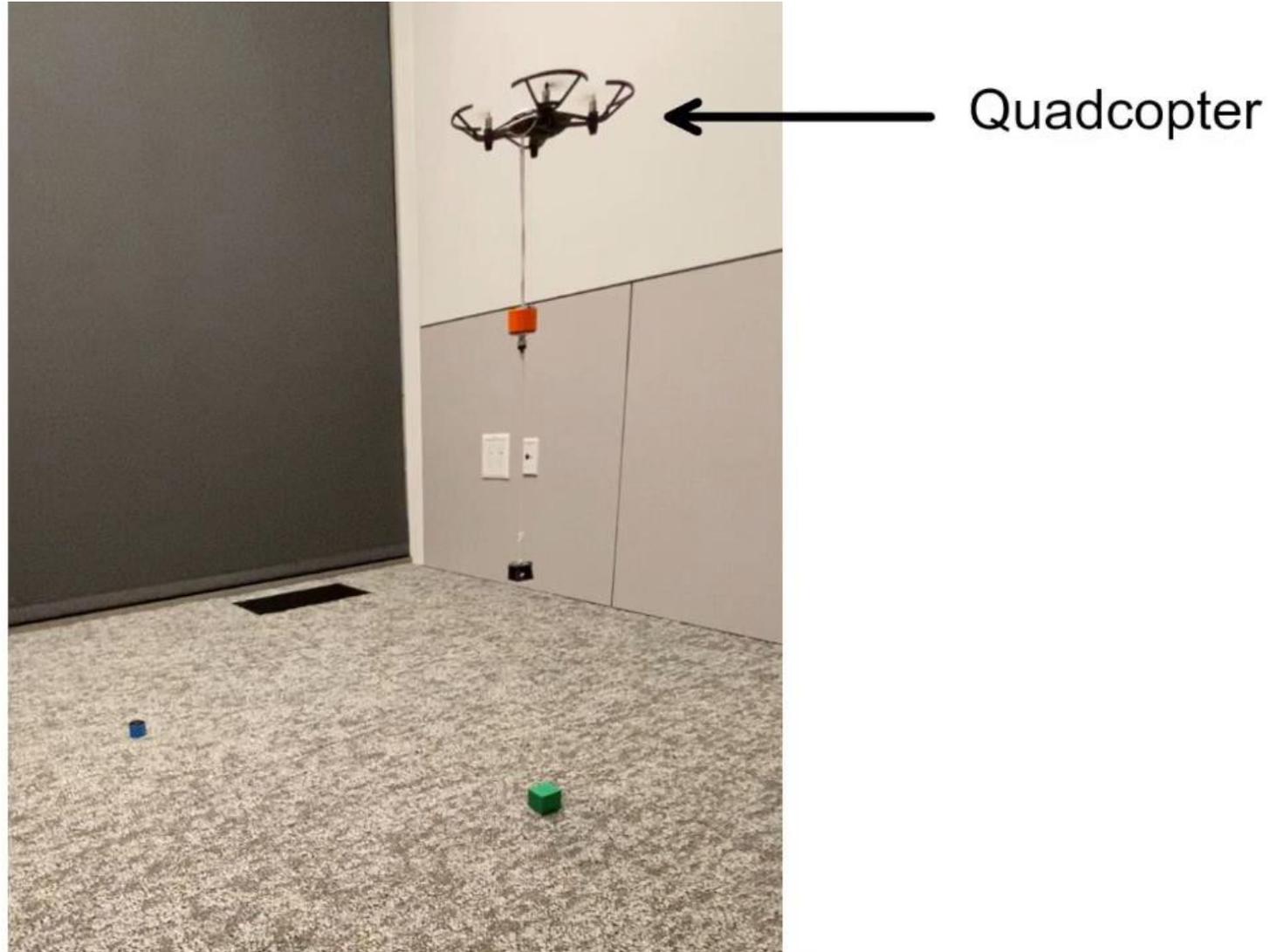
meta-test time

adaptive method:

1. take *one* step, get $\{s, a, s'\}$
2. $\theta \leftarrow \theta - \alpha \nabla_{\theta} \|d_{\theta}(s, a) - s'\|^2$
3. use d_{θ} to optimize a_t, \dots, a_{t+k} , take a_t

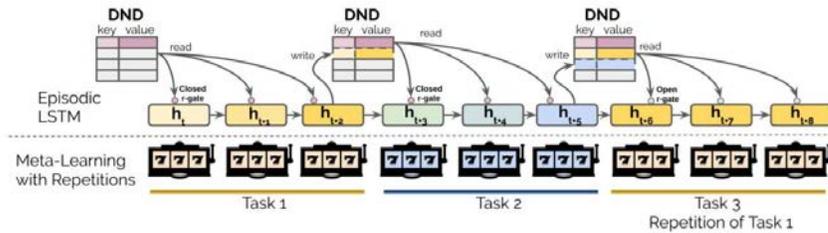


Model-Based Meta-RL for Quadrotor Control

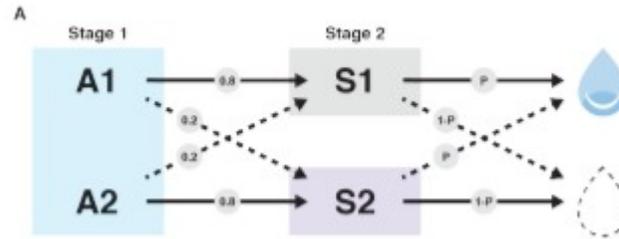


Meta-RL and emergent phenomena

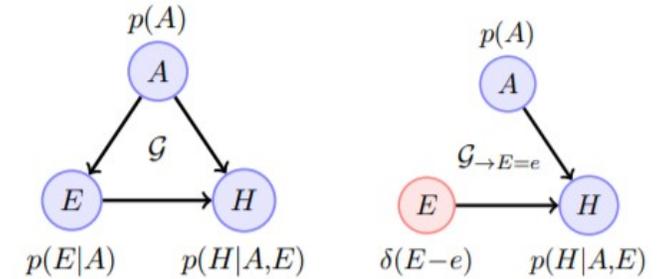
meta-RL gives rise to episodic learning



model-free meta-RL gives rise to model-based adaptation



meta-RL gives rise to causal reasoning (!)



Ritter, Wang, Kurth-Nelson, Jayakumar, Blundell, Pascanu, Botvinick. **Been There, Done That: Meta-Learning with Episodic Recall.**

Wang, Kurth-Nelson, Kumaran, Tirumala, Soyer, Leibo, Hassabis, Botvinick. **Prefrontal Cortex as a Meta-Reinforcement Learning System.**

Dasgupta, Wang, Chiappa, Mitrovic, Ortega, Raposo, Hughes, Battaglia, Botvinick, Kurth-Nelson. **Causal Reasoning from Meta-Reinforcement Learning.**

Humans and animals *seemingly* learn behaviors in a variety of ways:

- Highly efficient but (apparently) model-free RL
- Episodic recall
- Model-based RL
- Causal inference
- etc.

Perhaps each of these is a separate “algorithm” in the brain

But maybe these are all emergent phenomena resulting from meta-RL?