

Lecture outline

- recap: policy gradient RL and how it can be used to build meta-RL algorithms
- the exploration problem in meta-RL
- an approach to encourage better exploration

break

- meta-RL as a POMDP
- an approach for *off-policy* meta-RL and a different way to explore

Recap: meta-reinforcement learning

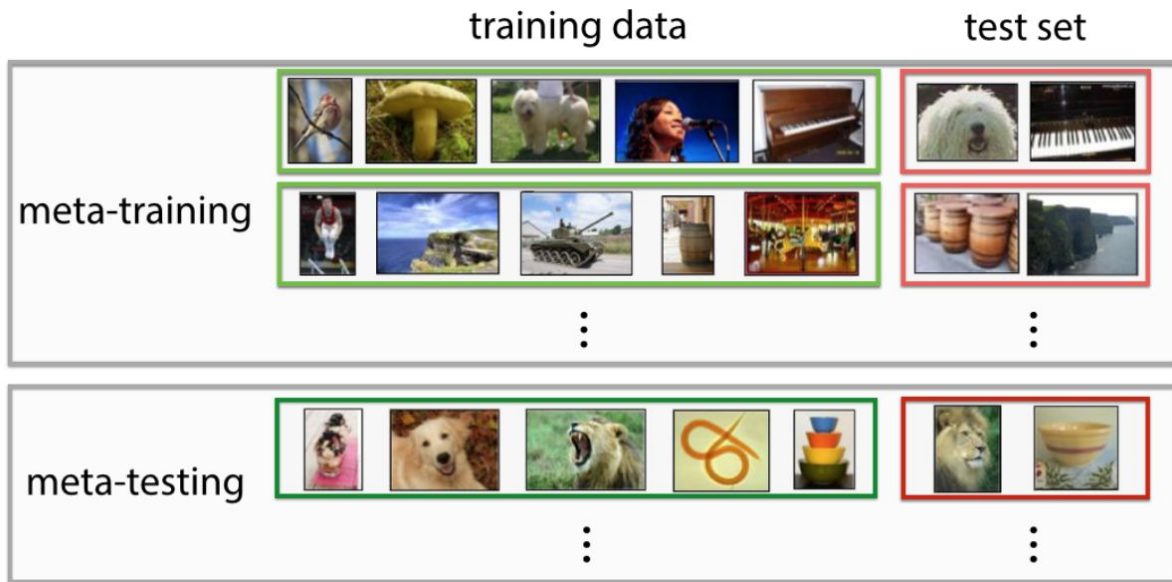


“Hula Beach”, “Never grow up”, “The Sled” - by artist Matt Spangler, mattspangler.com

Recap: meta-reinforcement learning

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

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



Recap: meta-reinforcement learning

Meta-training / outer loop

→ gradient descent

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}}) \quad \theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

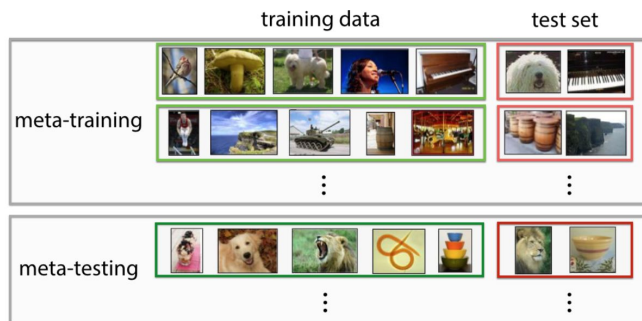
Adaptation / inner loop

→ lots of options

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

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

MDP for task i



M1

M2

M3

M_test

"Scooterrific!" by artist Matt Spangler

What's different in RL?

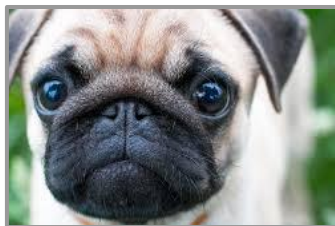
$$\theta^* = \arg \min_{\theta} \sum_{i=1}^n \mathcal{L}(\phi_i, \mathcal{D}_i^{\text{ts}})$$

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

Adaptation data is given to us!

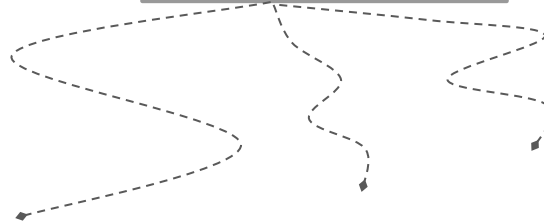
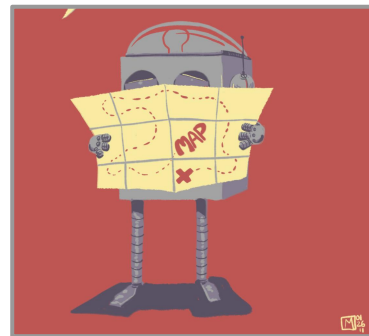
dalmation

german shepherd pug



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

where $\phi_i = f_{\theta}(\mathcal{M}_i)$ Agent has to collect adaptation data!



Recap: policy gradient RL algorithms

Direct policy search on $\pi_{\theta}(\mathbf{a}_t|\mathbf{s}_t)$

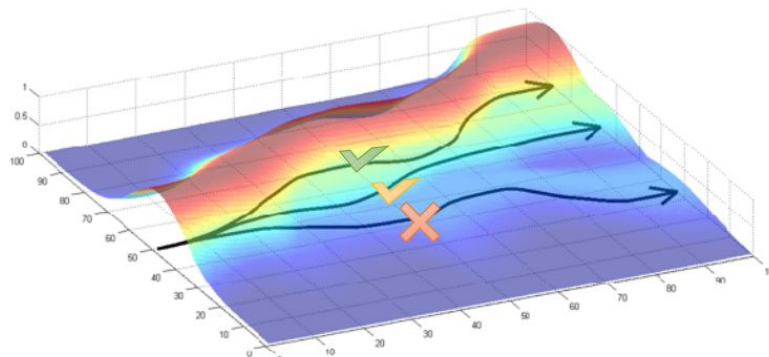
REINFORCE algorithm:

1. sample $\{\tau^i\}$ from $\pi_{\theta}(\mathbf{a}_t|\mathbf{s}_t)$ (run it on the robot)
2. $\nabla_{\theta} J(\theta) \approx \sum_i (\sum_t \nabla_{\theta} \log \pi_{\theta}(\mathbf{a}_t^i|\mathbf{s}_t^i)) (\sum_t r(\mathbf{s}_t^i, \mathbf{a}_t^i))$
3. $\theta \leftarrow \theta + \alpha \nabla_{\theta} J(\theta)$

Good stuff is made more likely

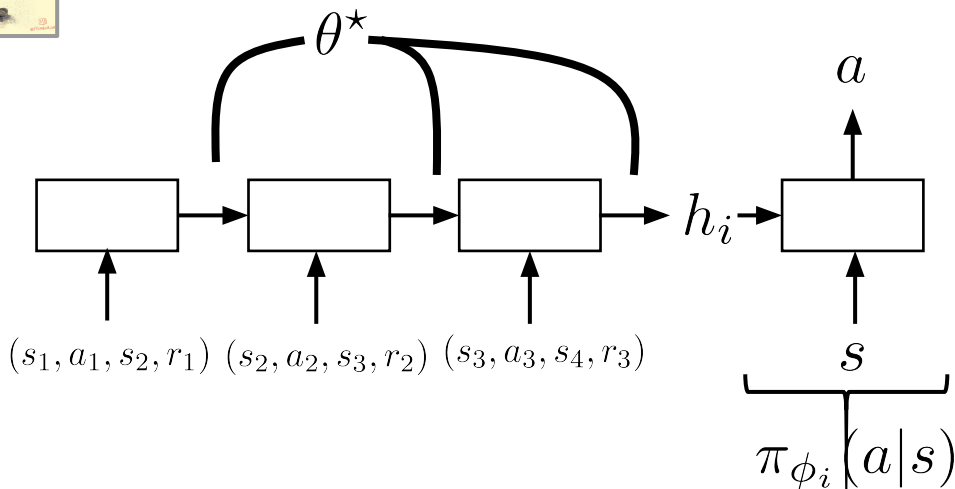
Bad stuff is made less likely

Formalizes the idea of “trial and error”



PG meta-RL algorithms: recurrent

Implement the policy as a recurrent network, train across a set of tasks



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

PG \nearrow

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

\uparrow
RNN

Pro: general, expressive

Con: not consistent

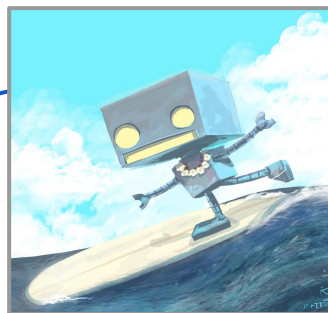
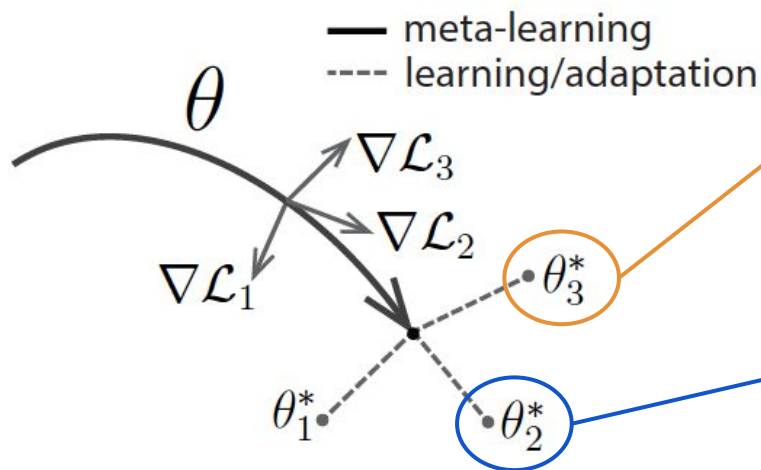
Persist the hidden state across episode boundaries for continued adaptation!

PG meta-RL algorithms: gradients

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

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

PG \nearrow

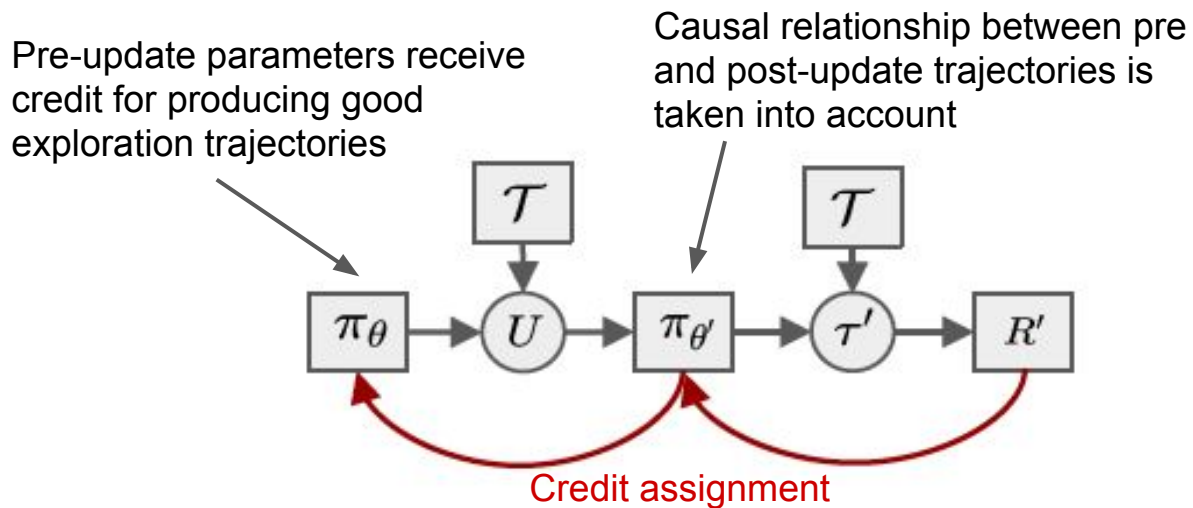


Pro: consistent!

Con: not expressive

Q: Can you think of an example in which recurrent methods are more expressive?

How these algorithms learn to explore



How well do they explore?

Recurrent approach explores in a new maze
(goal is to navigate from blue to red square)

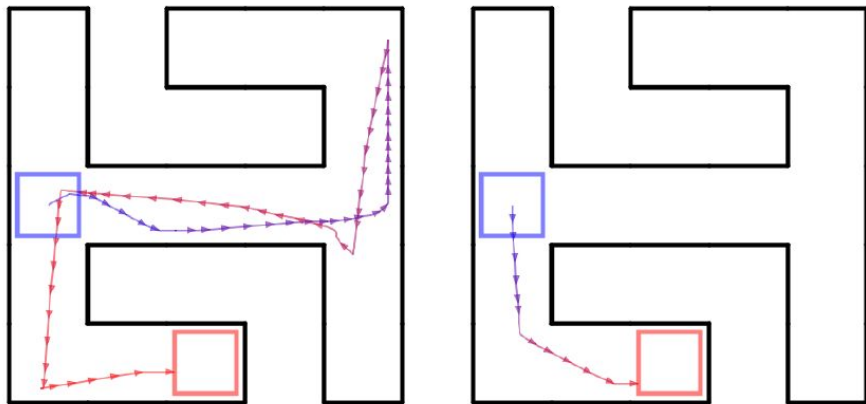


Fig adapted from RL2. Duan et al. 2016

Gradient-based approach explores in a point
robot navigation task

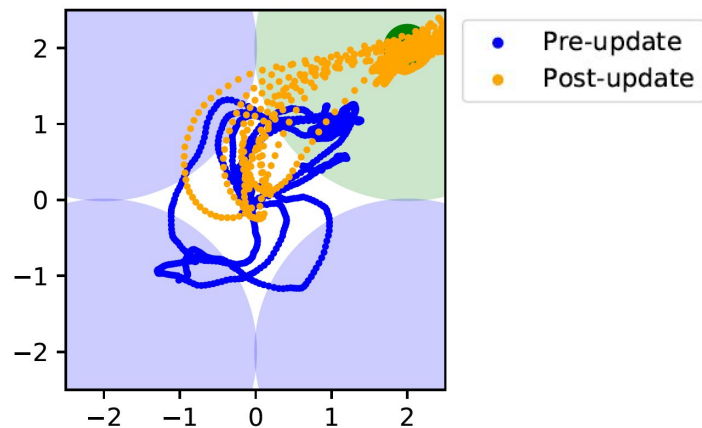
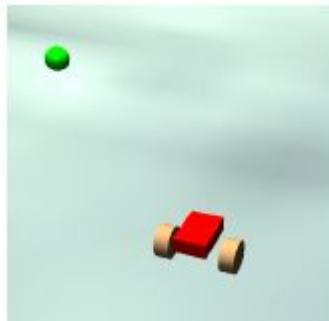


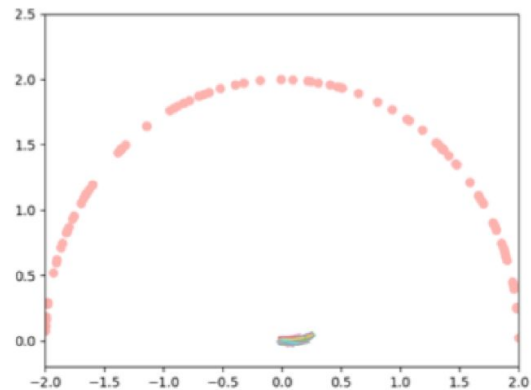
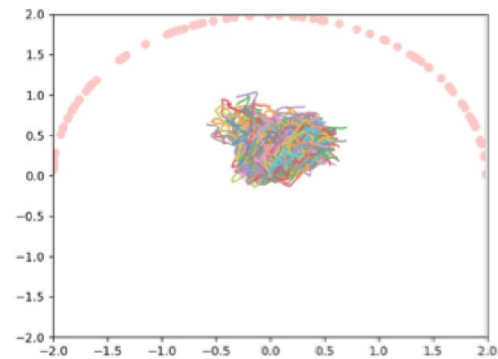
Fig adapted from ProMP Rothfuss et al. 2017

How well do they explore?

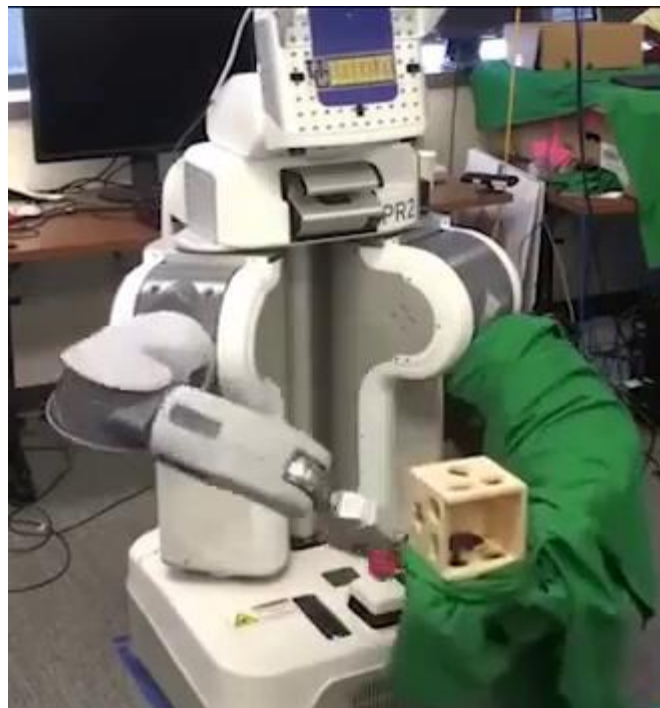
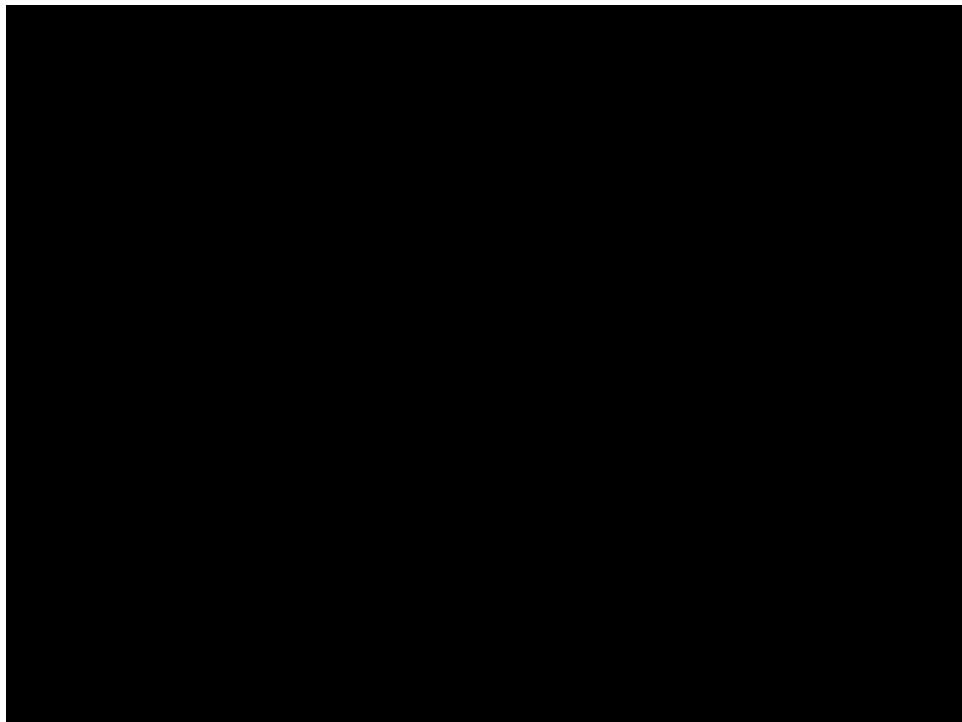
Here gradient-based meta-RL fails to explore in a sparse reward navigation task



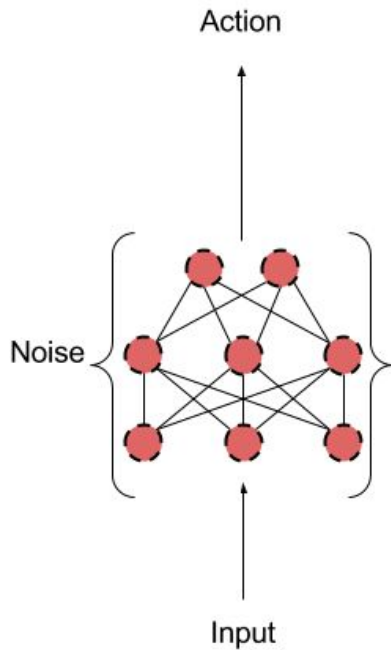
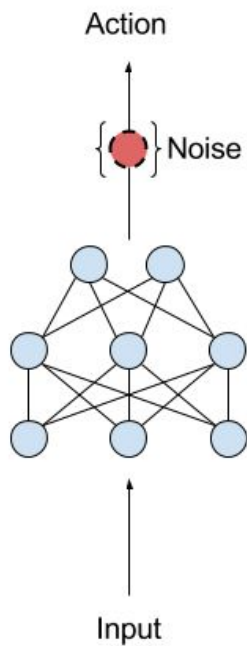
Exploration Trajectories



What's the problem?



What's the problem?

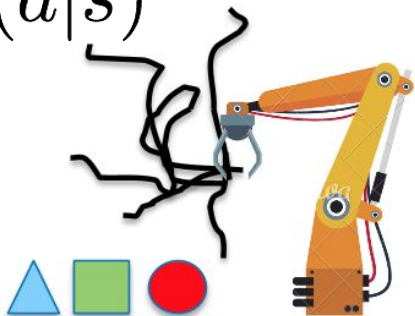


Exploration requires stochasticity,
optimal policies don't

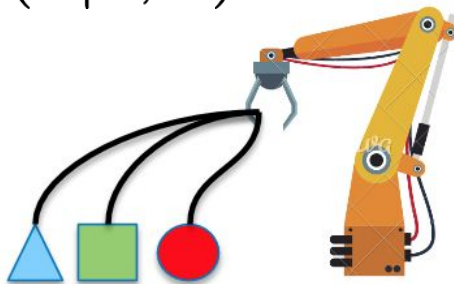
Typical methods of adding noise
are time-invariant

Temporally extended exploration

$\pi(a|s)$



$\pi(a|s, z)$



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

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

\uparrow PG on z

Sample z , hold constant during episode

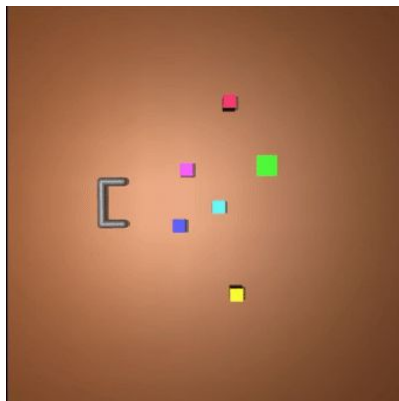
Adapt z to a new task with gradient descent

Pre-adaptation: good exploration

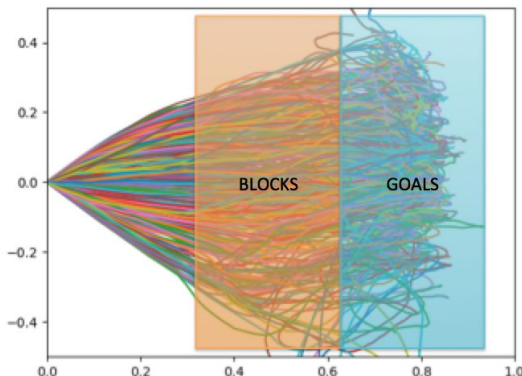
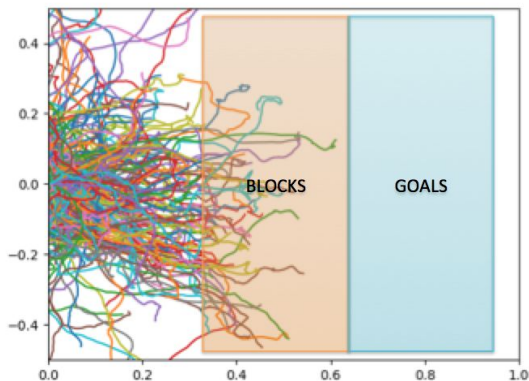
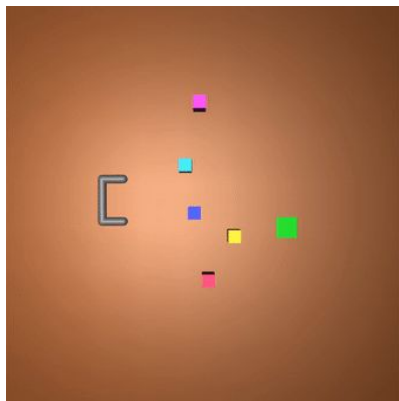
Post-adaptation: good task performance

Temporally extended exploration with MAESN

MAML Exploration






MAESN exploration









Meta-RL desiderata

	recurrent	gradient	structured exp










Meta-RL desiderata

	recurrent	gradient	structured exp
consistent			

Meta-RL desiderata

	recurrent	gradient	structured exp
consistent			
expressive			

Meta-RL desiderata

	recurrent	gradient	structured exp
consistent			
expressive			
structured exploration			

Meta-RL desiderata

	recurrent	gradient	structured exp
consistent	✗	✓	✓
expressive	✓	✗	✗
structured exploration	~	~	✓
efficient & off-policy	✗	✗	✗

In single-task RL, off-policy algorithms 1-2 orders of magnitude more efficient!
Huge difference for real-world applications (1 month -> 10 hours)

Why is off-policy meta-RL difficult?

Key characteristic of meta-learning: the conditions at meta-training time should closely match those at test time!

meta-train
classes



meta-test
classes



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

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

Train with off-policy data, but then f_{θ^*} is on-policy...

Note: this is very much an unresolved question

Break

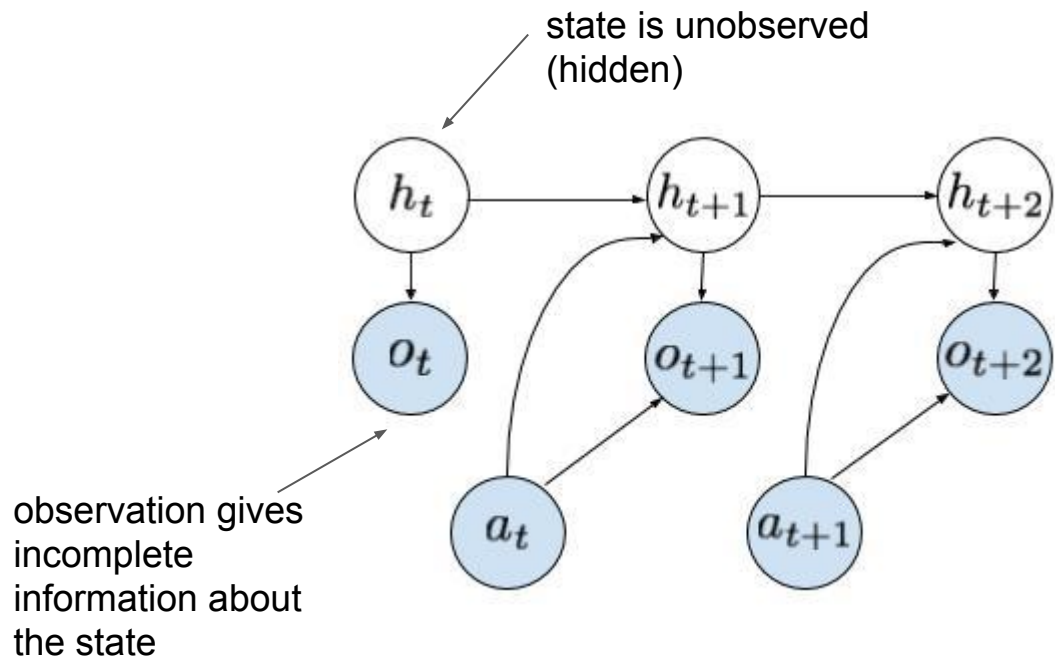
PEARL

Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables

Kate Rakelly*, Aurick Zhou*, Deirdre Quillen, Chelsea Finn, Sergey Levine



Aside: POMDPs



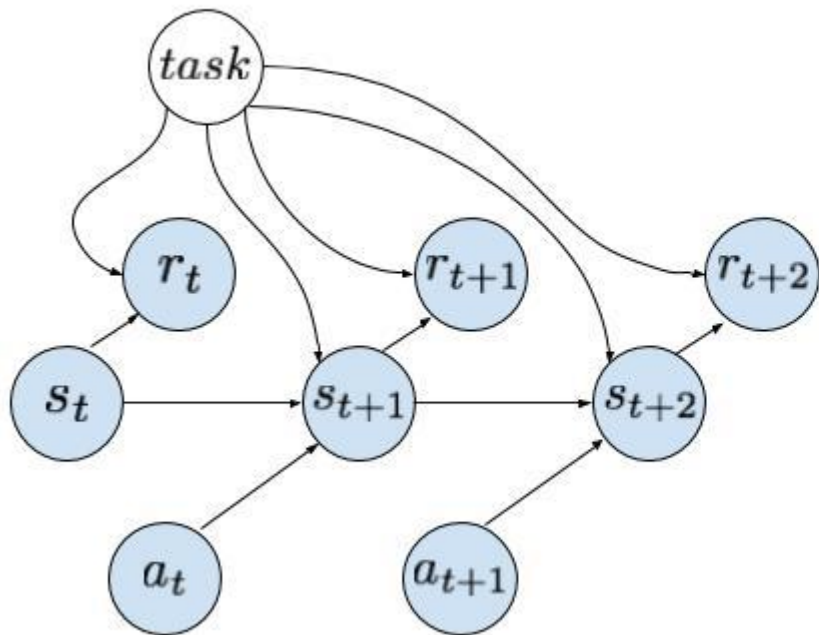
Example: incomplete sensor data



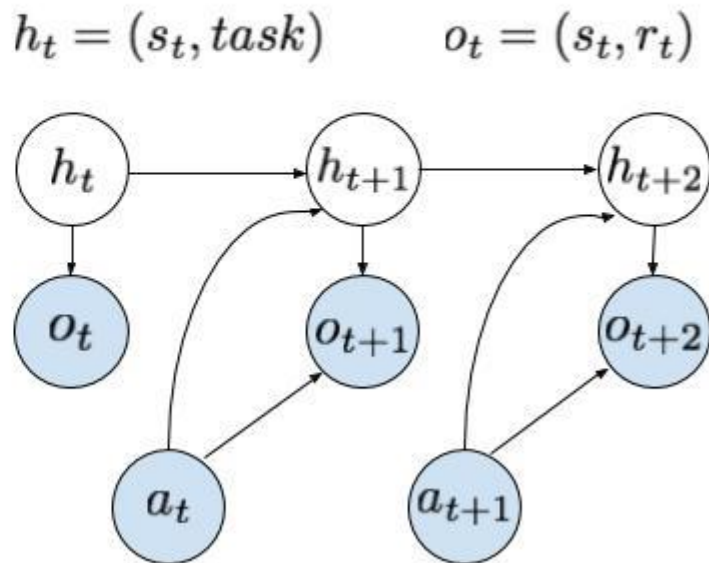
"That Way We Go" by Matt Spangler

The POMDP view of meta-RL

meta-RL...



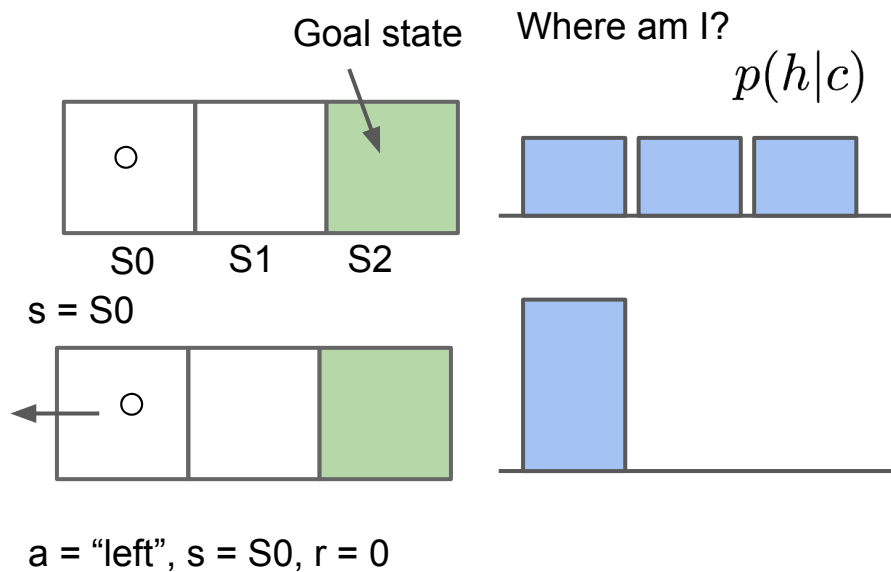
...as a POMDP



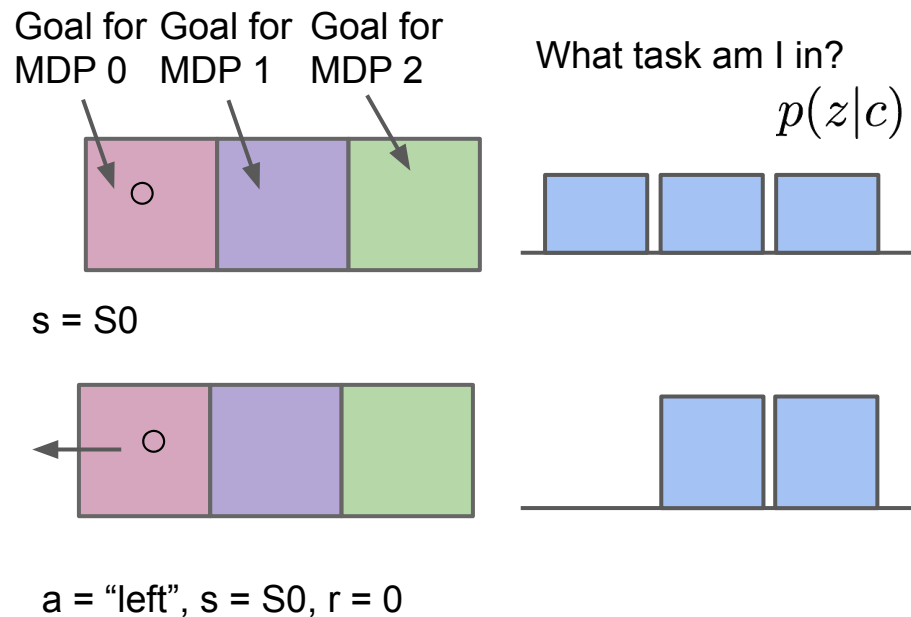
Can we leverage this connection to design a new meta-RL algorithm?

Model belief over latent task variables

POMDP for unobserved state

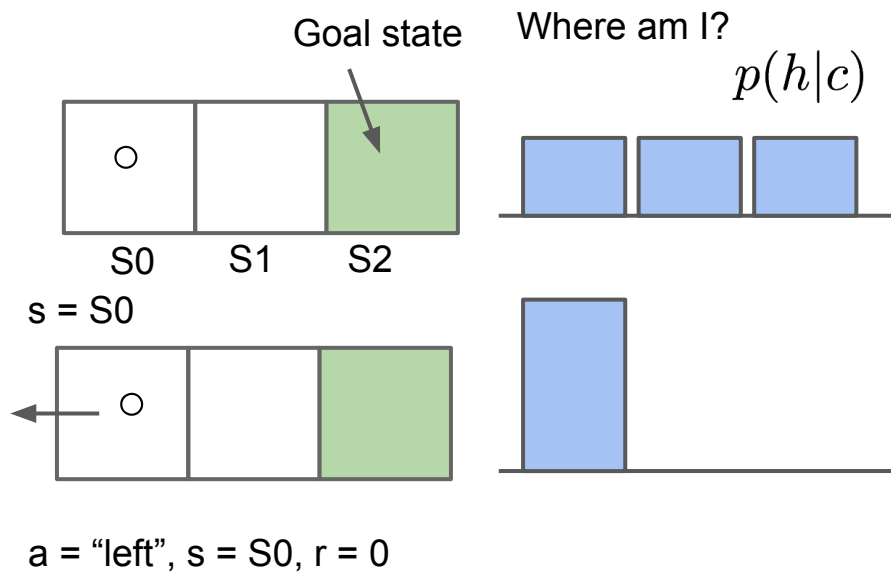


POMDP for unobserved task

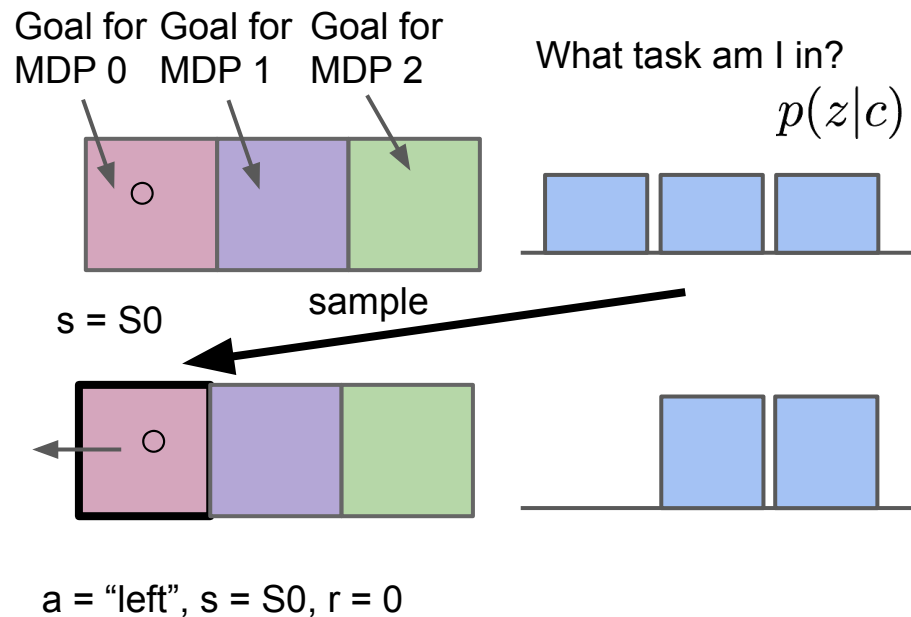


Model belief over latent task variables

POMDP for unobserved state

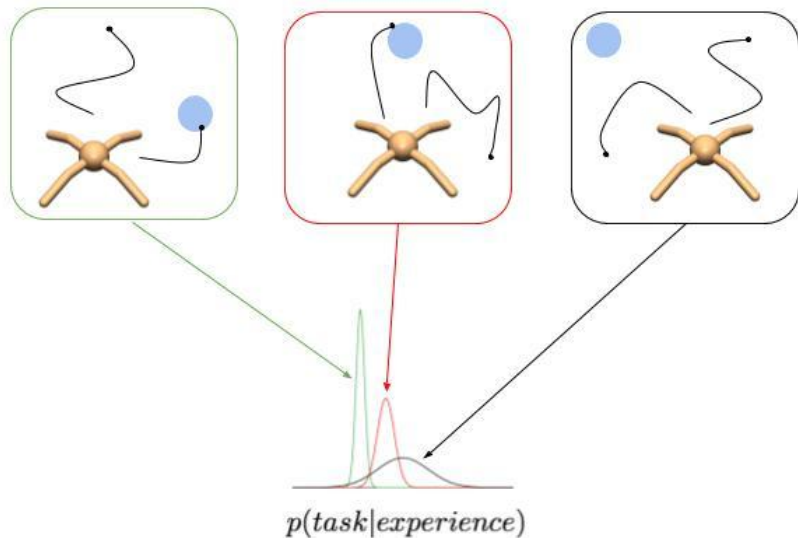


POMDP for unobserved task



RL with task-belief states

How do we learn this in a way that generalizes to new tasks?



“Task” can be supervised by reconstructing states and rewards

OR

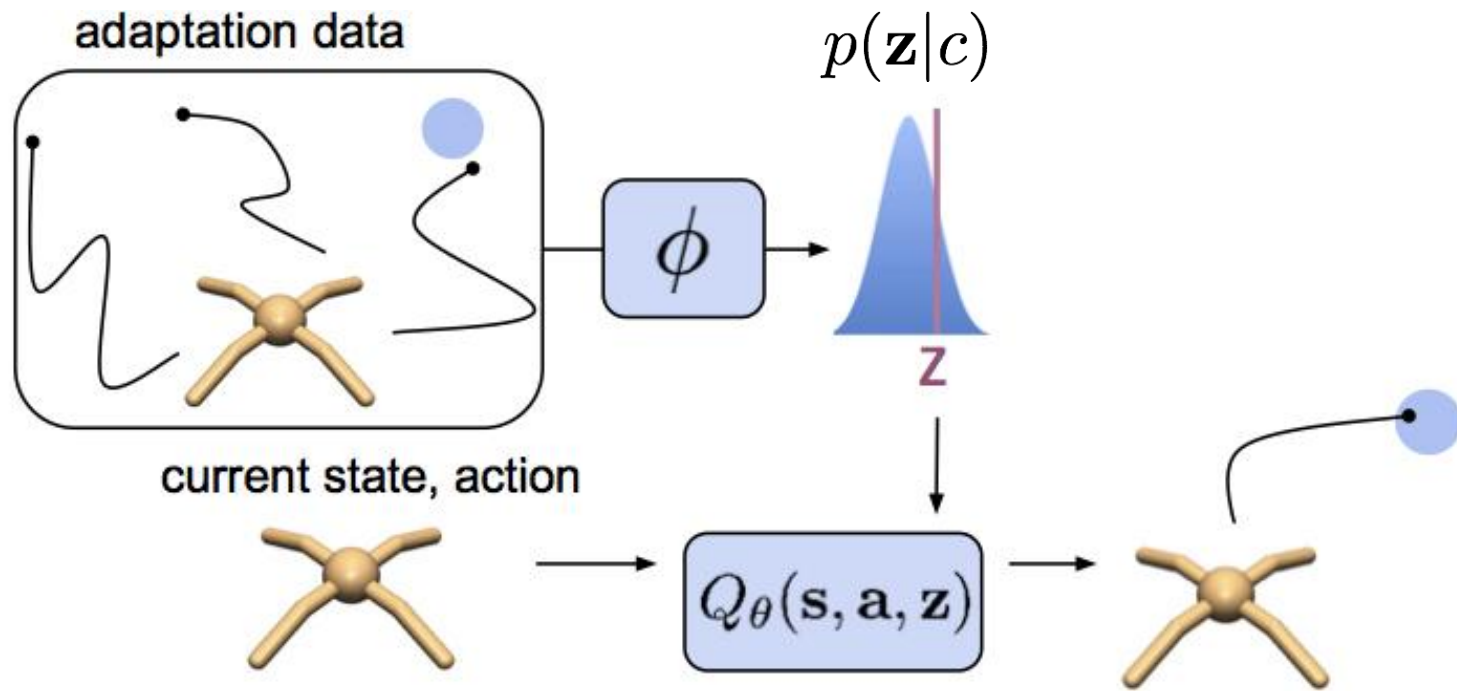
By minimizing Bellman error

Meta-RL with task-belief states

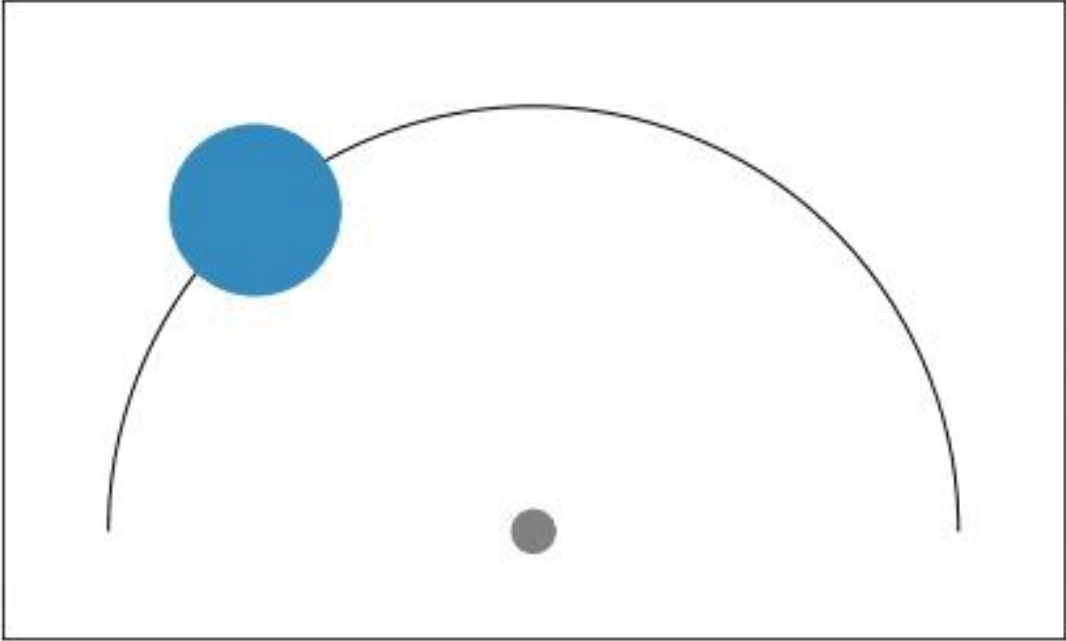
$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

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

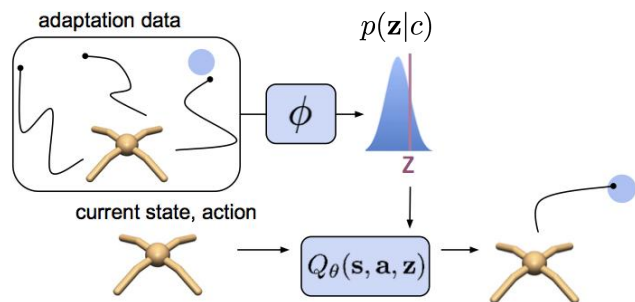
Stochastic
encoder



Posterior sampling in action



Meta-RL with task-belief states



“Likelihood” term (Bellman error)

$$\mathbb{E}_{\mathcal{T}} \left[\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{c}^{\mathcal{T}})} \left[\overbrace{R(\mathcal{T}, \mathbf{z})}^{\text{“Likelihood” term (Bellman error)}} + \beta \overbrace{D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{c}^{\mathcal{T}}) || p(\mathbf{z}))}^{\text{“Regularization” term / information bottleneck}} \right] \right]$$

“Regularization” term /
information bottleneck

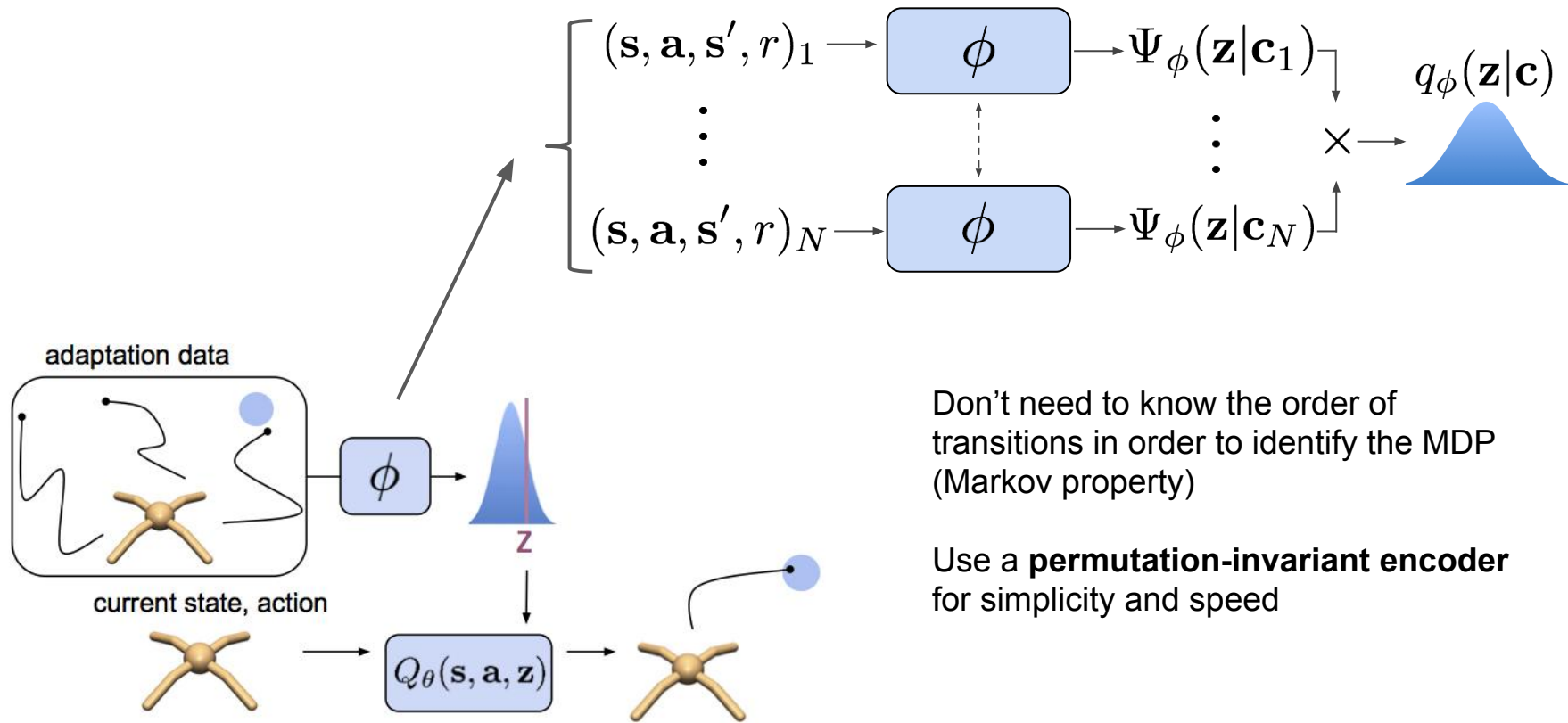
$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n E_{\pi_{\phi_i}(\tau)} [R(\tau)]$$

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

Stochastic
encoder

Variational approximations to
posterior and prior

Encoder design



Aside: Soft Actor-Critic (SAC)

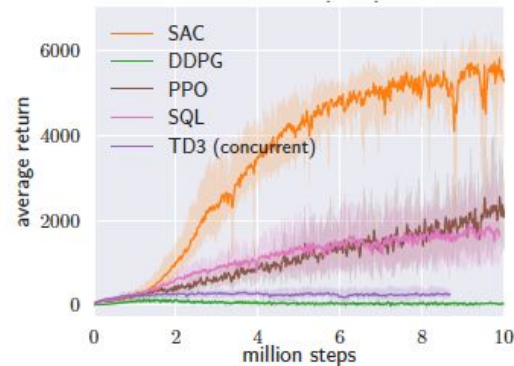
“Soft”: Maximize rewards *and* entropy of the policy
(higher entropy policies explore better)

$$J(\pi) = \sum_{t=0}^T \mathbb{E}_{(s_t, \mathbf{a}_t) \sim \rho_\pi} [r(s_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi(\cdot | s_t))]$$

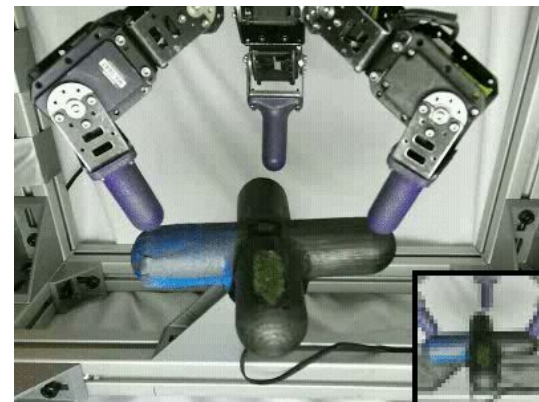
“Actor-Critic”: Model *both* the actor (aka the policy) and the critic (aka the Q-function)

$$J_Q(\theta) = \mathbb{E}_{(s_t, \mathbf{a}_t) \sim \mathcal{D}} \left[\frac{1}{2} \left(Q_\theta(s_t, \mathbf{a}_t) - \hat{Q}(s_t, \mathbf{a}_t) \right)^2 \right]$$

$$J_\pi(\phi) = \mathbb{E}_{s_t, a_t} [Q_\theta(s_t, a_t) + \alpha \mathcal{H}(\pi_\phi(\cdot | s_t))]$$

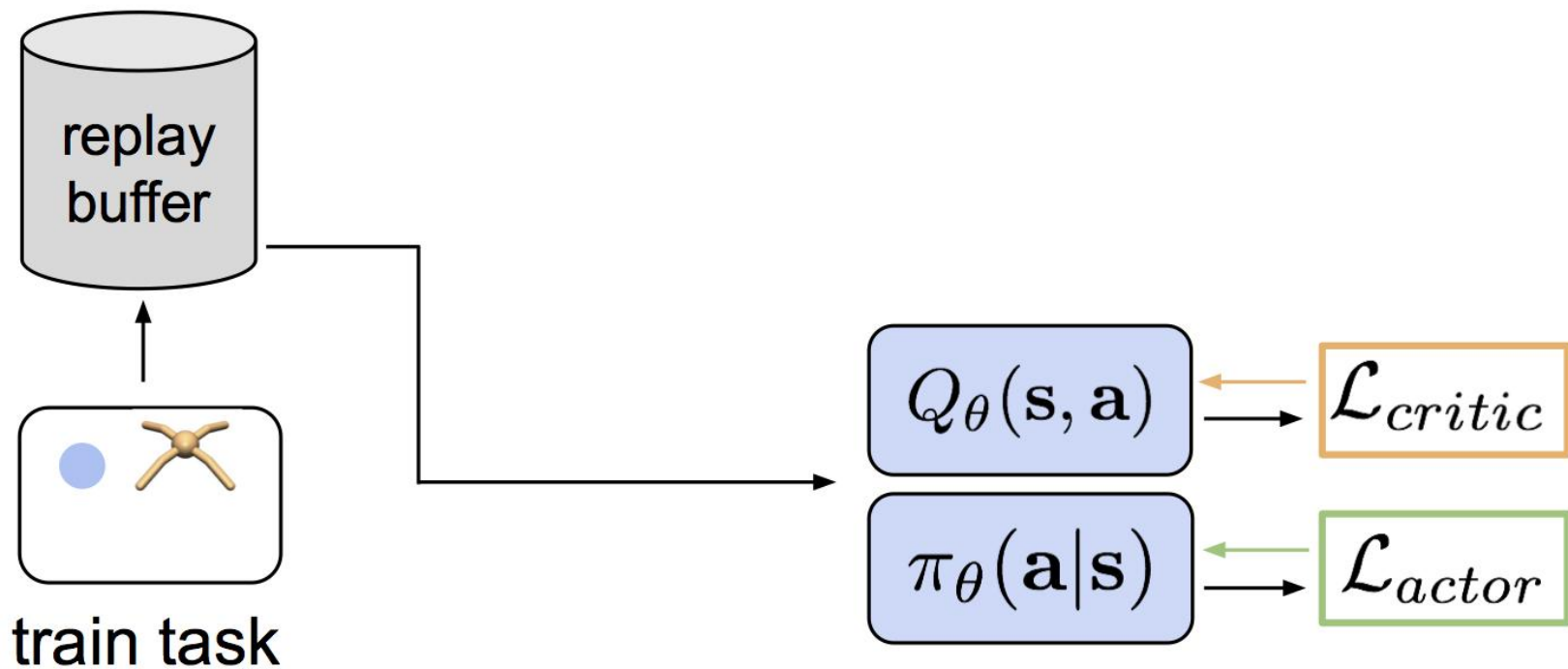


(f) Humanoid (rllab)



Dclaw robot turns valve from pixels

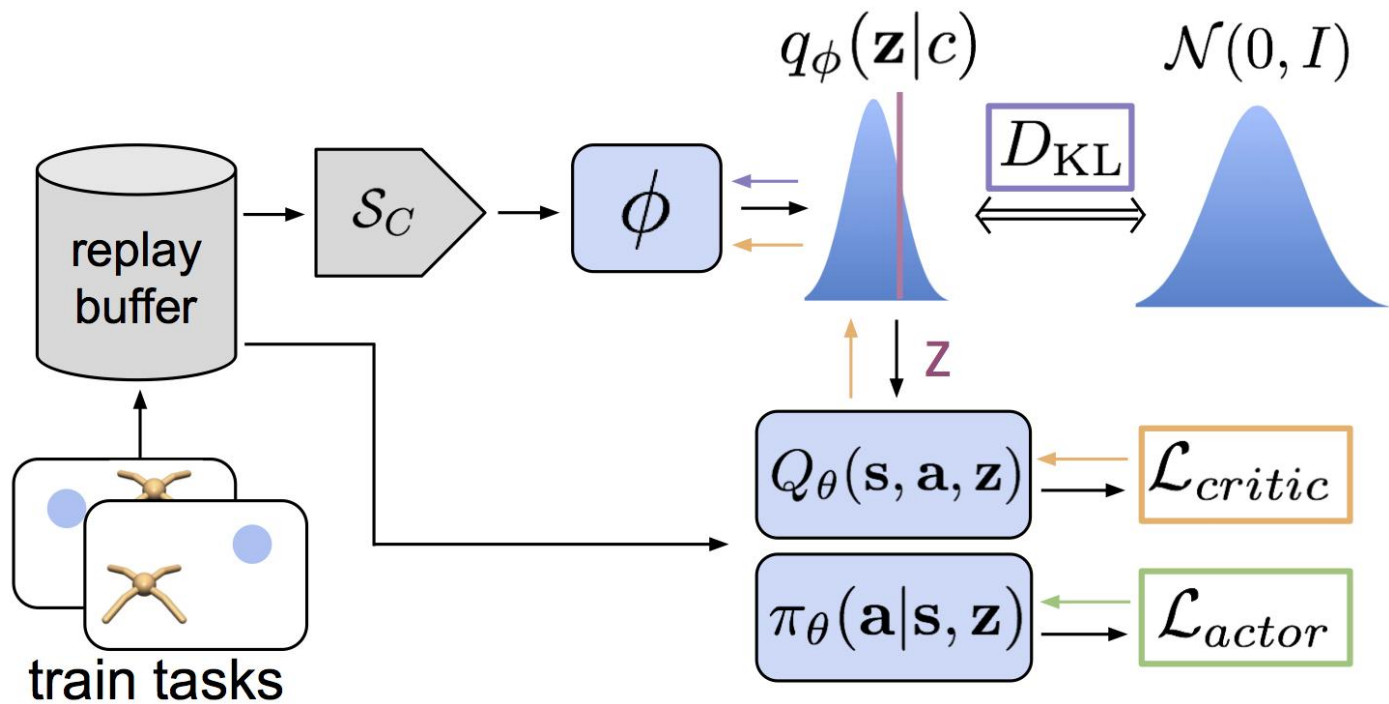
Soft Actor-Critic



Integrating task-belief with SAC

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

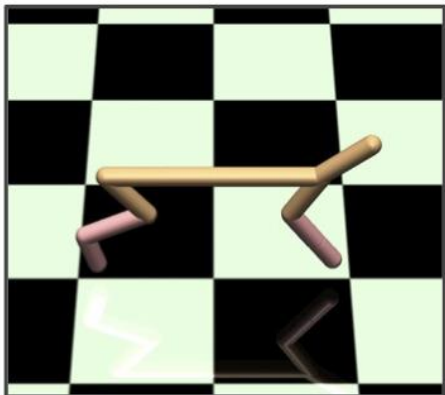
SAC where $\phi_i = f_{\theta}(\mathcal{M}_i)$



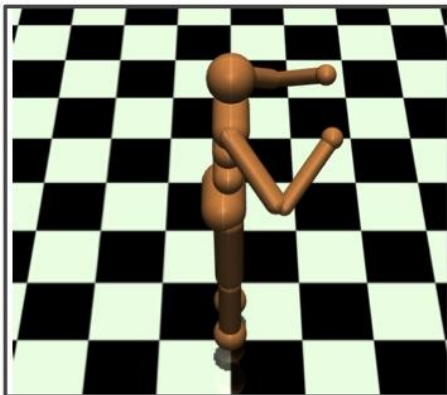
Stochastic encoder

Meta-RL experimental domains

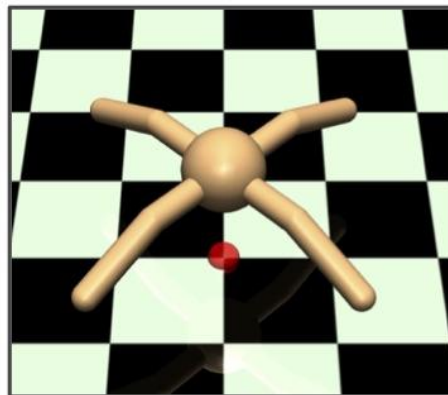
Half Cheetah



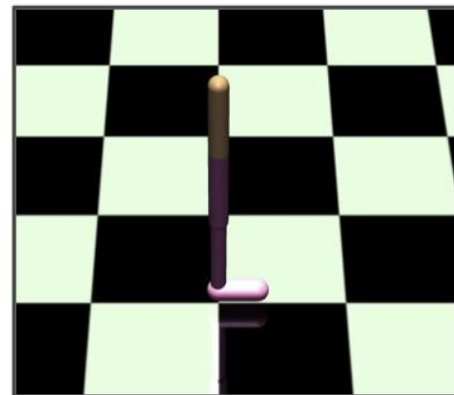
Humanoid



Ant

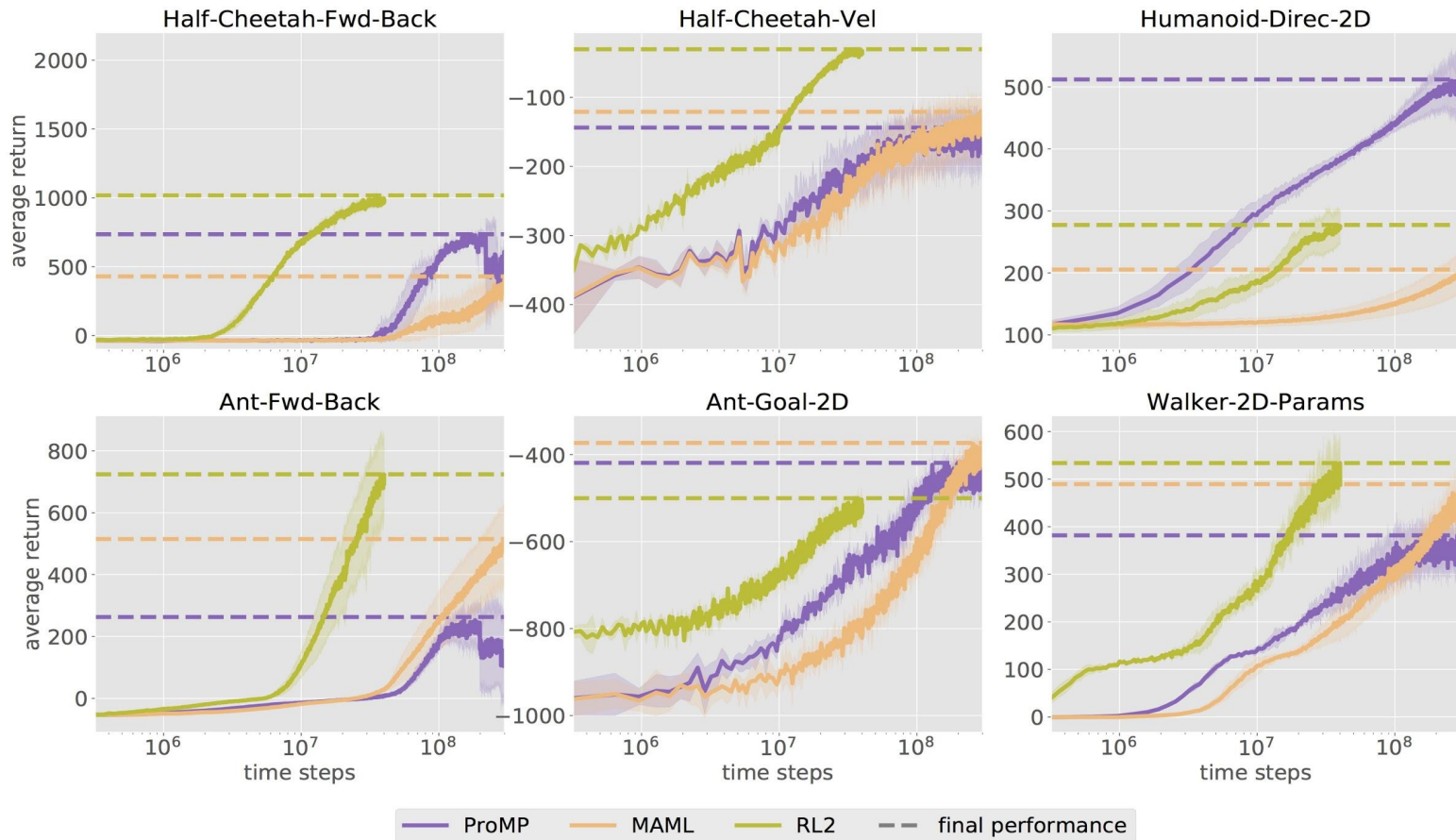


Walker

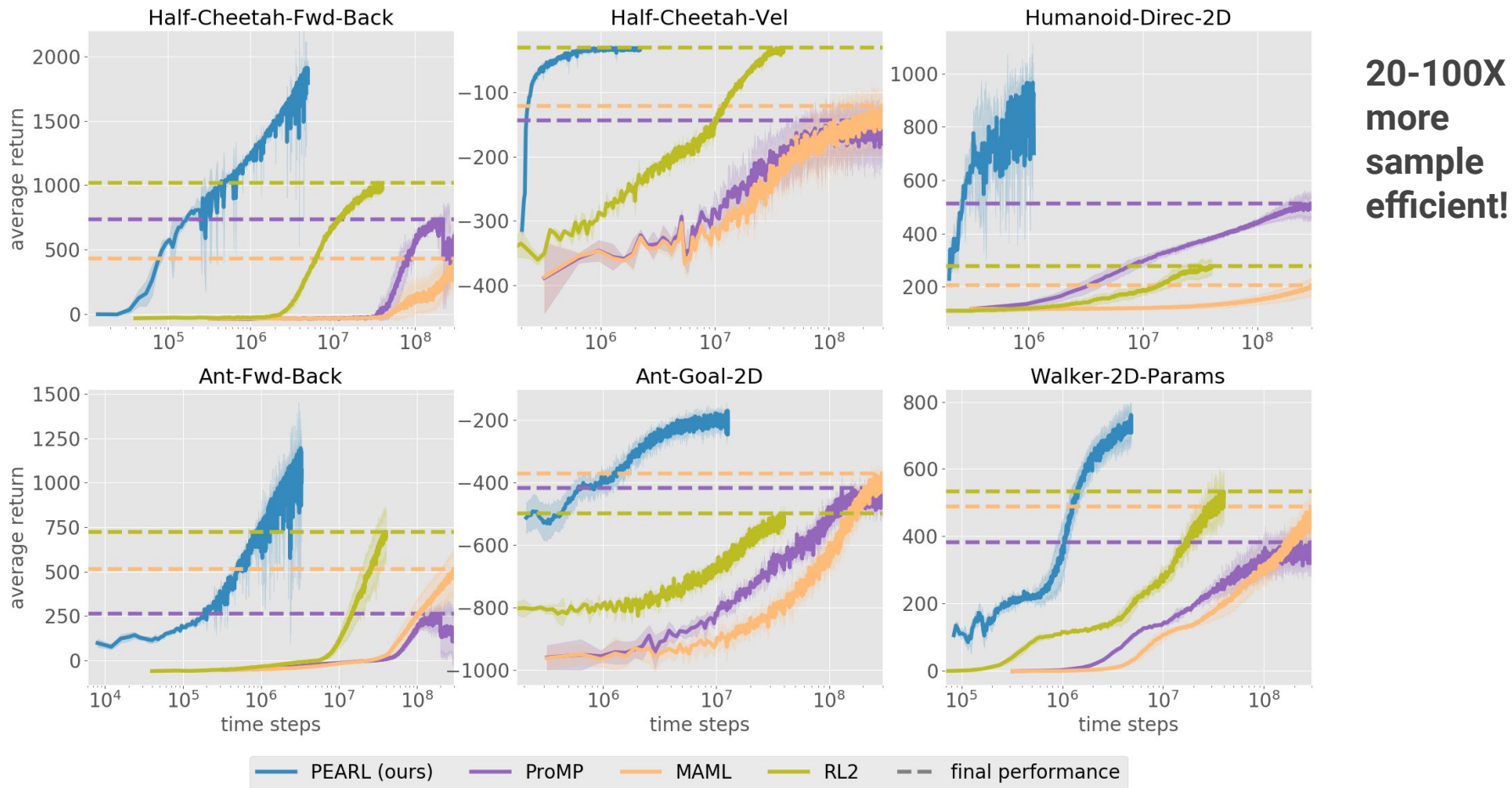


variable reward function
(locomotion direction, velocity, or goal)

variable dynamics
(joint parameters)



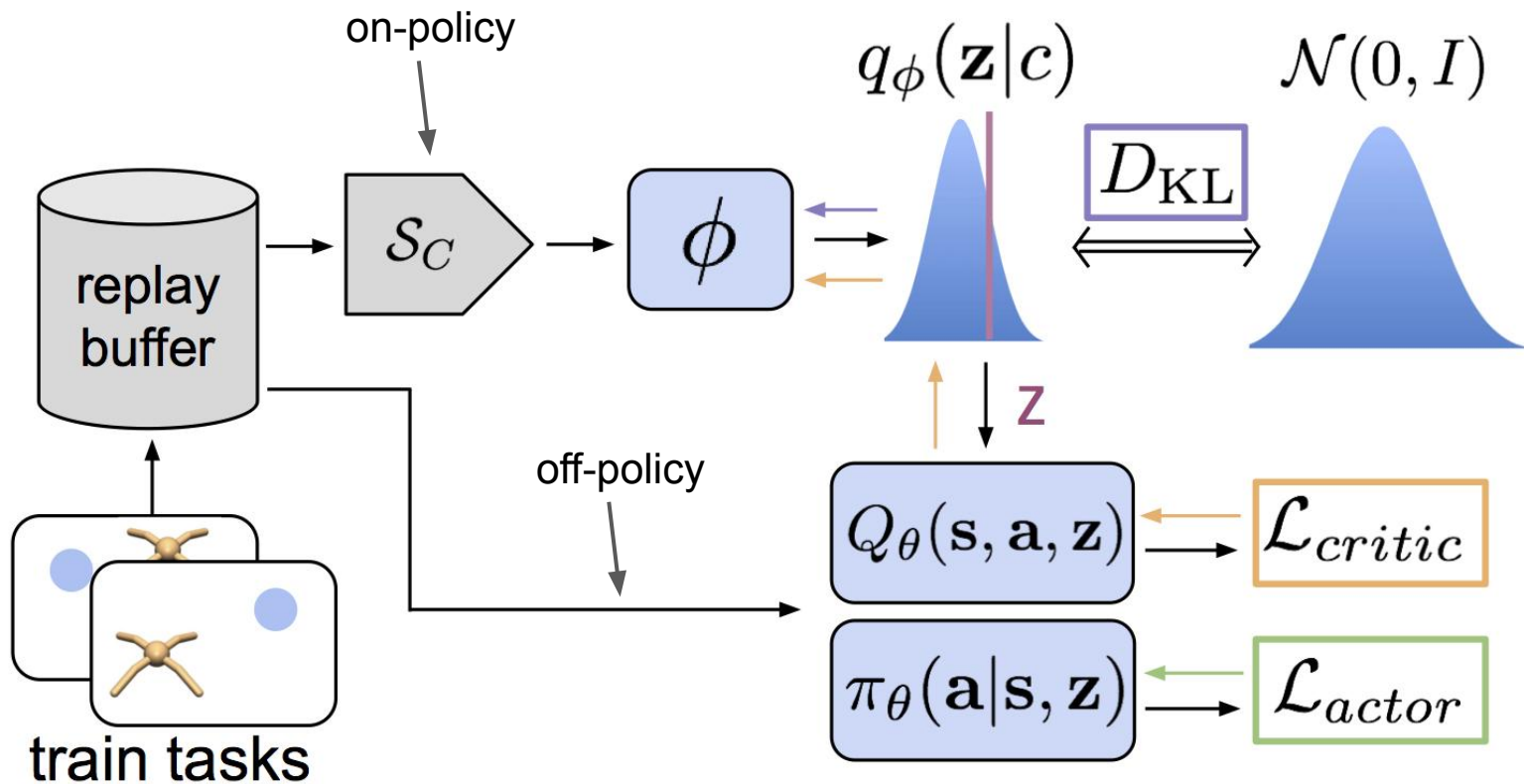
ProMP (Rothfuss et al. 2019), MAML (Finn et al. 2017), RL2 (Duan et al. 2016)



**20-100X
more
sample
efficient!**

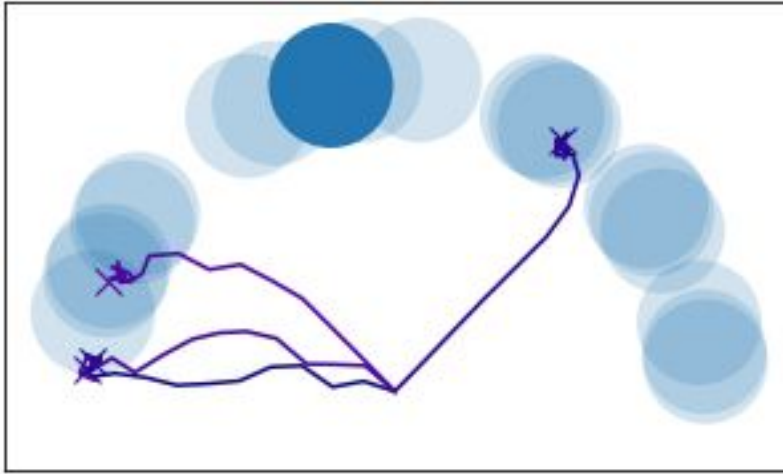
ProMP (Rothfuss et al. 2019), MAML (Finn et al. 2017), RL2 (Duan et al. 2016)

Separate task-Inference and RL data

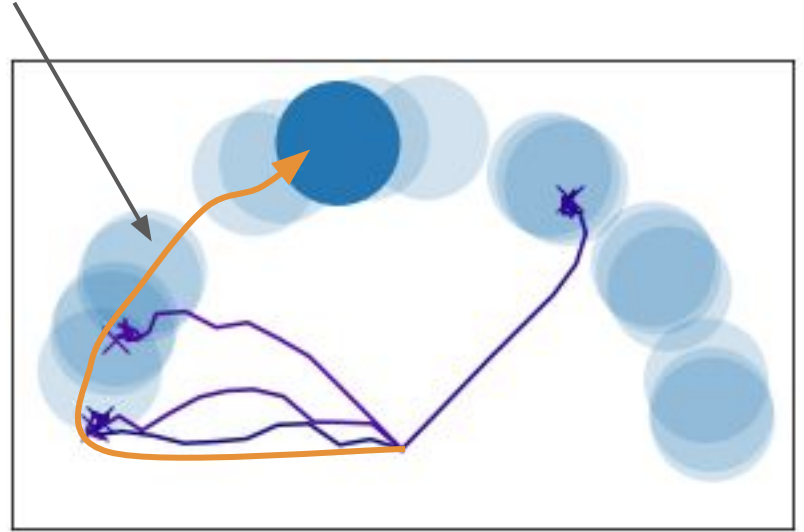


Limits of posterior sampling

Posterior sampling exploration strategy

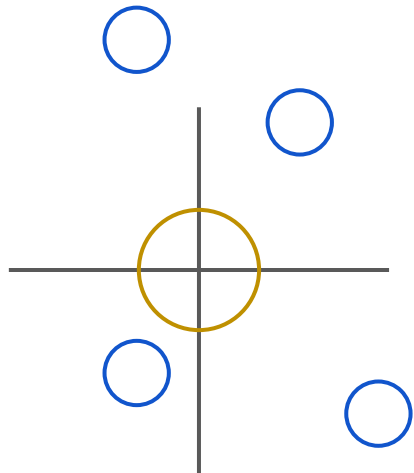


Optimal exploration strategy

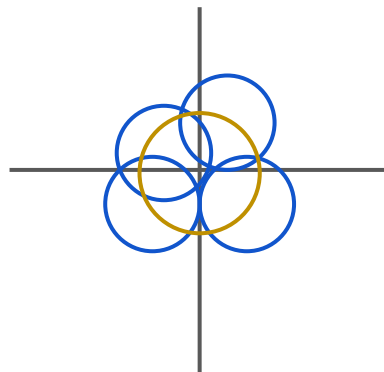




Limits of posterior sampling

MAESN (pre-adapted z constrained)



PEARL (post-adapted z constrained)



-  Prior distribution (pre-adaptation)
-  Posterior distribution (post-adaptation)

Summary

- Building on policy gradient RL, we can implement meta-RL algorithms via a recurrent network or gradient-based adaptation
- Adaptation in meta-RL includes both exploration as well as learning to perform well
- We can improve exploration by conditioning the policy on latent variables held constant across an episode, resulting in temporally-coherent strategies

Break

- meta-RL can be expressed as a particular kind of POMDP
- We can do meta-RL by inferring a belief over the task, explore via posterior sampling from this belief, and combine with SAC for a sample efficient alg.

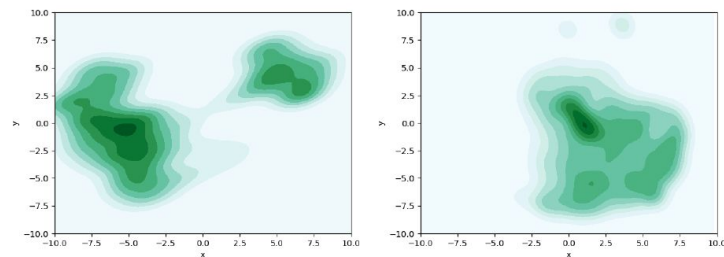
Explicitly Meta-Learn an Exploration Policy

Instantiate separate teacher (exploration) and student (target) policies

Train the exploration policy to maximize the increase in rewards earned by the target policy after training on the exploration policy's data

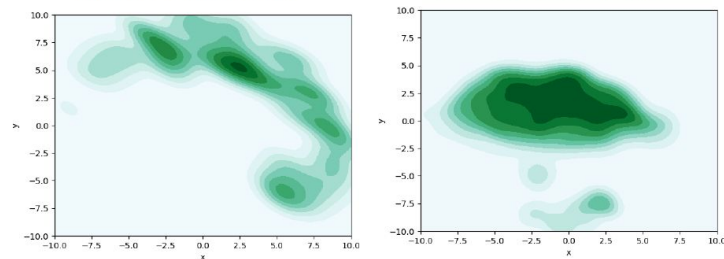
$$\hat{\mathcal{R}}(\pi, D_0) = \hat{R}_{\pi'} - \hat{R}_{\pi}$$

State visitation for student and teacher



(a) Meta-Teacher (early)

(b) Meta-Student (early)



(d) Meta-Teacher (late)

(e) Meta-Student (late)

References

Fast Reinforcement Learning via Slow Reinforcement Learning (RL2) (Duan et al. 2016), **Learning to Reinforcement Learn** (Wang et al. 2016), **Memory-Based Control with Recurrent Neural Networks** (Heess et al. 2015) - recurrent meta-RL

Model-Agnostic Meta-Learning (MAML) (Finn et al. 2017), **Proximal Meta-Policy Gradient (ProMP)** (Rothfuss et al. 2018) - gradient-based meta-RL (see ProMP for a breakdown of the gradient terms)

Meta-Learning Structured Exploration Strategies (MAESN) (Gupta et al. 2018) - temporally extended exploration with latent variables and MAML

Efficient Off-Policy Meta-RL via Probabilistic Context Variables (PEARL) (Rakelly et al. 2019) - off-policy meta-RL with posterior sampling

Soft Actor-Critic (Haarnoja et al. 2018) - off-policy RL in the maximum entropy framework

Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review (Levine 2018) - a framework for control as inference, good background for understanding SAC

(More) Efficient Reinforcement Learning via Posterior Sampling (Osband et al. 2013) - establishes a worse-case regret bound for posterior sampling that is similar to optimism-based exploration approaches

Further Reading

Stochastic Latent Actor-Critic (SLAC) (arXiv 2019) - do SAC in a latent state space inferred from image observations

Meta-Learning as Task Inference (arXiv 2019) - similar idea to PEARL and investigates different objectives to use for training the latent task space

VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning (arXiv 2019) - similar idea to PEARL and updates the latent state at every timestep rather than every trajectory, learns latent space a bit differently

Deep Variational Reinforcement Learning for POMDPs (Igl. et al. 2018) - variational inference approach for solving general POMDPs

Some Considerations on Learning to Explore with Meta-RL (Stadie et al. 2018) - does MAML but treats the adaptation step as part of the unknown dynamics of the environment (see ProMP for a good explanation of this difference)

Learning to Explore via Meta-Policy Gradient (Xu et al. 2018) - a different problem statement of learning to explore in a *single* task, an interesting approach of training the exploration policy based on differences in rewards accrued by the target policy