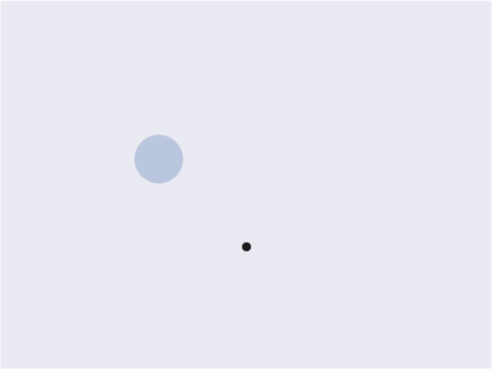
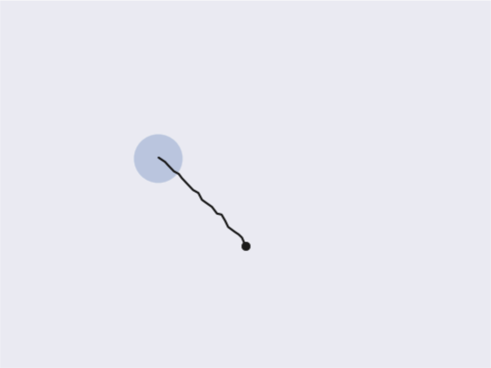
Offline Meta Reinforcement Learning

Ron Dorfman
M.Sc student, EE Department
Technion - Israel Institute of Technology

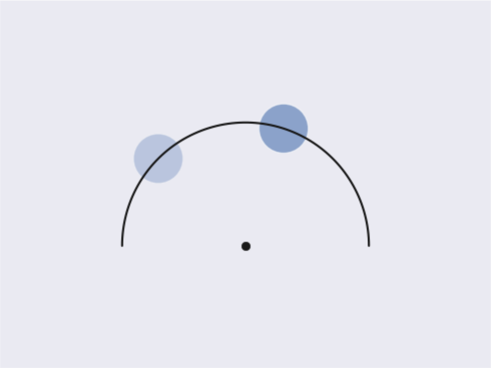
Joint work with Prof. Aviv Tamar



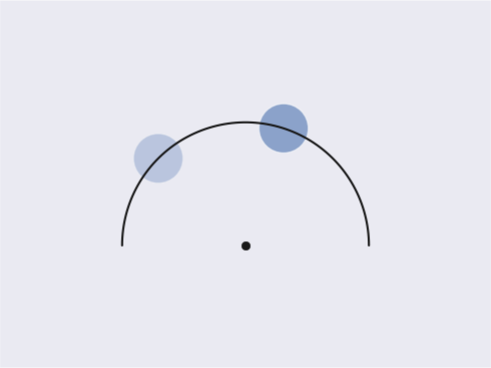
How to reach the goal?



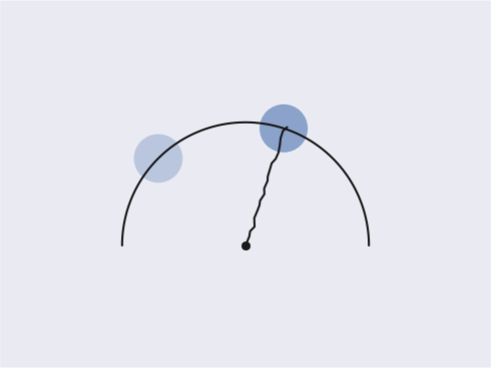
What if goal location is unknown, but we have **data** from agents trained to reach **different goals, all lie on a semi-circle**?



1. Sample possible goal

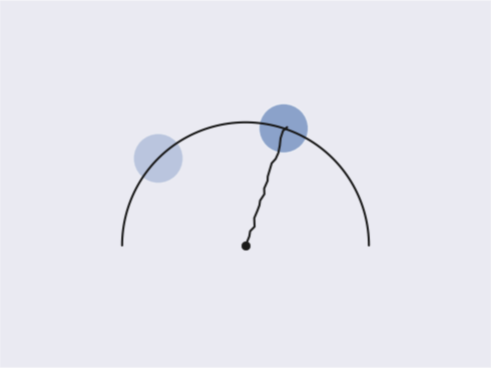


1. Sample possible goal
2. Pretend goal is correct and plan

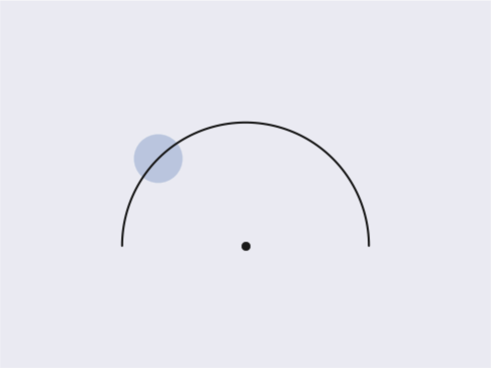


1. Sample possible goal
2. Pretend goal is correct and plan
3. Execute, observe evidence and update possible goals

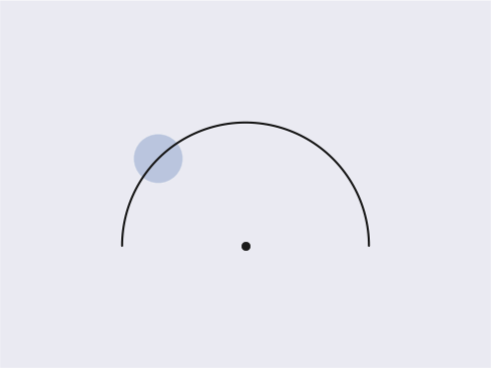
Thompson Sampling



1. Sample possible goal
2. Pretend goal is correct and plan
3. Execute, observe evidence and update possible goals



Is that the optimal thing to do?



Is that the optimal thing to do? No!

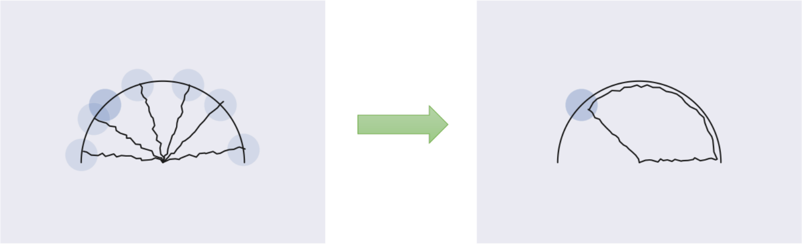
1. Search optimally across semi-circle

1. Search optimally across semi-circle
2. Go to found goal

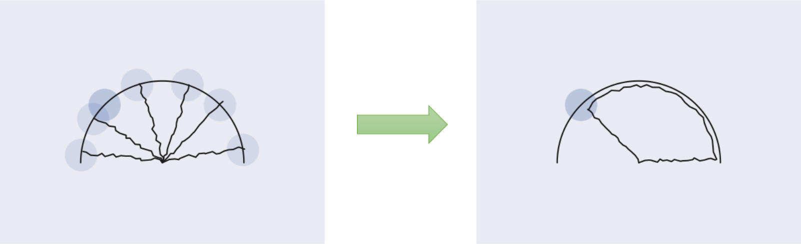
Bayes-optimal exploration

1. Search optimally across semi-circle
2. Go to found goal

Can we use collected data to learn Bayes-optimal behavior?



Can we use collected data to learn Bayes-optimal behavior?

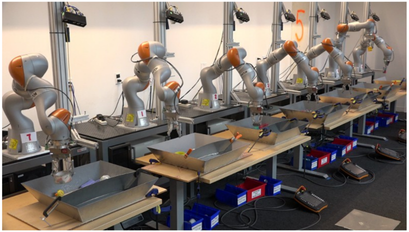


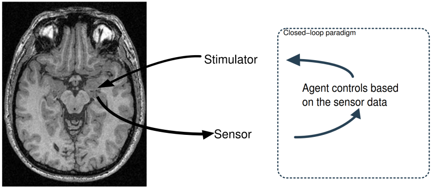
Suppose we can.. Why is it important?

Exploration generally requires **online** data collection.

Exploration generally requires **online** data collection.

Data collection can be expensive/unsafe: Robotics, Healthcare, AD, ...

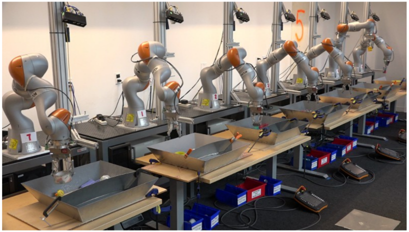


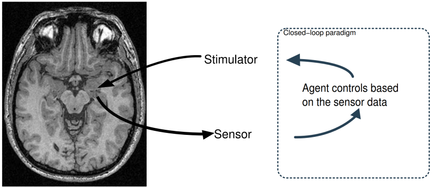


Levine et. al. (2020). Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems.

Exploration generally requires **online** data collection.

Data collection can be expensive/unsafe: Robotics, Healthcare, AD, ...





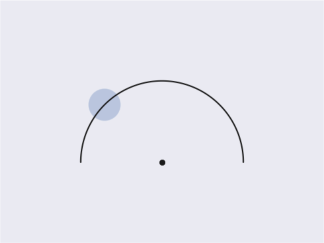
Learn to explore from **offline** data

Levine et. al. (2020). Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems.

Reinforcement Learning (RL)

- Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$.

$\mathcal{S}$ – state space
$\mathcal{A}$ – action space
$\mathcal{R}$ – reward function
$\mathcal{P}$ – transition function



- **Goal:** Find *policy* π that maximizes

$$\mathbb{E}\left[\sum_{t=0}^{H} \mathcal{R}(s_t, a_t)\right] .$$

- There exists an optimal policy π^* which is Markov, i.e., $\pi^* : \mathcal{S} \to \mathcal{A}$.

Exploration in RL

How to discover high reward strategies?

Exploration in RL

How to discover high reward strategies?

No prior information

- UCRL
- E^3
- R-max
- Exploration bonuses
- Count-based
- ...

Exploration in RL

How to discover high reward strategies?

No prior information

- UCRL
- E^3
- R-max
- Exploration bonuses
- Count-based
- ...

Efficiently search state space

Regret bounds, PAC bounds, ...

Exploration in RL

How to discover high reward strategies?

No prior information

- UCRL
- E^3
- R-max
- Exploration bonuses
- Count-based
- ...

Efficiently search state space

Regret bounds, PAC bounds, ...

Prior over MDPs

- Bayesian RL

Optimal exploration

Bayesian RL (BRL)

- Prior distribution over MDP parameters, $p(\mathcal{R}, \mathcal{P})$.
- **Goal:** Find *policy* π that maximizes

$$\mathbb{E}_{\mathcal{R},\mathcal{P}\sim p(\cdot,\cdot)}\left[\sum_{t=0}^{H}\mathcal{R}(s_t, a_t)\right] .$$

- In general, the optimal policy is **history-dependent**.
- Optimally balance exploration-exploitation: *An optimal agent takes actions that reduce its uncertainty, only if such leads to higher rewards.*

BRL as Partially-Observed MDP

- $\mathcal{R}, \mathcal{P}$ are unobserved variables.
- Collect samples:

$$h_{:t} = (s_0, a_0, r_1, s_1, \ldots, r_t, s_t) \ .$$

- Maintain *belief*:

$$b_{t+1}(\mathcal{R}, \mathcal{P}) = P(\mathcal{R}, \mathcal{P} | h_{:t+1}) \propto P(s_{t+1}, r_{t+1} | h_{:t}, \mathcal{R}, \mathcal{P}) b_t(\mathcal{R}, \mathcal{P}),$$
$$b_0(\mathcal{R}, \mathcal{P}) = p(\mathcal{R}, \mathcal{P}) \ .$$

- **Bayes-optimal policy** is of the form $\pi^*(s, b)$.

Bayes-Adaptive MDP (BAMDP)

- Hyper-state space:

$$\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$$

- Transition function:

$$\mathcal{P}^+(s^+_{t+1} | s^+_t, a_t) = \underbrace{\mathbb{E}_{b_t}\left[\mathcal{P}(s_{t+1} | s_t, a_t)\right]}_{\text{state transition}} \underbrace{\delta(b_{t+1} = P(\mathcal{R}, \mathcal{P} | h_{:t+1}))}_{\text{belief update}}$$

- Reward function:

$$\mathcal{R}^+(s^+_t, a_t) = \mathbb{E}_{b_t}\left[\mathcal{R}(s_t, a_t)\right]$$

- **Maximizing the BRL objective amounts to solving the BAMDP!**

Duff (2002). Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes.

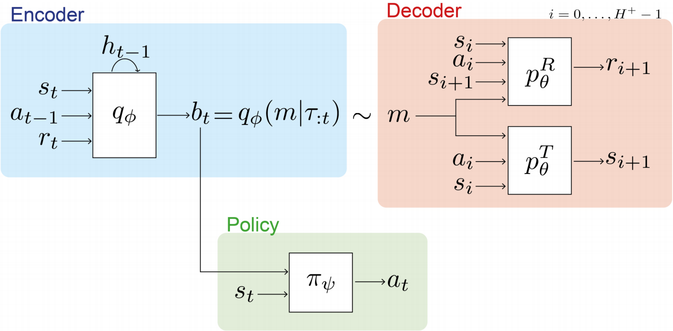
© Shimon Whiteson, NeurIPS 2019 Deep Reinforcement Learning Workshop.

- Intractable posterior update
- Intractable planning in belief space

Variational Bayes-Adaptive Deep RL (VariBAD)

Approximate using Meta-RL and Variational Inference.

- Train variational autoencoder to approximate belief.
- Train on-policy RL agent, conditioned on belief.



Zintgraf et. al. (2020). VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. ICLR.

Meta-RL

- Given access to train tasks $\mathcal{M}_1, \dots, \mathcal{M}_N \sim p(\mathcal{M}) = p(\mathcal{R}, \mathcal{P})$.
- Describe MDP $\mathcal{M}_i$ with learned latent variable m_i:

$$\mathcal{P}_i(s_{t+1}|s_t, a_t) \approx \mathcal{P}(s_{t+1}|s_t, a_t, m_i)$$
$$\mathcal{R}_i(s_t, a_t) \approx \mathcal{R}(s_t, a_t|m_i)$$

- Infer m_i by interaction with $\mathcal{M}_i$:

$$p(m_i|\tau^i_{:t}) = \mathbb{P}(m_i|s^i_0, a^i_0, r^i_1, s^i_1, \dots, r^i_t, s^i_t) \ .$$

Zintgraf et. al. (2020). VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. ICLR.

Variational Inference

Model trajectories using variational autoencoder.

- Generative model:

$$P(\tau_{:H}^{s,r}|a_{:H-1}) = \int p_\theta(m) p_\theta(\tau_{:H}^{s,r}|m, a_{:H-1}) dm$$

- Approximate posterior:

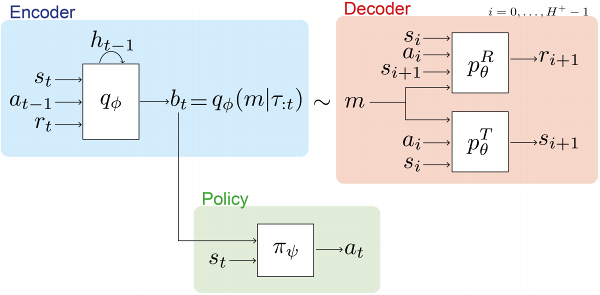
$$q_\phi(m|\tau_{:t}) = \mathcal{N}(\mu(\tau_{:t}), \Sigma(\tau_{:t}))$$

- Variational lower bound (ELBO):

$$\begin{aligned} \log P(\tau_{:H}^{s,r}|a_{:H-1}) &\geq \mathbb{E}_{q_\phi(m|\tau_{:t})}\left[\log p_\theta(\tau_{:H}^{s,r}|m, a_{:H-1})\right] - D_{KL}(q_\phi(m|\tau_{:t})||p_\theta(m)) \\ &\equiv ELBO_t \end{aligned}$$

Zintgraf et. al. (2020). VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. ICLR.

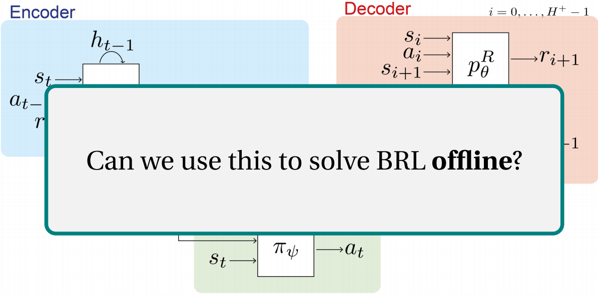
Training Procedure



1. For $i = 1, \dots, N$:
   Collect trajectories from $\mathcal{M}_i$
2. Optimize $\mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\text{VAE}}$

Zintgraf et. al. (2020). VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. ICLR.

Training Procedure



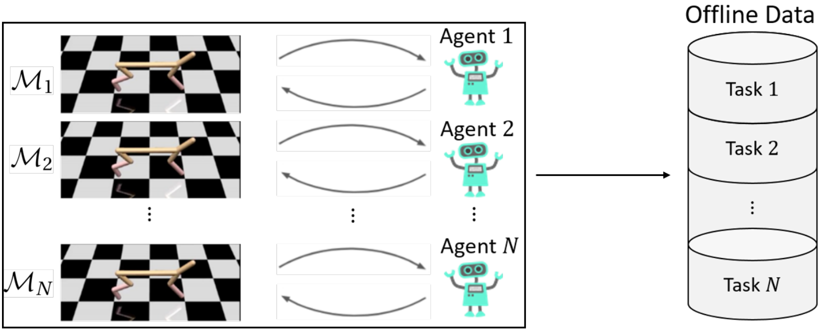
Can we use this to solve BRL **offline**?

1. For $i = 1, \ldots, N$:
   Collect trajectories from $\mathcal{M}_i$
2. Optimize $\mathcal{L}_{\text{RL}} + \lambda \mathcal{L}_{\text{VAE}}$

Zintgraf et. al. (2020). VariBAD: A Very Good Method for Bayes-Adaptive Deep RL via Meta-Learning. ICLR.

Offline Setting

In this work, offline data is **entire** training histories of RL agents.



State Relabelling

- Train VAE using trajectories in data.

State Relabelling

- Train VAE using trajectories in data.

- Relabel states:
  1. Run encoder on every partial trajectory $\tau_{:t}$. Obtain $b_t \approx (\mu_t, \Sigma_t)$.
  2. Replace each s_t in data with $s_t^+ = (s_t, \mu_t, \Sigma_t)$.

Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$

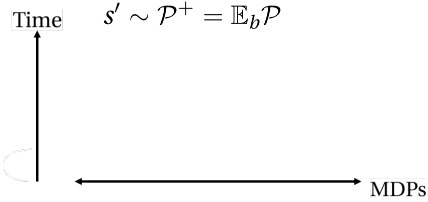
Reality

$$\begin{matrix} r \sim \mathcal{R}_\mathcal{M} \\ s' \sim \mathcal{P}_\mathcal{M} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

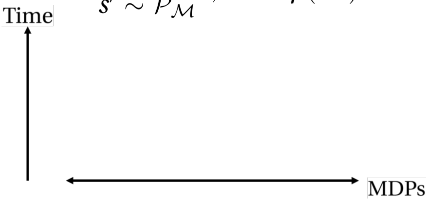
Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



Reality

$$\begin{matrix} r \sim \mathcal{R}_{\mathcal{M}} \\ s' \sim \mathcal{P}_{\mathcal{M}} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$

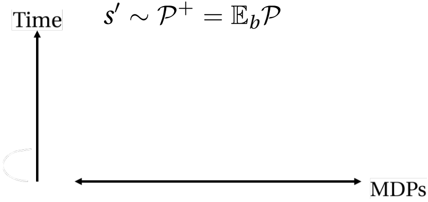


Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

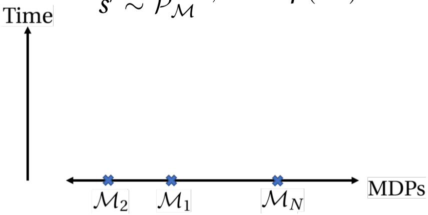
Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



Reality

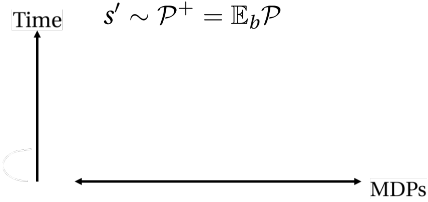
$$\begin{matrix} r \sim \mathcal{R}_{\mathcal{M}} \\ s' \sim \mathcal{P}_{\mathcal{M}} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$



Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

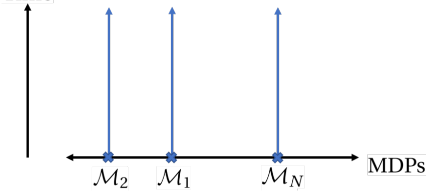
Ideal

$$r \sim \mathcal{R}^{+} = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^{+} = \mathbb{E}_b \mathcal{P}$$



Reality

$$\begin{matrix} r \sim \mathcal{R}_{\mathcal{M}} \\ s' \sim \mathcal{P}_{\mathcal{M}} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$

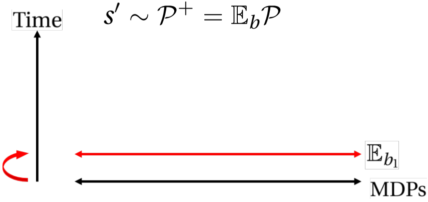


Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

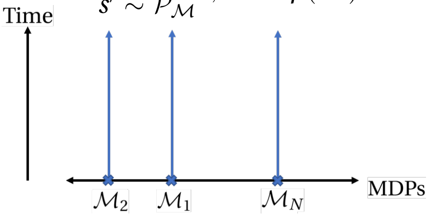
<u>Ideal</u>

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



<u>Reality</u>

$$\begin{matrix} r \sim \mathcal{R}_{\mathcal{M}} \\ s' \sim \mathcal{P}_{\mathcal{M}} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$

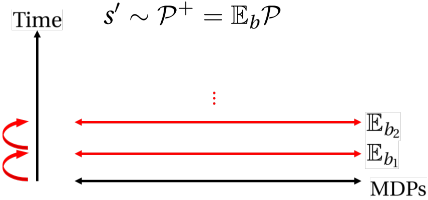


Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

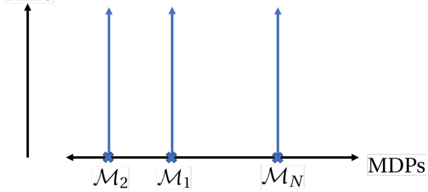
Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



Reality

$$\begin{matrix} r \sim \mathcal{R}_\mathcal{M} \\ s' \sim \mathcal{P}_\mathcal{M} \end{matrix}, \mathcal{M} \sim p(\mathcal{M})$$

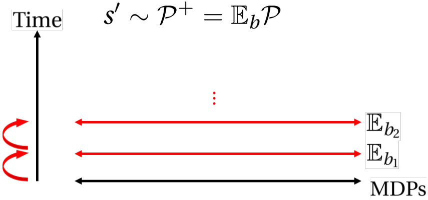


Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

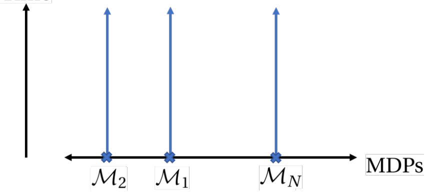
Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



Reality

$$r \sim \mathcal{R}_{\mathcal{M}}, \quad s' \sim \mathcal{P}_{\mathcal{M}}, \quad \mathcal{M} \sim p(\mathcal{M})$$



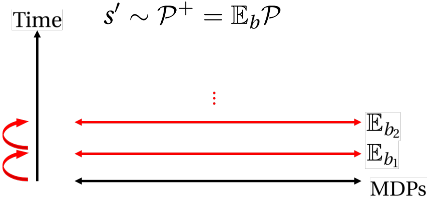
Equivalent!

Off-policy VariBAD

Can't use on-policy RL in offline setting! For off-policy, need tuples (s, a, r, s')

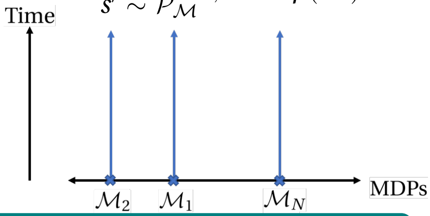
Ideal

$$r \sim \mathcal{R}^+ = \mathbb{E}_b \mathcal{R}$$
$$s' \sim \mathcal{P}^+ = \mathbb{E}_b \mathcal{P}$$



Reality

$$\left.\begin{array}{l} r \sim \mathcal{R}_\mathcal{M} \\ s' \sim \mathcal{P}_\mathcal{M} \end{array}\right., \mathcal{M} \sim p(\mathcal{M})$$



Proposition

Sample $\mathcal{R}, \mathcal{P} \sim p(\mathcal{R}, \mathcal{P})$. Collect trajectory according to $a_t \sim \pi(\cdot|h_{:t})$, $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and $r_{t+1} \sim \mathcal{R}(\cdot|s_t, a_t)$. Then,

$$\mathbb{P}(s_{t+1}|s_0, a_0, r_1, \ldots, s_t, a_t) = \mathbb{E}_{\mathcal{R},\mathcal{P}\sim b_t} \mathcal{P}(s_{t+1}|s_t, a_t),$$
$$\mathbb{P}(r_{t+1}|s_0, a_0, r_1, \ldots, s_t, a_t) = \mathbb{E}_{\mathcal{R},\mathcal{P}\sim b_t} \mathcal{R}(r_{t+1}|s_t, a_t).$$

Off-policy VariBAD

Conclusion

Any off-policy RL algorithm can be used on the hyper-state tuples in our data.

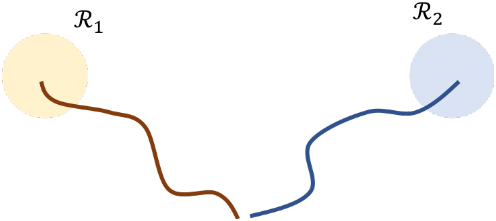
Off-policy VariBAD

Conclusion

Any off-policy RL algorithm can be used on the hyper-state tuples in our data.

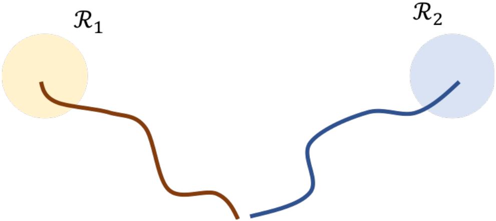
Is that it?

The MDP Ambiguity Problem



Two different MDPs or a single MDP with rewards at both circles?

The MDP Ambiguity Problem



Two different MDPs or a single MDP with rewards at both circles?

Unique problem to the offline Meta-RL setting!

The MDP Ambiguity Problem

Two different MDPs or a single MDP with rewards at both circles?

Unique problem to the offline Meta-RL setting!

During the VAE training

Reward Relabelling

- **Problem:** For each MDP, different part of state space is visited.

Reward Relabelling

- **Problem:** For each MDP, different part of state space is visited.
- Make state distribution uniform across MDPs:

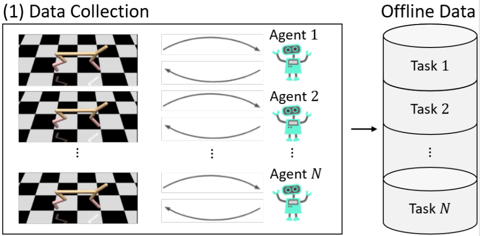
  1. Let $\tau^i = \left(s_0^i, a_0^i, r_1^i, s_1^i, \ldots, r_H^i, s_H^i\right)$ from $\mathcal{M}_i$.
  2. Sample randomly $i' \neq i$. Relabel rewards:

$$\hat{\tau}^i = \left(s_0^i, a_0^i, \hat{r}_1^i, s_1^i, \ldots, \hat{r}_H^i, s_H^i\right)$$

where $\hat{r}_{t+1}^i = \mathcal{R}_{i'}(s_t^i, a_t^i)$

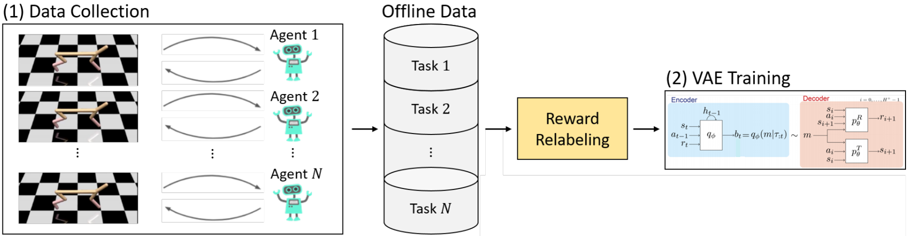
- Requires access to $\mathcal{R}_i$ for each $\mathcal{M}_i$.

Our Method

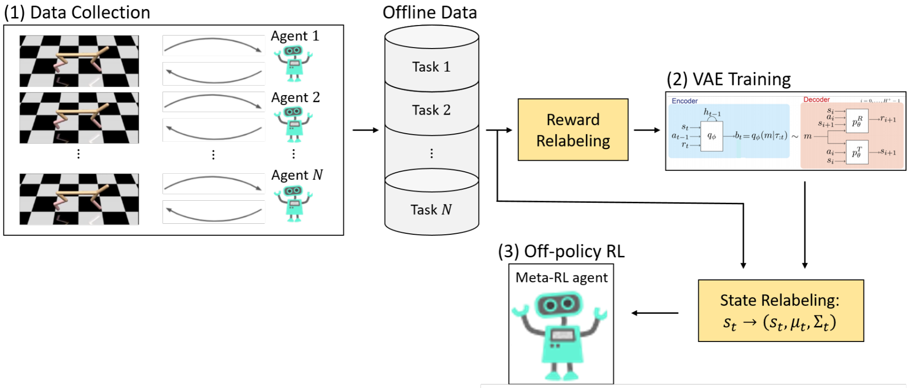


(1) Data Collection
Agent 1
Agent 2
Agent N
Offline Data
Task 1
Task 2
Task N

Our Method

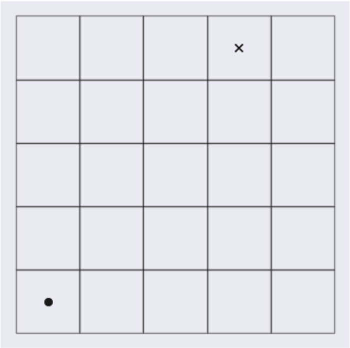

(1) Data Collection
Agent 1
Agent 2
Agent N
Offline Data
Task 1
Task 2
Task N
Reward Relabeling
(2) VAE Training
Encoder
Decoder

Our Method

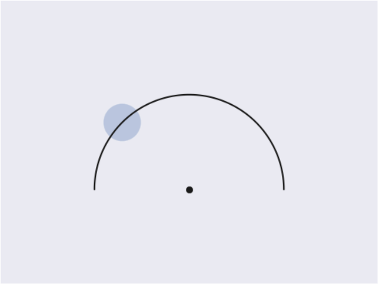

(1) Data Collection
Agent 1
Agent 2
Agent N
Offline Data
Task 1
Task 2
Task N
Reward Relabeling
(2) VAE Training
Encoder
Decoder
(3) Off-policy RL
Meta-RL agent
State Relabeling:
$s_t \rightarrow (s_t, \mu_t, \Sigma_t)$

Illustrative Domains

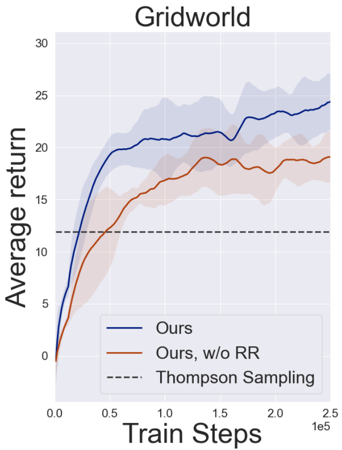
Gridworld

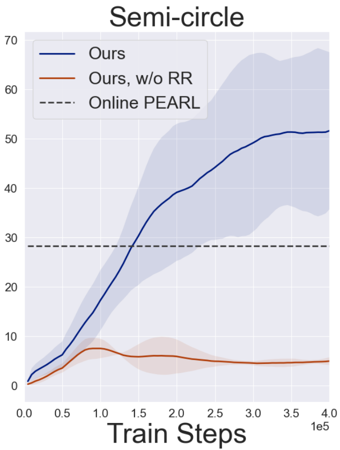


Semi-circle



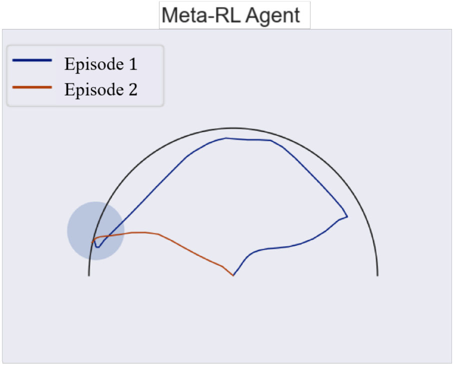
Illustrative Domains - Performance





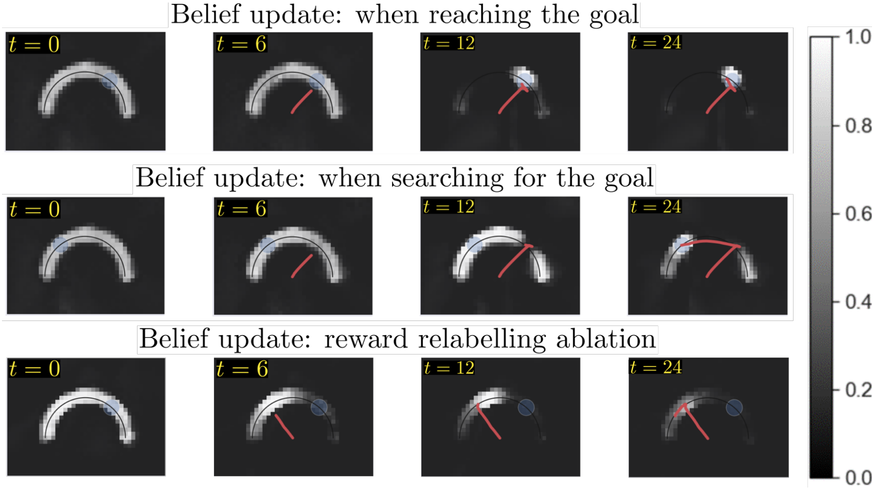
Semi-circle

Offline Data
Agent 1
Agent 2
Agent 3
Agent 4



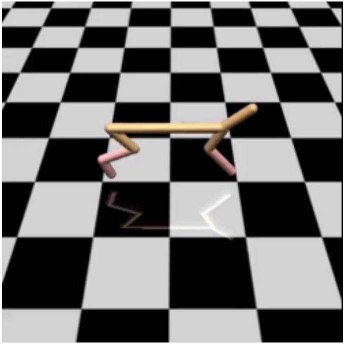
Meta-RL Agent
Episode 1
Episode 2

Semi-circle - Belief Visualization

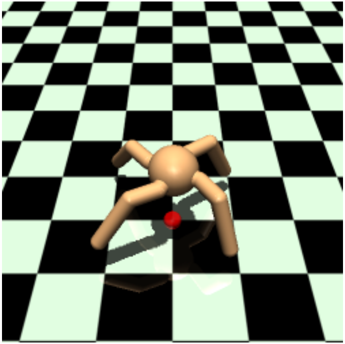


MuJoCo Domains

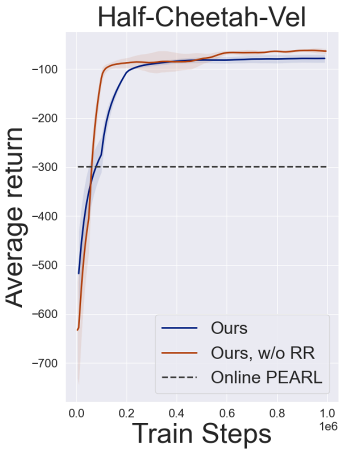
Half-Cheetah-Vel

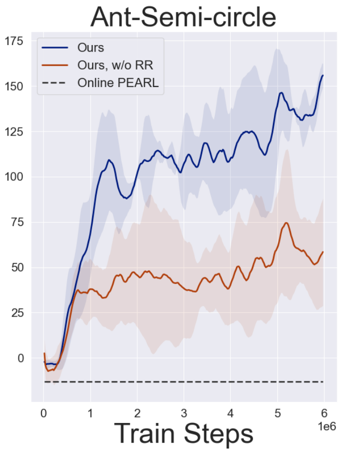


Ant-Semi-circle



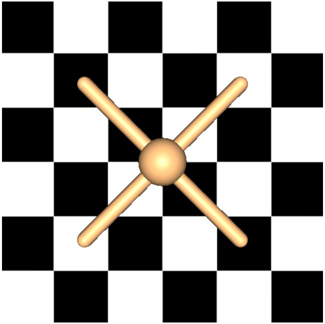
MuJoCo Domains - Performance



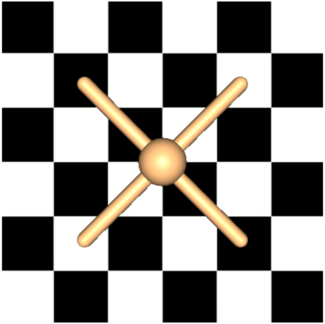


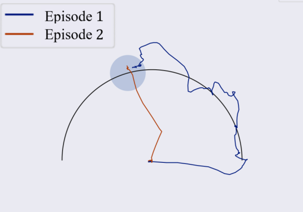
Ant-Semi-circle

Episode 1

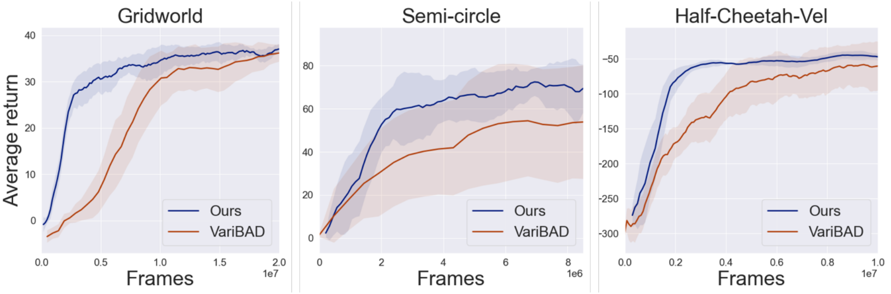


Episode 2



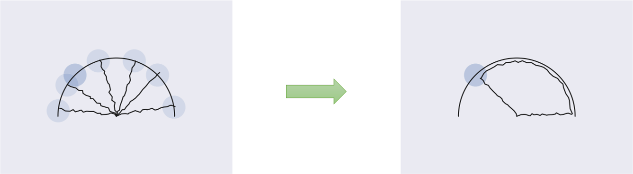


Online Setting Performance



Summary

- Formalized offline Meta-RL as BRL.



- Demonstrated learning an approximately Bayes-optimal policy.
- Sample efficient off-policy RL optimization.