KTH ROYAL INSTITUTE OF TECHNOLOGY

FDD3359 Reinforcement Learning Course
Offline RL

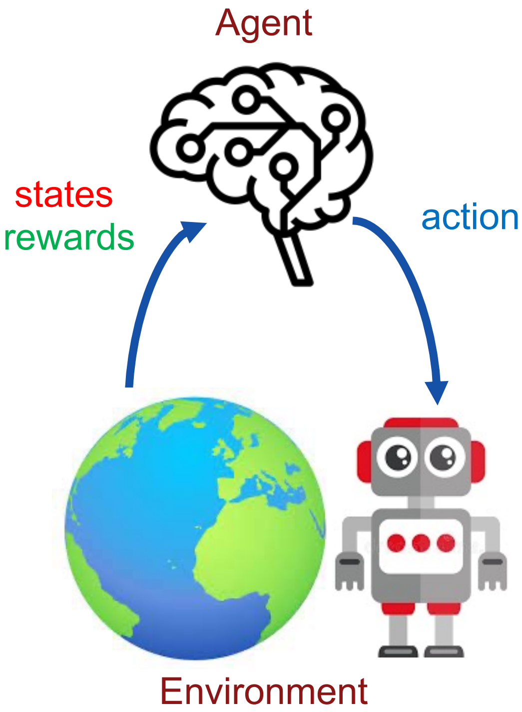
Ali Ghadirzadeh

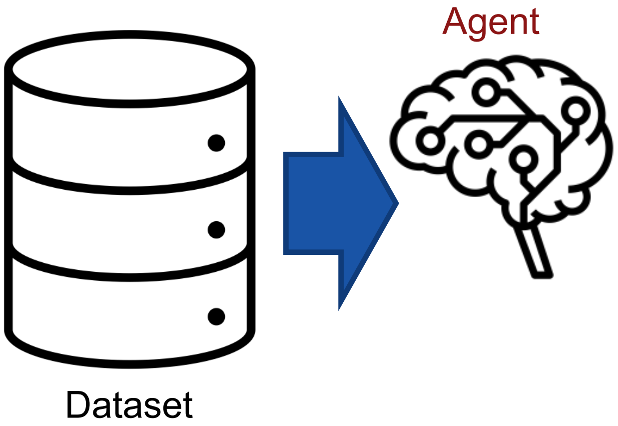
Offline RL?

A basic reinforcement learning agent AI interacts with its environment in discrete time steps. At each time t, the agent receives the current state s_t and reward r_t. It then chooses an action a_t from the set of available actions, which is subsequently sent to the environment. The environment moves to a new state s_{t+1} and the reward r_{t+1} associated with the *transition* (s_t, a_t, s_{t+1}) is determined. The goal of a reinforcement learning agent is to learn a *policy*: $\pi : A \times S \rightarrow [0,1], \pi(a,s) = \Pr(a_t = a \mid s_t = s)$ which maximizes the expected cumulative reward.

WIKIPEDIA
The Free Encyclopedia

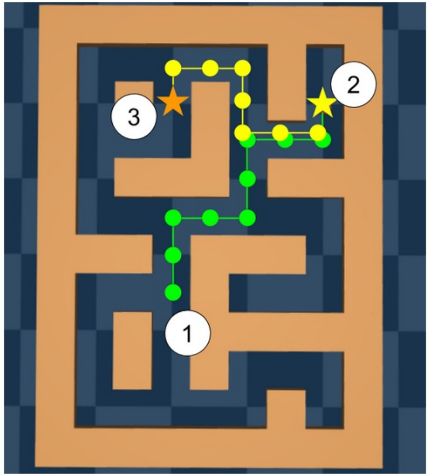
Offline RL?

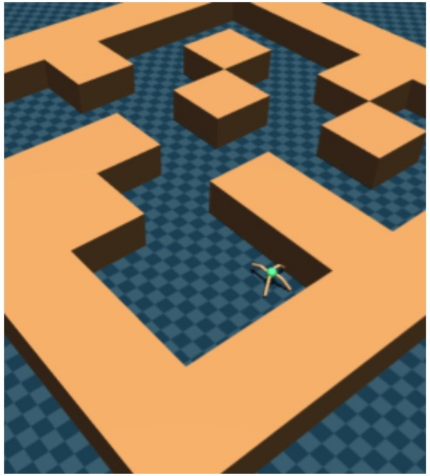




Maze2D and AntMaze

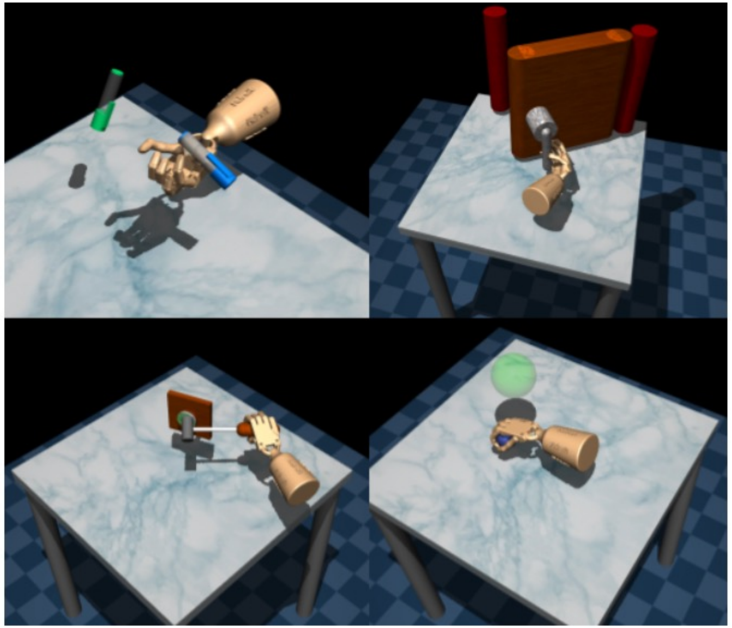
The maze environments are designed to test the ability of agents to recombine existing data in novel ways. For example, if an agent sees trajectories 1-2 and 2-3, it can form a shortest path from 1-3. Two robots are available - a simple ball and the "Ant" robot from the Gym benchmark.





Offline RL?

D4RL



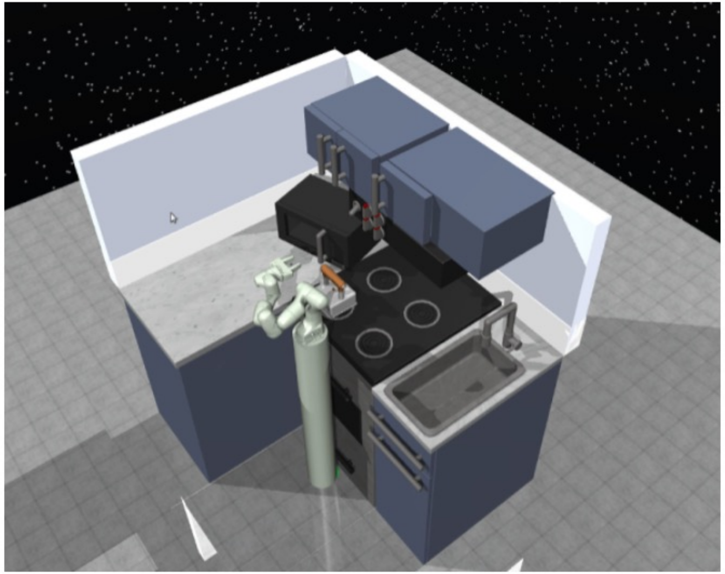
Adroit

The Adroit domain includes motion-captured human data on a realistic, high-DoF robotic hand. A variety of challenging tasks from the original paper are included, including pen twirling, opening a door, using a hammer, and relocating an object.

Offline RL?

FrankKitchen

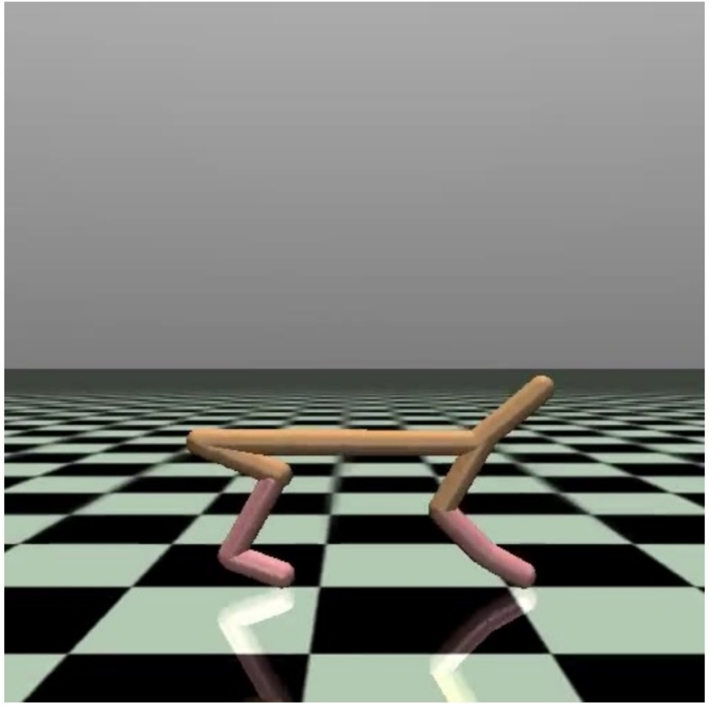
The FrankaKitchen domain is based on the Adept environment. This domain offers a challenging manipulation problem in an unstructured environment with many possible tasks to perform.



Offline RL?

Gym

Several OpenAI Gym benchmark tasks are included with data collected by a variety of pre-trained RL agents. This includes the Hopper, HalfCheetah, and Walker environments.



Why offline RL

Why offline RL

"Place Grapes in Ceramic Bowl"

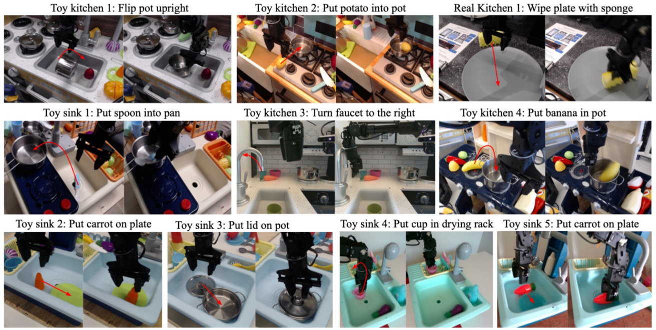
"Place Bottle In Tray"

"Push Purple Bowl Across The Table"

"Wipe Tray With Sponge"

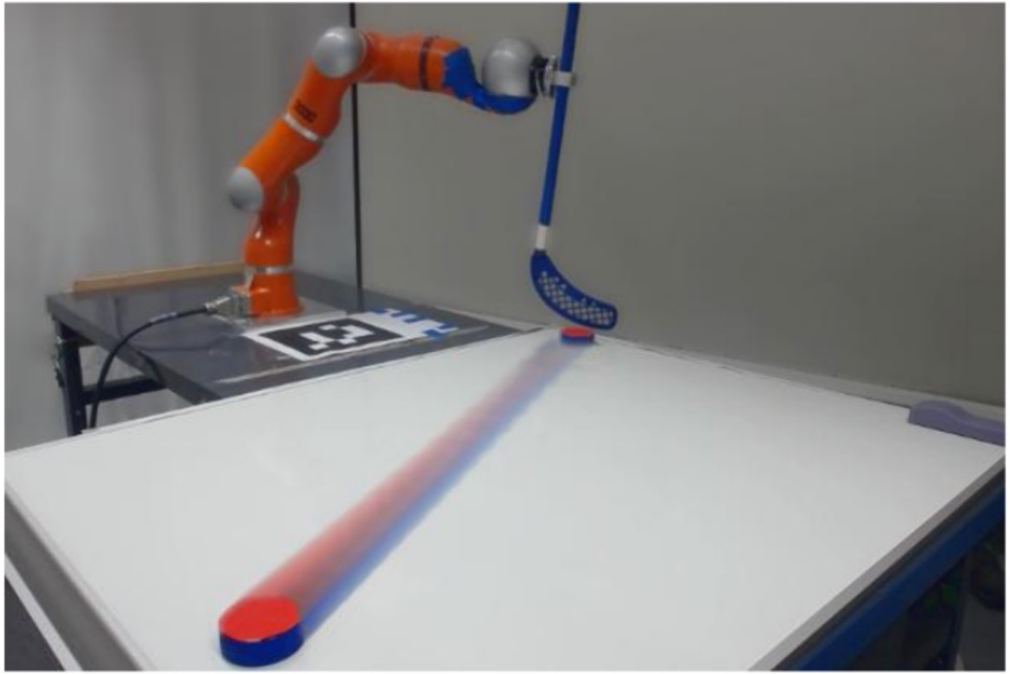
BC-Z dataset

RoboNet



Bridge Dataset

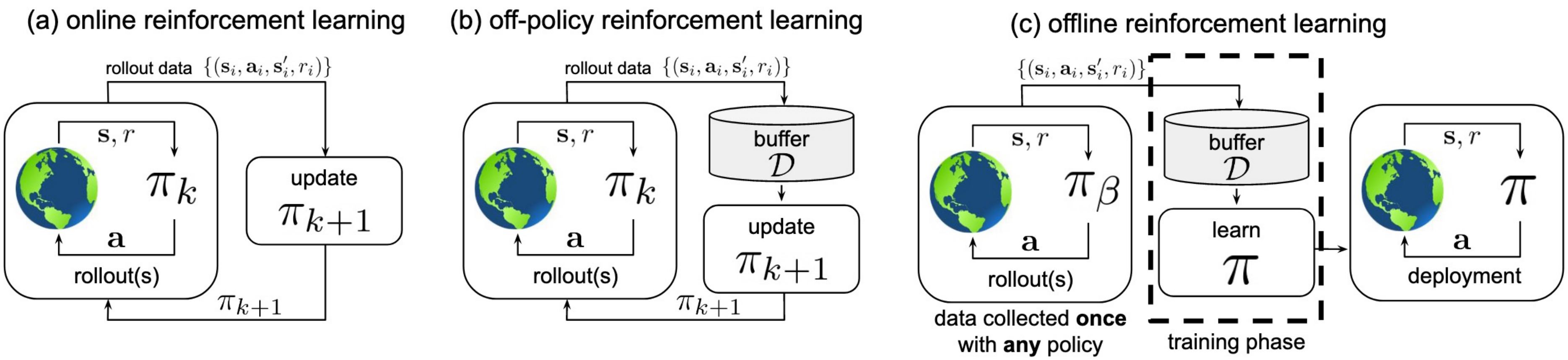
Why offline RL?



Off-policy Reinforcement Learning

Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

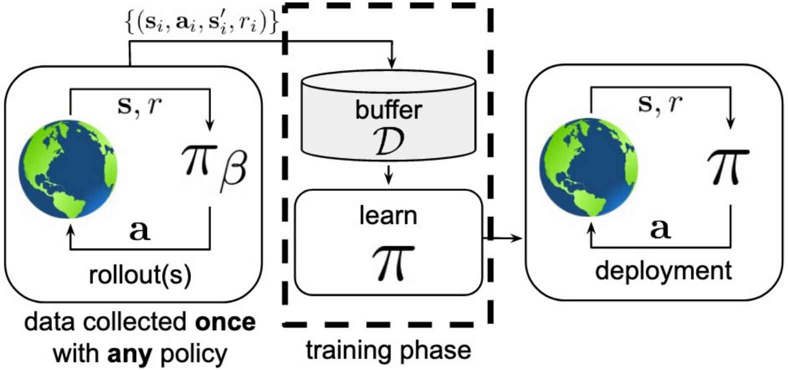
Sergey Levine[1,2], Aviral Kumar[1], George Tucker[2], Justin Fu[1]
[1]UC Berkeley, [2]Google Research, Brain Team



Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

Sergey Levine[1,2], Aviral Kumar[1], George Tucker[2], Justin Fu[1]
[1]UC Berkeley, [2]Google Research, Brain Team

Actor-Critic RL



$$\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)_j\}$$

Critic Update $$\mathcal{T}^\pi Q(s,a) = \mathbb{E}_{s'}[r + \gamma Q(s', \pi(s'))]$$

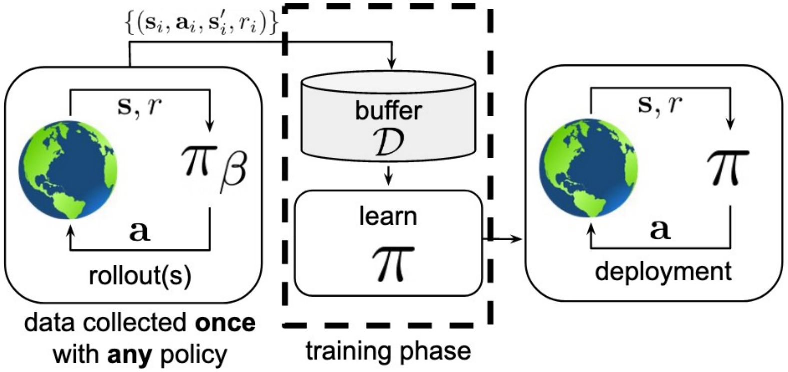
Critic Update $$\phi \leftarrow \operatorname{argmax}_\phi \mathbb{E}_{s \in \mathcal{B}}[Q_\theta(s, \pi_\phi(s))]$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\tau \sim p_\pi(\tau | \mathbf{s}_t, \mathbf{a}_t)}\left[\sum_{t'=t}^{H} \gamma^{t'-t} r(\mathbf{s}_t, \mathbf{a}_t)\right]$$

Policy Constraints Methods

Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

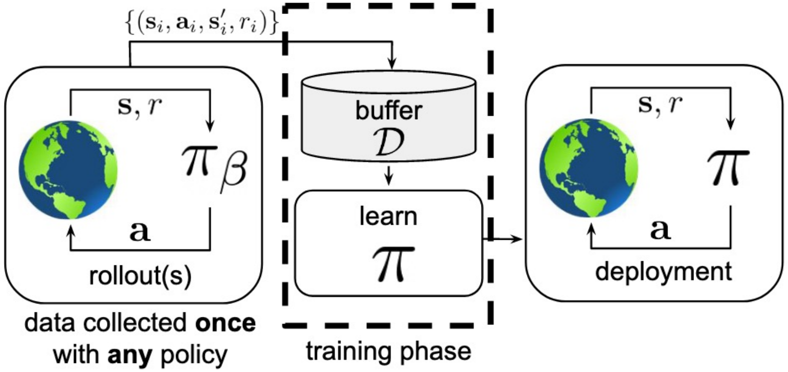
Sergey Levine[1,2], **Aviral Kumar**[1], **George Tucker**[2], **Justin Fu**[1]
[1]UC Berkeley, [2]Google Research, Brain Team



$$\hat{Q}^{\pi}_{k+1} \leftarrow \arg\min_{Q} \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi_k(\mathbf{a}'|\mathbf{s}')}[\hat{Q}^{\pi}_{k}(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right]$$

$$\pi_{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[\hat{Q}^{\pi}_{k+1}(\mathbf{s},\mathbf{a})]\right] \text{ s.t. } D(\pi,\pi_\beta) \leq \epsilon.$$

Policy Constraints Methods



Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

Sergey Levine[1,2], Aviral Kumar[1], George Tucker[2], Justin Fu[1]
[1]UC Berkeley, [2]Google Research, Brain Team

$$\hat{Q}^{\pi}_{k+1} \leftarrow \arg\min_{Q}$$

$$\mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi_k(\mathbf{a}'|\mathbf{s}')}[\hat{Q}^{\pi}_k(\mathbf{s}',\mathbf{a}')] - \alpha\gamma D(\pi_k(\cdot|\mathbf{s}'), \pi_\beta(\cdot|\mathbf{s}'))\right)\right)^2\right]$$

$$\pi_{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[\hat{Q}^{\pi}_{k+1}(\mathbf{s},\mathbf{a})] - \alpha D(\pi(\cdot|\mathbf{s}), \pi_\beta(\cdot|\mathbf{s}))\right].$$

Advantage Weighted Actor Critic

$$\pi_{k+1} = \arg\max_{\pi \in \Pi} \mathbb{E}_{\mathbf{a} \sim \pi(\cdot|\mathbf{s})}[A^{\pi_k}(\mathbf{s}, \mathbf{a})]$$

$$\text{s.t. } D_{\mathrm{KL}}(\pi(\cdot|\mathbf{s}) || \pi_\beta(\cdot|\mathbf{s})) \leq \epsilon$$

$$\int_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s}) d\mathbf{a} = 1.$$

$$V^\pi(\mathbf{s}_t) = \mathbb{E}_{\tau \sim p_\pi(\tau|\mathbf{s}_t)}\left[\sum_{t'=t}^{H} \gamma^{t'-t} r(\mathbf{s}_t, \mathbf{a}_t)\right]$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}_{\tau \sim p_\pi(\tau|\mathbf{s}_t, \mathbf{a}_t)}\left[\sum_{t'=t}^{H} \gamma^{t'-t} r(\mathbf{s}_t, \mathbf{a}_t)\right].$$

$$A(s, a) = Q_\phi(s, a) - V(s)$$

Advantage Weighted Actor Critic

$$
\begin{aligned}
\mathcal{L}(\pi, \lambda, \alpha) =& \mathbb{E}_{\mathbf{a}\sim\pi(\cdot|\mathbf{s})}[A^{\pi_k}(\mathbf{s}, \mathbf{a})] \\
&+ \lambda(\epsilon - D_{\mathrm{KL}}(\pi(\cdot|\mathbf{s})||\pi_\beta(\cdot|\mathbf{s}))) \\
&+ \alpha(1 - \int_{\mathbf{a}} \pi(\mathbf{a}|\mathbf{s})d\mathbf{a}).
\end{aligned}
$$

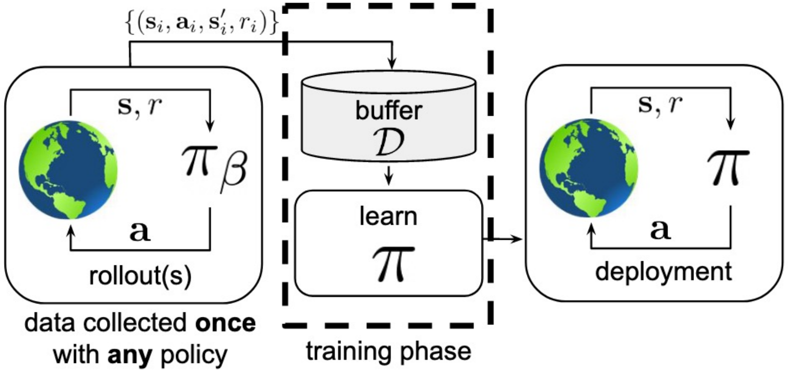
$$
\frac{\partial \mathcal{L}}{\partial \pi} = A^{\pi_k}(\mathbf{s}, \mathbf{a}) - \lambda \log \pi_\beta(\mathbf{a}|\mathbf{s}) + \lambda \log \pi(\mathbf{a}|\mathbf{s}) + \lambda - \alpha.
$$

Advantage Weighted Actor Critic

$$\pi^*(\mathbf{a}|\mathbf{s}) = \frac{1}{Z(\mathbf{s})}\pi_\beta(\mathbf{a}|\mathbf{s}) \exp\left(\frac{1}{\lambda}A^{\pi_k}(\mathbf{s},\mathbf{a})\right)$$

$$\begin{aligned}
&\arg\min_\theta \mathop{\mathbb{E}}_{\rho_{\pi_\beta}(\mathbf{s})} \left[D_{\mathrm{KL}}(\pi^*(\cdot|\mathbf{s}) || \pi_\theta(\cdot|\mathbf{s}))\right] \\
=&\arg\min_\theta \mathop{\mathbb{E}}_{\rho_{\pi_\beta}(\mathbf{s})} \left[\mathop{\mathbb{E}}_{\pi^*(\cdot|\mathbf{s})} \left[-\log \pi_\theta(\cdot|\mathbf{s})\right]\right]
\end{aligned}$$

Policy Constraints Methods



Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems

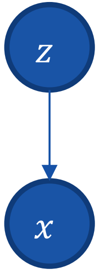
Sergey Levine[1,2], **Aviral Kumar**[1], **George Tucker**[2], **Justin Fu**[1]
[1]UC Berkeley, [2]Google Research, Brain Team

$$\hat{Q}^{\pi}_{k+1} \leftarrow \arg\min_{Q} \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim\mathcal{D}}\left[\left(Q(\mathbf{s},\mathbf{a}) - \left(r(\mathbf{s},\mathbf{a}) + \gamma\mathbb{E}_{\mathbf{a}'\sim\pi_k(\mathbf{a}'|\mathbf{s}')}[\hat{Q}^{\pi}_{k}(\mathbf{s}',\mathbf{a}')]\right)\right)^2\right]$$

$$\pi_{k+1} \leftarrow \arg\max_{\pi} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[\mathbb{E}_{\mathbf{a}\sim\pi(\mathbf{a}|\mathbf{s})}[\hat{Q}^{\pi}_{k+1}(\mathbf{s},\mathbf{a})]\right] \text{ s.t. } D(\pi,\pi_\beta) \le \epsilon.$$

Generative Models - Background

The variational lower bound



Latent variable

Data

Maximize the loglikelihood of the data

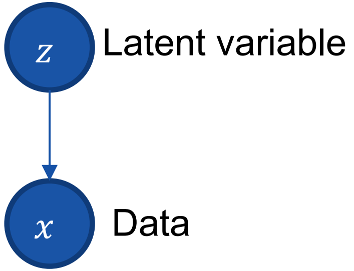
$$\begin{aligned}\log p(x) &= \log \int p(x,z)dz \\ &= \log \int_z p(x,z)\frac{q(z)}{q(z)}dz \\ &= \log \int_z p(x|z)p(z)\frac{q(z|x)}{q(z|x)}dz\end{aligned}$$

Jensen's inequality

$$\geq \mathbb{E}_{q(z|x)}[\log p(x|z)] - \mathbb{E}_{q(z)}[\log \frac{q(z)}{p(z)|x}]$$

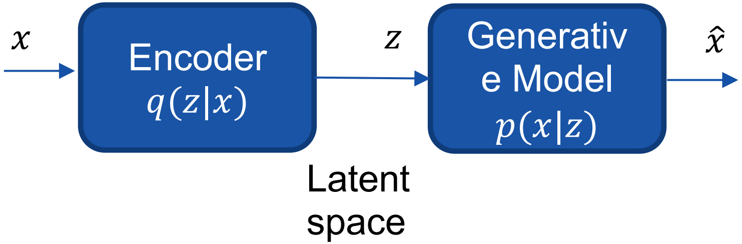
$$= \boxed{\mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z))}$$

Generative Models - Background



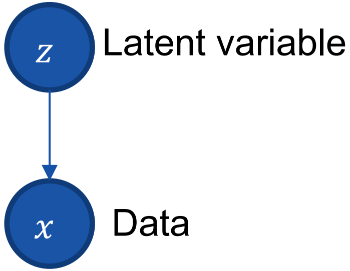
Maximize the variational lower bound

$$= \boxed{\mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}(q(z|x)||p(z))}$$

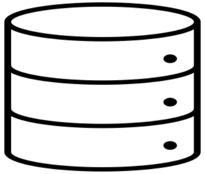


Variational Autoencoder Networks

Generative Models - Background

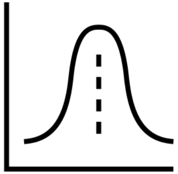


Offline RL with Generative Models

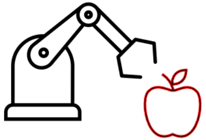


(offline data)

$$\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)_j\}$$



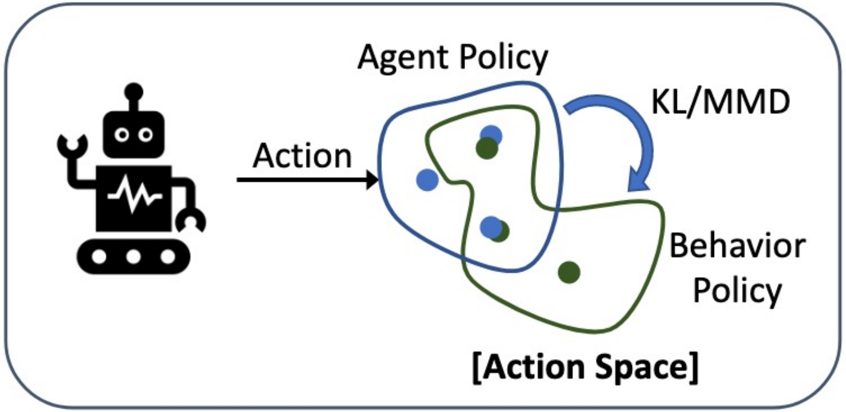
Learn the distribution
(Low-level policy)



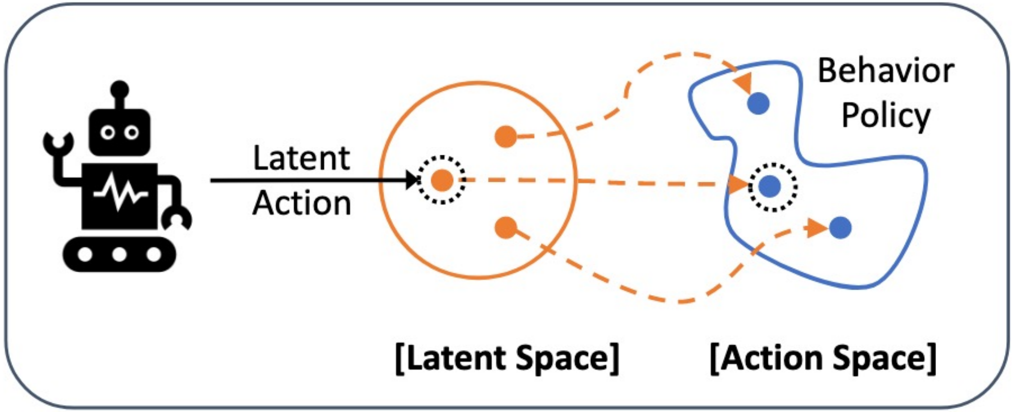
Online policy training
(High-level policy)

Policy in the Latent Action Space (PLAS)

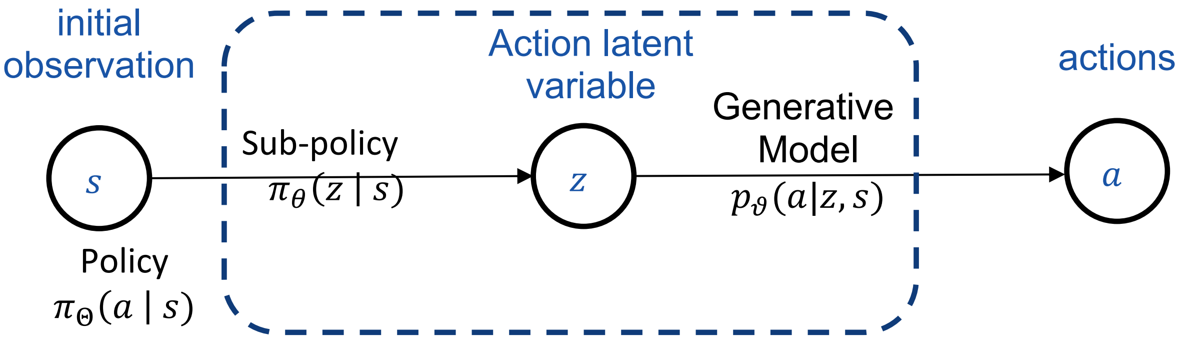
Explicit Policy Constrint



Implicit Policy Constrint using Generative Models

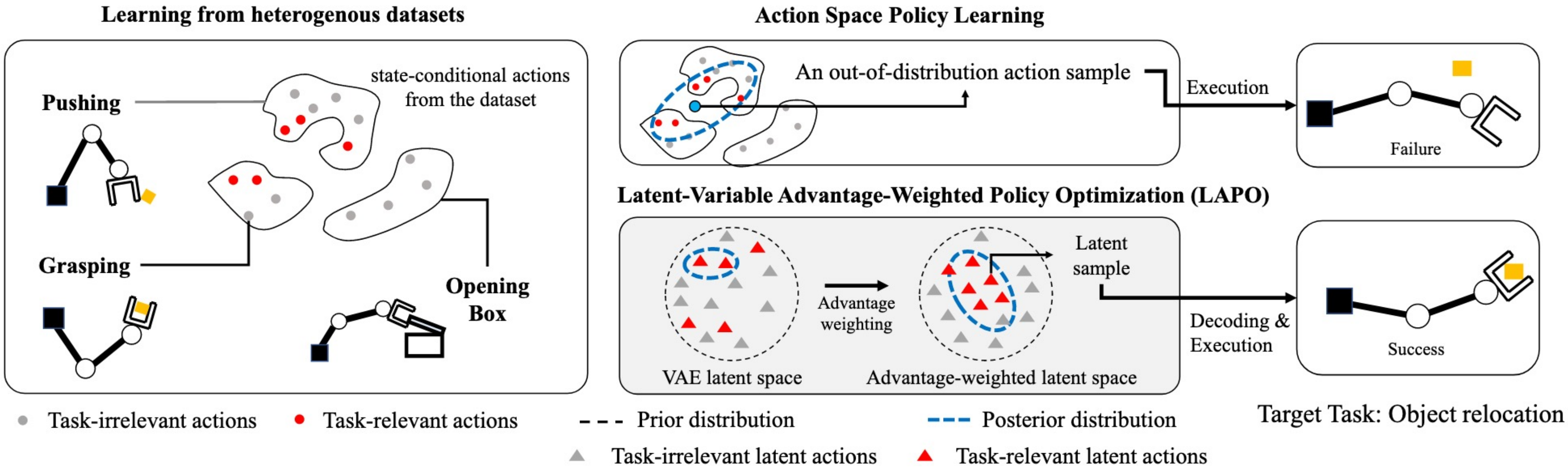


Policy in the Latent Action Space (PLAS)



$$\nabla_\vartheta J(\vartheta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_a Q(a,s)|_{a=\pi_\theta(s,z)} \nabla_z \pi_\theta(s,z)|_{z=\pi_\vartheta(s)} \nabla_\vartheta \pi_\vartheta(s)]$$

Latent-Variable Advantage-Weighted Policy Optimization

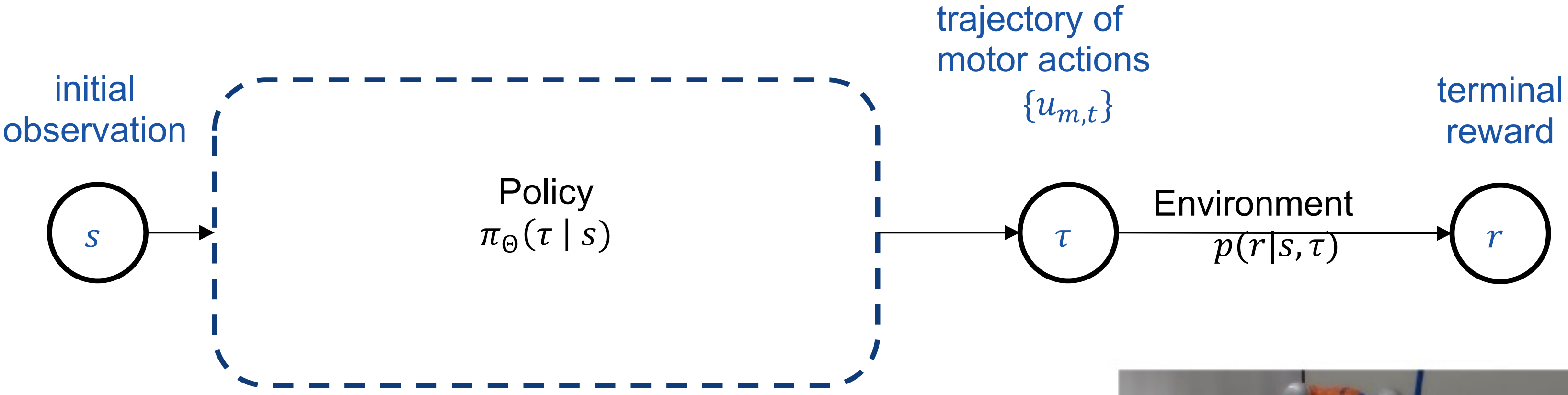


$$\pi^*(a|s) \propto \pi_\beta(a|s)\exp(A(s,a)/\lambda)$$

$$\omega = \exp(A(s,a)/\lambda)$$

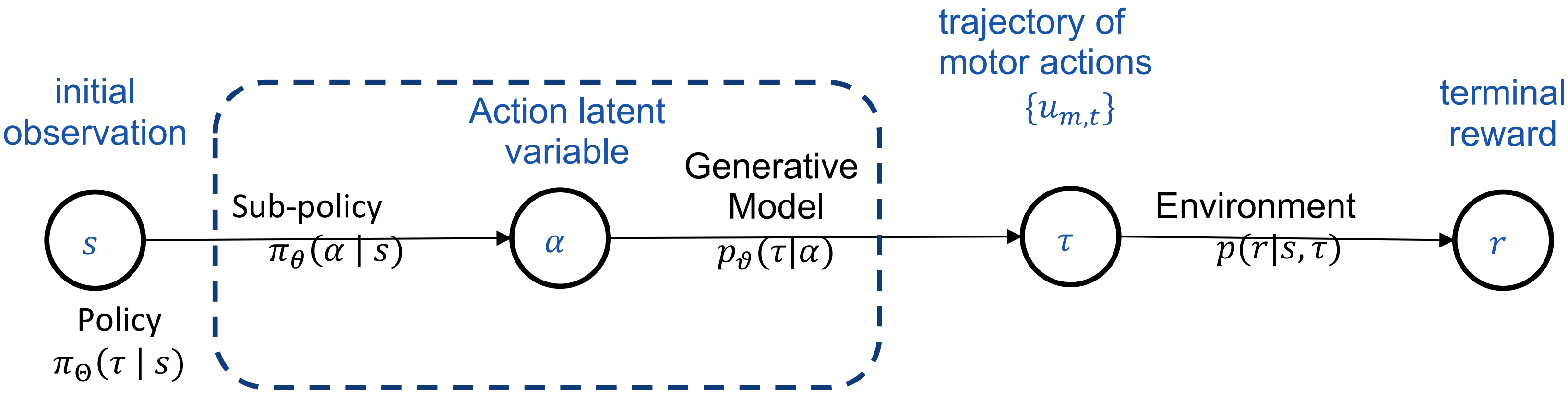
$$\max_{\pi_\theta, q_\psi} \mathbb{E}_{s,a\sim\mathcal{D}}[\omega\, \mathbb{E}_{q_\psi(z|s,a)}[\log(\pi_\theta(a|s,z)) - \beta\, \mathrm{D_{KL}}(q_\psi(z|s,a)\,||\,p(z))]$$

RL with Generative Models

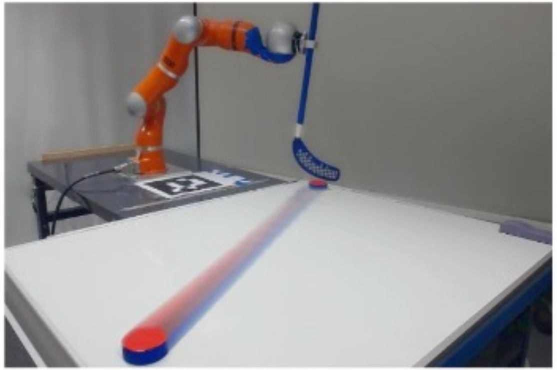


$$\log p(r|s,\Theta) = \log \int p(r|s,\tau)\pi_\Theta(\tau|s)d\tau$$

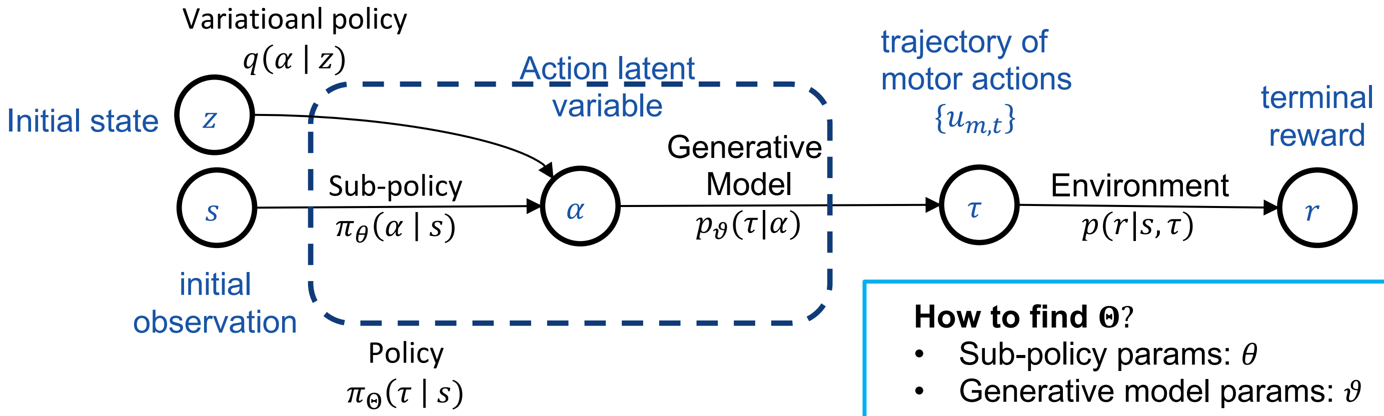
RL with Generative Models



$$\log p(r|s,\Theta) = \log \int p(r|s, p_\vartheta(\tau|\alpha))\pi_\theta(\alpha|s)d\alpha$$



RL with Generative Models

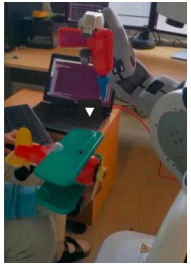


How to find Θ?

- Sub-policy params: θ
- Generative model params: ϑ

$q(\alpha|z)$

$\pi_\theta(\alpha|s)$

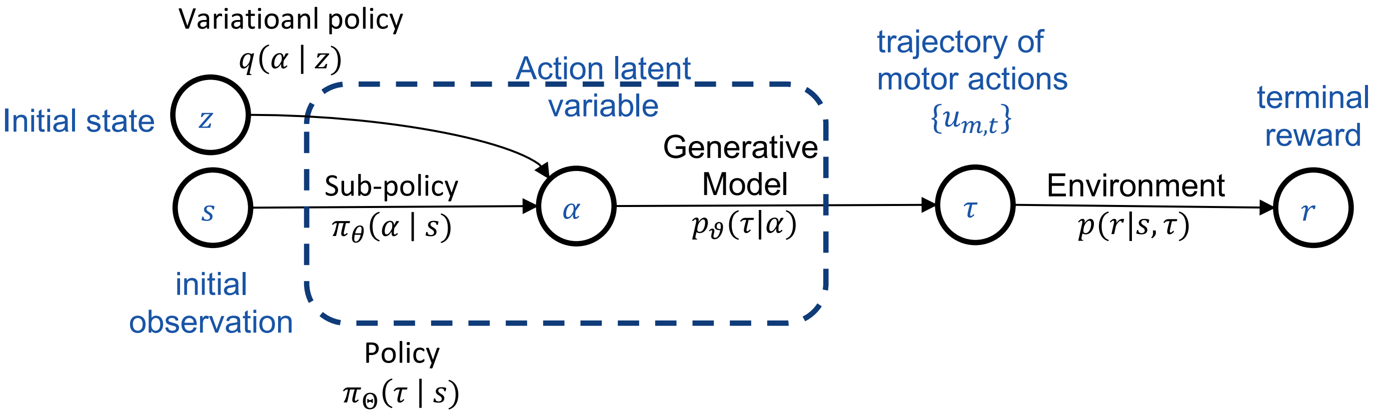


Input remapping trick

$$\log p(r|s,\theta) = \log p(r|s,\theta) \int \frac{\log q(\alpha|z)}{\log q(\alpha|z)} q(\alpha|z) d\alpha$$

Ghadirzadeh et al.,
Data-efficient visuomotor policy training
using reinforcement learning and generative models.

RL with Generative Models



1 Lower bound

2 KL divergence (non-negative)

$$\log p(r|s,\theta) = \boxed{\int q(\alpha|z) \log \frac{p(r,\alpha|s,\theta)}{q(\alpha|z)} d\alpha} + \boxed{\int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,s,\theta)} d\alpha}$$

Maximize reward log-likelihood iteratively in two steps:

1. Maximize the lower-bound by minimizing the KL divergence term (update $q(\alpha|z)$)
2. Maximize the lower bound directly (update θ)

RL with Generative Models

1 Lower bound 2 KL divergence (non-negative)

$$\log p(r|s,\theta) = \int q(\alpha|z) \log \frac{p(r,\alpha|s,\theta)}{q(\alpha|z)} d\alpha + \int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,s,\theta)} d\alpha$$

- Expectation step
  - Minimize the KL-divergence term by optimizing the variational policy $q(\alpha|z)$

$$q = \underset{q'}{\operatorname{argmin}} \int q'(\alpha|z) \log \frac{q'(\alpha|z)}{\pi_\theta(\alpha|s)} d\alpha - \int q'(\alpha|z) \log p(r|\alpha,s) d\alpha + \log p(r|s,\theta) \int q'(\alpha|z) d\alpha$$

$$q = \boxed{\underset{q'}{\operatorname{argmin}} D_{KL}(\, q'(\alpha|z) \;||\; \pi_\theta(\alpha|s) \,)} - \boxed{\mathbb{E}_{q'(\alpha|z)}[\log p(r|\alpha,s)]}$$

Trust region Reward seeking

RL with Generative Models

① Lower bound ② KL divergence (non-negative)

$$\log p(r|s,\theta) = \int q(\alpha|z) \log \frac{p(r,\alpha|s,\theta)}{q(\alpha|z)} d\alpha + \int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,s,\theta)} d\alpha$$

- Maximization step
  - Maximize the lower bound directly by updating the policy parameters θ

$$\theta = \underset{\theta'}{\operatorname{argmax}} \int q(\alpha|z) \log \frac{p(r,\alpha|s,\theta')}{q(\alpha|z)} d\alpha$$

$$= \underset{\theta'}{\operatorname{argmax}} \int q(\alpha|z) \log \frac{p(r|\alpha,s)\pi_{\theta'}(\alpha|s)}{q(\alpha|z)} d\alpha$$

$$= \underset{\theta'}{\operatorname{argmax}} \int q(\alpha|z) \log \frac{\pi_{\theta'}(\alpha|s)}{q(\alpha|z)} d\alpha + \int q(\alpha|z) \log p(r|\alpha,s) d\alpha$$

$$\theta = \boxed{\underset{\theta'}{\operatorname{argmin}} D_{KL}(\ q(\alpha|z) \ || \ \pi_{\theta'}(\alpha|s)\ )}$$

Supervised Learning

RL with Generative Models

1 Lower bound   2 KL divergence (non-negative)

$$\log p(r|s,\theta) = \int q(\alpha|z) \log \frac{p(r,\alpha|s,\theta)}{q(\alpha|z)} d\alpha + \int q(\alpha|z) \log \frac{q(\alpha|z)}{p(\alpha|r,s,\theta)} d\alpha$$

- Expectation step

$$q = \underbrace{\boxed{\operatorname*{argmin}_{q'} D_{KL}(\ q'(\alpha|z)\ ||\ \pi_\theta(\alpha|s)\ )}}_{\text{Trust region}} - \underbrace{\boxed{\mathbb{E}_{q'(\alpha|z)}[\log p(r|\alpha,s)]}}_{\text{Reward seeking}}$$

- Maximization step

$$\theta = \boxed{\operatorname*{argmin}_{\theta'} D_{KL}(\ q(\alpha|z)\ ||\ \pi_{\theta'}(\alpha|s)\ )}$$

Supervised Learning

RL with Generative Models