# Maximum-a-Posteriori (MAP) Policy Optimization

Mayank Mittal

# MAP Policy Optimization

**Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, Martin Riedmiller (2018)**

# V-MPO: On-Policy MAP Policy Optimization For Discrete and Continuous Control

**H. Francis Song* , Abbas Abdolmaleki* , Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin Riedmiller, Matthew M. Botvinick (2019)**

# Duality: Control and Estimation

- What are the actions which maximize future rewards?

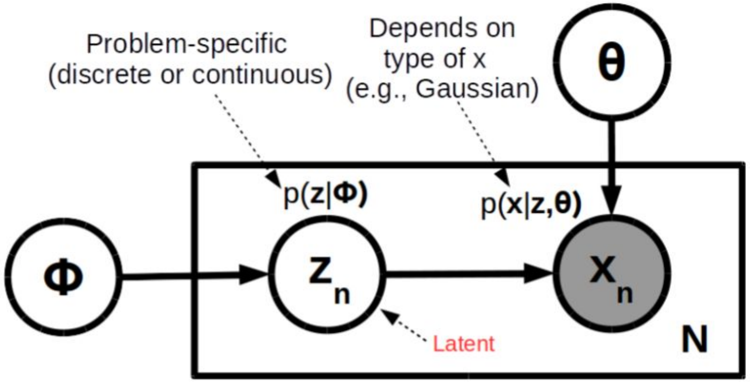
↕

- Assuming future success in maximizing rewards, what are the actions most likely to have been taken?

Solved using Expectation Maximization (EM)

# Expectation Maximization

- Consider a latent variable model:



*"global"* parameters

$$\Theta = (\theta, \phi)$$

- Generally, point estimation via MLE/MAP is not possible due to intractability

$$\Theta_{MLE} = \arg\max_{\Theta} \log p(\mathbf{X}|\Theta) = \arg\max_{\Theta} \log \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\Theta)$$

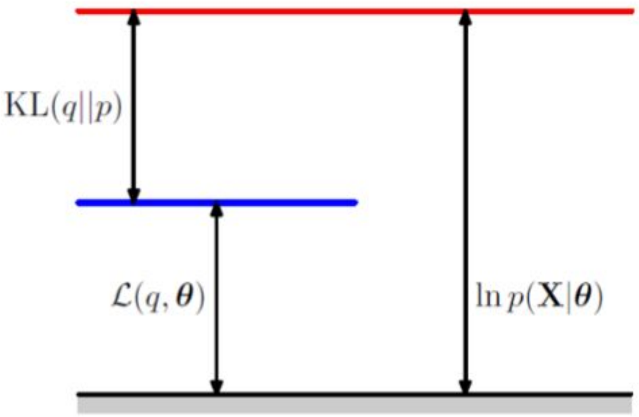
Slide from course: **Topics in Probabilistic Modeling and Inferences, Piyush Rai (Fall Semester, 2019), IIT Kanpur**

# Expectation Maximization

- Define $p_z = p(\mathbf{Z}|\mathbf{X}, \Theta)$ and let $q(\mathbf{Z})$ be some distribution over $\mathbf{Z}$
- Assume discrete $\mathbf{Z}$, the identity below holds for any choice of the distribution $q(\mathbf{Z})$

$$\boxed{\log p(\mathbf{X}|\Theta) = \mathcal{L}(q, \Theta) + \text{KL}(q||p_z)}$$

$$\begin{aligned} \mathcal{L}(q, \Theta) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\Theta)}{q(\mathbf{Z})} \right\} \\ \text{KL}(q||p_z) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{Z}|\mathbf{X}, \Theta)}{q(\mathbf{Z})} \right\} \end{aligned}$$



$\text{KL}(q||p)$
$\mathcal{L}(q, \boldsymbol{\theta})$
$\ln p(\mathbf{X}|\boldsymbol{\theta})$

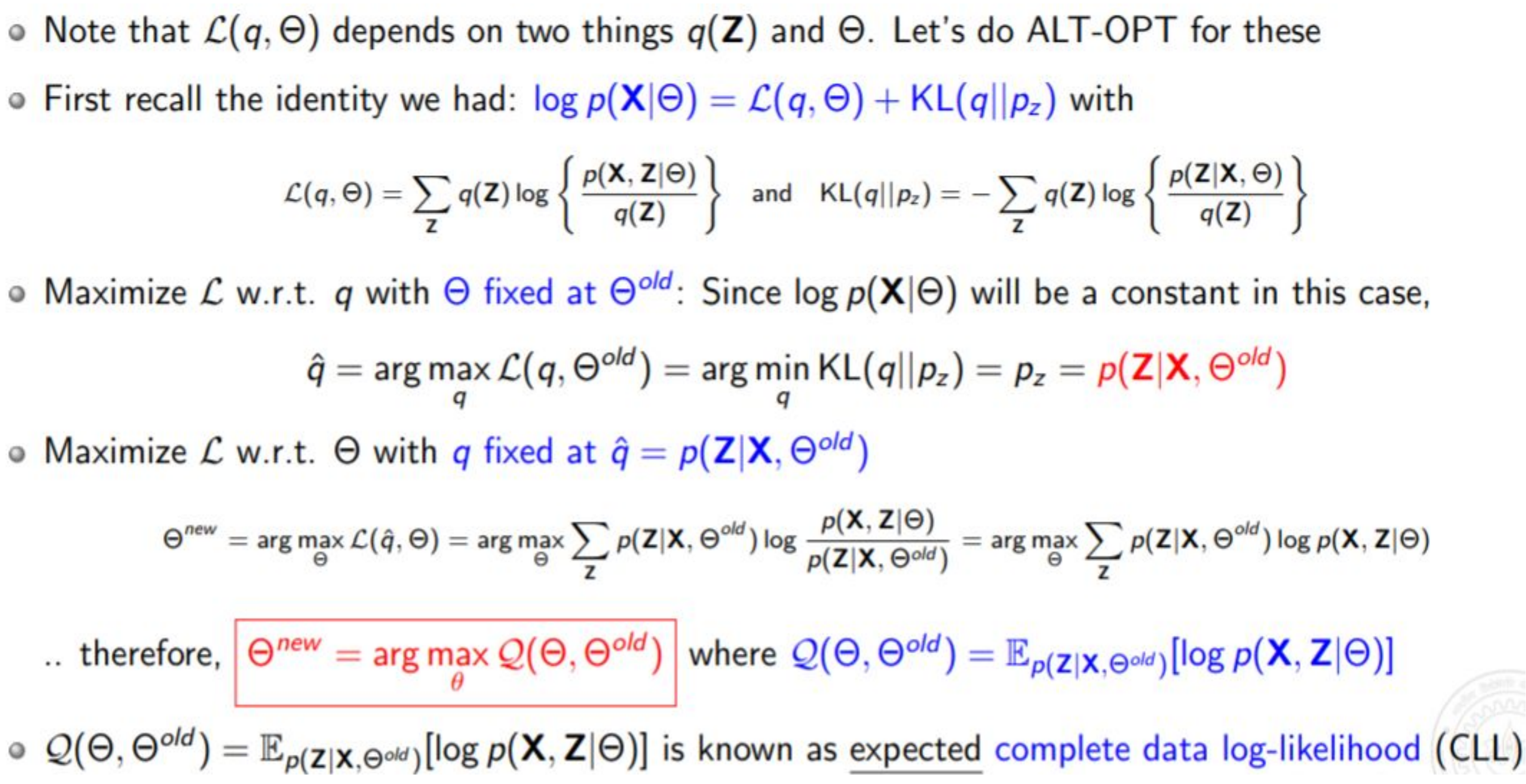
- Since $\text{KL}(q||p_z) \geq 0$, $\mathcal{L}(q, \Theta)$ is a lower-bound on $\log p(\mathbf{X}|\Theta)$

$$\log p(\mathbf{X}|\Theta) \geq \mathcal{L}(q, \Theta) \longleftarrow \text{ELBO}$$

- Maximizing $\mathcal{L}(q, \Theta)$ will also improve $\log p(\mathbf{X}|\Theta)$

Slide from course: **Topics in Probabilistic Modeling and Inferences, Piyush Rai (Fall Semester, 2019), IIT Kanpur**

# Expectation Maximization



- Note that $\mathcal{L}(q,\Theta)$ depends on two things $q(\mathbf{Z})$ and $\Theta$. Let's do ALT-OPT for these
- First recall the identity we had: $\log p(\mathbf{X}|\Theta) = \mathcal{L}(q,\Theta) + \text{KL}(q||p_z)$ with

$$\mathcal{L}(q,\Theta) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{X},\mathbf{Z}|\Theta)}{q(\mathbf{Z})} \right\} \quad \text{and} \quad \text{KL}(q||p_z) = -\sum_{\mathbf{Z}} q(\mathbf{Z}) \log \left\{ \frac{p(\mathbf{Z}|\mathbf{X},\Theta)}{q(\mathbf{Z})} \right\}$$

- Maximize $\mathcal{L}$ w.r.t. $q$ with $\Theta$ fixed at $\Theta^{old}$: Since $\log p(\mathbf{X}|\Theta)$ will be a constant in this case,

$$\hat{q} = \arg\max_q \mathcal{L}(q,\Theta^{old}) = \arg\min_q \text{KL}(q||p_z) = p_z = p(\mathbf{Z}|\mathbf{X},\Theta^{old})$$

- Maximize $\mathcal{L}$ w.r.t. $\Theta$ with $q$ fixed at $\hat{q} = p(\mathbf{Z}|\mathbf{X},\Theta^{old})$

$$\Theta^{new} = \arg\max_\Theta \mathcal{L}(\hat{q},\Theta) = \arg\max_\Theta \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\Theta^{old}) \log \frac{p(\mathbf{X},\mathbf{Z}|\Theta)}{p(\mathbf{Z}|\mathbf{X},\Theta^{old})} = \arg\max_\Theta \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X},\Theta^{old}) \log p(\mathbf{X},\mathbf{Z}|\Theta)$$

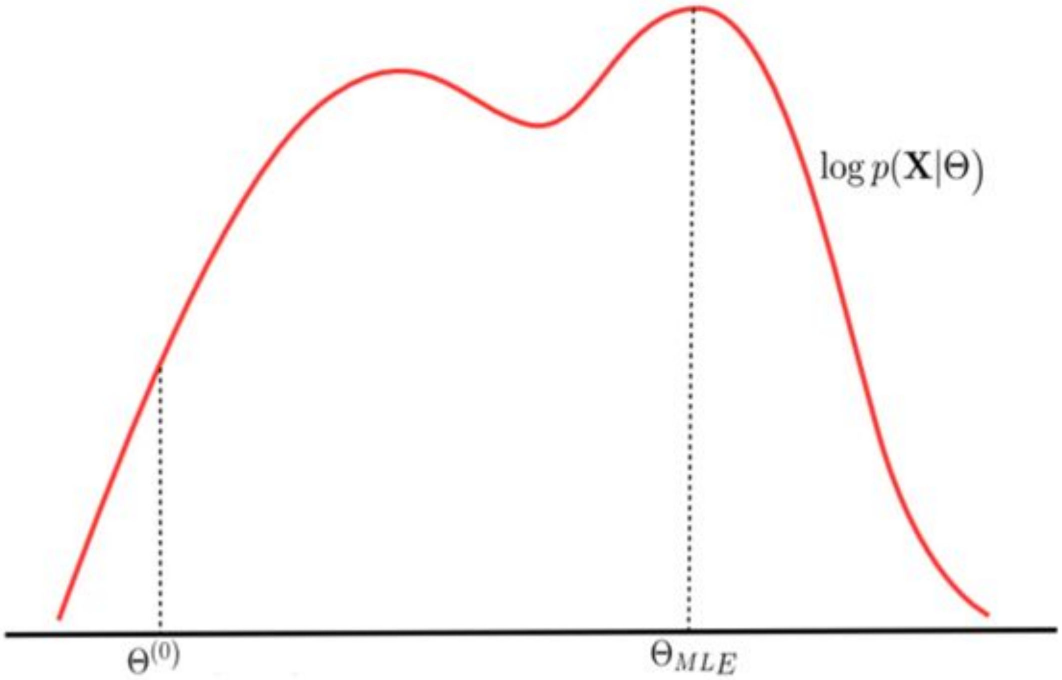
.. therefore, $\boxed{\Theta^{new} = \arg\max_\theta \mathcal{Q}(\Theta,\Theta^{old})}$ where $\mathcal{Q}(\Theta,\Theta^{old}) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\Theta^{old})}[\log p(\mathbf{X},\mathbf{Z}|\Theta)]$

- $\mathcal{Q}(\Theta,\Theta^{old}) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\Theta^{old})}[\log p(\mathbf{X},\mathbf{Z}|\Theta)]$ is known as expected complete data log-likelihood (CLL)

Slide from course: **Topics in Probabilistic Modeling and Inferences, Piyush Rai (Fall Semester, 2019), IIT Kanpur**

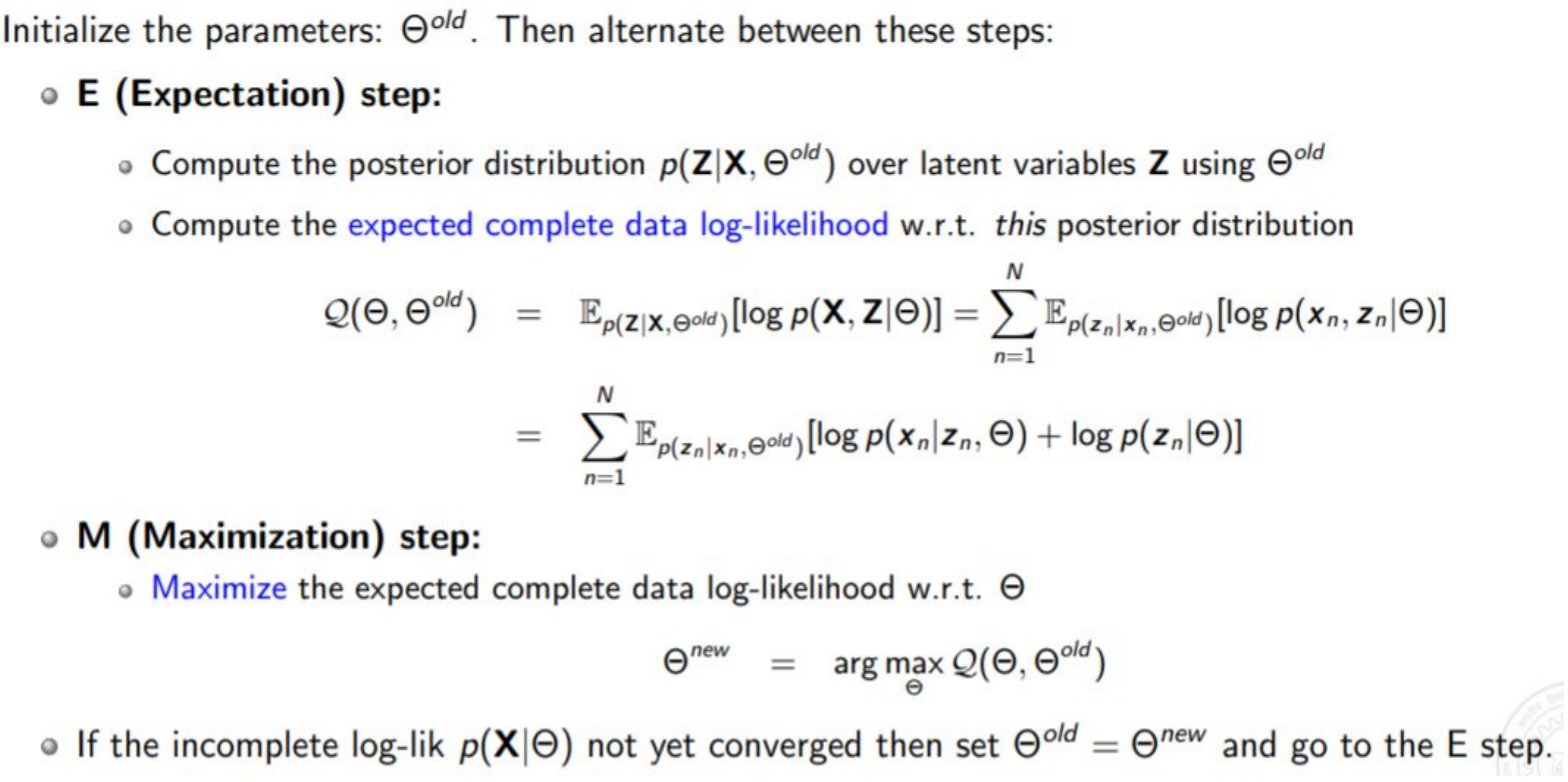
# Expectation Maximization

- Step 1: We set $\hat{q} = p(\mathbf{Z}|\mathbf{X}, \Theta^{old})$, $\mathcal{L}(\hat{q}, \Theta)$ touches $\log p(\mathbf{X}|\Theta)$ at $\Theta^{old}$
- Step 2: We maximize $\mathcal{L}(\hat{q}, \Theta)$ w.r.t. $\Theta$ (equivalent to maximizing $\mathcal{Q}(\Theta, \Theta^{old})$)



Slide from course: **Topics in Probabilistic Modeling and Inferences, Piyush Rai (Fall Semester, 2019), IIT Kanpur**

# Expectation Maximization



Initialize the parameters: $\Theta^{old}$. Then alternate between these steps:

- **E (Expectation) step:**
  - Compute the posterior distribution $p(\mathbf{Z}|\mathbf{X},\Theta^{old})$ over latent variables $\mathbf{Z}$ using $\Theta^{old}$
  - Compute the expected complete data log-likelihood w.r.t. *this* posterior distribution

$$
\begin{aligned}
\mathcal{Q}(\Theta,\Theta^{old}) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\Theta^{old})}[\log p(\mathbf{X},\mathbf{Z}|\Theta)] = \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\Theta^{old})}[\log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\Theta)] \\
&= \sum_{n=1}^{N} \mathbb{E}_{p(\boldsymbol{z}_n|\boldsymbol{x}_n,\Theta^{old})}[\log p(\boldsymbol{x}_n|\boldsymbol{z}_n,\Theta) + \log p(\boldsymbol{z}_n|\Theta)]
\end{aligned}
$$

- **M (Maximization) step:**
  - Maximize the expected complete data log-likelihood w.r.t. $\Theta$

$$
\Theta^{new} = \arg\max_{\Theta} \mathcal{Q}(\Theta,\Theta^{old})
$$

- If the incomplete log-lik $p(\mathbf{X}|\Theta)$ not yet converged then set $\Theta^{old} = \Theta^{new}$ and go to the E step.

Slide from course: **Topics in Probabilistic Modeling and Inferences, Piyush Rai (Fall Semester, 2019), IIT Kanpur**

# Duality: Control and Estimation

- What are the actions which maximize future rewards?

↕

- Assuming future success in maximizing rewards, what are the actions most likely to have been taken?

Solved using Expectation Maximization (EM)

# MAP Policy Optimization

**Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, Martin Riedmiller (2018)**

# V-MPO: On-Policy MAP Policy Optimization For Discrete and Continuous Control

**H. Francis Song* , Abbas Abdolmaleki* , Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin Riedmiller, Matthew M. Botvinick (2019)**

# Inference for Optimal Control

- Given a prior distribution over trajectories

$$p_\pi(\tau) = p(s_0) \prod_{t>0} p(s_{t+1}|s_t, a_t)\pi(a_t|s_t)$$

- Estimate the posterior distribution over trajectories consistent with desired outcome, *O* (such as achieving a goal)

$$p_\pi(\tau|O = 1) \propto p_\pi(\tau)p_\pi(O = 1|\tau)$$

interpreted as event of
succeeding at RL task

# Inference for Optimal Control

**Likelihood Function:**
(for undiscounted case)

$$p(O = 1|\tau) \propto \exp\left(\frac{\sum_t r_t}{\alpha}\right)$$

interpreted as event of succeeding at RL task

temperature

**Likelihood Objective:**

$$\begin{aligned}\pi^* &= \text{argmax}_\pi \ \log p_\pi(O = 1) \\ &= \text{argmax}_\pi \ \log \int_\tau p_\pi(\tau) p(O = 1|\tau) d\tau\end{aligned}$$

# Inference for Optimal Control

**Likelihood Objective:** $\pi^* = \mathrm{argmax}_\pi \; \log p_\pi(O=1)$

$$\begin{aligned}
\log p_\pi(O=1) &= \log \int p_\pi(\tau) p_\pi(O=1|\tau) d\tau \\
&= \log \int q(\tau) \frac{p_\pi(\tau)}{q(\tau)} p(O=1|\tau) d\tau \\
&= \log \mathbb{E}_{\tau\sim q}\left[\frac{p_\pi(\tau)}{q(\tau)} p(O=1|\tau)\right] \\
&\geq \mathbb{E}_{\tau\sim q}\Big[\log p(O=1|\tau)\Big] + \mathbb{E}_{\tau\sim q}\left[\log \frac{p_\pi(\tau)}{q(\tau)}\right] \\
&\geq \underbrace{\mathbb{E}_{\tau\sim q}\Big[\log p(O=1|\tau)\Big] - \mathrm{KL}(q(\tau)||p_\pi(\tau))}_{\mathcal{J}(q,\pi)} \\
&\geq \mathcal{J}(q,\pi)
\end{aligned}$$

Auxiliary Distribution

ELBO

**E-step:**
Improves ELBO w.r.t. *q*

**M-step:**
Improves ELBO w.r.t. policy

# Inference for Optimal Control

**Likelihood Function:**
(for undiscounted case)

$$p(O = 1|\tau) \propto \exp\left(\frac{\sum_t r_t}{\alpha}\right)$$

interpreted as event of succeeding at RL task

temperature

**Likelihood Objective:**

$$\begin{aligned}
\pi^* &= \mathrm{argmax}_\pi \;\; \log p_\pi(O = 1) \\
&= \mathrm{argmax}_\pi \;\; \mathcal{J}(q, \pi) \\
&= \mathrm{argmax}_\pi \;\; \mathbb{E}_{\tau \sim q}\Big[\log p(O = 1|\tau)\Big] - \mathrm{KL}(q(\tau)||p_\pi(\tau)) \\
&= \mathrm{argmax}_\pi \;\; \mathbb{E}_{\tau \sim q}\left[\frac{\sum_t r_t}{\alpha}\right] - \mathrm{KL}(q(\tau)||p_\pi(\tau))
\end{aligned}$$

# Inference for Optimal Control

**Definition of variational distribution** →

$$q(\tau) = p(s_0) \prod_{t>0} p(s_{t+1}|s_t, a_t) q(a_t|s_t)$$

**Likelihood Objective:**

For undiscounted case:

$$\mathcal{J}(q, \pi) = \mathbb{E}_{\tau \sim q}\Big[\sum_t r_t\Big] - \alpha \mathrm{KL}(q(\tau)||p_\pi(\tau))$$

For discounted case:

$$\mathcal{J}(q, \boldsymbol{\theta}) = \mathbb{E}_{\tau \sim q}\Bigg[\sum_{t=0}^{\infty} \gamma^t \Big[r_t - \alpha \mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t, \boldsymbol{\theta})\big)\Big]\Bigg] + \log p(\boldsymbol{\theta})$$

policy parameters

discount factor

prior

# Regularized RL

## Likelihood Objective:

For discounted case:

$$\mathcal{J}(q,\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim q}\left[\sum_{t=0}^{\infty}\gamma^{t}\underbrace{\Big[r_t - \alpha\mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t,\boldsymbol{\theta})\big)\Big]}_{}\right] + \log p(\boldsymbol{\theta})$$

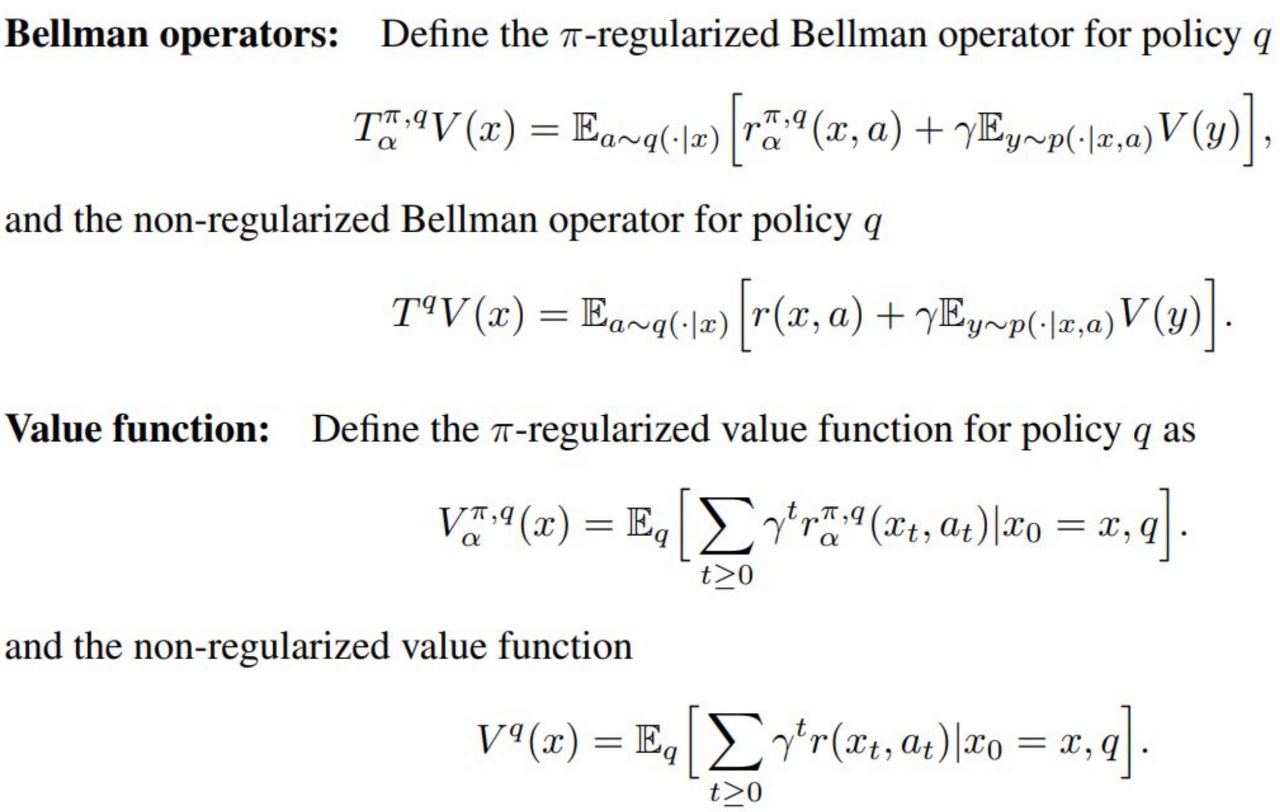
$$\text{$\pi$-regularized reward for policy $q$} \longrightarrow r_{\alpha}^{\pi,q}(x,a) = r(x,a) - \alpha\log\frac{q(a|x)}{\pi(a|x)}$$

## Regularized Q-value function:

$$Q_{\theta}^{q}(s,a) = r_0 + \mathbb{E}_{q(\tau),s_0=s,a_0=a}\left[\sum_{t\geq 1}^{\infty}\gamma^{t}\left[r_t - \alpha\mathrm{KL}(q_t\|\pi_t)\right]\right]$$

# Regularized RL

$\pi$-regularized reward
for policy $q$ $\longrightarrow$ $$r_\alpha^{\pi,q}(x,a) = r(x,a) - \alpha \log \frac{q(a|x)}{\pi(a|x)}$$



**Bellman operators:** Define the $\pi$-regularized Bellman operator for policy $q$

$$T_\alpha^{\pi,q} V(x) = \mathbb{E}_{a\sim q(\cdot|x)}\Big[r_\alpha^{\pi,q}(x,a) + \gamma \mathbb{E}_{y\sim p(\cdot|x,a)} V(y)\Big],$$

and the non-regularized Bellman operator for policy $q$

$$T^q V(x) = \mathbb{E}_{a\sim q(\cdot|x)}\Big[r(x,a) + \gamma \mathbb{E}_{y\sim p(\cdot|x,a)} V(y)\Big].$$

**Value function:** Define the $\pi$-regularized value function for policy $q$ as

$$V_\alpha^{\pi,q}(x) = \mathbb{E}_q\Big[\sum_{t\geq 0} \gamma^t r_\alpha^{\pi,q}(x_t,a_t)|x_0 = x, q\Big].$$

and the non-regularized value function

$$V^q(x) = \mathbb{E}_q\Big[\sum_{t\geq 0} \gamma^t r(x_t,a_t)|x_0 = x, q\Big].$$

**Proposition**

$$V_\alpha^{\pi,q} \leq V^q$$

$$T_\alpha^{\pi,q} V \leq T^q V$$

# Objective of MPO

- Uses EM-style coordinate ascent to maximize estimation objective

$$\mathcal{J}(q,\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim q}\left[\sum_{t=0}^{\infty}\gamma^t\left[r_t - \alpha\mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t,\boldsymbol{\theta}))\big)\right]\right] + \log p(\boldsymbol{\theta})$$

- Proposes off-policy algorithm that is
  - scalable, robust and insensitive to hyperparameters ← on-policy algorithms
  - offers data-efficiency ← off-policy algorithms

# Regularized RL

## Likelihood Objective:

For discounted case:

$$\mathcal{J}(q,\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim q}\left[\sum_{t=0}^{\infty}\gamma^t \underbrace{\left[r_t - \alpha\mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t,\boldsymbol{\theta})\big)\right]}_{}\right] + \log p(\boldsymbol{\theta})$$

$$\begin{array}{c}\pi\text{-regularized reward}\\ \text{for policy } q\end{array} \longrightarrow r_\alpha^{\pi,q}(x,a) = r(x,a) - \alpha\log\frac{q(a|x)}{\pi(a|x)}$$

## Regularized Q-value function:

$$Q_\theta^q(s,a) = r_0 + \mathbb{E}_{q(\tau),s_0=s,a_0=a}\left[\sum_{t\geq 1}^{\infty}\gamma^t\left[r_t - \alpha\mathrm{KL}(q_t\|\pi_t)\right]\right]$$

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

1. Set $q = \pi_{\boldsymbol{\theta}_i}$ $\mathrm{KL}(q||\pi_i) = 0$

2. Estimate unregularized action value:

$$Q^q_{\boldsymbol{\theta}_i}(s,a) = Q_{\boldsymbol{\theta}_i}(s,a) = \mathbb{E}_{\tau_{\pi_i}, s_0=s, a_0=a}\left[\sum_t^{\infty} \gamma^t r_t\right]$$

Using Retrace Algorithm

$$\min_{\phi} L(\phi) = \min_{\phi} \mathbb{E}_{\mu_b(s), b(a|s)}\left[\left(Q_{\theta_i}(s_t, a_t, \phi) - Q_t^{\mathrm{ret}}\right)^2\right]$$

**Off-policy!**

Munos, Remi, et al. ‘Safe and Efficient Off-Policy Reinforcement Learning’. Advances in Neural Information Processing Systems, 2016, pp. 1054–1062. Neural Information Processing Systems,

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

3. Maximize *one-step* objective:

$$
\begin{aligned}
\max_q \; \bar{\mathcal{J}}_s(q, \theta_i) &= \max_q T^{\pi, q} Q_{\boldsymbol{\theta}_i}(s, a) \\
&= \max_q \mathbb{E}_{\underbrace{\mu(s)}_{\substack{\text{Stationary since samples} \\ \text{from replay buffer}}}} \Big[ \mathbb{E}_{q(\cdot|s)} [\underbrace{Q_{\boldsymbol{\theta}_i}(s, a)}_{\text{constant w.r.t } q}] - \alpha \mathrm{KL}(q \| \pi_i) \Big]
\end{aligned}
$$

**INTERPRETATION:**

Policy $q$ chooses soft-optimal action for one step and then resorts to executing policy $\pi$

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

3. Maximize *one-step* objective:

$$
\begin{aligned}
\max_q \; \bar{\mathcal{J}}_s(q, \theta_i) &= \max_q T^{\pi,q} Q_{\boldsymbol{\theta}_i}(s,a) \\
&= \max_q \mathbb{E}_{\mu(s)}\left[\underline{\mathbb{E}_{q(\cdot|s)}[Q_{\boldsymbol{\theta}_i}(s,a)]} - \alpha\underline{\mathrm{KL}(q\|\pi_i)}\right]
\end{aligned}
$$

arbitrary scale!

***(Hard)* Constrained E-step:**

$$
\begin{aligned}
&\max_q \mathbb{E}_{\mu(s)}\left[\mathbb{E}_{q(a|s)}\left[Q_{\theta_i}(s,a)\right]\right] \\
&s.t. \mathbb{E}_{\mu(s)}\left[\mathrm{KL}(q(a|s), \pi(a|s,\boldsymbol{\theta}_i))\right] < \epsilon
\end{aligned}
$$

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

3. Maximize *one-step* objective:

   ***(Hard)* Constrained E-step:**

$$\max_{q} \mathbb{E}_{\mu(s)}\left[\mathbb{E}_{q(a|s)}\left[Q_{\theta_i}(s,a)\right]\right]$$

$$s.t. \mathbb{E}_{\mu(s)}\left[\mathrm{KL}(q(a|s), \pi(a|s,\boldsymbol{\theta}_i))\right] < \epsilon$$

**Method 1**

Use parametric variational distribution

Similar to TRPO/PPO

**Method 2**

Use non-parametric variational distribution

sample based distribution over actions for a state

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

3. Maximize *one-step* objective:

***(Hard)* Constrained E-step:**

$$\max_{q} \mathbb{E}_{\mu(s)}\Big[\mathbb{E}_{q(a|s)}\Big[Q_{\theta_i}(s,a)\Big]\Big]$$

$$s.t. \mathbb{E}_{\mu(s)}\Big[\mathrm{KL}(q(a|s), \pi(a|s, \boldsymbol{\theta}_i))\Big] < \epsilon$$

$$q_i(a|s) \propto \pi(a|s, \boldsymbol{\theta}_i) \exp\left(\frac{Q_{\theta_i}(s,a)}{\eta^*}\right)$$

Lagrangian Formulation

**Method 2**

Use non-parametric variational distribution

sample based distribution over actions for a state

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

1. Set $q = \pi_{\boldsymbol{\theta}_i}$
2. Estimate unregularized action value:

$$Q^q_{\boldsymbol{\theta}_i}(s,a) = Q_{\boldsymbol{\theta}_i}(s,a) = \mathbb{E}_{\tau_{\pi_i}, s_0=s, a_0=a}\left[\sum_t^\infty \gamma^t r_t\right] \longleftarrow$$

Using Retrace Algorithm

3. Maximize "one-step" KL regularized objective to obtain:

$$q_i(a|s) \propto \pi(a|s,\boldsymbol{\theta}_i)\exp\left(\frac{Q_{\theta_i}(s,a)}{\eta^*}\right)$$

# M-step: Maximization w.r.t $\theta$

## Likelihood Objective:

For discounted case:

$$\mathcal{J}(q, \boldsymbol{\theta}) = \mathbb{E}_{\tau \sim q}\left[\sum_{t=0}^{\infty} \gamma^t \Big[r_t - \alpha \mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t, \boldsymbol{\theta}))\big)\Big]\right] + \log p(\boldsymbol{\theta})$$

## M-step: Partial Maximization w.r.t policy

$$\max_{\boldsymbol{\theta}} \mathcal{J}(q_i, \theta) = \max_{\boldsymbol{\theta}} \mathbb{E}_{\mu_q(s)}\Big[\mathbb{E}_{q(a|s)}\Big[\log \pi(a|s, \boldsymbol{\theta})\Big]\Big] + \log p(\boldsymbol{\theta})$$

**Looks similar to supervised learning!**

# M-step: Maximization w.r.t $\theta$

## Likelihood Objective:

For discounted case:

$$\mathcal{J}(q,\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim q}\left[\sum_{t=0}^{\infty}\gamma^{t}\Big[r_t - \alpha\mathrm{KL}\big((q(a_t|s_t)||\pi(a_t|s_t,\boldsymbol{\theta}))\big)\Big]\right] + \log p(\boldsymbol{\theta})$$

## M-step: Partial Maximization w.r.t policy

$$\max_{\boldsymbol{\theta}}\mathcal{J}(q_i,\theta) = \max_{\boldsymbol{\theta}}\mathbb{E}_{\mu_q(s)}\Big[\underbrace{\mathbb{E}_{q(a|s)}\Big[\log\pi(a|s,\boldsymbol{\theta})\Big]}_{\text{samples weighted by variational distribution from E-step}}\Big] + \log p(\boldsymbol{\theta})$$

**Looks similar to supervised learning!**

# M-step: Maximization w.r.t $\theta$

## M-step: Partial Maximization w.r.t policy

$$\max_{\boldsymbol{\theta}} \mathcal{J}(q_i, \theta) = \max_{\boldsymbol{\theta}} \mathbb{E}_{\mu_q(s)}\Big[\underline{\mathbb{E}_{q(a|s)}\Big[\log \pi(a|s, \boldsymbol{\theta})\Big]}\Big] + \underline{\log p(\boldsymbol{\theta})}$$

**Looks similar to supervised learning!**

$$p(\boldsymbol{\theta}) \approx \mathcal{N}\left(\mu = \boldsymbol{\theta}_i, \Sigma = \frac{F_{\boldsymbol{\theta}_i}}{\lambda}\right)$$

For generalized case:

$$\max_{\pi} \mathbb{E}_{\mu_q(s)}\left[\mathbb{E}_{q(a|s)}\Big[\log \pi(a|s, \boldsymbol{\theta})\Big] - \lambda \mathrm{KL}\Big(\pi(a|s, \boldsymbol{\theta}_i), \pi(a|s, \boldsymbol{\theta})\Big)\right]$$

# M-step: Maximization w.r.t $\theta$

## M-step: Partial Maximization w.r.t policy

For generalized case:

$$\max_{\pi} \mathbb{E}_{\mu_q(s)} \Big[ \mathbb{E}_{q(a|s)} \Big[ \log \pi(a|s, \boldsymbol{\theta}) \Big] - \lambda \mathrm{KL}\Big( \pi(a|s, \boldsymbol{\theta}_i), \pi(a|s, \boldsymbol{\theta}) \Big) \Big]$$

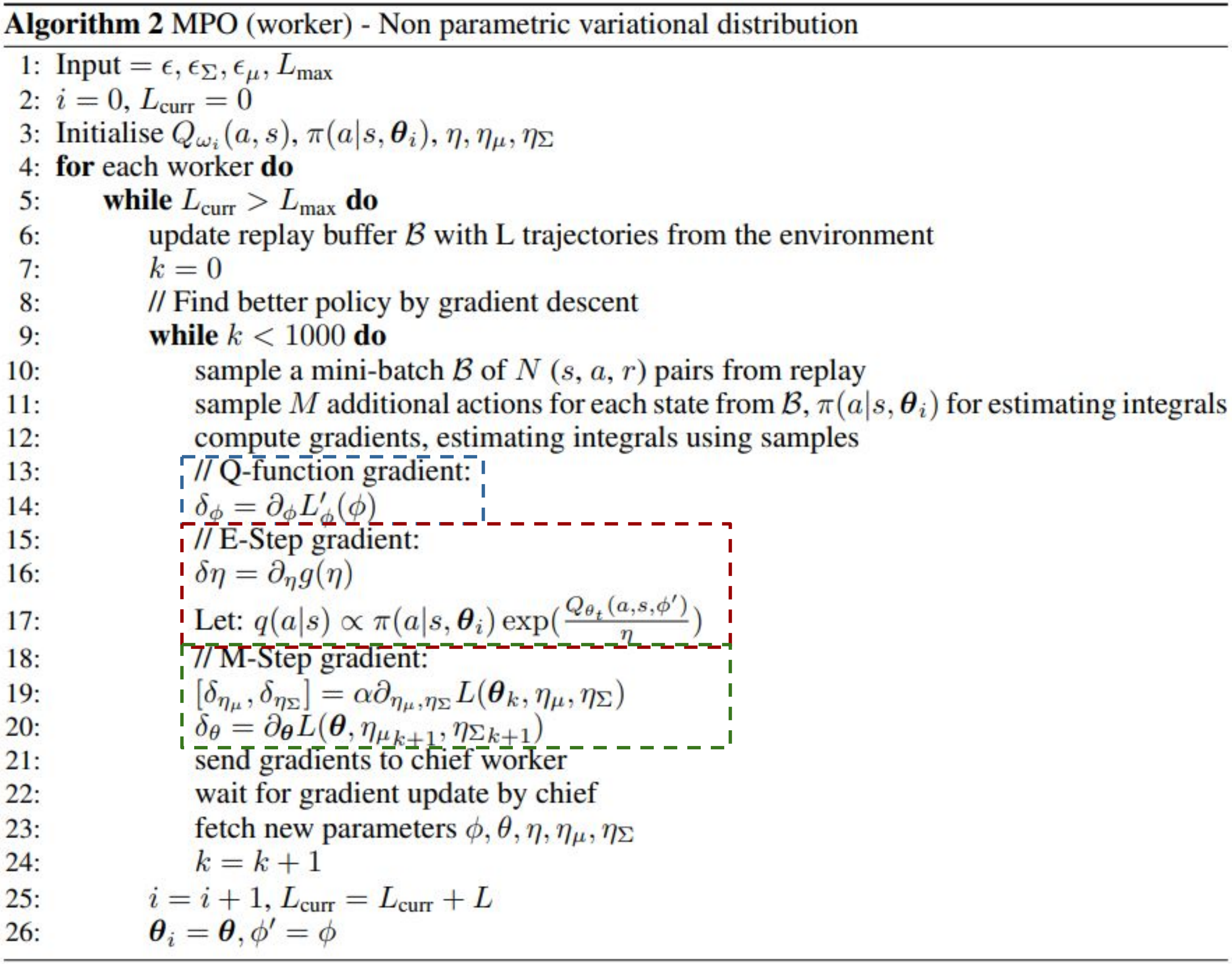
## *(Hard)* Constrained M-step:

$$\max_{\pi} \mathbb{E}_{\mu_q(s)} \Big[ \mathbb{E}_{q(a|s)} \Big[ \log \pi(a|s, \boldsymbol{\theta}) \Big] \Big]$$

$$s.t.\ \underline{\mathbb{E}_{\mu_q(s)} \Big[ \mathrm{KL}(\pi(a|s, \boldsymbol{\theta}_i), \pi(a|s, \boldsymbol{\theta})) \Big] < \epsilon.}$$

prevents overfitting on the samples since the constraint decreases tendency of the entropy of policy to collapse

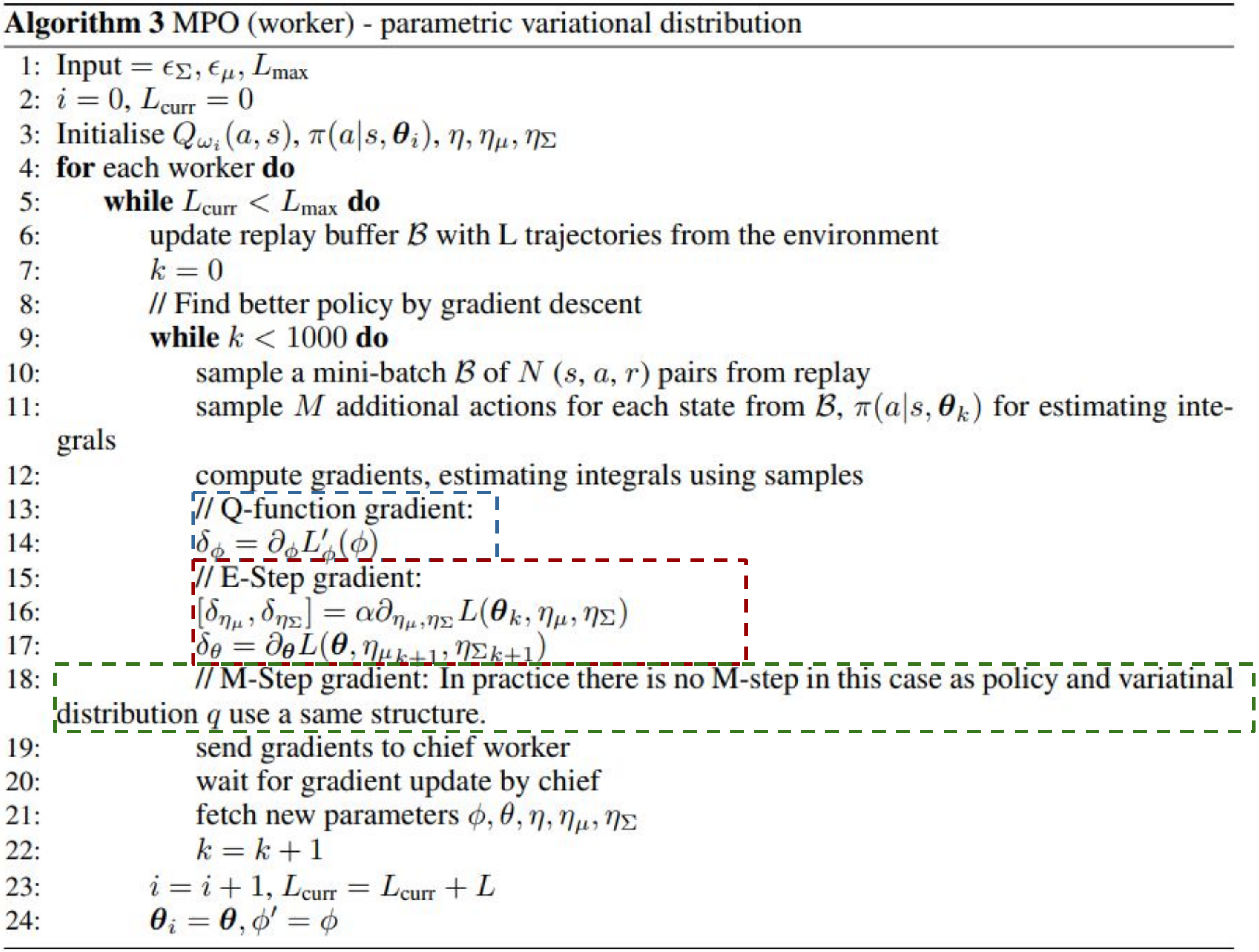
# Algorithm



**Algorithm 2** MPO (worker) - Non parametric variational distribution

```
Input = $\epsilon, \epsilon_\Sigma, \epsilon_\mu, L_{\text{max}}$
$i = 0$, $L_{\text{curr}} = 0$
Initialise $Q_{\omega_i}(a, s)$, $\pi(a|s, \boldsymbol{\theta}_i)$, $\eta, \eta_\mu, \eta_\Sigma$
for each worker do
    while $L_{\text{curr}} > L_{\text{max}}$ do
        update replay buffer $\mathcal{B}$ with L trajectories from the environment
        $k = 0$
        // Find better policy by gradient descent
        while $k < 1000$ do
            sample a mini-batch $\mathcal{B}$ of $N$ $(s, a, r)$ pairs from replay
            sample $M$ additional actions for each state from $\mathcal{B}$, $\pi(a|s, \boldsymbol{\theta}_i)$ for estimating integrals
            compute gradients, estimating integrals using samples
            // Q-function gradient:
            $\delta_\phi = \partial_\phi L'_\phi(\phi)$
            // E-Step gradient:
            $\delta\eta = \partial_\eta g(\eta)$
            Let: $q(a|s) \propto \pi(a|s, \boldsymbol{\theta}_i) \exp(\frac{Q_{\theta_t}(a,s,\phi')}{\eta})$
            // M-Step gradient:
            $[\delta_{\eta_\mu}, \delta_{\eta_\Sigma}] = \alpha\partial_{\eta_\mu,\eta_\Sigma} L(\boldsymbol{\theta}_k, \eta_\mu, \eta_\Sigma)$
            $\delta_\theta = \partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \eta_{\mu_{k+1}}, \eta_{\Sigma_{k+1}})$
            send gradients to chief worker
            wait for gradient update by chief
            fetch new parameters $\phi, \theta, \eta, \eta_\mu, \eta_\Sigma$
            $k = k + 1$
        $i = i + 1$, $L_{\text{curr}} = L_{\text{curr}} + L$
        $\boldsymbol{\theta}_i = \boldsymbol{\theta}$, $\phi' = \phi$
```

# Algorithm



**Algorithm 3** MPO (worker) - parametric variational distribution

```
Input = $\epsilon_\Sigma, \epsilon_\mu, L_{\text{max}}$
$i = 0$, $L_{\text{curr}} = 0$
Initialise $Q_{\omega_i}(a, s), \pi(a|s, \boldsymbol{\theta}_i), \eta, \eta_\mu, \eta_\Sigma$
for each worker do
    while $L_{\text{curr}} < L_{\text{max}}$ do
        update replay buffer $\mathcal{B}$ with L trajectories from the environment
        $k = 0$
        // Find better policy by gradient descent
        while $k < 1000$ do
            sample a mini-batch $\mathcal{B}$ of $N$ $(s, a, r)$ pairs from replay
            sample $M$ additional actions for each state from $\mathcal{B}$, $\pi(a|s, \boldsymbol{\theta}_k)$ for estimating integrals
            compute gradients, estimating integrals using samples
            // Q-function gradient:
            $\delta_\phi = \partial_\phi L'_\phi(\phi)$
            // E-Step gradient:
            $[\delta_{\eta_\mu}, \delta_{\eta_\Sigma}] = \alpha \partial_{\eta_\mu, \eta_\Sigma} L(\boldsymbol{\theta}_k, \eta_\mu, \eta_\Sigma)$
            $\delta_\theta = \partial_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \eta_{\mu_{k+1}}, \eta_{\Sigma_{k+1}})$
            // M-Step gradient: In practice there is no M-step in this case as policy and variatinal distribution $q$ use a same structure.
            send gradients to chief worker
            wait for gradient update by chief
            fetch new parameters $\phi, \theta, \eta, \eta_\mu, \eta_\Sigma$
            $k = k + 1$
        $i = i + 1$, $L_{\text{curr}} = L_{\text{curr}} + L$
        $\boldsymbol{\theta}_i = \boldsymbol{\theta}, \phi' = \phi$
```

# Experimental Evaluation

- Gaussian parametrization of policy
- Benchmark on continuous control tasks

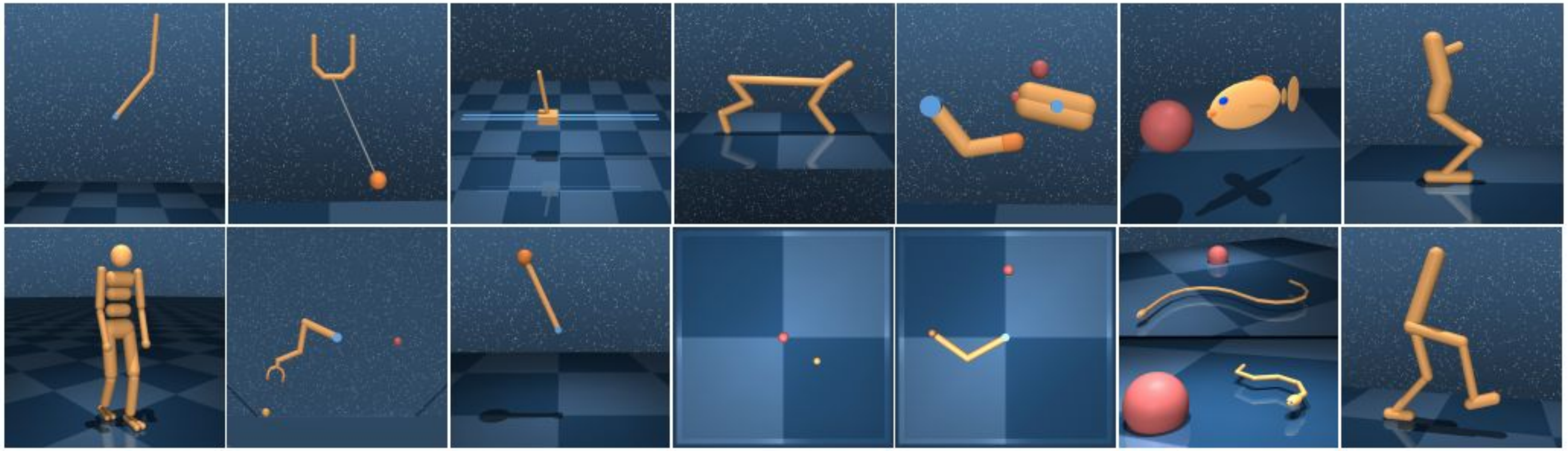
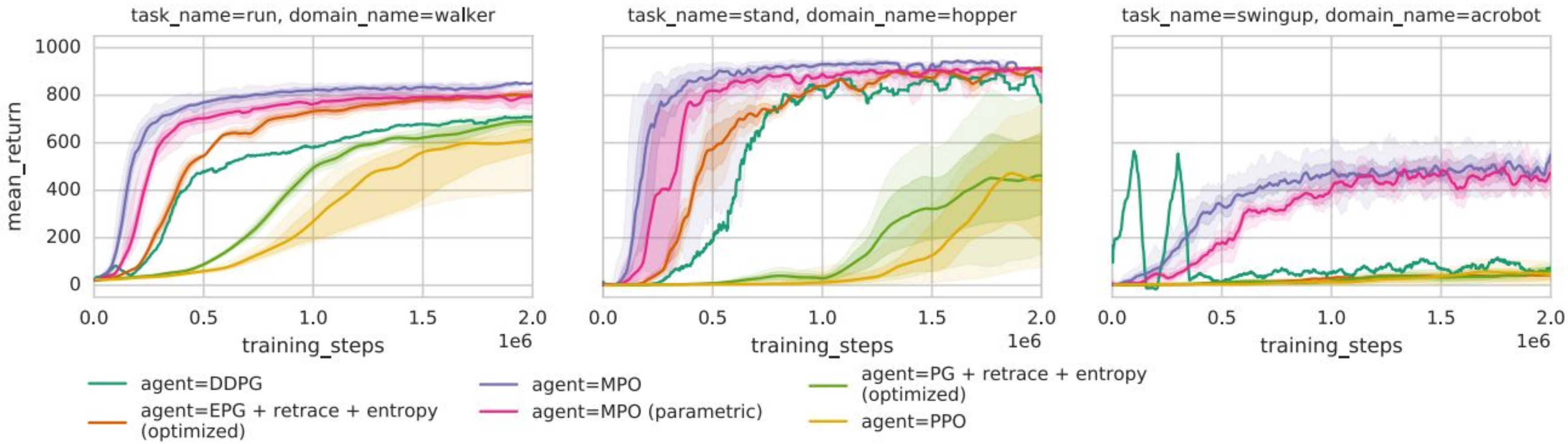


Figure 1: Control Suite domains used for benchmarking. *Top*: Acrobot, Ball-in-cup, Cart-pole, Cheetah, Finger, Fish, Hopper. *Bottom*: Humanoid, Manipulator, Pendulum, Point-mass, Reacher, Swimmers (6 and 15 links), Walker.

# Experimental Evaluation

- Stable learning on all tasks
- Significant sample efficiency

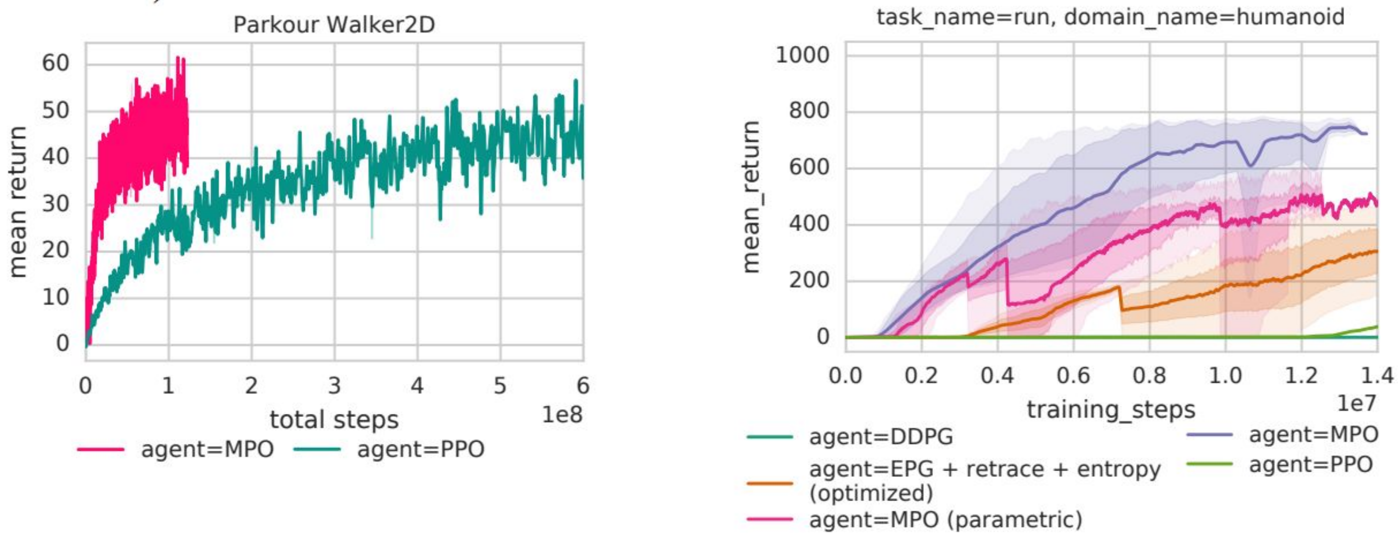
Figure 2: Ablation study of the MPO algorithm and comparison to common baselines from the literature on three domains from the control suite. We plot the median performance over 10 experiments with different random seeds.



# Experimental Evaluation

- Stable learning on all tasks
- Significant sample efficiency

Figure 3: MPO on high-dimensional control problems (Parkour Walker2D and Humanoid walking from control suite).



## MAP Policy Optimization

**Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, Martin Riedmiller (2018)**

## V-MPO: On-Policy MAP Policy Optimization For Discrete and Continuous Control

**H. Francis Song* , Abbas Abdolmaleki* , Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin Riedmiller, Matthew M. Botvinick (2019)**

# Objective of V-MPO

- Uses EM-style coordinate ascent to maximize estimation objective

$$\mathcal{L}_{\text{V-MPO}}(\theta, \eta, \alpha) = \mathcal{L}_{\pi}(\theta) + \mathcal{L}_{\eta}(\eta) + \mathcal{L}_{\alpha}(\theta, \alpha)$$

- Proposes on-policy algorithm
  - replaces state-action value function in MPO with state value function
  - scalable to multi-task setting without population-based tuning of hyperparameters

# Objective of V-MPO

- Uses EM-style coordinate ascent to maximize estimation objective

$$\mathcal{L}_{\text{V-MPO}}(\theta, \eta, \alpha) = \mathcal{L}_{\pi}(\theta) + \mathcal{L}_{\eta}(\eta) + \mathcal{L}_{\alpha}(\theta, \alpha)$$

- Proposes on-policy algorithm
  - replaces state-action value function in MPO with state value function
  - scalable to multi-task setting without population-based tuning of hyperparameters

# Inference for Optimal Control

- In MPO:

$$p(O = 1|\tau) \propto \exp\left(\frac{\sum_t r_t}{\alpha}\right)$$

interpreted as event of
succeeding at RL task

temperature

- In V-MPO:

$$p_{\boldsymbol{\theta}}(\mathcal{I} = 1|s, a) \propto \exp\left(\frac{A^{\pi_{\boldsymbol{\theta}}}(s, a)}{\eta}\right)$$

interpreted as relative
improvement in policy
over previous policy

temperature

# Inference for Control

**MAP Objective:** $\theta^* = \arg\max_\theta \Big[ \log p_\theta(\mathcal{I} = 1) + \log p(\theta) \Big]$

Identity: $\log p(X) = \mathbb{E}_{\psi(Z)}\Big[\log \frac{p(X,Z)}{\psi(Z)}\Big] + D_{\mathrm{KL}}\big(\psi(Z) \| p(Z|X)\big)$

$$\log p_\theta(\mathcal{I} = 1) = \sum_{s,a} \psi(s,a) \log \frac{p_\theta(\mathcal{I} = 1, s, a)}{\psi(s,a)} + D_{\mathrm{KL}}\big(\psi(s,a) \| p_\theta(s, a | \mathcal{I} = 1)\big)$$

**E-step:**
Improves ELBO w.r.t. $\psi(s,a)$

**M-step:**
Improves ELBO w.r.t. policy

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

1. Set $\psi(s,a) = p_{\theta_{\text{old}}}(s, a|\mathcal{I}=1)$

2. Estimate value function $V_\phi^\pi(s)$ :

$$\mathcal{L}_V(\phi) = \frac{1}{2|\mathcal{D}|}\sum_{s_t\sim\mathcal{D}}\left(V_\phi^\pi(s_t) - G_t^{(n)}\right)^2$$

← Using n-step targets

↑ **On-policy!**

3. Calculate advantages:

$$A^\pi(s_t, a_t) = G_t^{(n)} - V_\phi^\pi(s_t)$$

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

4. Maximize objective:

$$\mathcal{J}(\psi(s,a)) = D_{\mathrm{KL}}\left(\psi(s,a) \| p_{\theta_{\text{old}}}(s,a|\mathcal{I}=1)\right)$$
$$\propto -\sum_{s,a} \psi(s,a) A^{\pi_{\theta_{\text{old}}}}(s,a) + \eta \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)} + \lambda \sum_{s,a} \psi(s,a)$$

***(Hard)* Constrained E-step:**

$$\psi(s,a) = \arg\max_{\psi(s,a)} \sum_{s,a} \psi(s,a) A^{\pi_{\theta_{\text{old}}}}(s,a)$$

$$\text{s.t.} \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)} < \epsilon_\eta \text{ and } \sum_{s,a} \psi(s,a) = 1$$

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

4. Maximize objective:

**(*Hard*) Constrained E-step:**

$$\psi(s,a) = \arg\max_{\psi(s,a)} \sum_{s,a} \psi(s,a) A^{\pi\theta_{\text{old}}}(s,a)$$

$$\text{s.t.} \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)} < \epsilon_\eta \text{ and } \sum_{s,a} \psi(s,a) = 1$$

$$\psi(s,a) = \frac{p_{\theta_{\text{old}}}(s,a) \exp\left(\frac{A^{\pi\theta_{\text{old}}}(s,a)}{\eta}\right)}{\sum_{s,a} p_{\theta_{\text{old}}}(s,a) \exp\left(\frac{A^{\pi\theta_{\text{old}}}(s,a)}{\eta}\right)}$$

Lagrangian
Formulation

**Method**

Use non-parametric variational distribution

sample based distribution over actions for a state

# E-step: Maximization w.r.t. *q*

Consider iteration *i*:

4. Maximize objective:

***(Hard)* Constrained E-step:**

$$\psi(s,a) = \arg\max_{\psi(s,a)} \sum_{s,a} \psi(s,a) A^{\pi_{\theta_{\text{old}}}}(s,a)$$

$$\text{s.t.} \sum_{s,a} \psi(s,a) \log \frac{\psi(s,a)}{p_{\theta_{\text{old}}}(s,a)} < \epsilon_\eta \text{ and } \sum_{s,a} \psi(s,a) = 1$$

***Engineering:***

learning improves substantially if samples corresponding to the highest 50% of the advantages in each batch are taken

# M-step: Maximization w.r.t $\theta$

## M-step: Partial Maximization w.r.t policy

Here, minimization (due to negative sign):

$$\mathcal{L}(\theta) = \underline{-\sum_{s,a} \psi(s,a) \log \frac{p_\theta(\mathcal{I}=1, s, a)}{\psi(s,a)}} - \log p(\theta)$$

$$\uparrow$$

$$\mathcal{L}_\pi(\theta) = -\sum_{s,a} \psi(s,a) \log \pi_\theta(a|s)$$

**Weighted maximum likelihood policy loss!**

***Assumption:***

During sample-based computation of the loss, any state-action pairs not in the batch of trajectories have zero weight

# M-step: Maximization w.r.t $\theta$

## M-step: Partial Maximization w.r.t policy

For generalized case (minimization, due to negative sign):

$$\min_{\boldsymbol{\theta}} -\sum_{s,a} \psi(s,a) \log \pi_{\boldsymbol{\theta}}(a|s) + \lambda \mathbb{E}_{p(s)} \Big[ \mathrm{D}_{\mathrm{KL}} \big( \pi_{\boldsymbol{\theta}_{\mathrm{old}}}(a|s) || \pi_{\boldsymbol{\theta}}(a|s) \big) \Big]$$

## *(Hard)* Constrained M-step:

$$\theta^* = \arg\min_{\theta} -\sum_{s,a} \psi(s,a) \log \pi_{\theta}(a|s)$$

$$\text{s.t.} \quad \mathop{\mathbb{E}}_{s\sim p(s)} \Big[ D_{\mathrm{KL}} \big( \pi_{\theta_{\mathrm{old}}}(a|s) \| \pi_{\theta}(a|s) \big) \Big] < \epsilon_{\alpha}$$

prevents overfitting on the samples since the constraint decreases tendency of the entropy of policy to collapse

# Experimental Evaluation

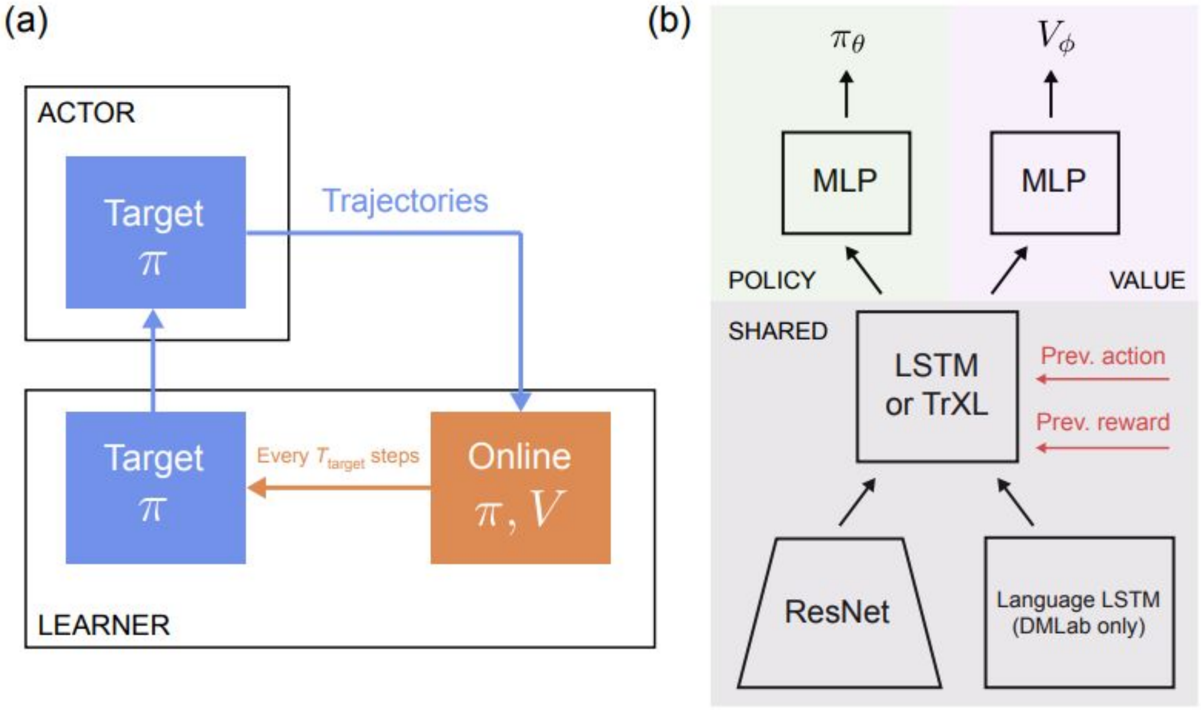
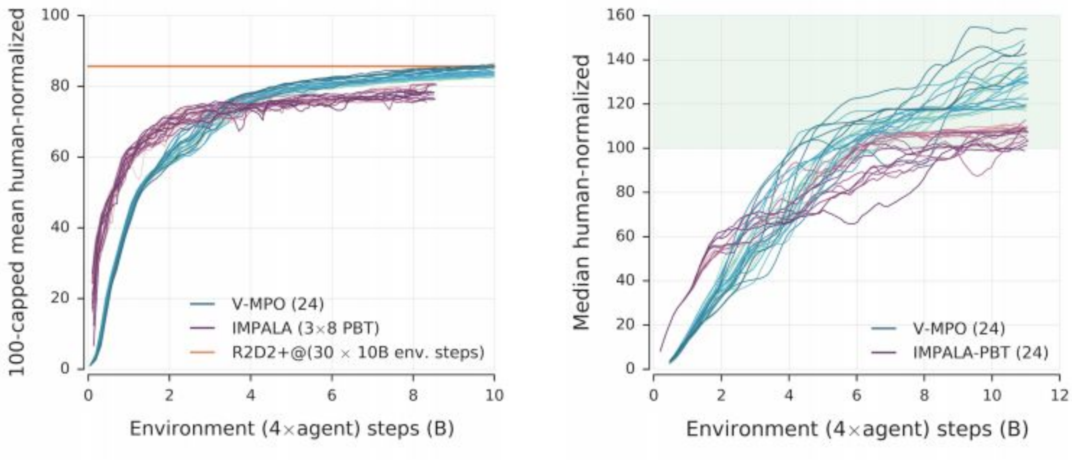


Figure 5: (a) Actor-learner architecture with a target network, which is used to generate agent experience in the environment and is updated every $T_{\text{target}}$ learning steps from the online network. (b) Schematic of the agents, with the policy ($\theta$) and value ($\phi$) networks sharing most of their parameters through a shared input encoder and LSTM [or Transformer-XL (TrXL) for single Atari levels]. The agent also receives the action and reward from the previous step as an input to the LSTM. For DMLab an additional LSTM is used to process simple language instructions.

# Experimental Evaluation

## Multi-task Control: DMLab-30



(a) Multi-task DMLab-30.

(b) Multi-task Atari-57.

Figure 1: (a) Multi-task DMLab-30. IMPALA results show 3 runs of 8 agents each; within a run hyperparameters were evolved via PBT. For V-MPO each line represents a set of hyperparameters that are fixed throughout training. The final result of R2D2+ trained for 10B environment steps on individual levels (Kapturowski et al., 2019) is also shown for comparison (orange line). (b) Multi-task Atari-57. In the IMPALA experiment, hyperparameters were evolved with PBT. For V-MPO each of the 24 lines represents a set of hyperparameters that were fixed throughout training, and all runs achieved a higher score than the best IMPALA run. Data for IMPALA ("Pixel-PopArt-IMPALA" for DMLab-30 and "PopArt-IMPALA" for Atari-57) was obtained from the authors of Hessel et al. (2018). Each environment frame corresponds to 4 agent steps due to the action repeat.

# Experimental Evaluation

## Discrete Control: Atari

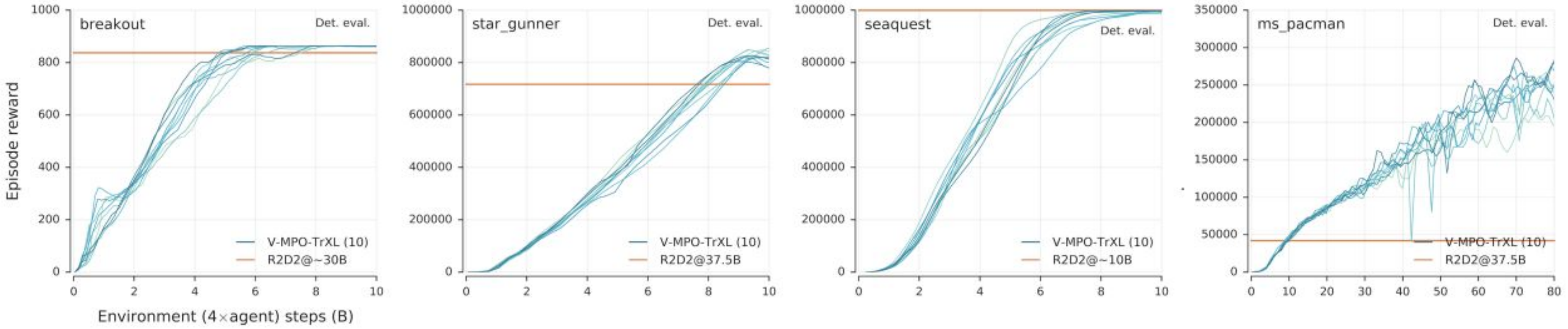
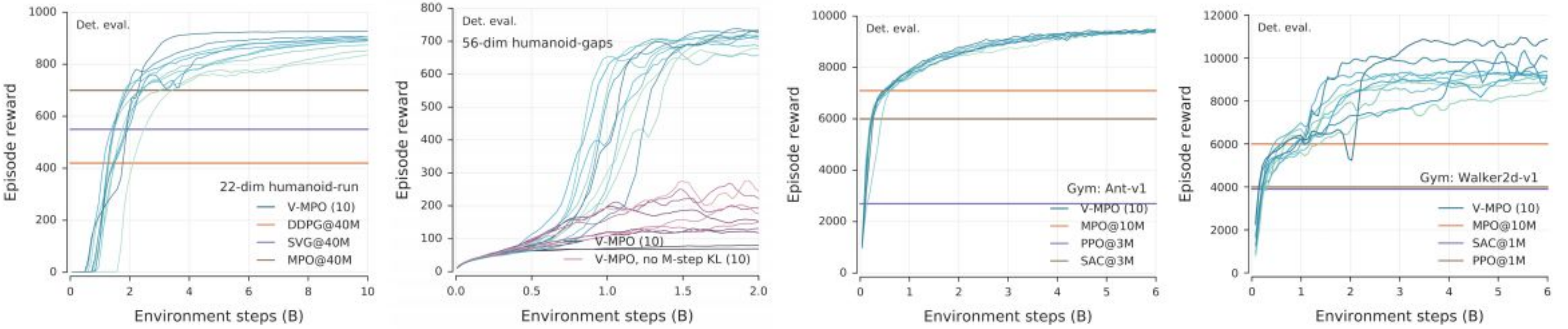


Figure 3: Example levels from Atari. In Breakout, V-MPO achieves the maximum score of 864 in every episode. No reward clipping was applied, and the maximum length of an episode was 30 minutes (108,000 frames). Supplementary video for Ms. Pacman: `https://bit.ly/2lWQBy5`

# Experimental Evaluation

## Continuous Control



(a) (b) (c) (d)

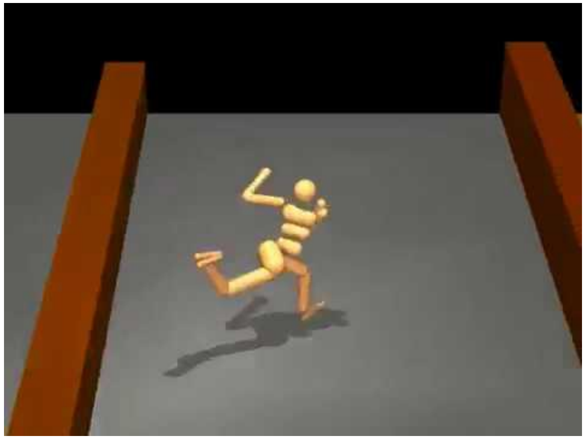
Figure 4: (a) Humanoid "run" from full state (Tassa et al., 2018) and (b) humanoid "gaps" from pixel observations (Merel et al., 2019). Purple curves are the same runs but without parametric KL constraints. Det. eval.: deterministic evaluation. Supplementary video for humanoid gaps: `https://bit.ly/2L9KZdS`. (c)-(d) Example OpenAI Gym tasks.

# Summary

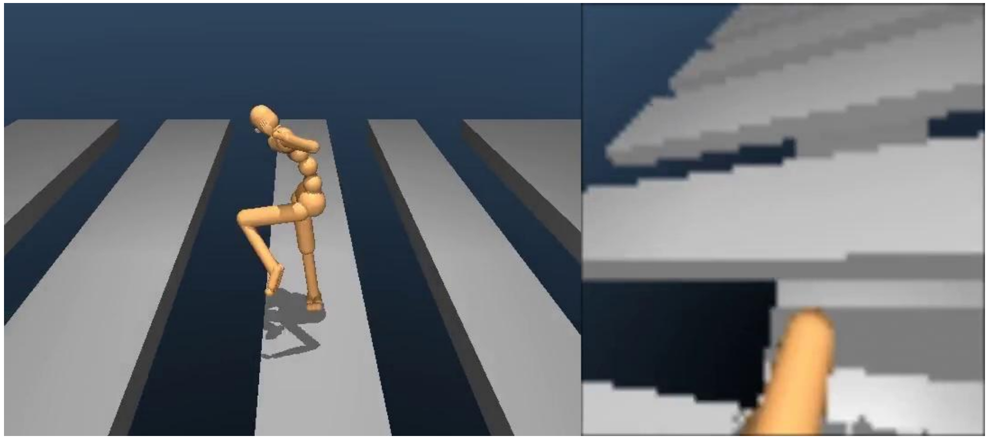
- Formulation of RL optimization problem into an inference problem

- Two particular formulations:
  - **MPO:** off-policy algorithm
  - **V-MPO:** on-policy algorithm

**MPO**



**V-MPO**



# Thank you!

# References

- Abdolmaleki, A., *et al*. (2018). Maximum a Posteriori Policy Optimisation. ArXiv, abs/1806.06920.
- Song, F., Abdolmaleki, A., *et al.* (2019), V-MPO: On-Policy MAP Policy Optimization for Discrete and Continuous Control. ArXiv, abs/1909.12238.