Maximum a Posteriori Policy Optimisation

- Abdolmaleki et al

Theory review by Ashish Kumar

03/25/2024

Theory

RL, MDP, Return and Value Functions

- Markov Decision Process or finite MDP – A mathematical framework used for modeling decision making in a dynamic system governed by a probabilistic dynamics and defines an environment for reinforcement learning.
- It is defined by *S, A and R* which are the set of states, actions and rewards. Additionally, it has a state transition probability P, a reward function r and a discount factor γ, for episodic environments.
- Return or discounted return is defined as the sum of discounted rewards from current time to end of horizon T.

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- State value for a state *s* at time *t* under a policy π is defined as the expected discounted reward if π is followed until horizon

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S}$$

- State, action value for a state *s* and action *a* at time *t* under a policy π is defined as the expected discounted reward if π is followed until horizon

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s, A_t = a\right]$$

Definitions, images and equations from Sutton and Barto 2018

RL Objectives and Inference Problem

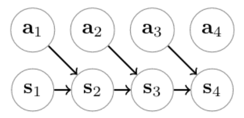
- Q Learning – Learn a q-value function, tabular or approximation, such that the learned q function is same or similar to the true q function under a policy π.

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t) \Big]$$

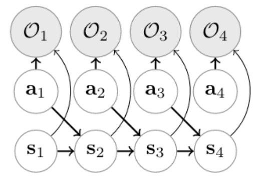
- Policy Gradient methods – Parameterize a policy π(θ) and learn the parameter by maximizing an objective, which in most cases is the true value function under a policy.

$$J(\boldsymbol{\theta}) \doteq v_{\pi_{\boldsymbol{\theta}}}(s_0)$$

- In this settings, the goal for a policy to generate actions that maximizes the future discounted rewards for optimal control.
- A different view is to treat RL for optimal control as an inference problem. That is, given that a policy is optimal at every time step, what are the likely actions generated by the policy at each time step.



(a) graphical model with states and actions



(b) graphical model with optimality variables

$$p(\mathcal{O}_t = 1 | \mathbf{s}_t, \mathbf{a}_t) = \exp(r(\mathbf{s}_t, \mathbf{a}_t)).$$

$$p(\tau | \mathbf{o}_{1:T}) \propto \mathbb{1}[p(\tau) \neq 0] \exp\left(\sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t) \right)$$

Definitions, images and equations from Sutton and Barto 2018, Levine 2018

Expectation Maximization

- Expectation Maximization is a general technique used to find the best parameters for a model that has latent variables to model data.
- EM finds the most likelihood solutions for data.
- Considering a model with observed variables, X, and latent variables, Z. Maximum likelihood solution requires maximizing the likelihood P(X | θ) which is more difficult than

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$$

- If the distribution of the latent variable is q(Z), then

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

where we have defined

$$\mathcal{L}(q, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})} \right\}$$

- EM finds the maximum likelihood solution by maximizing the lower bound L with respect to q in the E-step, keeping θ fixed. In the M step, the L is maximized with respect to θ keeping q fixed. Intuitively, move L closer to true likelihood and find the best θ for L.

Definitions, images and equations from Bishop 2006

Variational Inference

- In case of continuous variables, the likelihood and lower bound change to

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

where we have defined

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- The above can be intractable for some choices of q.
- Restrict the choices for q with an approximation.

Definitions, images and equations from Bishop 2006

Maximum a Posteriori Optimization RL

- This paper defines an optimal control problem in RL setting as an inference problem similar to Levine 2018.
- Likelihood of optimality for a policy π is given by:

Why?

Trajectory distribution under π

$$\log p_\pi(O=1) = \log \int p_\pi(\tau) p(O=1|\tau) d\tau \geq \int q(\tau)\Big[\log p(O=1|\tau) + \log \frac{p_\pi(\tau)}{q(\tau)}\Big] d\tau \tag{1}$$

$$= \mathbb{E}_q\Big[\sum_t r_t/\alpha\Big] - \mathrm{KL}\Big(q(\tau)||p_\pi(\tau)\Big) = \mathcal{J}(q,\pi), \tag{2}$$

- The proposed algorithm convert the traditional RL an inference problem and employs EM on the lower bound or the ELBO J. The E-step improves J with respect to q and the M step optimizes J with respect to the policy parameters keeping the q fixed.
- Why q ? This is used to decompose the log evidence. From Neumann 2011,

$$p(R;\theta) = \int_\tau p(R|\tau) p(\tau;\theta) d\tau, \quad \log p(R;\theta) = \mathcal{L}(q,\theta) + \mathrm{KL}(q||p_R)$$

- Why EM? This is because the state and action are hidden variables in the inference problem and the optimality random variable O is observed.
- Why variational inference? This is due to the use variational distribution q.

Definitions, images and equations from Abdolmaleki 2018, Neumann 2011, Levine et al 2011

MPO – Policy Improvement

- With the lower bound to the optimal posterior defined, EM is used to find the best policy.
- KL Regularized lower bound rewritten as:

$$\mathcal{J}(q,\boldsymbol{\theta}) = \mathbb{E}_q\Big[\sum_{t=0}^{\infty}\gamma^t\big[r_t - \alpha \mathrm{KL}\big(q(a|s_t)\|\pi(a|s_t,\boldsymbol{\theta})\big)\big]\Big] + \underbrace{\log p(\boldsymbol{\theta})}_{\text{Prior over policy parameters}}.$$

- KL Regularized Q-value function defined as:

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

- Consider an augmented reward term r' which is equal to the inner square bracket term. In that case, J is an objective function for policy optimization and Q is a state value function for an MDP with r'.
- Maximizing J with respect to q, brings q close to the current best policy π.

Definitions, images and equations from Abdolmaleki 2018

E - Step

- The ELBO J is a function of q and θ. In the expectation stage, the lower bound J is optimized with respect to the variational distribution q.
- The authors first expand the regularized Q function via regularized bellman operator:

$$\mathbb{E}_{q(a|s)}\Big[r(s,a) - \alpha \mathbf{KL}(q\|\pi_i) + \gamma \mathbb{E}_{p(s'|s,a)}[V_{\boldsymbol{\theta}_i}(s')]\Big].$$

- The E– Step is then performed by optimizing the KL regularized objective.

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

Stationary distribution in replay buffer

- The above formulation allows off-policy learning.
- The above formulation has soft constraints. The authors provide a formulation with a hard constraint as choosing a is non-trivial.

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

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

where we can obtain η^* by minimising the following convex dual function,

$$g(\eta) = \eta\epsilon + \eta\int \mu(s)\log\int \pi(a|s,\boldsymbol{\theta}_i)\exp\left(\frac{Q_{\theta_i}(s,a)}{\eta}\right)dads.$$

- Choice of parametric q or non-parametric q. Parametric q converts into a TRPO/PPO like algorithm. In non-parametric, no need to commit to a parametric distribution. It also helps convert the hard constrained optimization problem into a non-constrained convex optimization problem.

M - Step

- The next step in the EM algorithm is to maximize J with respect to θ given the new q.
- The authors define the maximization problem as

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

- Here p(θ) is defined as

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

where, θ_i are the parameters of the current policy, F is the empirical Fisher Information matrix and λ is a positive scalar.

- The authors derive a generalized formulation of the above maximization problem as

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

which can be re-written as the hard constrained version:

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

- The additional constraint minimizes the risk of overfitting the samples.

Definitions, images and equations from Abdolmaleki 2018

Policy Iteration

- After the policy has been improved, use the improved policy to improve the state-action value function.
- The improved state action function is then used to improve the policy in the next iteration.
- This is called policy iteration as defined in the Sutton and Barto 2018.

$$\pi_0 \xrightarrow{\text{E}} v_{\pi_0} \xrightarrow{\text{I}} \pi_1 \xrightarrow{\text{E}} v_{\pi_1} \xrightarrow{\text{I}} \pi_2 \xrightarrow{\text{E}} \cdots \xrightarrow{\text{I}} \pi_* \xrightarrow{\text{E}} v_*,$$

where $\xrightarrow{\text{E}}$ denotes a policy *evaluation* and $\xrightarrow{\text{I}}$ denotes a policy *improvement*

- The next step is to then perform the policy evaluation, which is to improved the Q function.
- The authors use the retrace algorithm for stability. The improvement is performed as below

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

$$Q_t^{\text{ret}} = Q_{\phi'}(s_t,a_t) + \sum_{j=t}^{\infty} \gamma^{j-t}\Big(\prod_{k=t+1}^{j} c_k\Big)\Big[r(s_j,a_j) + \mathbb{E}_{\pi(a|s_{j+1})}[Q_{\phi'}(s_{j+1},a)] - Q_{\phi'}(s_j,a_j)\Big]$$

$$c_k = \min\Big(1, \frac{\pi(a_k|s_k)}{b(a_k|s_k)}\Big),$$

Experiments

- The authors performed experiments in both continuous action setting as well as discrete action setting.
- For the continuous control tasks, DeepMind Control Suite environments were used and for the discrete control tasks, ATARI's Arcade Learning Environment (ALE) was used.

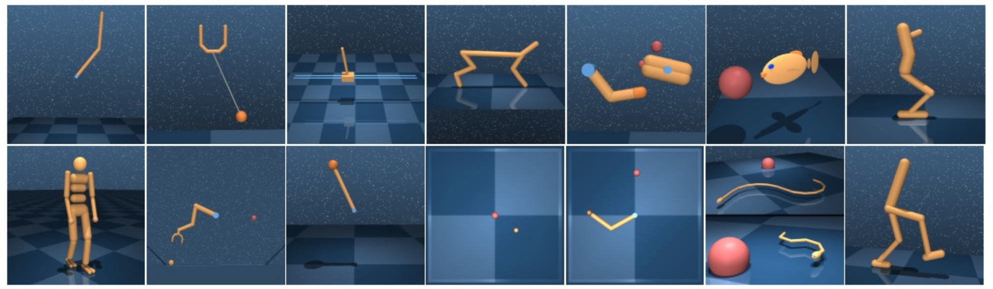


Figure 1: Control Suite domains used for benchmarking. *Top*: Acrobot, Ball-in-cup, Cart-pole, Cheetah, Finger, Fish, Hopper. *Bottom*: Humanoid, Manipulator, Pendulum, Point-mass, Reacher, Swimmers (6 and 15 links), Walker.

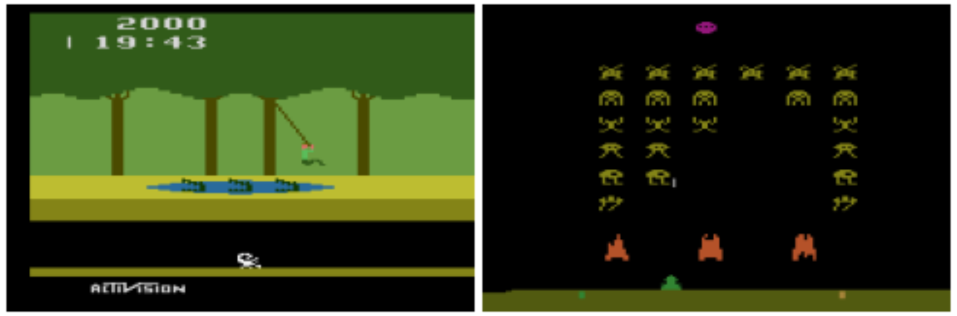
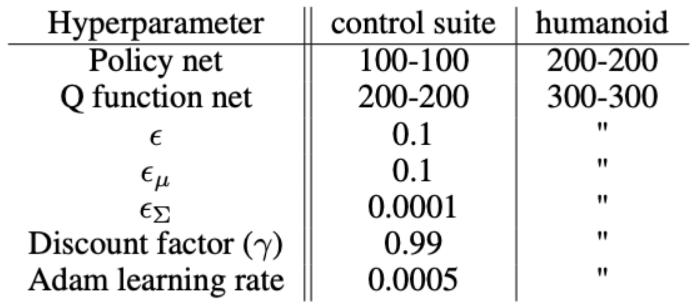


Figure 1: Screenshots of PITFALL! and SPACE INVADERS.

- Pong, Breakout, Q*bert, Tennis and Boxing ATARI ALE environments were used.
- For continuous control, a suite of 18 environments were used.

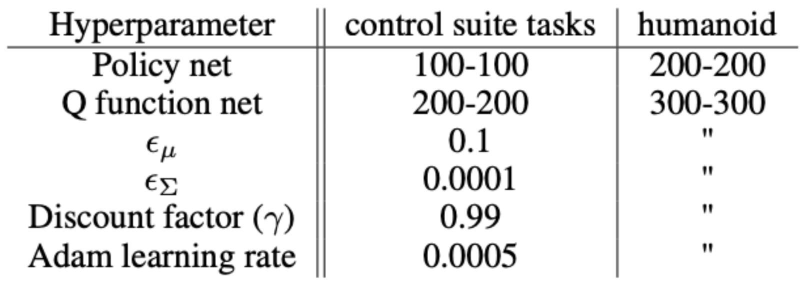
Experiments Details

- For the continuous tasks and the discrete tasks, the same KL contrains and the same learning rates were used.
- For the ATARI tasks, the authors changed the network architecture to match the architecture in the original DQN paper.



Hyperparameter	control suite	humanoid
Policy net	100-100	200-200
Q function net	200-200	300-300
ϵ	0.1	"
ϵ_μ	0.1	"
ϵ_Σ	0.0001	"
Discount factor (γ)	0.99	"
Adam learning rate	0.0005	"

Table 2: Parameters for non-parametric variational distribution



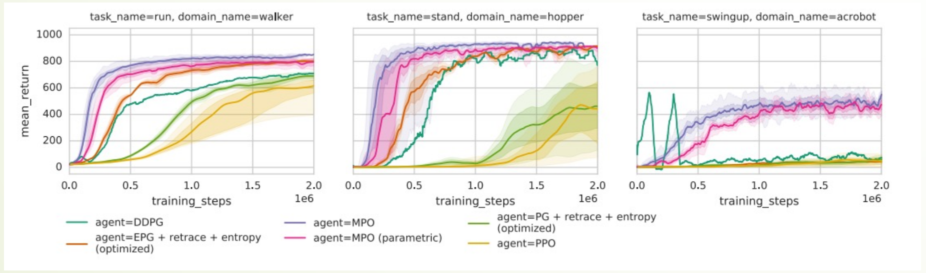
Hyperparameter	control suite tasks	humanoid
Policy net	100-100	200-200
Q function net	200-200	300-300
ϵ_μ	0.1	"
ϵ_Σ	0.0001	"
Discount factor (γ)	0.99	"
Adam learning rate	0.0005	"

Table 3: Parameters for parametric variational distribution

Definitions, images and equations from Abdolmaleki 2018

Results

- The authors performed an ablation study by comparing MPO, PPO and DDPG on classical Acrobot task, the 2D walker and the hopper standing task.
- The study compares the performance of MPO against Proximal Policy Optimization (PPO) and DDPG baselines, highlighting MPO's stable learning and higher sample efficiency due to its off-policy nature.
- Variations in the algorithm, such as changing the variational distribution from non-parametric to parametric, adjusting the KL constraint, and employing different policy gradient methods like EPG + Retrace, are explored for their impact on performance and sample efficiency.
- The findings underscore MPO's effectiveness and the nuanced impacts of algorithmic adjustments on task-specific performance.

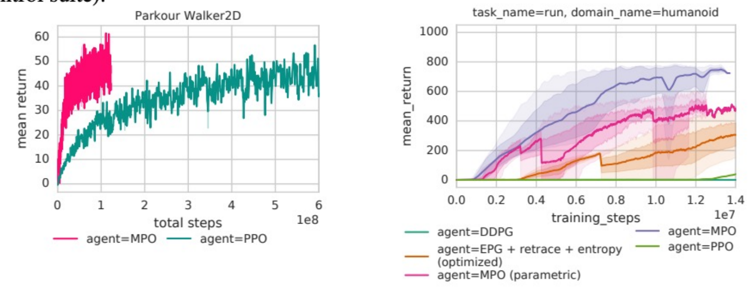


Definitions, images and equations from Abdolmaleki 2018

Results

- Noteworthy is the comparison of MPO and PPO on high dimensional control tasks.

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



- In ATARI environments, C51 performed the best. However, the idea was to show that MPO can be used in discrete setting. Noteworthy, the authors did not fine tune the parameters of the algorithm for ATARI environments.

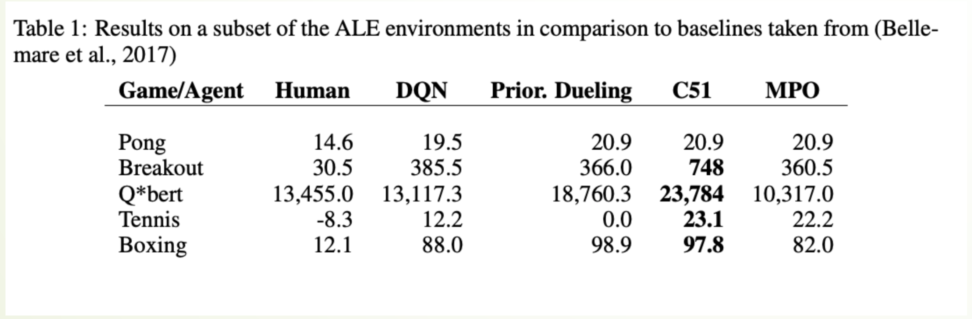


Table 1: Results on a subset of the ALE environments in comparison to baselines taken from (Bellemare et al., 2017)

Game/Agent	Human	DQN	Prior. Dueling	C51	MPO
Pong	14.6	19.5	20.9	20.9	20.9
Breakout	30.5	385.5	366.0	**748**	360.5
Q*bert	13,455.0	13,117.3	18,760.3	**23,784**	10,317.0
Tennis	-8.3	12.2	0.0	**23.1**	22.2
Boxing	12.1	88.0	98.9	**97.8**	82.0

Definitions, images and equations from Abdolmaleki 2018

DERIVATION OF UPDATE RULES

- For the E-Step

From

$$\max_q \int \mu_q(s) \int q(a|s) Q_{\theta_i}(s,a)\,dads$$
$$s.t. \int \mu_q(s)\mathrm{KL}(q(a|s), \pi(a|s,\boldsymbol{\theta}_i))da < \epsilon,$$
$$\iint \mu_q(s) q(a|s)\,dads = 1.$$

To

$$L(q,\eta,\gamma) = \int \mu_q(s) \int q(a|s) Q_{\boldsymbol{\theta}_i}(s,a)\,dads +$$
$$\eta\left(\epsilon - \int \mu_q(s) \int q(a|s) \log \frac{q(a|s)}{\pi(a|s,\boldsymbol{\theta}_i)}\right) + \gamma\left(1 - \iint \mu_q(s) q(a|s)\,dads\right)$$

Next we maximise the Lagrangian L w.r.t the primal variable q. The derivative w.r.t q reads,

$$\partial q L(q,\eta,\gamma) = Q_{\boldsymbol{\theta}_i}(a,s) - \eta \log q(a|s) + \eta \log \pi(a|s,\boldsymbol{\theta}_i) - (\eta - \gamma).$$

Setting it to zero and rearranging terms we get

$$q(a|s) = \pi(a|s,\boldsymbol{\theta}_i) \exp\left(\frac{Q_{\boldsymbol{\theta}_i}(a,s)}{\eta}\right) \exp\left(-\frac{\eta-\gamma}{\eta}\right).$$

However the last exponential term is a normalisation constant for q. Therefore we can write,

$$\exp(-\frac{\eta-\gamma}{\eta}) = \int \pi(a|s,\boldsymbol{\theta}_i) \exp(\frac{Q_{\boldsymbol{\theta}_i}(a,s)}{\eta})da,$$

$$\gamma = \eta - \eta \log\left(\int \pi(a|s,\boldsymbol{\theta}_i) \exp(\frac{Q_{\boldsymbol{\theta}_i}(a,s)}{\eta})da\right).$$

Note that we could write γ based on π and η. At this point we can derive the dual function,

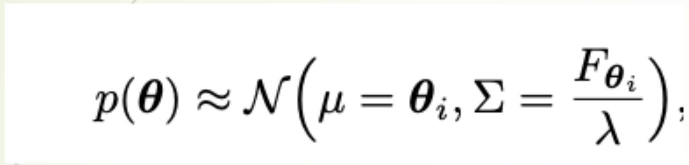
$$g(\eta) = \eta\epsilon + \eta \int \mu_q(s) \log\left(\int \pi(a|s,\boldsymbol{\theta}_i) \exp(\frac{Q(a,s)}{\eta})da\right).$$

Definitions, images and equations from Abdolmaleki 2018

DERIVATION OF UPDATE RULES

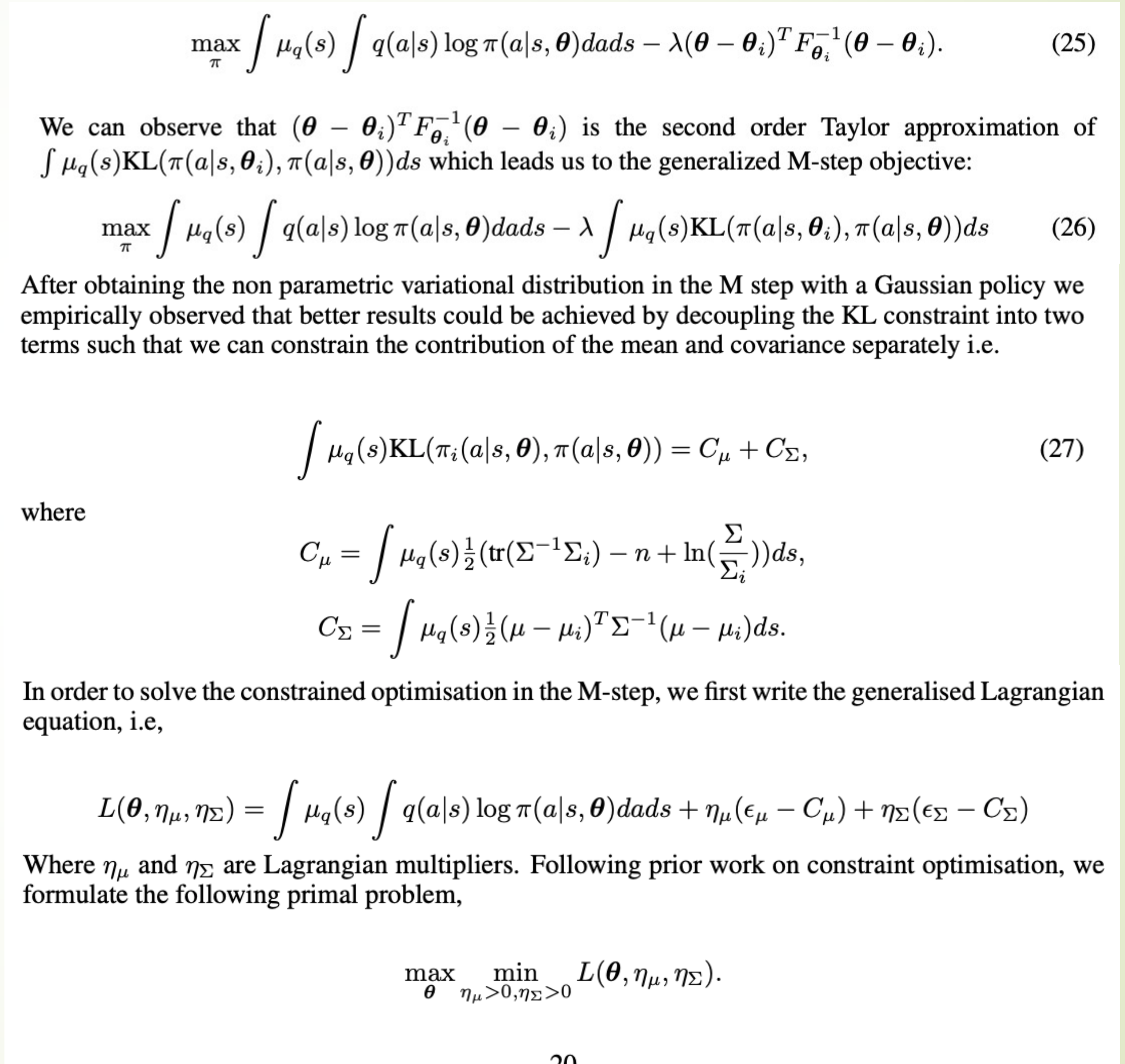
- For the M-Step

From



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

To



$$\max_{\pi} \int \mu_q(s) \int q(a|s) \log \pi(a|s,\boldsymbol{\theta}) da ds - \lambda(\boldsymbol{\theta} - \boldsymbol{\theta}_i)^T F_{\boldsymbol{\theta}_i}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_i). \tag{25}$$

We can observe that $(\boldsymbol{\theta} - \boldsymbol{\theta}_i)^T F_{\boldsymbol{\theta}_i}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_i)$ is the second order Taylor approximation of $\int \mu_q(s) \mathrm{KL}(\pi(a|s,\boldsymbol{\theta}_i), \pi(a|s,\boldsymbol{\theta})) ds$ which leads us to the generalized M-step objective:

$$\max_{\pi} \int \mu_q(s) \int q(a|s) \log \pi(a|s,\boldsymbol{\theta}) da ds - \lambda \int \mu_q(s) \mathrm{KL}(\pi(a|s,\boldsymbol{\theta}_i), \pi(a|s,\boldsymbol{\theta})) ds \tag{26}$$

After obtaining the non parametric variational distribution in the M step with a Gaussian policy we empirically observed that better results could be achieved by decoupling the KL constraint into two terms such that we can constrain the contribution of the mean and covariance separately i.e.

$$\int \mu_q(s) \mathrm{KL}(\pi_i(a|s,\boldsymbol{\theta}), \pi(a|s,\boldsymbol{\theta})) = C_\mu + C_\Sigma, \tag{27}$$

where

$$C_\mu = \int \mu_q(s) \tfrac{1}{2}(\mathrm{tr}(\Sigma^{-1}\Sigma_i) - n + \ln(\frac{\Sigma}{\Sigma_i})) ds,$$

$$C_\Sigma = \int \mu_q(s) \tfrac{1}{2}(\mu - \mu_i)^T \Sigma^{-1}(\mu - \mu_i) ds.$$

In order to solve the constrained optimisation in the M-step, we first write the generalised Lagrangian equation, i.e,

$$L(\boldsymbol{\theta}, \eta_\mu, \eta_\Sigma) = \int \mu_q(s) \int q(a|s) \log \pi(a|s,\boldsymbol{\theta}) da ds + \eta_\mu(\epsilon_\mu - C_\mu) + \eta_\Sigma(\epsilon_\Sigma - C_\Sigma)$$

Where η_μ and η_Σ are Lagrangian multipliers. Following prior work on constraint optimisation, we formulate the following primal problem,

$$\max_{\boldsymbol{\theta}} \min_{\eta_\mu > 0, \eta_\Sigma > 0} L(\boldsymbol{\theta}, \eta_\mu, \eta_\Sigma).$$

Definitions, images and equations from Abdolmaleki 2018

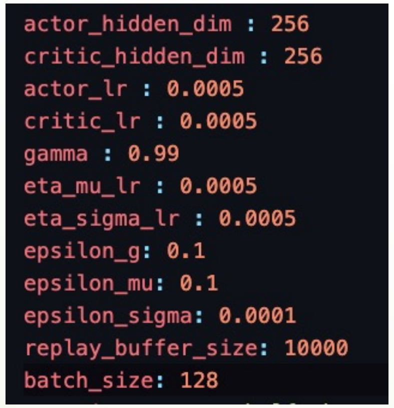
Practice

CHALLENGES

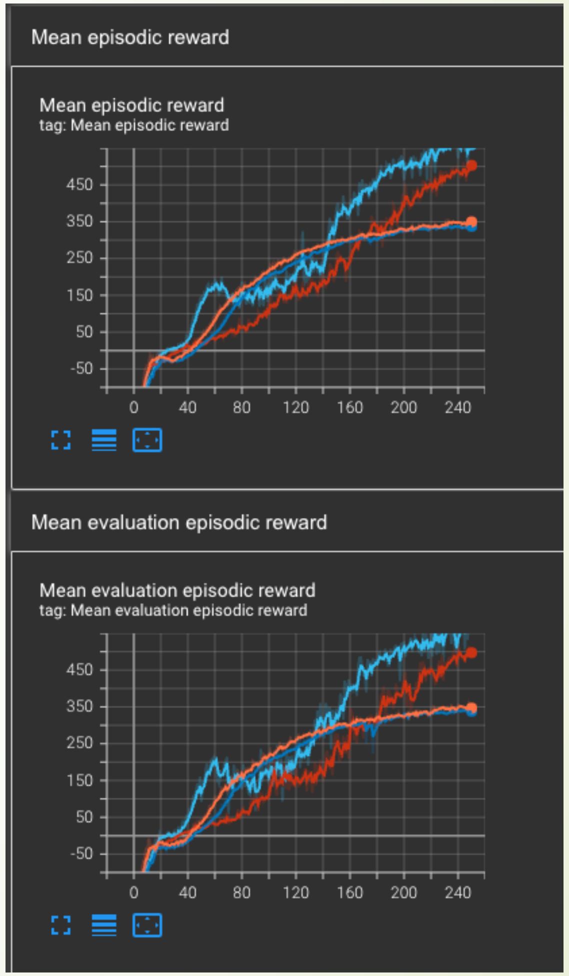
- There were some challenges before were replicated.
- The well-known RL open-source libraries do not contain an implementation of the MPO algorithm.
- An implementation of the algorithm from the authors is no longer available. (I could be wrong)
- The current best implementation is available for discrete environment and very hard to follow. (https://github.com/acyclics/MPO)
- The solution was to implement MPO using Torch and Torch RL.

RESULTS

- These are plots from 4 experiments performed in the HalfCheetah-v4 environment.
- The y-axes represents mean of the episodic rewards during the training and evaluation steps. The x-axis represents iteration count in policy iteration.
- After each policy iteration, evaluation is performed.
- Experiment details:

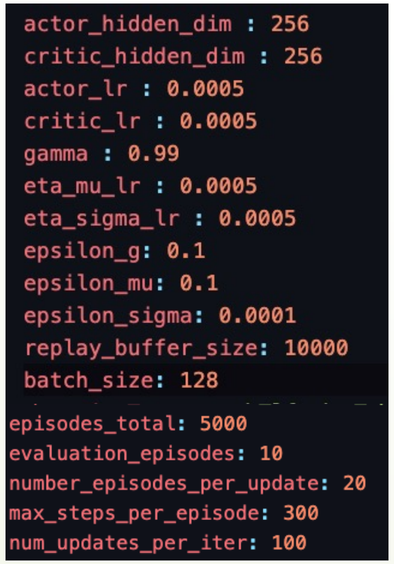


```
actor_hidden_dim : 256
critic_hidden_dim : 256
actor_lr : 0.0005
critic_lr : 0.0005
gamma : 0.99
eta_mu_lr : 0.0005
eta_sigma_lr : 0.0005
epsilon_g: 0.1
epsilon_mu: 0.1
epsilon_sigma: 0.0001
replay_buffer_size: 10000
batch_size: 128
```

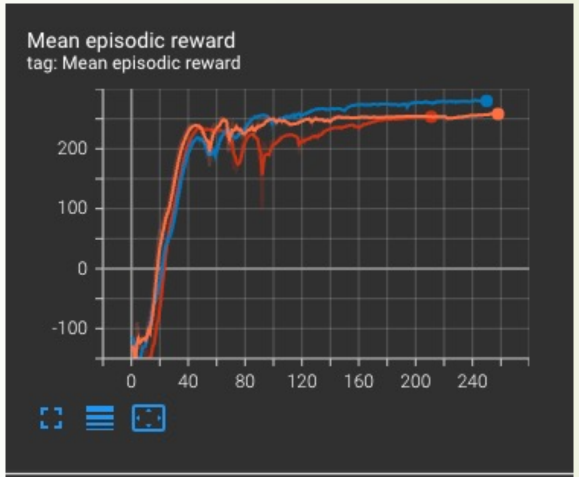


RESULTS

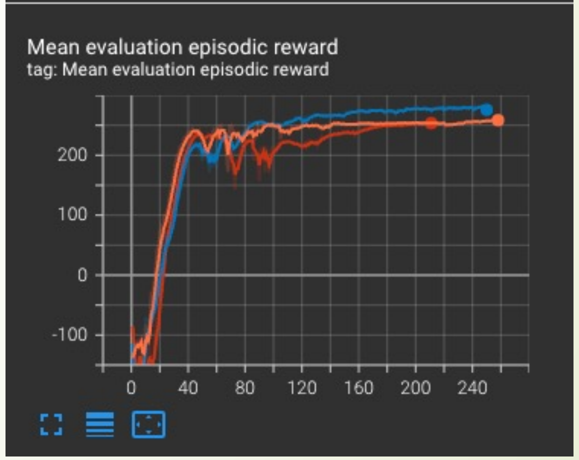
- These are plots from 3 experiments performed in the Ant-v4 environment.
- The y-axes represents mean of the episodic rewards during the training and evaluation steps. The x-axis represents iteration count in policy iteration.
- After each policy iteration, evaluation is performed.
- Experiment details:



```
actor_hidden_dim : 256
critic_hidden_dim : 256
actor_lr : 0.0005
critic_lr : 0.0005
gamma : 0.99
eta_mu_lr : 0.0005
eta_sigma_lr : 0.0005
epsilon_g: 0.1
epsilon_mu: 0.1
epsilon_sigma: 0.0001
replay_buffer_size: 10000
batch_size: 128
episodes_total: 5000
evaluation_episodes: 10
number_episodes_per_update: 20
max_steps_per_episode: 300
num_updates_per_iter: 100
```

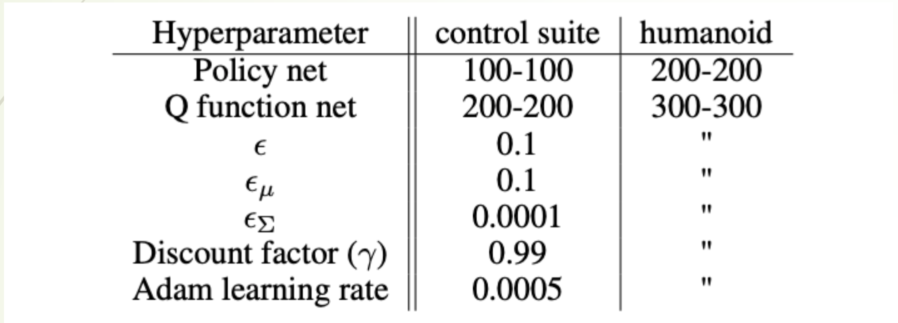


Mean evaluation episodic reward



PARAMETER TUNING

- Parameters of the algorithms were selected based on the paper for continuous control tasks.

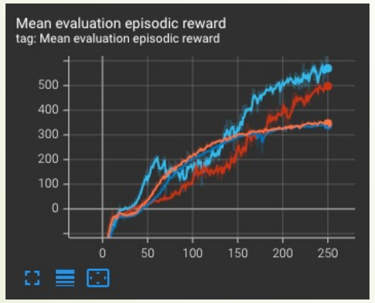


Hyperparameter	control suite	humanoid
Policy net	100-100	200-200
Q function net	200-200	300-300
ϵ	0.1	"
ϵ_μ	0.1	"
ϵ_Σ	0.0001	"
Discount factor (γ)	0.99	"
Adam learning rate	0.0005	"

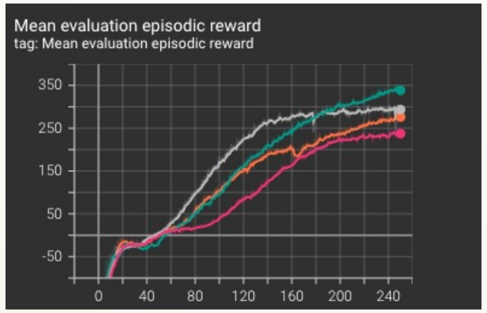
Table 2: Parameters for non-parametric variational distribution

- Sensitivity analysis performed by changing the batch_size, the learning rates, and the epsilon values on the HalfCheetah-v4 environment

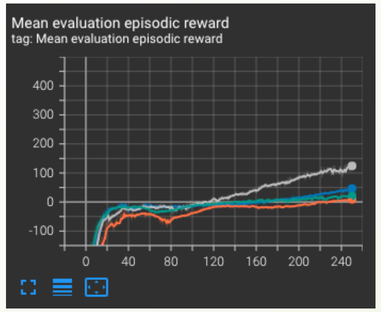
batch_size: 256,
all other same



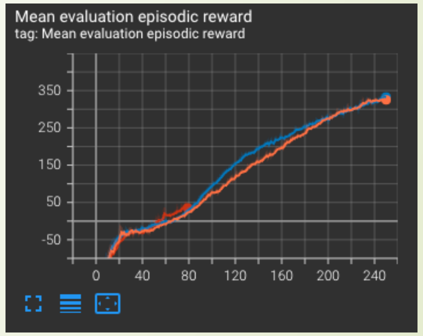
batch_size: 128,
all other same



All lr's : 0.0001,
all other same



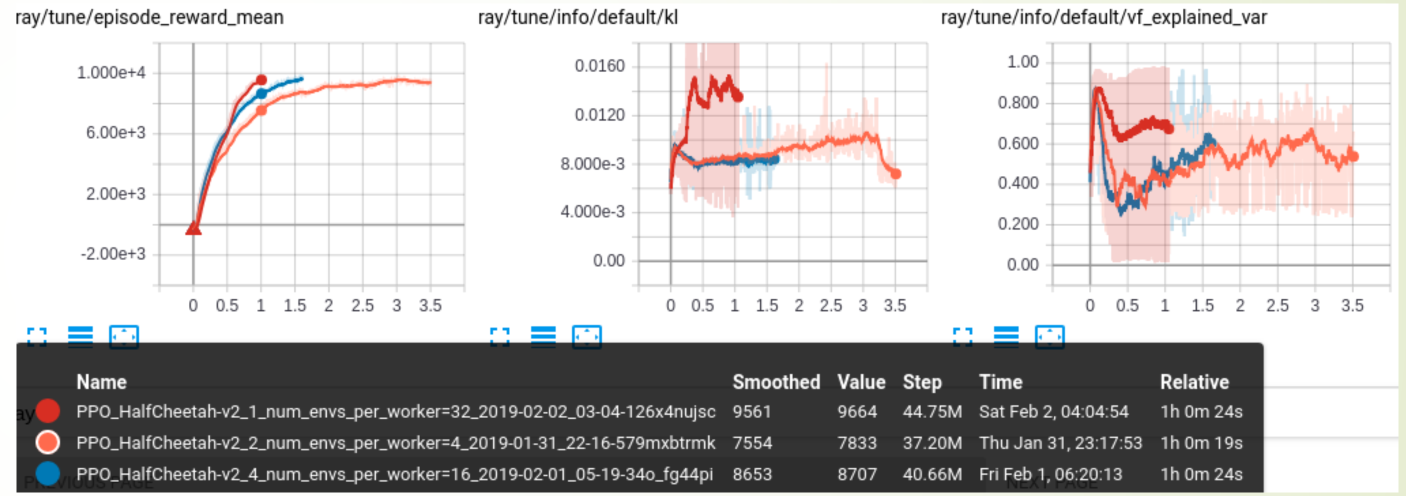
epsilon_g: 0.01
epsilon_mu: 0.01
epsilon_sigma:
0.001
, all other same



Hyperparameter diagram from Abdolmaleki 2018

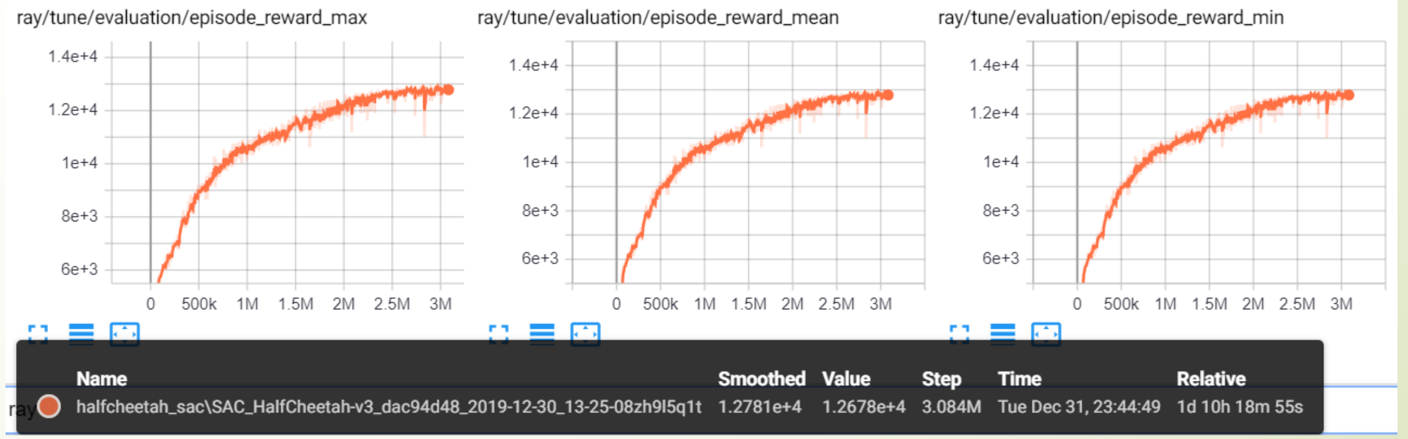
COMPARISON WITH PPO and SAC

- No experiments were performed on SAC and PPO. However, results from well tuned PPO and SAC agents using rllib are available here: https://github.com/ray-project/rl-experiments
- PPO results:



	Name	Smoothed	Value	Step	Time	Relative
●	PPO_HalfCheetah-v2_1_num_envs_per_worker=32_2019-02-02_03-04-126x4nujsc	9561	9664	44.75M	Sat Feb 2, 04:04:54	1h 0m 24s
●	PPO_HalfCheetah-v2_2_num_envs_per_worker=4_2019-01-31_22-16-579mxbtrmk	7554	7833	37.20M	Thu Jan 31, 23:17:53	1h 0m 19s
●	PPO_HalfCheetah-v2_4_num_envs_per_worker=16_2019-02-01_05-19-34o_fg44pi	8653	8707	40.66M	Fri Feb 1, 06:20:13	1h 0m 24s

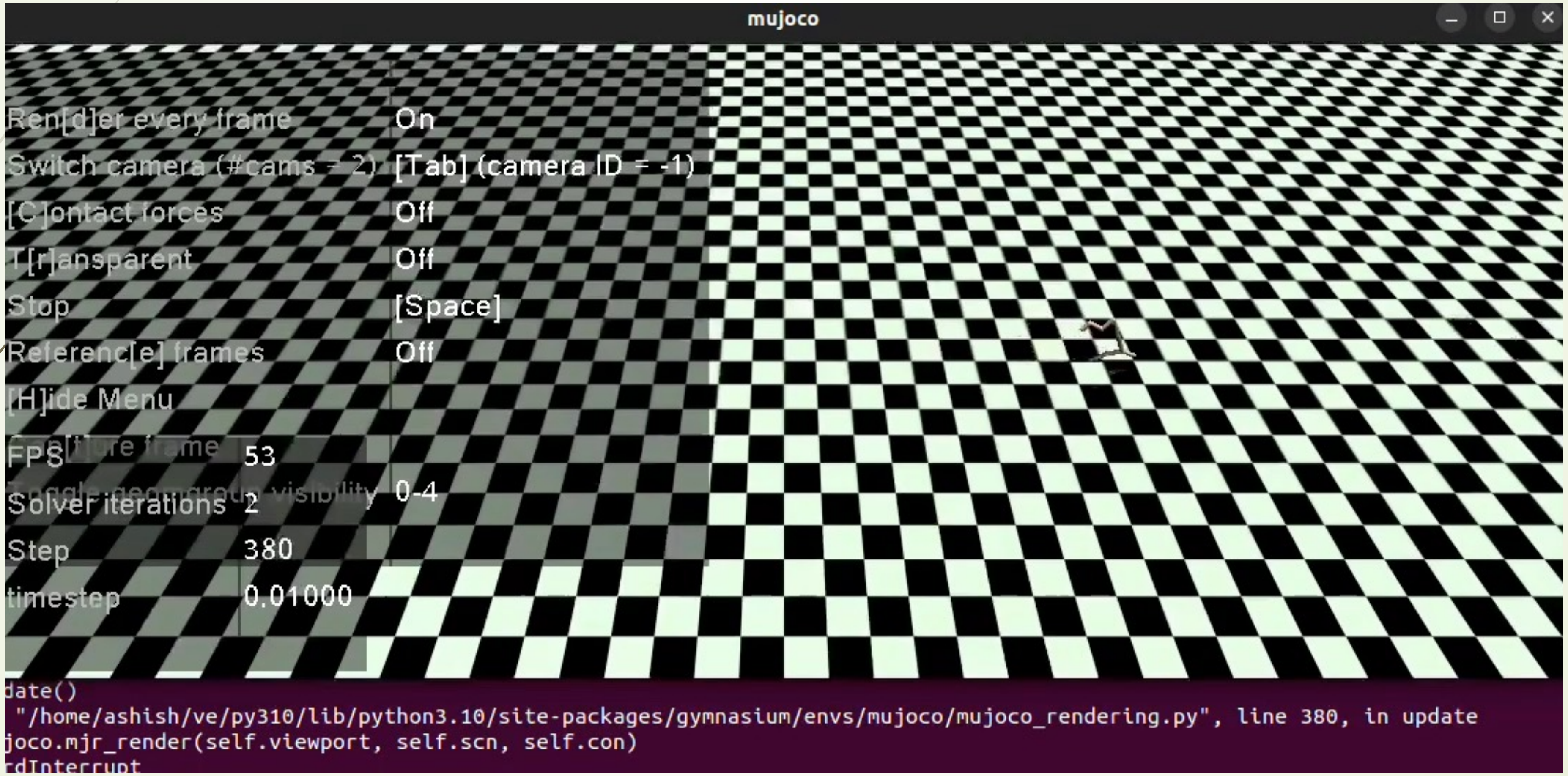
- SAC results:



	Name	Smoothed	Value	Step	Time	Relative
●	halfcheetah_sac\SAC_HalfCheetah-v3_dac94d48_2019-12-30_13-25-08zh9l5q1t	1.2781e+4	1.2678e+4	3.084M	Tue Dec 31, 23:44:49	1d 10h 18m 55s

BEHAVIOR

- Runs on one legs for most part



```
Ren[d]er every frame         On
Switch camera (#cams = 2)    [Tab] (camera ID = -1)
[C]ontact forces             Off
T[r]ansparent                Off
Stop                         [Space]
Referenc[e] frames           Off
[H]ide Menu
Cap[t]ure frame
Toggle geomgroup visibility  0-4

FPS                53
Solver iterations  2
Step               380
timestep           0.01000
```

```
date()
 "/home/ashish/ve/py310/lib/python3.10/site-packages/gymnasium/envs/mujoco/mujoco_rendering.py", line 380, in update
joco.mjr_render(self.viewport, self.scn, self.con)
rdInterrupt
```

FUTURE WORK

- Implement retrace.
- Perform experiments in real world environments.
- Implement the parametric variational distribution version.
- Implement distributed learning algorithm.
- Tune parameters to match well tuned PPO and SAC.
- Read the MPO and related paper to gain more in depth understanding.