Policy Optimization via Importance Sampling

Alberto Maria Metelli **Matteo Papini**
Francesco Faccio Marcello Restelli

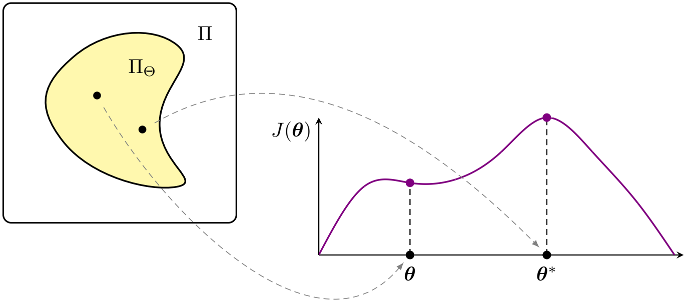
5th December 2018
Thirty-second Conference on Neural Information Processing Systems, Montréal, Canada

Policy Search

- **RL problem (?)**: find the optimal policy

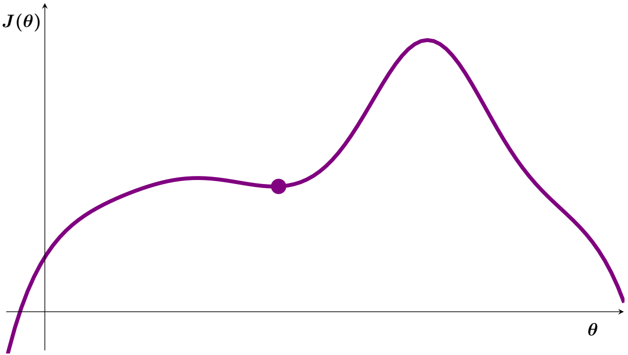
$$\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta(\mathcal{A}) \qquad \tau = [s_0, a_0, r_1, s_1, a_1, r_2 \dots] \sim p(\cdot|\boldsymbol{\theta})$$

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \right]$$



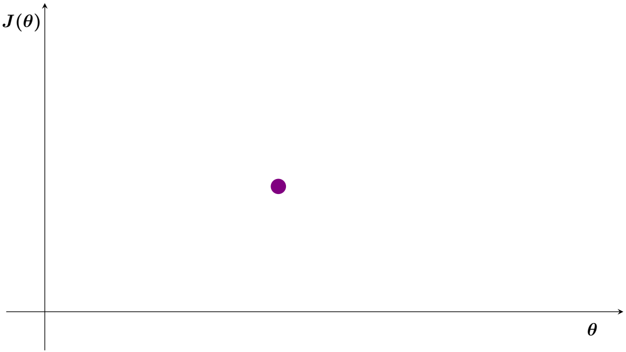
The problem with RL

- **Collecting data** is expensive
- Each policy induces a **different distribution** over data
- How to evaluate many policies with the same data?



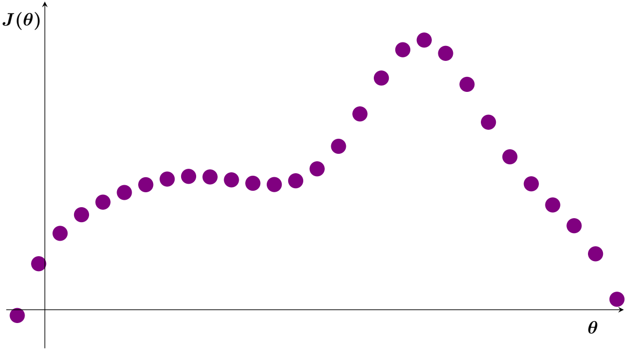
The problem with RL

- **Collecting data** is expensive
- Each policy induces a **different distribution** over data
- How to evaluate many policies with the same data?



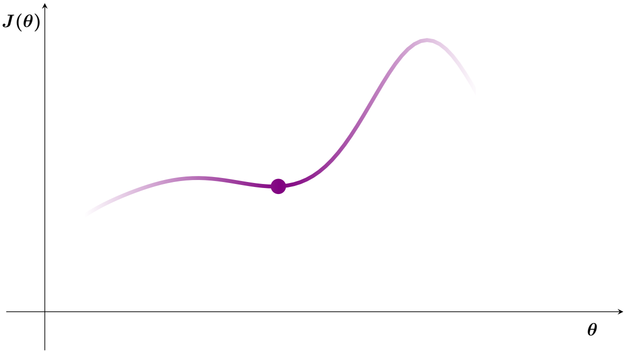
The problem with RL

- **Collecting data** is expensive
- Each policy induces a **different distribution** over data
- How to evaluate many policies with the same data?

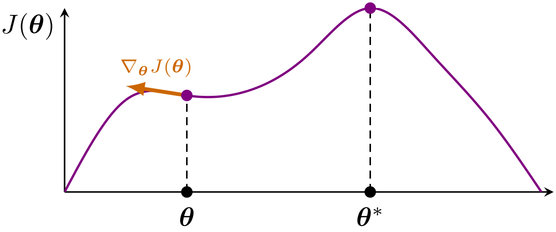


The problem with RL

- **Collecting data** is expensive
- Each policy induces a **different distribution** over data
- How to evaluate many policies with the same data? $\implies$ **off-policy** learning

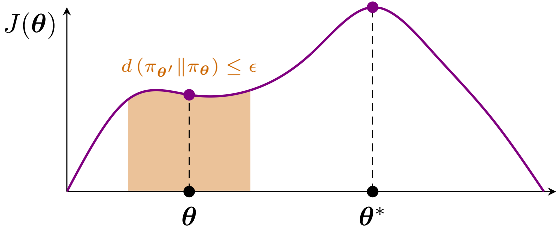


- Follow the **gradient** direction



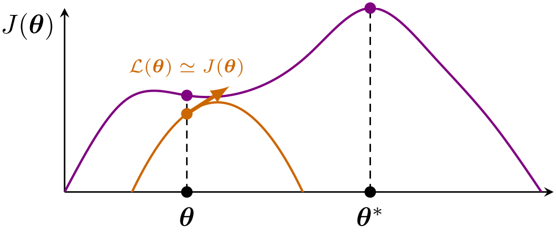
- REINFORCE (?)
- G(PO)MDP (?)
- PGPE (?)
- NAC (?)
- ...

- **Constrain** the new policy $\pi_{\boldsymbol{\theta}'}$ to be close to the current policy $\pi_{\boldsymbol{\theta}}$



- REPS (**?**)
- TRPO (**?**)
- ...

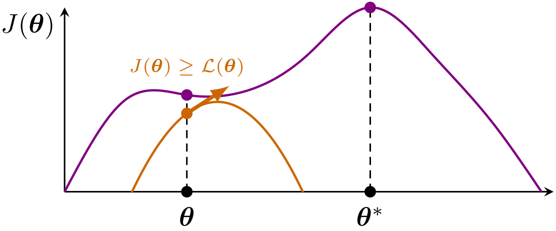
- Optimize a **surrogate** function $\mathcal{L}$



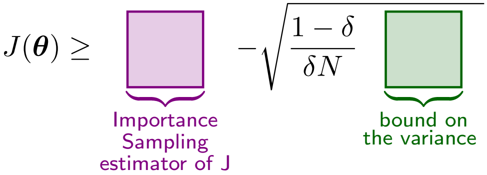
- PPO (?)
- EM (??)
- ...
- **Our algorithm**

Policy Optimization via Importance Sampling

- Optimize a **statistical lower bound** on the *estimated* J
- *Off–policy estimation* via **Importance Sampling**



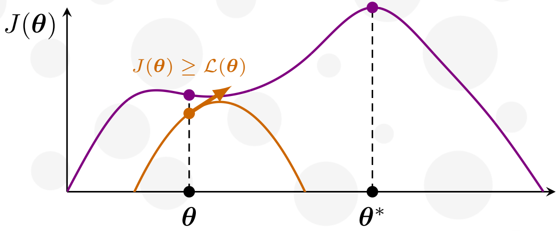
Cantelli's Inequality



Policy Optimization via Importance Sampling

POIS

- Optimize a **statistical lower bound** on the *estimated J*
- *Off–policy estimation* via **Importance Sampling**



Cantelli's Inequality

$$J(\boldsymbol{\theta}) \geq \boxed{} - \sqrt{\frac{1-\delta}{\delta N} \boxed{}}$$

Importance Sampling estimator of J

bound on the variance

Off–Distribution Estimation

- **Problem** (?): estimate the expectation of a function f under a *target* distribution P given samples from a *behavioral* distribution Q
- **Importance Sampling** (?)

$$\widehat{\mu}_{P/Q} = \frac{1}{N}\sum_{i=1}^{N}\frac{p(x_i)}{q(x_i)}f(x_i) = \frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i)f(x_i) \qquad x_i \sim Q$$

- $\widehat{\mu}_{P/Q}$ is **unbiased** but its **variance** grows proportionally to the *exponentiated Rényi divergence* between P and Q.

$$\underset{x\sim Q}{\mathbb{V}\mathrm{ar}}\left[\widehat{\mu}_{P/Q}\right] \leq \frac{1}{N}\|f\|_\infty^2 d_2\left(P\|Q\right) \qquad d_2\left(P\|Q\right) = \underset{x\sim Q}{\mathbb{E}}\left[w_{P/Q}(x)^2\right]$$

Off–Distribution Estimation

- **Problem** (?): estimate the expectation of a function f under a *target* distribution P given samples from a *behavioral* distribution Q
- **Importance Sampling** (?)

$$\widehat{\mu}_{P/Q} = \frac{1}{N}\sum_{i=1}^{N} \frac{p(x_i)}{q(x_i)} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i) \qquad x_i \sim Q$$

- $\widehat{\mu}_{P/Q}$ is **unbiased** but its **variance** grows proportionally to the *exponentiated Rényi divergence* between P and Q.

$$\underset{\mathbf{x}\sim Q}{\mathbb{V}\mathrm{ar}} \left[\widehat{\mu}_{P/Q}\right] \le \frac{1}{N} \|f\|_\infty^2 d_2\left(P\|Q\right) \qquad d_2\left(P\|Q\right) = \underset{x\sim Q}{\mathbb{E}} \left[w_{P/Q}(x)^2\right]$$

Off–Distribution Learning

- **Problem** (?): find a *target* distribution P that maximizes the expectation of a function f, given samples from a *behavioral* distribution Q

Theorem

$$\mathop{\mathbb{E}}_{x\sim P}[f(x)] \simeq \underbrace{\boxed{\frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}}_{\text{Importance Sampling estimator of } \mathbb{E}_{x\sim P}[f(x)]}$$

Importance Sampling estimator of $\mathbb{E}_{x\sim P}[f(x)]$

Off–Distribution Learning

- **Problem (?)**: find a *target* distribution P that maximizes the expectation of a function f, given samples from a *behavioral* distribution Q

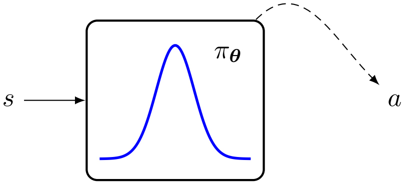
Theorem

For any $0 < \delta \leq 1$ *and* $N > 0$ *with probability at least* $1 - \delta$ *it holds that:*

$$\mathop{\mathbb{E}}_{x \sim P}[f(x)] \geq \underbrace{\frac{1}{N}\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}_{\text{Importance Sampling estimator of } \mathbb{E}_{x\sim P}[f(x)]} - \sqrt{\frac{1-\delta}{\delta N}\underbrace{\|f\|_\infty^2 d_2(P\|Q)}_{\text{bound on the variance}}}$$

Two Policy Search Paradigms

Action-based

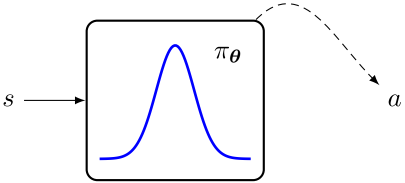


(??)

- Find the *policy* parameters $\boldsymbol{\theta}^*$ that maximize $J(\boldsymbol{\theta})$

$$J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [R(\tau)]$$

Action-based



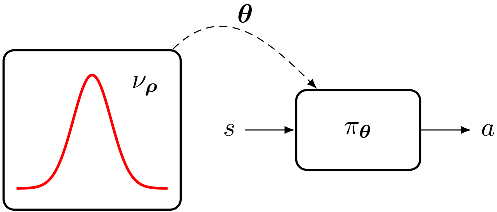
(??)

- Find the *policy* parameters $\boldsymbol{\theta}^*$ that maximize $J(\boldsymbol{\theta})$

$$J(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [R(\tau)]$$

Parameter-based

(?)



- Find the *hyperpolicy* parameters $\boldsymbol{\rho}^*$ that maximize $J(\boldsymbol{\rho})$

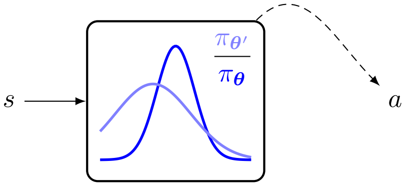
$$J(\boldsymbol{\rho}) = \mathop{\mathbb{E}}_{\boldsymbol{\theta} \sim \nu_{\boldsymbol{\rho}}} \mathop{\mathbb{E}}_{\tau \sim p(\cdot|\boldsymbol{\theta})} [R(\tau)]$$

Action-Based POIS (A-POIS)

$$\mathcal{L}_{\lambda}^{\mathrm{A-POIS}}(\boldsymbol{\theta}'/\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{t=0}^{H-1}\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau_i,t}|s_{\tau_i,t})}R(\tau_i) - \lambda\sqrt{\frac{\widehat{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)}{N}}$$

- The term $d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right)$ needs to be **estimated** from samples
- Affected by the **task horizon** H
- λ is a regularization hyperparameter

$$\lambda = \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1-\delta}{\delta}}$$

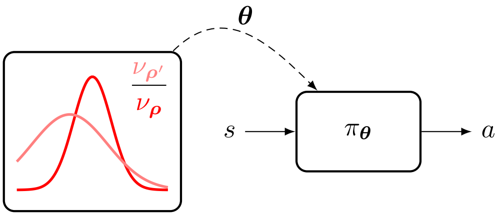


Parameter-Based POIS (P-POIS)

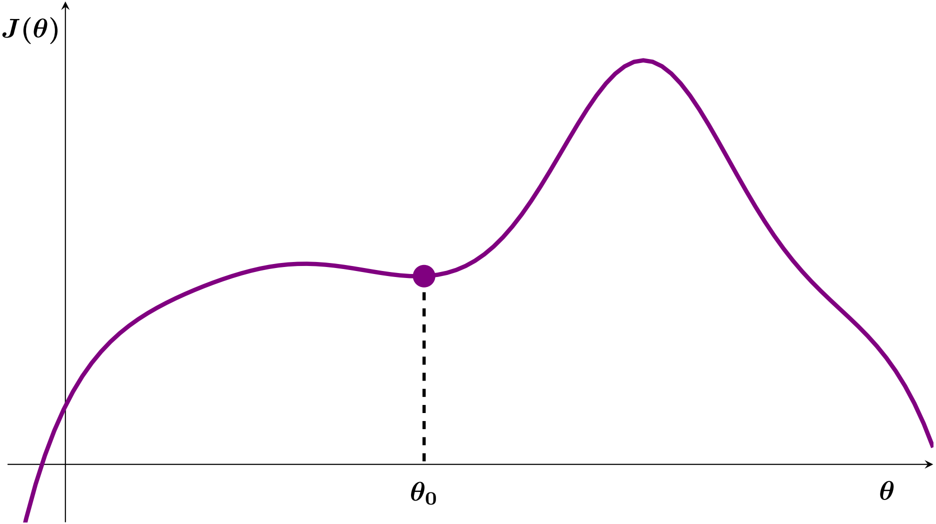
$$\mathcal{L}_{\lambda}^{\mathrm{P-POIS}}(\boldsymbol{\rho}'/\boldsymbol{\rho}) = \frac{1}{N}\sum_{i=1}^{N} \frac{\nu_{\boldsymbol{\rho}'}(\boldsymbol{\theta}_i)}{\nu_{\boldsymbol{\rho}}(\boldsymbol{\theta}_i)} R(\tau_i) - \lambda \sqrt{\frac{d_2\left(\nu_{\boldsymbol{\rho}'} \| \nu_{\boldsymbol{\rho}}\right)}{N}}$$

- The term $d_2\left(\nu_{\boldsymbol{\rho}'} \| \nu_{\boldsymbol{\rho}}\right)$ can be computed **exactly**
- Affected by the parameter space dimension $\mathbf{dim}(\boldsymbol{\theta})$
- λ is a regularization hyperparameter

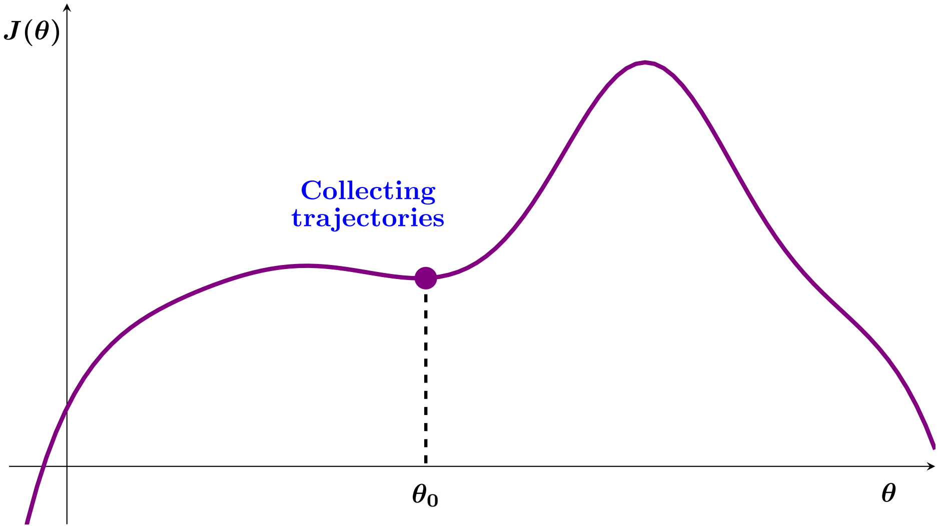
$$\lambda = \frac{R_{\max}}{1-\gamma}\sqrt{\frac{1-\delta}{\delta}}$$



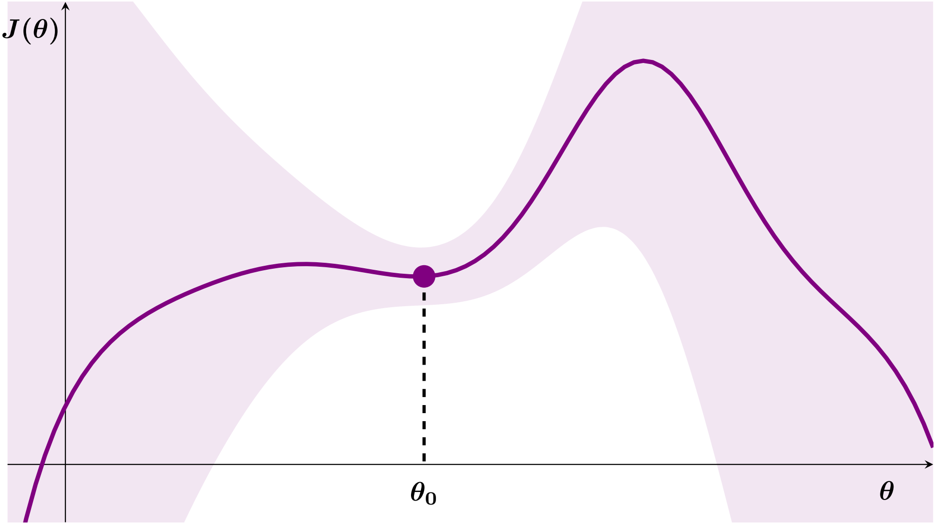
Algorithm



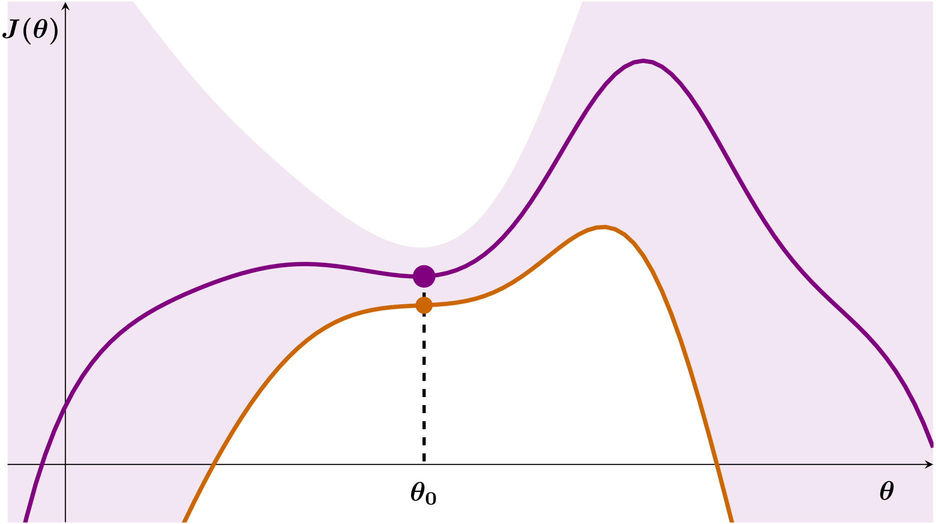
Algorithm



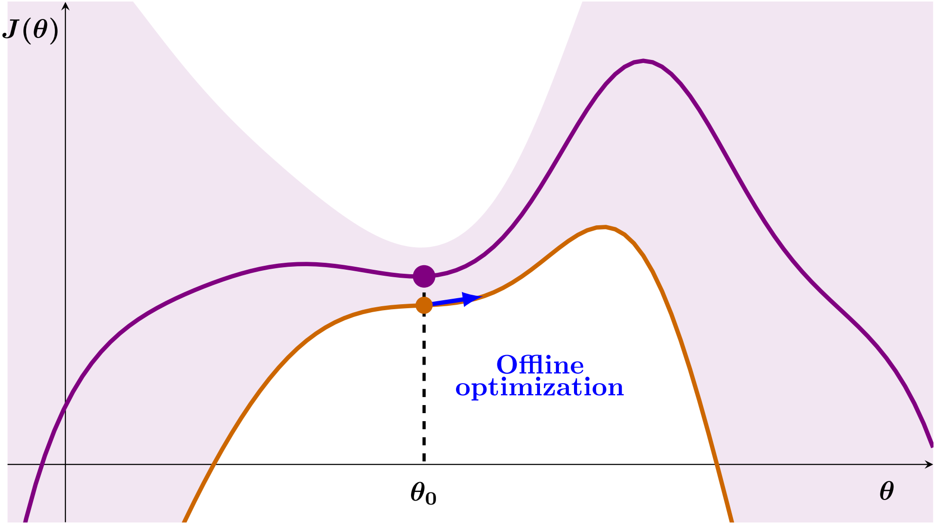
Algorithm



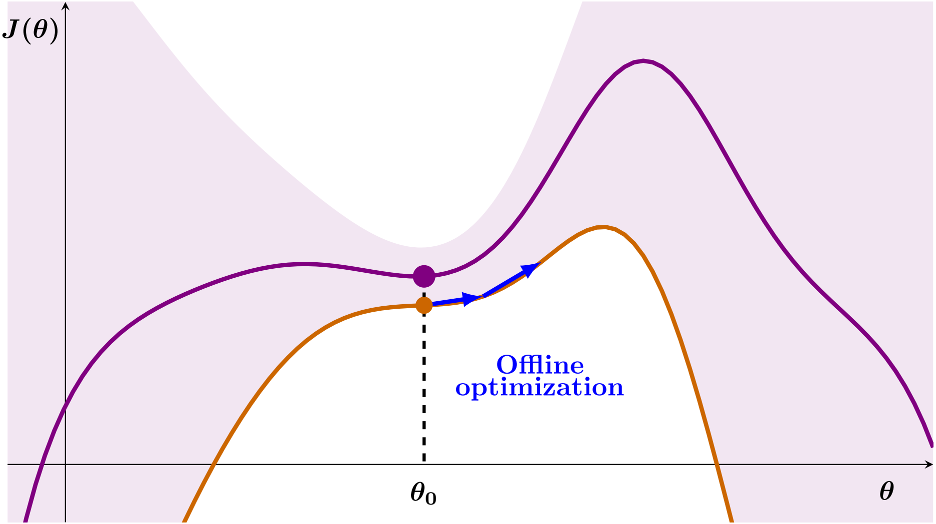
Algorithm



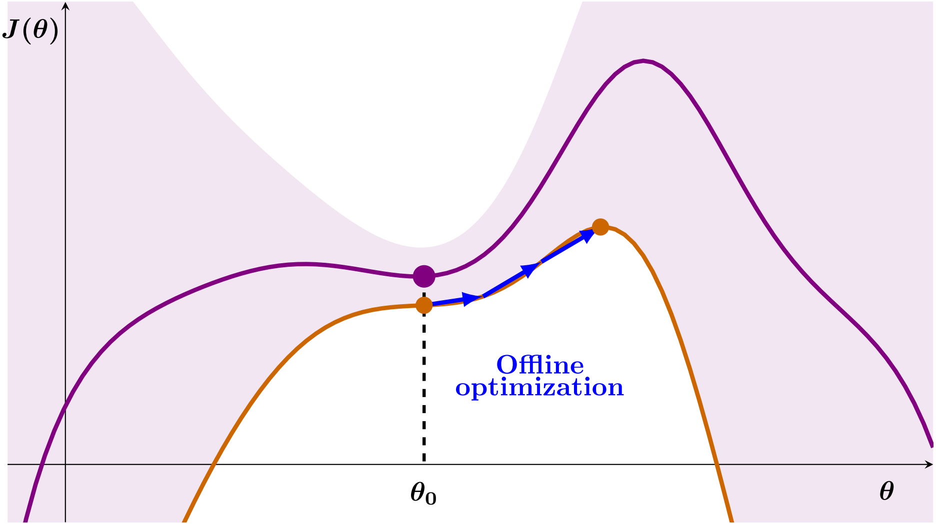
Algorithm



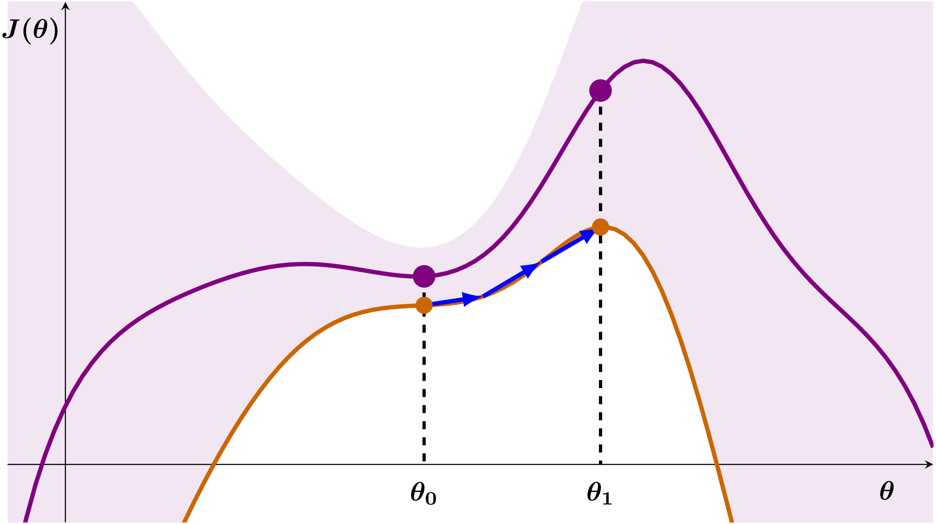
Algorithm



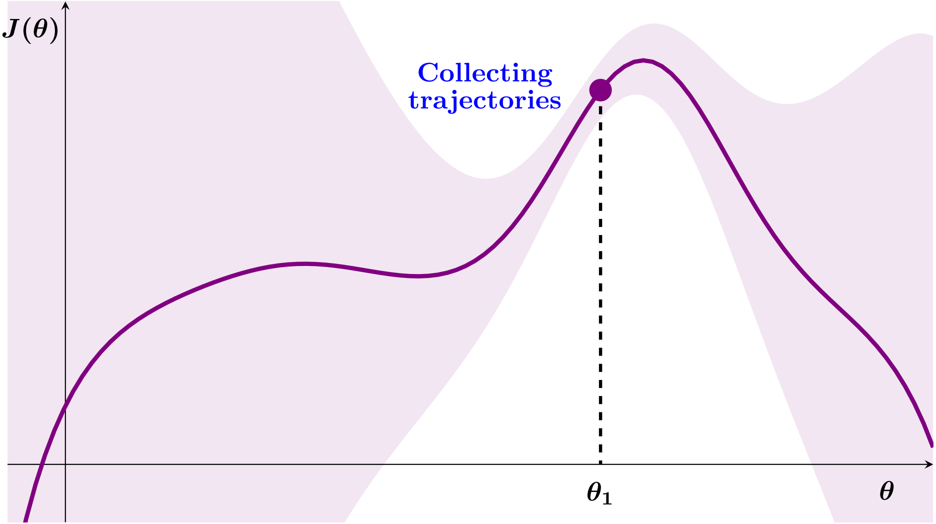
Algorithm



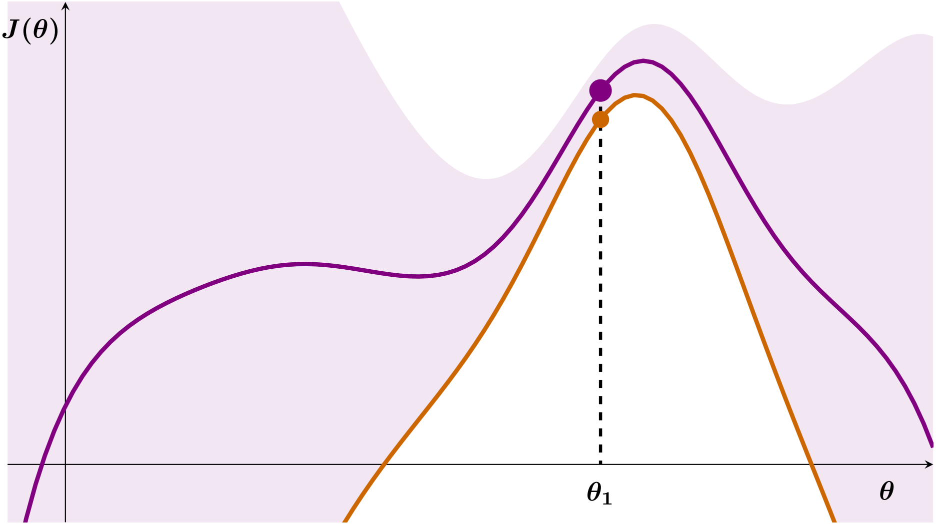
Algorithm



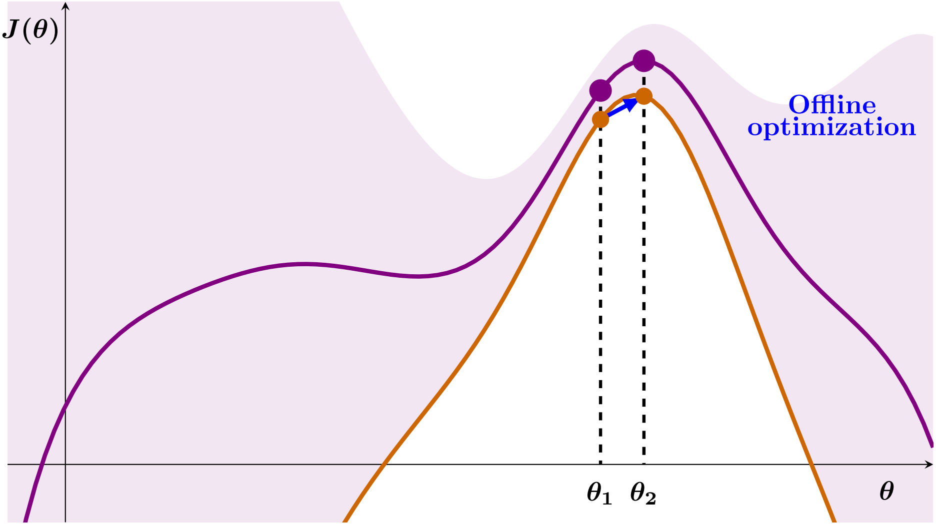
Algorithm



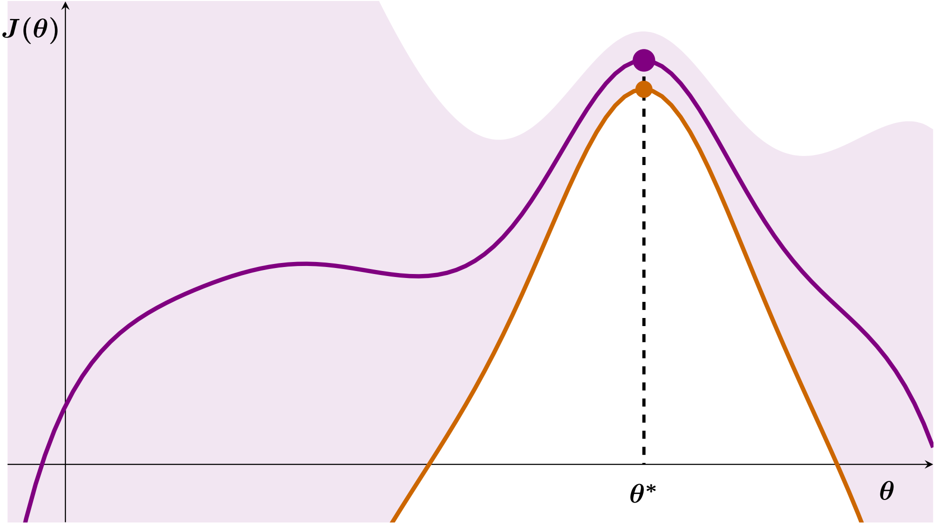
Algorithm



Algorithm



Algorithm



Algorithm Details

- *Self–normalized* importance sampling (**?**)

$$\widetilde{\mu}_{P/Q} = \frac{\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^{N} w_{P/Q}(x_i)} \qquad x_i \sim Q$$

- Effective Sample Size vs d_2

$$\mathrm{ESS} = \frac{N}{d_2(P\|Q)} \approx \frac{\|w_{P/Q}\|_1^2}{\|w_{P/Q}\|_2^2} = \widehat{\mathrm{ESS}}$$

- Gradient optimization of the bound using *line search*
- Natural gradient for P-POIS

Algorithm Details

- *Self–normalized* importance sampling (**?**)

$$\widetilde{\mu}_{P/Q} = \frac{\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^{N} w_{P/Q}(x_i)} \qquad x_i \sim Q$$

- Effective Sample Size vs d_2

$$\mathrm{ESS} = \frac{N}{d_2(P\|Q)} \approx \frac{\left\|\boldsymbol{w}_{P/Q}\right\|_1^2}{\left\|\boldsymbol{w}_{P/Q}\right\|_2^2} = \widehat{\mathrm{ESS}}$$

- Gradient optimization of the bound using *line search*
- Natural gradient for P-POIS

Algorithm Details

- *Self–normalized* importance sampling (**?**)

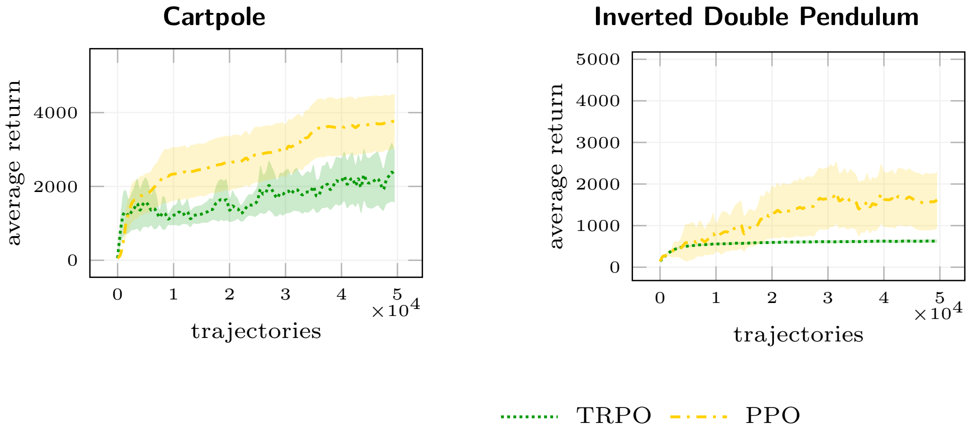
$$\widetilde{\mu}_{P/Q} = \frac{\sum_{i=1}^{N} w_{P/Q}(x_i) f(x_i)}{\sum_{i=1}^{N} w_{P/Q}(x_i)} \qquad x_i \sim Q$$

- Effective Sample Size vs d_2

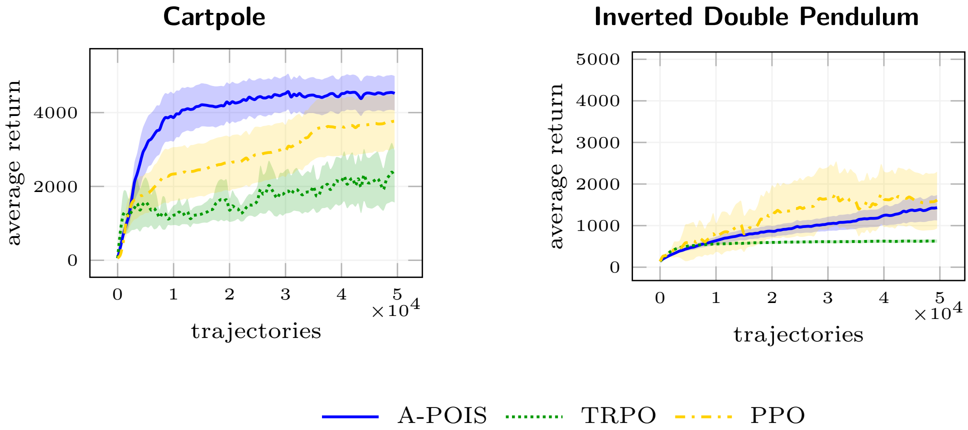
$$\mathrm{ESS} = \frac{N}{d_2(P\|Q)} \approx \frac{\left\|\boldsymbol{w}_{P/Q}\right\|_1^2}{\left\|\boldsymbol{w}_{P/Q}\right\|_2^2} = \widehat{\mathrm{ESS}}$$

- Gradient optimization of the bound using *line search*
- Natural gradient for P-POIS

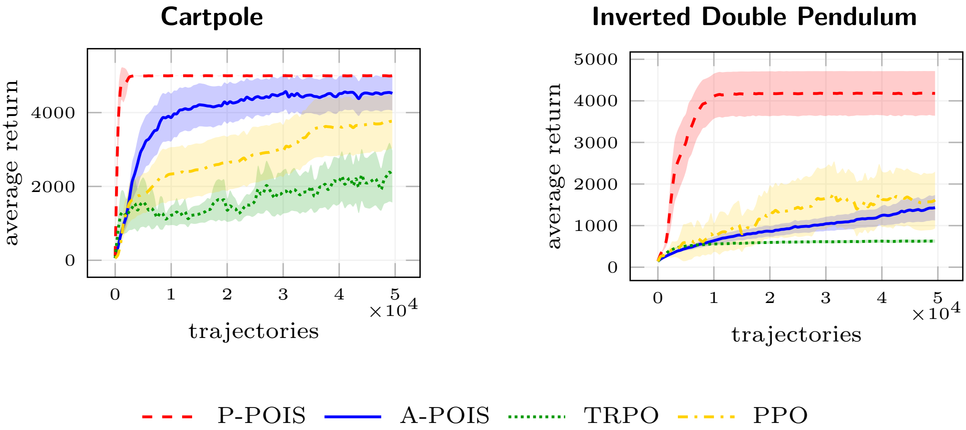
- Comparison with TRPO (?) and PPO (?)



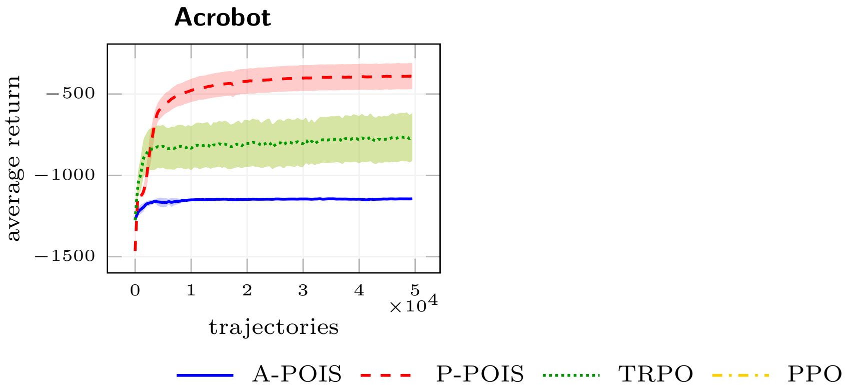
- Comparison with TRPO (?) and PPO (?)



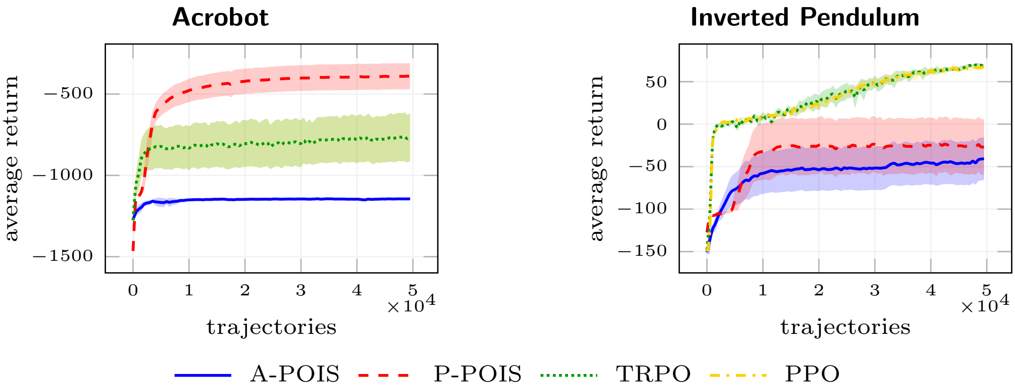
- Comparison with TRPO (?) and PPO (?)



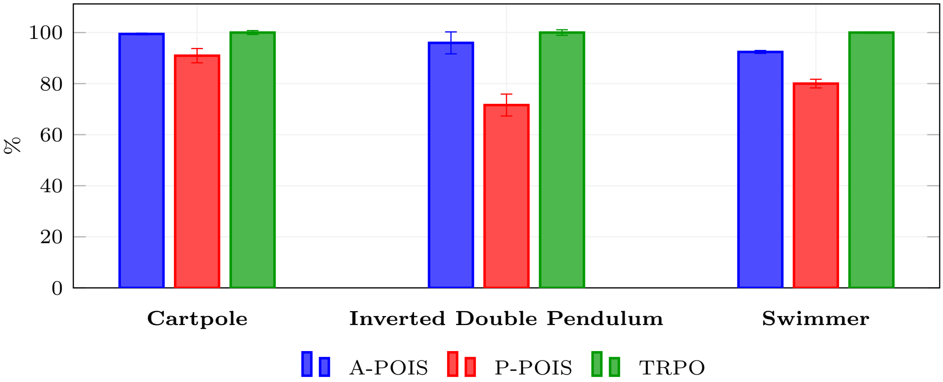
- Comparison with TRPO (?) and PPO (?)



- Comparison with TRPO (?) and PPO (?)



- Continuous control benchmark (?)



Conclusions

- Contributions
  - A novel statistical lower bound on off-policy evaluations
  - Action-based and parameter-based POIS versions

- Future works
  - Per–decision importance sampling
  - Multiple/Mixture importance sampling

Conclusions

- Contributions
  - A novel statistical lower bound on off-policy evaluations
  - Action-based and parameter-based POIS versions
- Future works
  - Per–decision importance sampling
  - Multiple/Mixture importance sampling

Thank You for Your Attention!

- Poster **#109** @ 517 AB (upstairs!)
- Code: `https://github.com/T3p/pois`
- Contact: matteo.papini@polimi.it
- Web page: `t3p.github.io/NeurIPS18`

References

Natural Gradient

- The (exponentiated) Renyi divergence induces a Riemannian metric given by the **Fisher information** (?):

$$d_2(\nu_{\boldsymbol{\rho}'}|\nu_{\boldsymbol{\rho}}) = 1 + (\boldsymbol{\rho}' - \boldsymbol{\rho})^T \mathcal{F}(\boldsymbol{\rho})(\boldsymbol{\rho}' - \boldsymbol{\rho}) + o(\|\boldsymbol{\rho}' - \boldsymbol{\rho}\|^2)$$

- This means $\mathbb{V}\mathrm{ar}[\boldsymbol{w}] \simeq (\boldsymbol{\rho}' - \boldsymbol{\rho})^T \mathcal{F}(\boldsymbol{\rho})(\boldsymbol{\rho}' - \boldsymbol{\rho})$
- We can use a normalized **Natural Gradient** (?) update to keep the variance under control:

$$\boldsymbol{\rho}' = \boldsymbol{\rho} + \frac{\alpha}{\nabla_{\boldsymbol{\rho}} J(\boldsymbol{\rho})^T \mathcal{F}^{-1}(\boldsymbol{\rho}) \nabla_{\boldsymbol{\rho}} J(\boldsymbol{\rho})} \mathcal{F}^{-1}(\boldsymbol{\rho}) \nabla_{\boldsymbol{\rho}} J(\boldsymbol{\rho}) \implies \mathbb{V}\mathrm{ar}[\boldsymbol{w}] \simeq \alpha^2$$

- This is more feasible in PB-POIS, where the Fisher matrix can be easily computed (at least for Gaussian hyperpolicies)

Estimating the Rényi Divergence

- Loose bound

$$d_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) \le \left(\sup_{s\in\mathcal{S}} d_2\left(\pi_{\boldsymbol{\theta}'}(\cdot|s)\|\pi_{\boldsymbol{\theta}}(\cdot|s)\right)\right)^H$$

- Monte Carlo Estimator

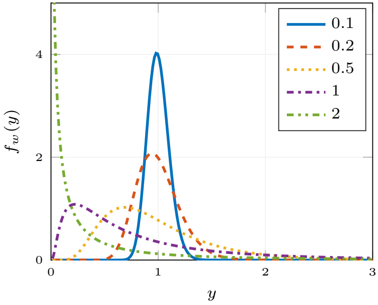
$$\widehat{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) = \frac{1}{N}\sum_{i=1}^{N}\prod_{t=0}^{H-1}\left(\frac{\pi_{\boldsymbol{\theta}'}(a_{\tau_i,t}|s_{\tau_i,t})}{\pi_{\boldsymbol{\theta}}(a_{\tau_i,t}|s_{\tau_i,t})}\right)^2$$

- Exploiting the fact that we know $\pi_{\boldsymbol{\theta}}$

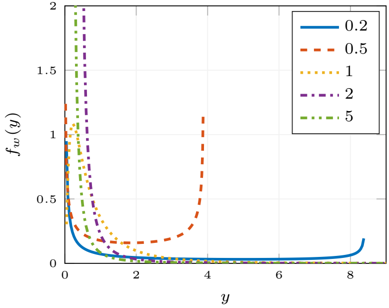
$$\widehat{d}_2\left(p(\cdot|\boldsymbol{\theta}')\|p(\cdot|\boldsymbol{\theta})\right) = \frac{1}{N}\sum_{i=1}^{N}\prod_{t=0}^{H-1} d_2\left(\pi_{\boldsymbol{\theta}'}(\cdot|s_{\tau_i,t})\|\pi_{\boldsymbol{\theta}}(\cdot|s_{\tau_i,t})\right)$$

Importance Weights Distribution

- Gaussian distributions: $P \sim \mathcal{N}(\mu_P, \sigma_P^2)$ (target) and $Q \sim \mathcal{N}(\mu_Q, \sigma_Q^2)$ (behavioral)
  - $\sigma_Q^2 > \sigma_P^2 \implies w_{P/Q}$ is bounded
  - $\sigma_Q^2 = \sigma_P^2 \implies w_{P/Q}$ admits all finite moments
  - $\sigma_Q^2 < \sigma_P^2 \implies w_{P/Q}$ admits only few finite moments (heavy-tailed)



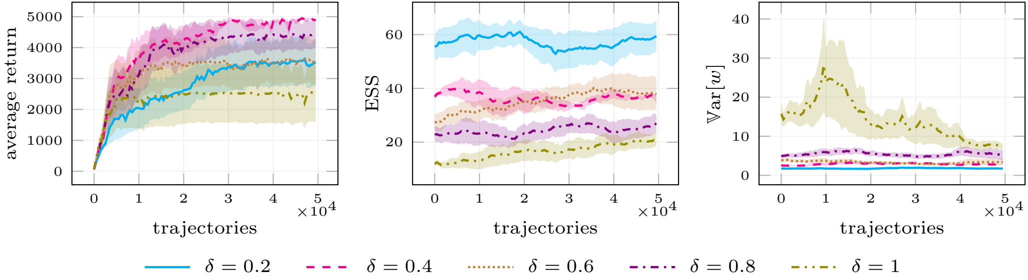
$Q \sim \mathcal{N}(0,1) \qquad \sigma_P = 1$



$Q \sim \mathcal{N}(0,1) \qquad \mu_P = 1$

Experiments (cont.)

- A-POIS with different values of δ in the Cartpole environment



Neuron-Based POIS

- The penalization term of PB-POIS is amplified by the **dimensionality** of policy parameters
- As a result, PB-POIS with deep neural policies is **overly conservative**
- A possible workaround is to **group** policy parameters into smaller blocks, learned independently
- We group network weights by the **neuron** they activate