Entropy Regularization in RL through Interpolation

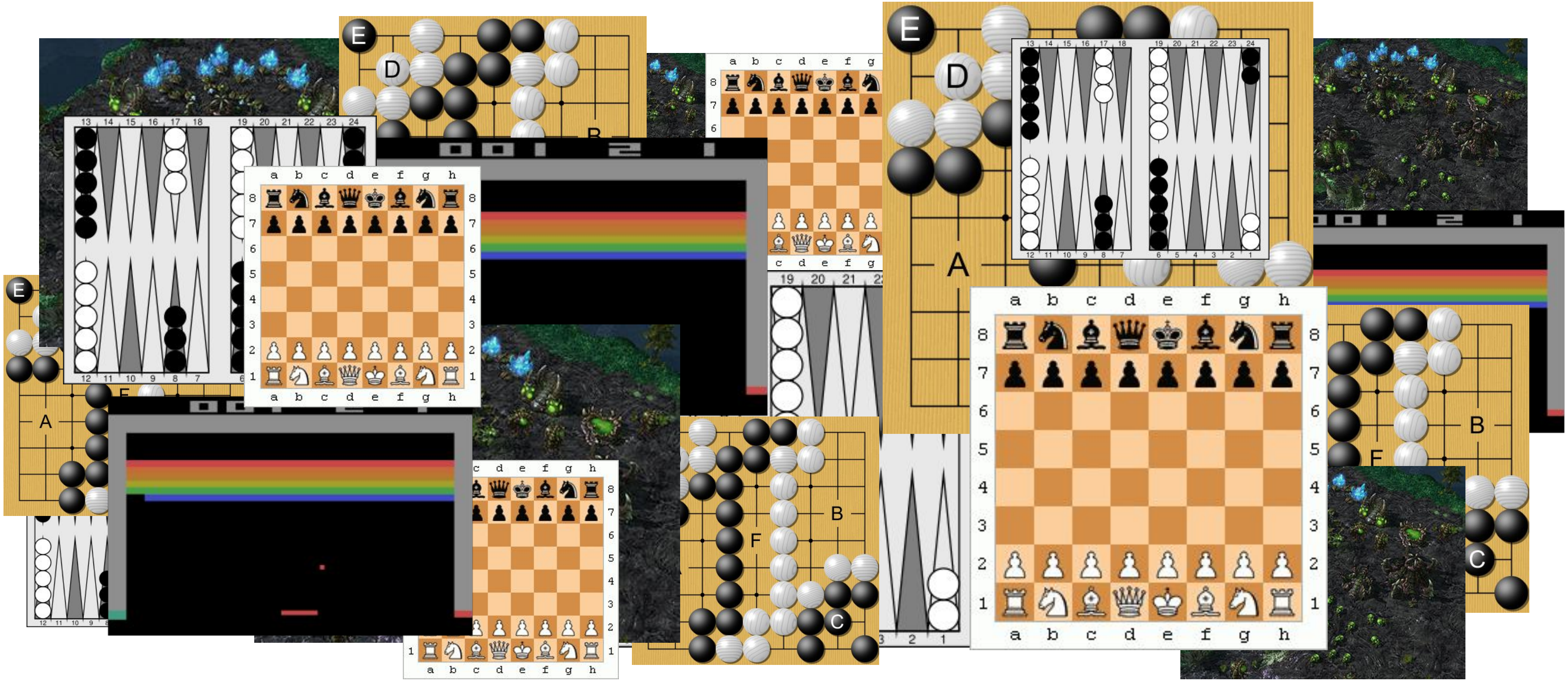
Riad Akrour

TECHNISCHE UNIVERSITÄT DARMSTADT

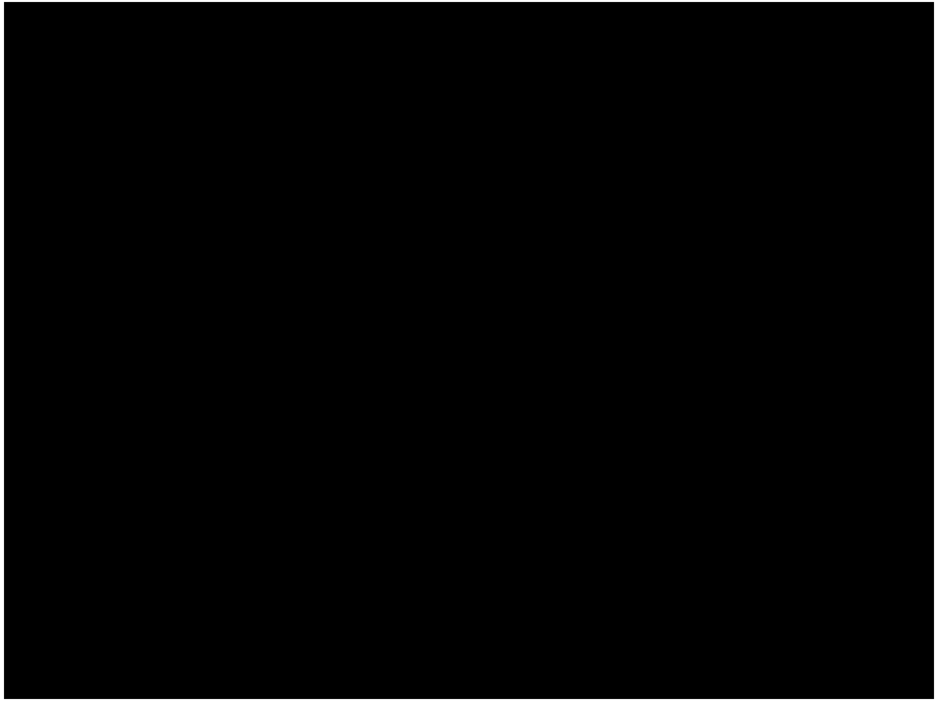
Outline

- **Entropy Regularization in RL**
  - Why it is important
  - How to solve the regularized problem
- Application: **Explainable RL**

Successes of RL



RL in the Physical World



High Acceleration Reinforcement Learning for Real-World Juggling with Binary Rewards
K. Ploeger, M. Lutter, J. Peters
CoRL20

- **Small number of parameters (<20)**
- **Open loop policy**
- **Black-box optimization**
- Requires expert demonstrations

Relative Entropy Policy Search (REPS)

- For Gaussian policies $\pi_k(\theta) = \mathcal{N}(\theta|\mu_k, \Sigma_k)$

$$\max_{\pi_k} \quad \mathbb{E}_{\theta\sim\pi_k}\left[R(\theta)\right]$$
$$s.t. \quad \mathrm{KL}\left(\pi_k||\pi_{k-1}\right) \leq \epsilon$$

- Closed-form solution

$$\pi_k \propto \pi_{k-1} \exp\left(\frac{R}{\eta}\right)$$

- **Small number of parameters (<20)**
- **Open loop policy**
- **Black-box optimization**
- Requires expert demonstrations

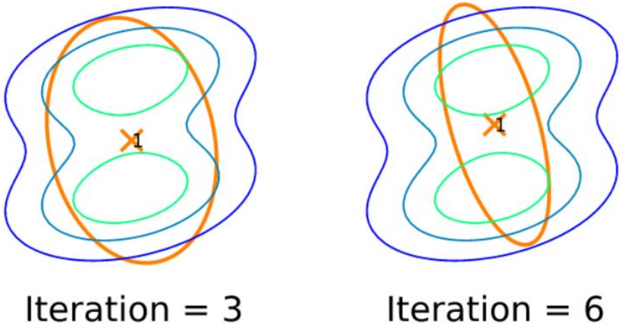
Outline

- **Entropy Regularization in RL**
    - How to solve this regularized problem
        - From black-box optimization
        - To Reinforcement Learning
        - To Deep RL
        - To convex optimization
    - Why it is important
- Application: **Explainable RL**

Model-based REPS (MORE)

- For Gaussian policies $\pi_k(\theta) = \mathcal{N}(\theta | \mu_k, \Sigma_k)$

$$\begin{aligned} \max_{\pi_k} \quad & \mathbb{E}_{\theta \sim \pi_k}\left[\hat{R}(\theta)\right] \\ s.t. \quad & \mathrm{KL}\left(\pi_k || \pi_{k-1}\right) \leq \epsilon \\ & \mathcal{H}(\pi_{k-1}) - \mathcal{H}(\pi_k) \leq \beta \end{aligned}$$



Deisenroth et al.

- Closed-form solution in probability space

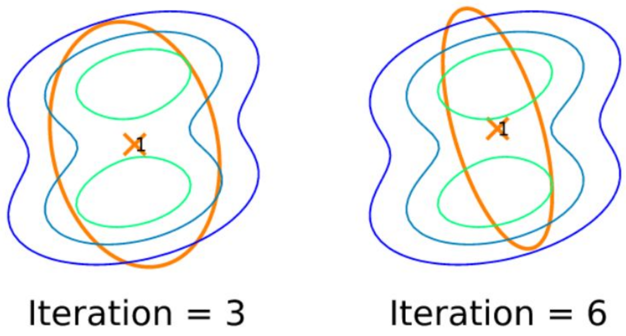
$$\pi_k \propto \pi_{k-1}^{\eta/(\eta+\omega)} \exp\left(\frac{\hat{R}}{\eta+\omega}\right)$$

Model-Based Relative Entropy Stochastic Search; A. Abdolmaleki, R. Lioutikov, N. Lau, L. Reis, J. Peters, G. Neumann; NeurIPS15

Model-based REPS (MORE)

- For Gaussian policies $\pi_k(\theta) = \mathcal{N}(\theta|\mu_k, \Sigma_k)$

$$\begin{aligned} \max_{\pi_k} \quad & \mathbb{E}_{\theta\sim\pi_k}\left[\hat{R}(\theta)\right] \\ s.t. \quad & \mathrm{KL}\left(\pi_k||\pi_{k-1}\right) \leq \epsilon \\ & \mathcal{H}(\pi_{k-1}) - \mathcal{H}(\pi_k) \leq \beta \end{aligned}$$



- Closed-form solution in probability space

$$\pi_k \propto \pi_{k-1}^{\eta/(\eta+\omega)} \exp\left(\frac{\hat{R}}{\eta+\omega}\right)$$

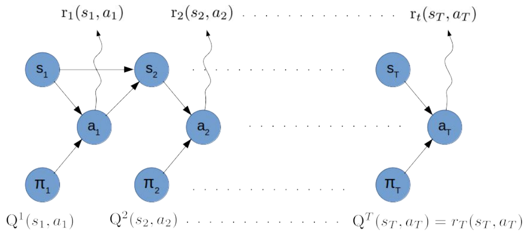
- Closed-form solution in parameter space for quad. $\hat{R}$

Model-Based Relative Entropy Stochastic Search; A. Abdolmaleki, R. Lioutikov, N. Lau, L. Reis, J. Peters, G. Neumann; NeurIPS15

Step-based MORE (MOTO)

- For linear-Gaussian policies $\pi_k^t(a_t|s_t) = \mathcal{N}\left(a_t|K_t s_t, \Sigma_t\right)$

$$\max_{\pi_k^t} \quad \mathbb{E}_{s\sim p_{k-1}^t, a\sim\pi_k^t(.|s)}\left[\hat{Q}_{k-1}^t(s,a)\right]$$
$$s.t. \quad \mathbb{E}_{s\sim p_{k-1}^t}\left[\mathrm{KL}\left(\pi_k^t(.|s)||\pi_{k-1}^t(.|s)\right)\right] \leq \epsilon$$
$$\mathbb{E}_{s\sim p_{k-1}^t}\left[\mathcal{H}(\pi_{k-1}^t(.|s)) - \mathcal{H}(\pi_k^t(.|s))\right] \leq \beta$$



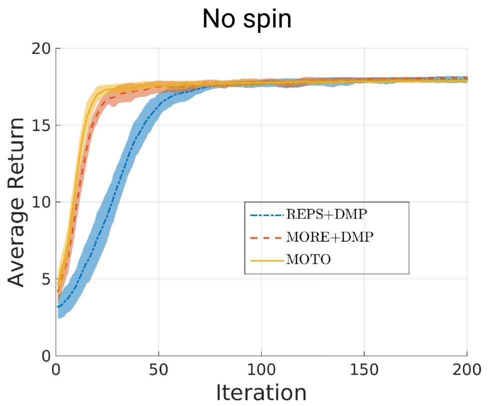
- Closed-form solution in probability space

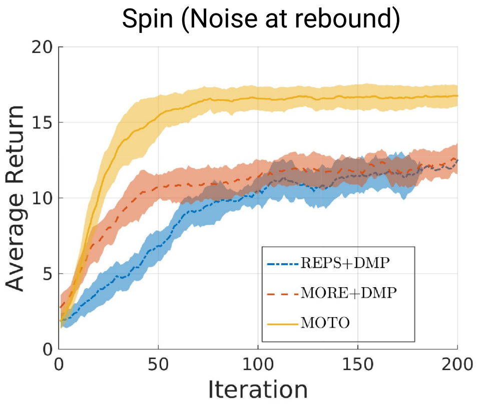
$$\pi_k^t(.|s) \propto \pi_{k-1}^t(.|s)^{\eta/(\eta+\omega)} \exp\left(\frac{\hat{Q}_{k-1}^t(s,.)}{\eta+\omega}\right)$$

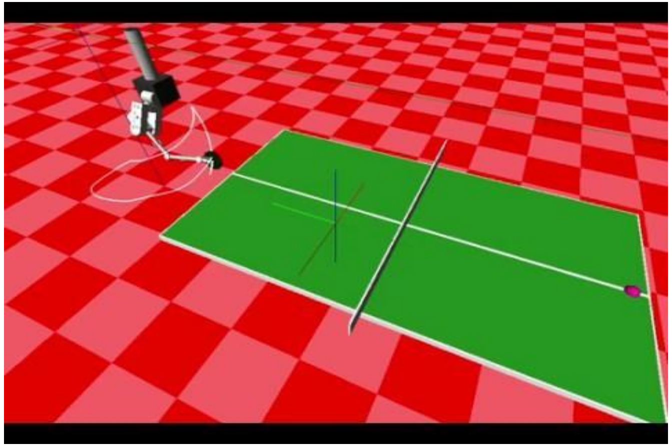
- Closed-form solution in parameter space for quad. $\hat{Q}$
- Additional Gaussian approx. of p_{k-1}^t

Model-free Trajectory Optimization for Reinforcement Learning; R. Akrour, A. Abdolmaleki, H. Abdulsamad, G. Neumann; ICML16

MOTO vs Black-box



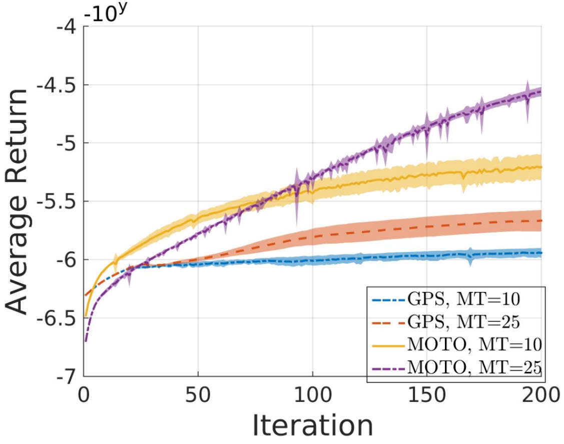




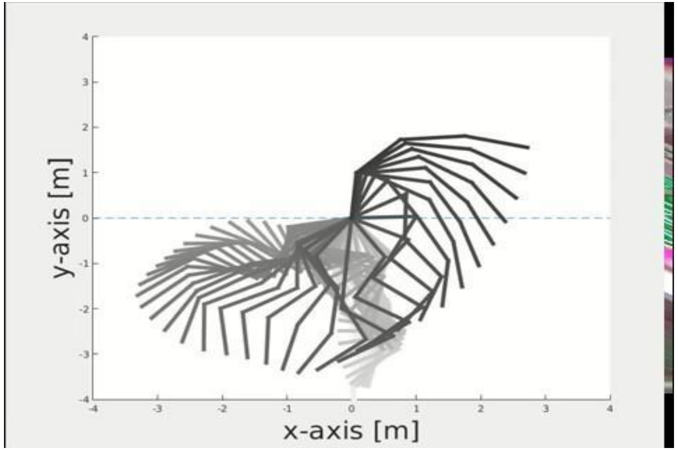
- Comparable to black-box methods with open-loop policies despite much larger search space
- Closed-loop policy adapts to changes in the environment

Model-free Trajectory Optimization for Reinforcement Learning; R. Akrour, A. Abdolmaleki, H. Abdulsamad, G. Neumann; ICML16

MOTO vs trajectory optimization



GPS = trajectory optimization algorithm used in:
Learning Neural Network Policies with Guided Policy Search under Unknown Dynamics
S. Levine and P. Abbeel
NeurIPS14



- Back propagating approx. quad. Q functions > forward propagating approx. linear dynamics models

Model-free Trajectory Optimization for Reinforcement Learning; R. Akrour, A. Abdolmaleki, H. Abdulsamad, G. Neumann; ICML16

Why does MOTO work?

- MOTO policy update

$$
\begin{aligned}
\max_{\pi_k^t} \quad & \mathbb{E}_{s\sim p_{k-1}^t, a\sim\pi_k^t(.|s)}\left[\hat{Q}_{k-1}^t(s,a)\right] \\
s.t. \quad & \mathbb{E}_{s\sim p_{k-1}^t}\left[\mathrm{KL}\left(\pi_k^t(.|s)||\pi_{k-1}^t(.|s)\right)\right] \leq \epsilon \\
& \mathbb{E}_{s\sim p_{k-1}^t}\left[\mathcal{H}(\pi_{k-1}^t(.|s)) - \mathcal{H}(\pi_k^t(.|s))\right] \leq \beta
\end{aligned}
$$

- Strict compliance with KL-divergence cst. important in practice… why?

- Objective and constraints expressed in terms of p_{k-1}^t… is it reasonable?

Policy improvement

- In tabular RL, $\pi_k(s) = \arg\max Q_{k-1}(s, .)$
  - Take better action in **all states**
- Relaxation when using function approximators
  - $\pi_k = \arg\max_\pi \quad \mathbb{E}_{s \sim p_{k-1}, a \sim \pi(.|s)} \left[Q_{k-1}(s, a) \right]$
  - Take better actions in average of **previous state** distribution
  - What about average under the **current state** distribution $\mathbb{E}_{s \sim p_k, a \sim \pi(.|s)} \left[Q_{k-1}(s, a) \right]$?

Notation

- Π matrix representation of policy π

- Π matrix of size $|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|$

$$\Pi_{(s,(s',a))} = \pi(a|s) \text{ if } s = s', 0 \text{ else}$$

Notation

- Π matrix representation of policy π
- P transition matrix

- P matrix of size $|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|$

  $P_{((s,a),s')} = p(s'|s,a)$

Notation

- Π matrix representation of policy π
- P transition matrix
- Value function $V^\pi = \Pi R + \gamma \Pi P V^\pi$

- R matrix of size $|\mathcal{S}||\mathcal{A}| \times 1$

  $R_{((s,a),1)} = r(s,a)$

Notation

- Π matrix representation of policy π
- P transition matrix
- Value function $V^{\pi} = (I - \gamma \Pi P)^{-1} \Pi R$

- Policy induced state distribution

$$(I - \gamma \Pi P)^{-1}_{(s,s')} = \sum_{t=0}^{\infty} \gamma^t \left(\Pi P\right)^t_{(s,s')},$$
$$= \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s' | s_0 = s; \pi),$$

- We define $\Pi_s = (I - \gamma \Pi P)^{-1}$

Notation

- Π matrix representation of policy π
- P transition matrix
- Value function $V^\pi = \Pi_s \Pi R$

Notation

- Π matrix representation of policy π
- P transition matrix
- Value function $V^{\pi} = \Pi_s \Pi R$
- Policy return $J(\pi) = \mu^T V^{\pi}$ for initial state distribution matrix μ

Performance difference lemma

- $V^\pi - V^{\pi'} = \Pi_s \Pi A^{\pi'}$

Performance difference lemma

- $V^{\pi} - V^{\pi'} = \Pi_s \Pi A^{\pi'}$

- Value difference

$$
\begin{aligned}
V^{\pi} - V^{\pi'} &= \Pi\left(\mathcal{R} + \gamma P V^{\pi}\right) - V^{\pi'}, \\
&= \Pi\left(\mathcal{R} + \gamma P\left(V^{\pi} + V^{\pi'} - V^{\pi'}\right)\right) - V^{\pi'}, \\
&= \gamma \Pi P\left(V^{\pi} - V^{\pi'}\right) + \Pi\left(\mathcal{R} + \gamma P V^{\pi'}\right) - V^{\pi'} \\
&= \gamma \Pi P\left(V^{\pi} - V^{\pi'}\right) + \Pi Q^{\pi'} - V^{\pi'}, \\
&= \gamma \Pi P\left(V^{\pi} - V^{\pi'}\right) + \Pi A^{\pi'}, \\
&= \left(I - \gamma \Pi P\right)^{-1} \Pi A^{\pi'}.
\end{aligned}
$$

Performance difference lemma

- $$V^\pi - V^{\pi'} = \Pi_s \Pi A^{\pi'}$$
  $$= \Pi'_s \Pi A^{\pi'} + (\Pi_s - \Pi'_s)\, \Pi A^{\pi'}$$

Performance difference lemma

- $$V^{\pi} - V^{\pi'} = \Pi_s \Pi A^{\pi'}$$
  $$= \Pi'_s \Pi A^{\pi'} + (\Pi_s - \Pi'_s)$$

- State distribution difference

$$\begin{aligned}
\Pi_s - \Pi'_s &= \gamma \Pi P \Pi_s - \gamma \Pi' P \Pi'_s \\
&= \gamma \left(\Pi - \Pi' + \Pi'\right) P \Pi_s - \gamma \Pi' P \Pi'_s \\
&= \gamma \Pi' P (\Pi_s - \Pi'_s) + \gamma (\Pi - \Pi') P \Pi_s \\
&= \gamma \Pi'_s (\Pi - \Pi') P \Pi_s
\end{aligned}$$

Performance difference lemma

- $$\begin{aligned} V^{\pi} - V^{\pi'} &= \Pi_s \Pi A^{\pi'} \\ &= \Pi'_s \Pi A^{\pi'} + \left(\Pi_s - \Pi'_s\right) \Pi A^{\pi'} \\ &= \Pi'_s \Pi A^{\pi'} + \gamma \Pi'_s \left(\Pi - \Pi'\right) P \Pi_s \Pi A^{\pi'} \end{aligned}$$

Performance difference lemma

- $$\begin{aligned} V^{\pi} - V^{\pi'} &= \Pi_s \Pi A^{\pi'} \\ &= \Pi'_s \Pi A^{\pi'} + (\Pi_s - \Pi'_s)\, \Pi A^{\pi'} \\ &= \Pi'_s \Pi A^{\pi'} + \gamma \Pi'_s \left(\Pi - \Pi'\right) P \Pi_s \Pi A^{\pi'} \end{aligned}$$
- $$J^{\pi} - J^{\pi'} = \mu^T \Pi'_s \Pi A^{\pi'} + \gamma \mu^T \Pi'_s \left(\Pi - \Pi'\right) P \Pi_s \Pi A^{\pi'}$$
- Expressed policy return as a function of old advantage under old state distribution
  - + term small when new policy is close to old one

Lower bounding the policy return

- Previous expression contains Π_s which is hard to quantify
    - Prior work will mainly differ in bounding $||\mu^T \left(\Pi_s - \Pi'_s\right)||_\infty$

Lower bounding the policy return

- Previous expression contains Π_s which is hard to quantify
  - Prior work will mainly differ in bounding $||\mu^T \left(\Pi_s - \Pi'_s\right)||_\infty$
- CPI (Kakade et al. ICML02): $||\mu^T \left(\Pi_s - \Pi'_s\right)||_\infty \leq \frac{2\alpha\gamma}{(1-\gamma)^2}$
  - Where $\Pi = \alpha\Pi^g - (1-\alpha)\Pi'$ mixes previous policy with policy maximizing old advantage
  - Improvement of policy return can be guaranteed for small enough α

$$J^\pi - J^{\pi'} \geq \frac{\left(\mu^T \Pi'_s \Pi^g A^{\pi'}\right)^2}{8}$$

Lower bounding the policy return

- Previous expression contains Π_s which is hard to quantify
  - Prior work will mainly differ in bounding $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty$
- CPI (Kakade et al. ICML02): $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty \leq \dfrac{2\alpha\gamma}{(1-\gamma)^2}$
- USPI (Pirotta et al. ICML13): $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty \leq \dfrac{\gamma}{(1-\gamma)^2}||\Pi - \Pi'||_\infty$
  - $$\begin{aligned}\left\|\Pi - \Pi'\right\|_\infty &= \max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} |\pi(a|s) - \pi'(a|s)| \\ &= 2\max_{s\in\mathcal{S}} \mathrm{TV}(\pi(.|s) \parallel \pi'(.|s))\end{aligned}$$

Lower bounding the policy return

- Previous expression contains Π_s which is hard to quantify
  - Prior work will mainly differ in bounding $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty$
- CPI (Kakade et al. ICML02): $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty \leq \frac{2\alpha\gamma}{(1-\gamma)^2}$
- USPI (Pirotta et al. ICML13): $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty \leq \frac{2\gamma}{(1-\gamma)^2}\max_{s\in\mathcal{S}}\mathrm{TV}(\pi(.|s) \parallel \pi'(.|s))$
- TRPO (Schulman et al. ICML15): $||\mu^T\left(\Pi_s - \Pi'_s\right)||_\infty \leq \frac{2\gamma}{(1-\gamma)^2}\max_{s\in\mathcal{S}}\sqrt{\frac{1}{2}\mathrm{KL}(\pi(.|s) \parallel \pi'(.|s))}$
  - Pinsker's inequality: $\mathrm{TV} \leq \sqrt{\frac{1}{2}\mathrm{KL}}$

Lower bounding the policy return

- Previous expression contains Π_s which is hard to quantify
  - Prior work will mainly differ in bounding $||\mu^T\left(\Pi_s - \Pi_s'\right)||_\infty$
- CPI (Kakade et al. ICML02): $||\mu^T\left(\Pi_s - \Pi_s'\right)||_\infty \leq \dfrac{2\alpha\gamma}{(1-\gamma)^2}$
- USPI (Pirotta et al. ICML13): $||\mu^T\left(\Pi_s - \Pi_s'\right)||_\infty \leq \dfrac{2\gamma}{(1-\gamma)^2}\max_{s\in\mathcal{S}} \mathrm{TV}(\pi(.|s) \parallel \pi'(.|s))$
- TRPO (Schulman et al. ICML15): $||\mu^T\left(\Pi_s - \Pi_s'\right)||_\infty \leq \dfrac{2\gamma}{(1-\gamma)^2}\max_{s\in\mathcal{S}} \sqrt{\dfrac{1}{2}\mathrm{KL}(\pi(.|s) \parallel \pi'(.|s))}$
- CPO (Achiam et al. ICML17): $||\mu^T\left(\Pi_s - \Pi_s'\right)||_\infty \leq \dfrac{2\gamma}{(1-\gamma)^2}\mathbb{E}_{s\sim\pi'}\left[\mathrm{TV}(\pi(.|s) \parallel \pi'(.|s))\right]$

Lower bounds for MOTO's setting

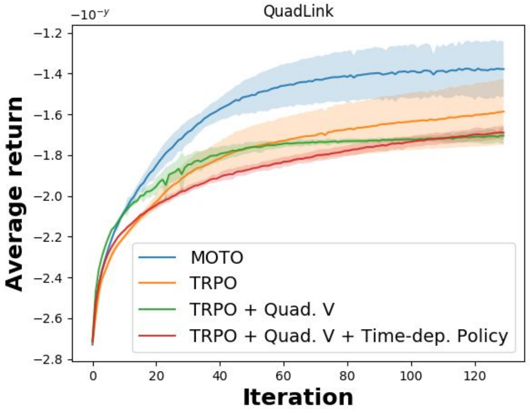
- Continuous state/action spaces
- We want to exploit specific model assumptions (Quad. Q and Gaussian state distributions)
- Same order of magnitude in terms of KL and discount in the lower bound
- As with prior work, lower bound too pessimistic to be used in practice
  - Despite using more specific assumptions on the MDP!

Model-Free Trajectory-based Policy Optimization with Monotonic Improvement; R. Akrour, A. Abdolmaleki, H. Abdulsamad, J. Peters, G. Neumann; JMLR18

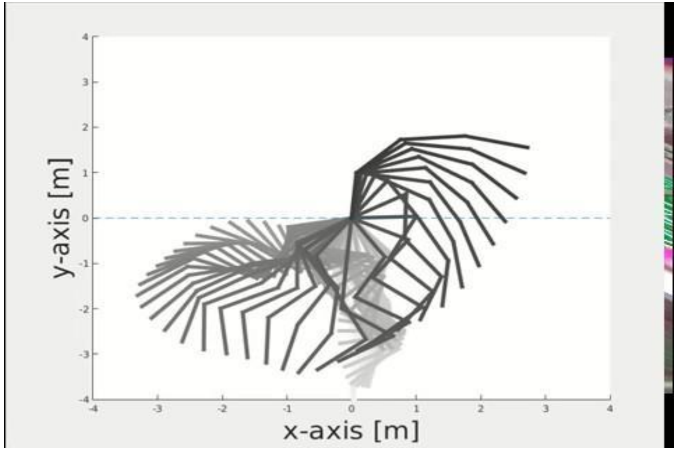
Perspectives

- Progress can be made when bounding $||(\Pi - \Pi')P||_\infty$
  - Current work: same actions lead to same states
    - If action not supported by previous policy = maximum penalty
  - Transition function assumed completely unknown
- Approach between optimal control and RL: P unknown but some of its property known
  - True for many physical systems (e.g. robots)
  - **Inject human knowledge** to obtain more practical predictions of policy return
    - Useful for **safety guarantees**

Comparison to deep RL



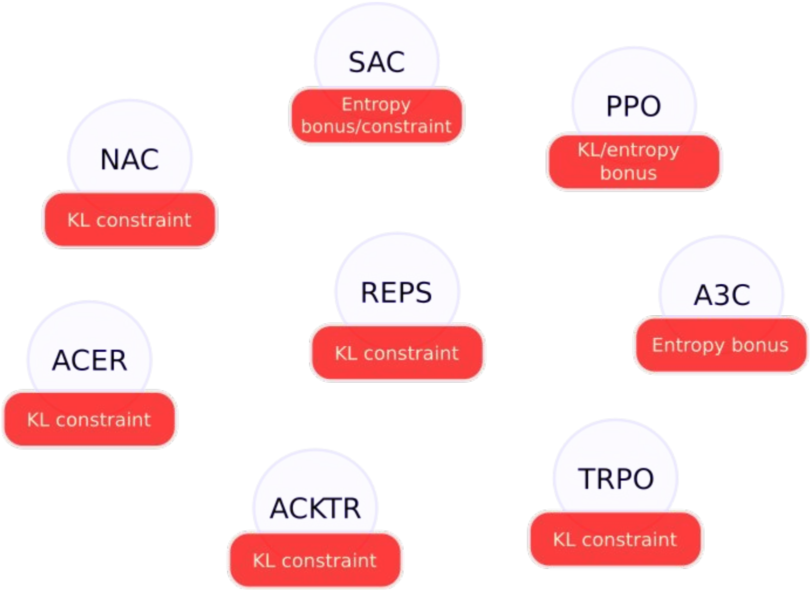
TRPO:
Trust Region Policy Optimization
J. Schulman, S. Levine, P. Moritz,
M. Jordan, P. Abbeel
ICML15



- Specialized models of MOTO work well on control problems
- What if the state is high dimensional?

Model-Free Trajectory-based Policy Optimization with Monotonic Improvement; R. Akrour, A. Abdolmaleki, H. Abdulsamad, J. Peters, G. Neumann; JMLR18

Entropy regularization in (deep) RL



- Can we transfer anything from the simpler setting of MOTO to deep RL?

MOTO's closed form solutions

- New policy $\pi_k^t(a|s) = (a|FLs + Ff, F(\eta^* + \omega^*)),$
  - $F = (\eta^* \Sigma_t^{-1} - Q_{aa})^{-1},$ $L = \eta^* \Sigma_t^{-1} K_t + Q_{as},$
    $f = \eta^* \Sigma_t^{-1} k_t + q_a.$

Old policy

$$\pi_{k-1}^t(a|s) \sim (K_t s + k_t, \Sigma_t)$$

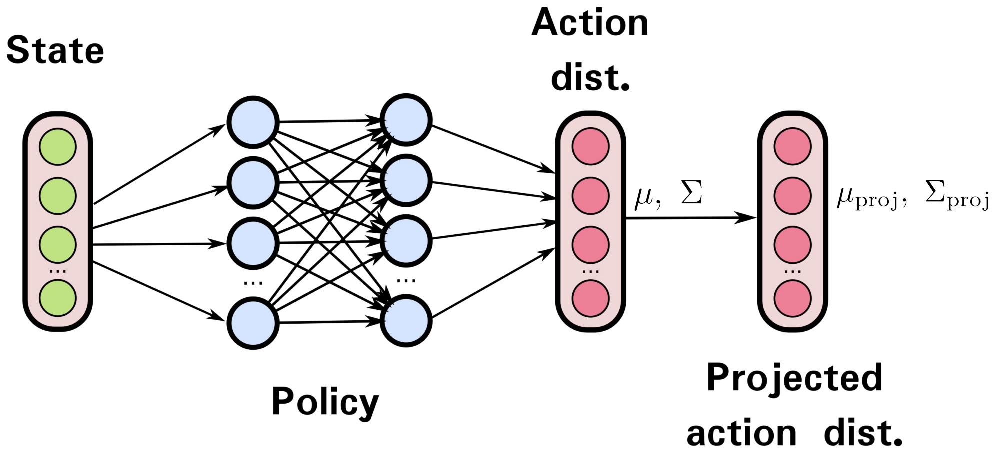
Q function

$$\hat{Q}_{k-1}^t(s,a) = \frac{1}{2} a^T Q_{aa} a + a^T Q_{as} s + a^T q_a + q(s)$$

MOTO's closed form solutions

- New policy $\pi_k^t(a|s) = (a|FLs + Ff, F(\eta^* + \omega^*)),$
  - $F = (\eta^* \Sigma_t^{-1} - Q_{aa})^{-1}, \qquad L = \eta^* \Sigma_t^{-1} K_t + Q_{as},$
    $f = \eta^* \Sigma_t^{-1} k_t + q_a.$
- Dual param. entropy cst. ω^*: scale covariance matrix
- Dual param. KL-divergence cst. η^*: interpolate params of policy and Q function
- Given an input covariance, can we compute in closed form the scaling satisfying an entropy constraint?

Entropy projections



- Solve for ω, $\mathcal{H}\left(\mathcal{N}\left(\mu, \omega\Sigma\right)\right) = \beta$
- Solve for η, $\mathrm{KL}\left(\mathcal{N}\left(\eta\mu + (1-\eta)\mu_{k-1}, \eta\Sigma + (1-\eta)\Sigma_{k-1}\right) | \mathcal{N}\left(\mu_{k-1}, \Sigma_{k-1}\right)\right) = \epsilon$
- Closed form and differentiable solution => Use gradient descent to solve the policy update

Entropy projections

- **Can solve** the entropy eq. in closed form
- **No closed** form solution for the KL (would require solving x + log x = a)
  - Replace with an upper bound of the KL, quadratic in η
  - Similar approach for entropy/KL of soft-max distribution (discrete action space)

Projections for Approximate Policy Iteration Algorithms; R. Akrour, J. Pajarinen, G. Neumann, J. Peters; ICML19

Entropy projections

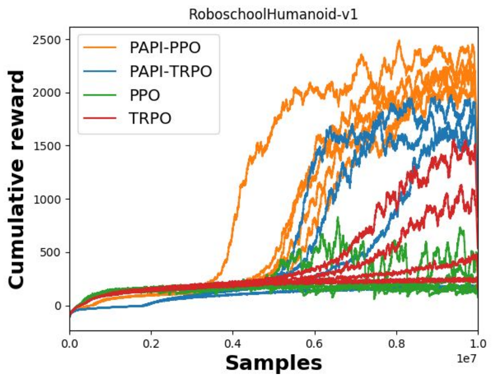
- **Can solve** the entropy eq. in closed form
- **No closed** form solution for the KL (would require solving x
  - Replace with an upper bound of the KL, quadrati
  - Similar approach for entropy/KL of soft-max

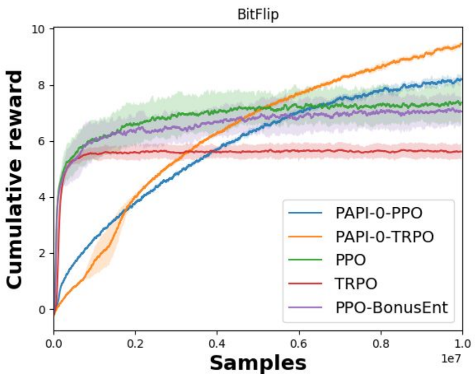
Algorithm 2 API linear-Gaussian policy projection

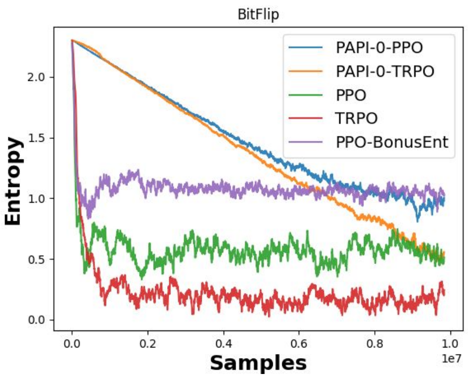
Input: A', λ, $\lambda_{\text{off\_diag}}$, $q(.|s) = (A_q^T \psi_q(s), \Sigma_q)$, A^T, ψ, ϵ and β
Output: $\pi(.|s) = \mathcal{N}(A'^T \psi(s), \Sigma)$ complying with KL (6) and entropy (7) constraints
Σ = Entropy_projection(λ, $\lambda_{\text{off\_diag}}$, β)
if $\mathbb{E}_s \mathrm{KL}(\mathcal{N}(A'^T \psi(s), \Sigma) \parallel q(.|s)) > \epsilon$ **then**
$\quad \eta_g = \frac{\epsilon - m_q(A)}{m_q(A') + r_q(\Sigma) + e_q(\Sigma)}$
$\quad \Sigma = \eta_g \Sigma + (1 - \eta_g)\Sigma_q$
end if
if $\mathbb{E}_s \mathrm{KL}(\mathcal{N}(A'^T \psi(s), \Sigma) \parallel q(.|s)) > \epsilon$ **then**
$\quad a = .5\mathbb{E}_s ||A'^T \psi(s) - A^T \psi(s)||^2_{\Sigma_q^{-1}}$
$\quad b = .5\mathbb{E}_s[(A'^T \psi(s) - A^T \psi(s))^T \Sigma_q^{-1}(A^T \psi(s) - A_q^T \psi_q(s))]$
$\quad c = m_q(A) + r_q(\Sigma) + e_q(\Sigma) - \epsilon$
$\quad \eta_m = \frac{-b + \sqrt{b^2 - ac}}{a}$
$\quad A' = \eta_m A' + (1 - \eta_m)A$
end if

Projections for Approximate Policy Iteration Algorithms; R. Akrour, J. Pajarinen,

Results





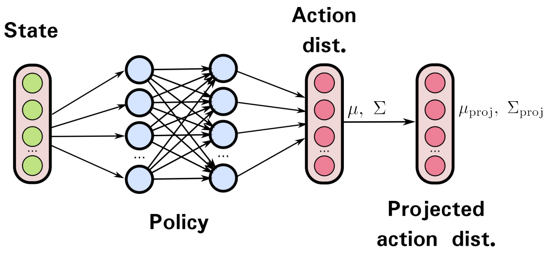


- **Pros**: entropy constraint (Gaussian and soft-max) very **easy to implement and offers large gains**
- **Caveat**: KL requires to store previous policies or only optimize last layer of the neural network

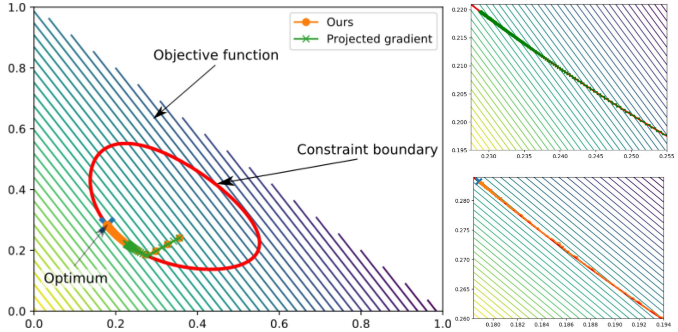
Projections for Approximate Policy Iteration Algorithms; R. Akrour, J. Pajarinen, G. Neumann, J. Peters; ICML19

Projection...?

- **General algorithm**: upper bound KL/entropy with a function ‘simple’ in an interpolation parameter
  - Solve for the interpolation parameter and return a distribution complying with cst.
- Is this a projection??
- What is the ‘error’ of the projection?
- Will gradient descent converge to a reasonable solution?



Convergence on toy problem



- Problem $\max_{p \in \text{Simplex}} \quad f(p) = c^T p$

$$s.t. \quad \mathcal{H}(p) \geq \beta$$

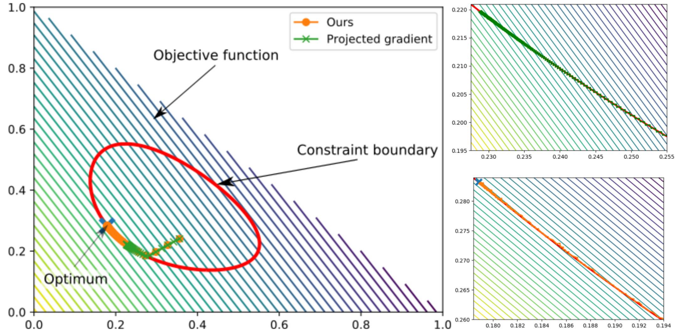
- Projection $g(p)$: solve for η

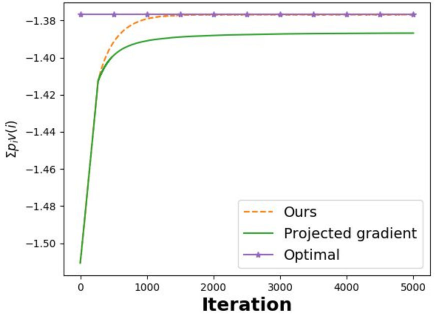
$$\mathcal{H}(\eta p + (1-\eta)\text{Unif.}) \geq \eta \mathcal{H}(p) + (1-\eta)\mathcal{H}(\text{Unif.}) = \beta$$

$$\eta = \frac{\log |\mathcal{A}| - \beta}{\log |\mathcal{A}| - \mathcal{H}(p)}$$

Projections for Approximate Policy Iteration Algorithms; R. Akrour, J. Pajarinen, G. Neumann, J. Peters; ICML19

Convergence on toy problem





- Problem $\max_{p \in \text{Simplex}} \quad f(p) = c^T p$
$$s.t. \quad \mathcal{H}(p) \geq \beta$$
- Projection $g(p)$: solve for η

$$\mathcal{H}(\eta p + (1-\eta)\text{Unif.}) \geq \eta \mathcal{H}(p) + (1-\eta)\mathcal{H}(\text{Unif.}) = \beta$$

$$\eta = \frac{\log |\mathcal{A}| - \beta}{\log |\mathcal{A}| - \mathcal{H}(p)}$$

GD on composition $p_{k+1} = p_k - \alpha \nabla f \circ g(p_k)$
- converges

PGD $p_{k+1} = g(p_k - \alpha \nabla f(p_k))$
- No convergence because not an orthogonal projection

Projections for Approximate Policy Iteration Algorithms; R. Akrour, J. Pajarinen, G. Neumann, J. Peters; ICML19

Convergence in a convex optimization setting

- Problem: $$\min_{x \in \mathbb{R}^d} \quad f(x) = c^T x, \qquad \text{s.t.} \quad h(x) \le 0.$$

- Projection: $g(x) = \begin{cases} x & \text{if } h(x) \le 0, \\ \eta_x x + (1 - \eta_x) x_0 & \text{else,} \end{cases}$ with $\eta_x = \frac{h(x_0)}{h(x_0) - h(x)}$ and $h(x_0) < 0$

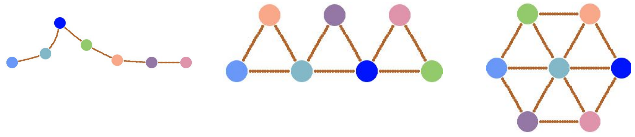
- Algorithm: $x_{k+1} = x_k - \alpha \nabla f \circ g(x_k)$, with gradient $\nabla f \circ g(x_k) = \eta_k \left(\nabla f(g(x_k)) + \frac{\nabla f(g(x_k))^T (g(x_k) - x_0)}{h(x_0)} \nabla h(x_k) \right)$

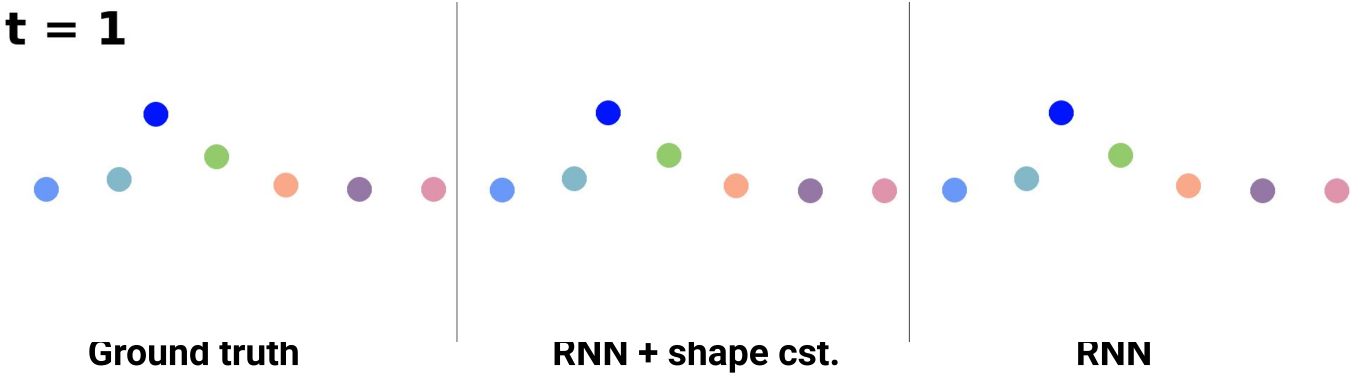
- Convergence rate $\mathcal{O}(\frac{1}{\sqrt{K}})$ (interpolation is non-smooth)

Convex Optimization with an Interpolation-based Projection and its Application to Deep Learning; R. Akrour, A. Atamna, J. Peters; MACH (submitted)

Applications of the interpolation projection

- Applications: RL, inductive bias for learning dynamics, …



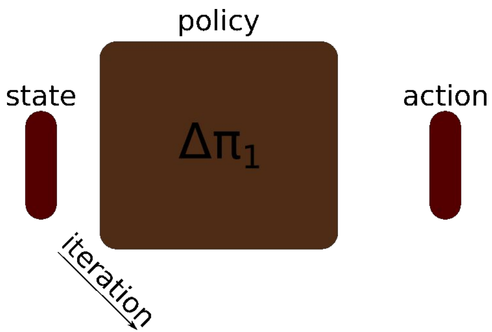


- About x200 faster than orthogonal projection layers (**Differentiable Convex Optimization Layers;** Agrawal et al.; NeurIPS19)

Convex Optimization with an Interpolation-based Projection and its Application to Deep Learning; R. Akrour, A. Atamna, J. Peters; MACH (submitted)

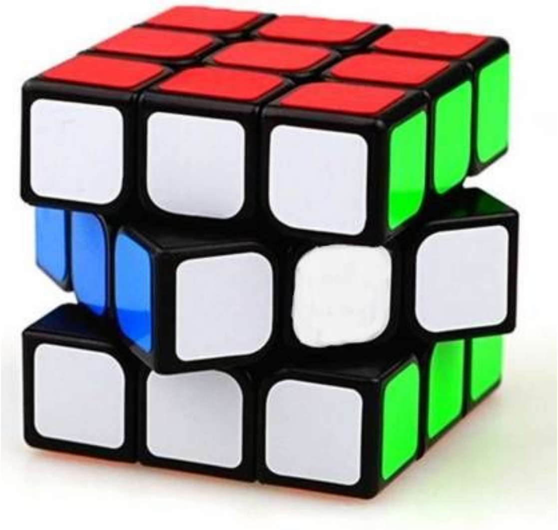
Perspectives

- Non i.i.d. RL setting alternates data collection and model update
  - Very important to **update the policy gracefully**
    - Doable with linear policies, trees (MCTS), not so much with NNs
- Updating all NN params. at once **not reasonable**?
- **Incremental** construction of the policy
  - Stack smaller **policy 'delta'** networks at each update
    - Easy to control KL/TV
  - How to forget?



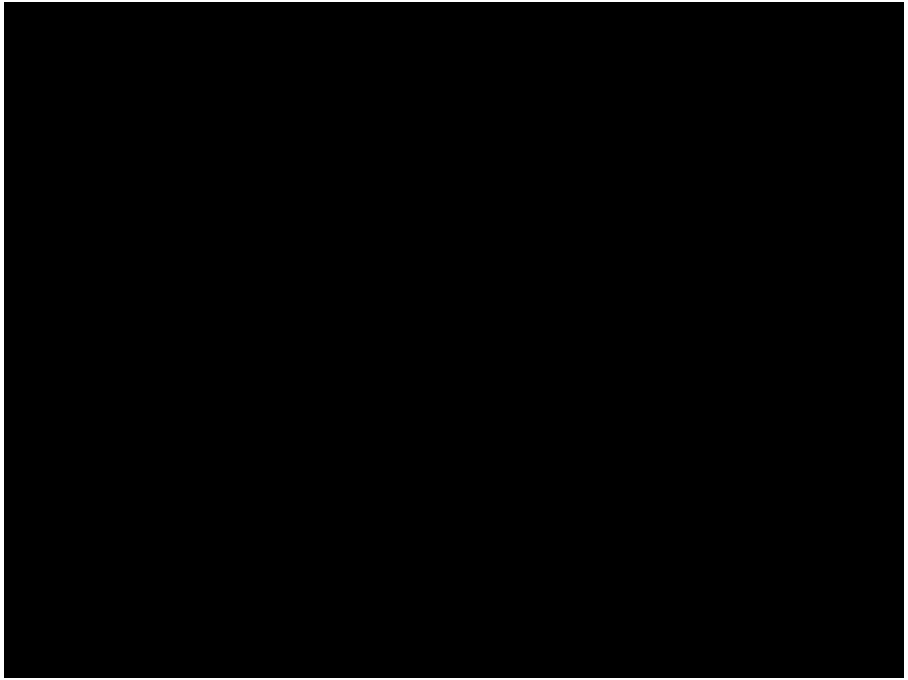
Examples of explainable RL policies

- **Example of an interpretable policy**: solving the Rubik's cube
- **Policy structure**: IF State similar to Prototype DO Action
- Can we extract similar policies for other decision problems?



Limits of differentiable programming?

- IF State similar to Prototype DO Action
- **Naive approach**: differentiable policy with prototypical states = basis functions
- Hard to relate cluster centers and behavior

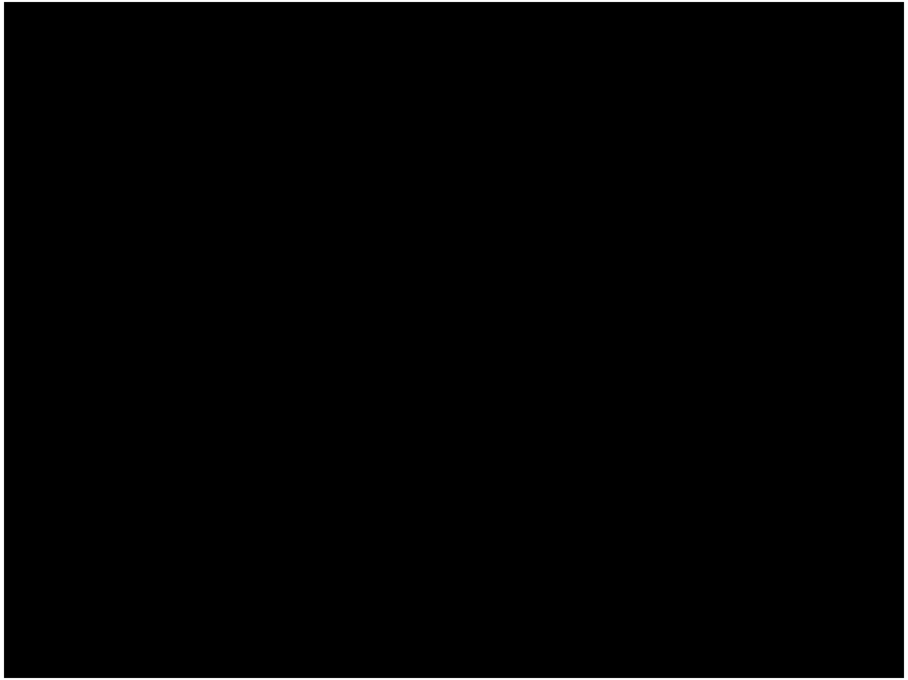


Proposed solution

- Stay in the **manifold** of (interpretable) states

- **Pick prototypes** from trajectory data (non-differentiable operation)

- Mixture of **discrete optimization** (search heuristics) and **continuous optimization** (gradient descent + interpolation projections)



Reinforcement Learning from a Mixture of Interpretable Experts; R. Akrour, D. Tateo, J. Peters; TPAMI (submitted)

Perspectives

- **Limitation**: similarity to prototype based on Euclidean distance
  - How to scale to more complex inputs and keep the similarity function interpretable?

Perspective 1: Ensure temporal coherency

- Higher similarity between states occurring closely after each other in the MDP

Perspective 2: Semantic understanding of the state (e.g. vision) + state factorization + simple distance