

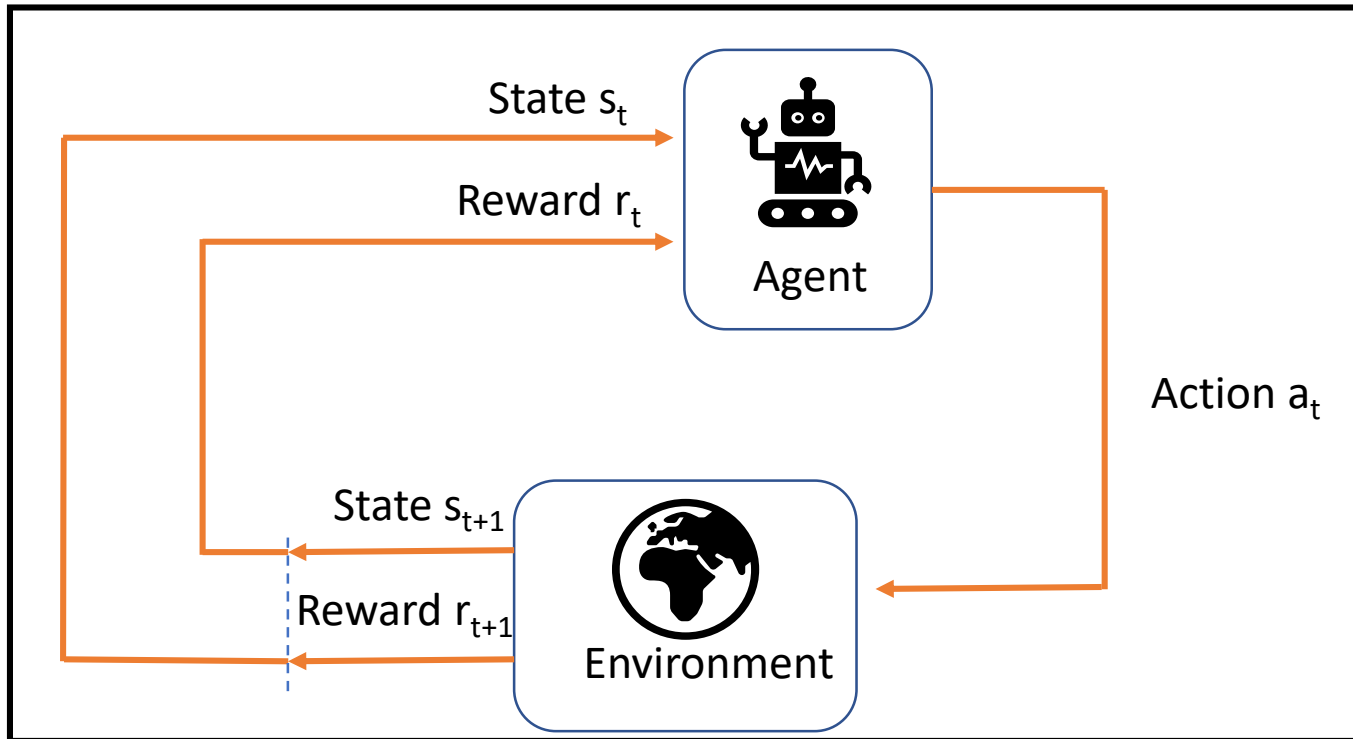


CAMEO: Curiosity Augmented Metropolis for Exploratory Optimal Policies

*Mohamed Alami Chehboune,^(1, 2)
Fernando Llorente,⁽³⁾
Rim Kaddah,⁽²⁾
Luca Martino,⁽⁴⁾
Jesse Read⁽¹⁾*

- (1) LIX, Ecole Polytechnique, IP-Paris
- (2) IRT SystemX
- (3) Universidad Carlos III de Madrid
- (4) Universidad Rey Juan Carlos

Optimal Policies in Reinforcement Learning



$$G = \mathbb{E}[r_0 + \gamma r_1 + \gamma^2 r_2 + \dots \gamma^t r_t + \dots | s_0]$$



Converges towards an optimal policy π^* maximizing the return G

Contribution

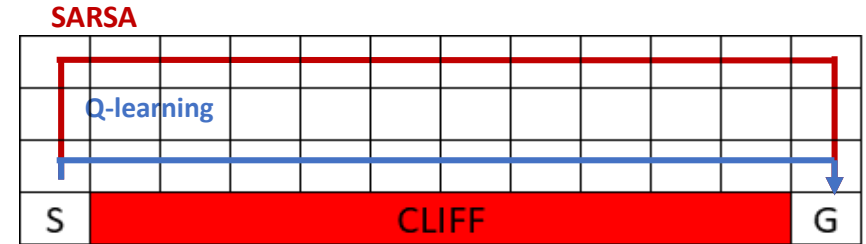
- There is no guarantee that π^* is actually unique
- Different policies can correspond to different behaviours with different risk profiles

➤ There should exist a distribution of policies

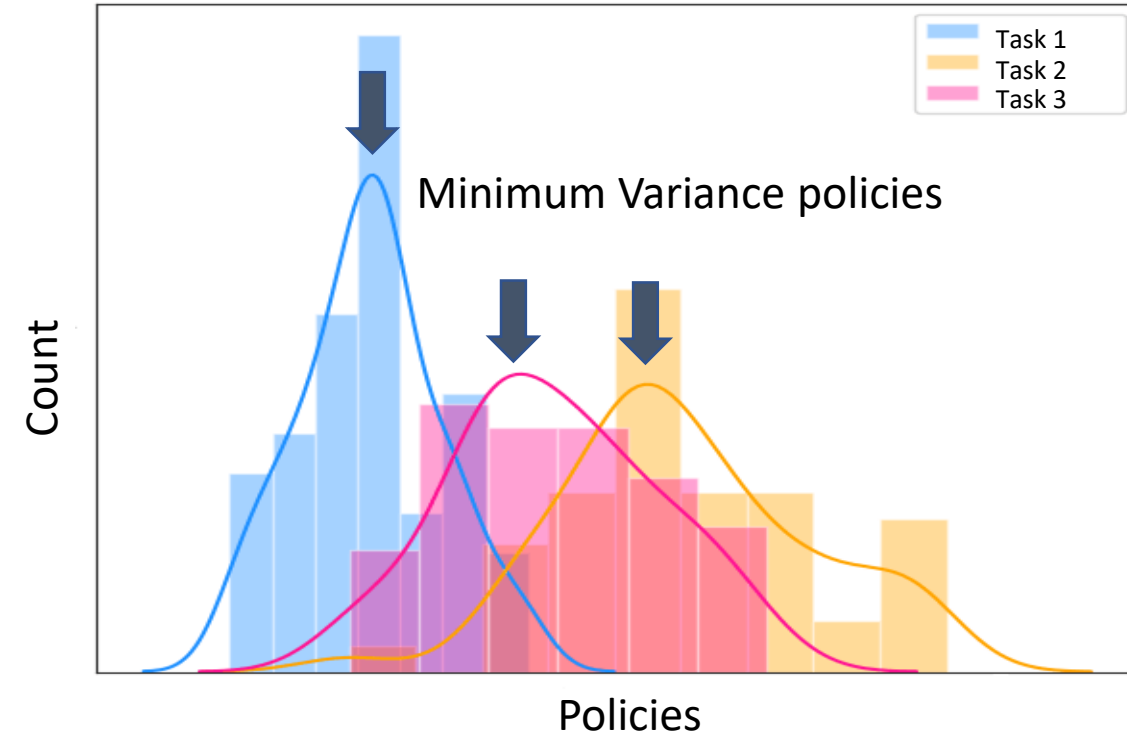
Learning such a distribution would allow to sample different optimal policies and choose the one which profile suits the best the task needs.

To learn the policies distribution, it is necessary to sample different policies with different behaviours on the fly.

We propose a Metropolis based method to generate optimal policies while using a curiosity mechanism to ensure that the optimal policies generated adopt different behaviours



Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An introduction, The MIT Press, second edition 2018



Metropolis Algorithm Principle

The objective: MH algorithm is a MCMC method which aim is to generate samples from a distribution when direct sampling is difficult or not feasible

Given a distribution f in Ω the algorithm defines a markov chain which stationary distribution is f .

- Only requires to know a distribution that is proportional to f
- Used for sampling from multi-dimensional distributions, in particular when the number of dimensions is high

Let $h \propto f$. We choose an arbitrary x_0 as a starting point and a transition distribution $g(x|x_0)$. At each iteration:

- $x \sim g(\cdot | x_i)$ and calculate the acceptance rate: $\alpha = \frac{h(x)g(x_i, x)}{h(x_i)g(x, x_i)}$
- As h is proportional to f : $\alpha = \frac{h(x)g(x_i, x)}{h(x_i)g(x, x_i)} = \frac{f(x)g(x_i, x)}{f(x_i)g(x, x_i)}$
- We can choose g invertible such as: $g(x, x_i) = g(x_i, x)$

$$\alpha = \frac{h(x)}{h(x_i)} = \frac{f(x)}{f(x_i)}$$

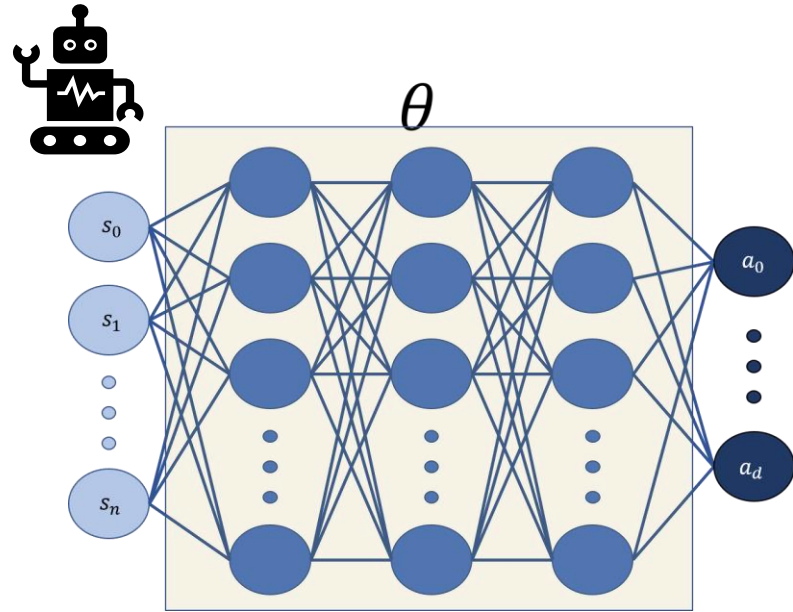
$$u \sim U[0,1]$$



If $\alpha \geq u$, $x_{i+1} = x$, otherwise $x_{i+1} = x_i$

$x_n, x_{n+1}, \dots, x_{n+m} \sim f$

First Adaptation to Reinforcement Learning



Let θ the parameters of a policy, corresponding to the parameters of a neural network which input is the state and outputs an action.

Let π_θ a policy parameterized by θ and $f(\theta)$ the target distribution.

Safe hypothesis: Given a policy:

- The higher its mean return the more probable it is that it is an optimal policy
- The lower its variance, the more probable it is that it is a mode of the distribution

Let U a positive monotonically increasing function of the return that is maximal when the return is maximal, we have:

$$f(\theta) \propto U$$

$$U(\tau) = \exp(\tilde{G}(\tau)/T)$$

τ : a trajectory

G : the mean return

T : a temperature parameter

First Adaptation to Reinforcement Learning

$f(\theta) \propto p(\theta)\eta(\theta)$ where :

- $p(\theta)$ is a prior over θ (uniform in the simple case)
- $\eta(\theta)$ the performance depending on U

$$\eta(\theta) = \mathbb{E}_{p(\tau|\theta)}[U(\tau)] = \int U(\tau)p(\tau|\theta)d\tau$$

To estimate $\eta(\theta)$, we sample several trajectories:



$$\begin{aligned}\eta(\theta) &= \mathbb{E}_{p(\tau|\theta)}[U(\tau)] \\ &\approx \bar{U}_N(\theta) = \frac{1}{N} \sum_{i=1}^N U(\tau_i), \quad \tau_i \sim p(\tau|\theta),\end{aligned}$$

Algorithm 1 Monte Carlo-within-Metropolis for RL

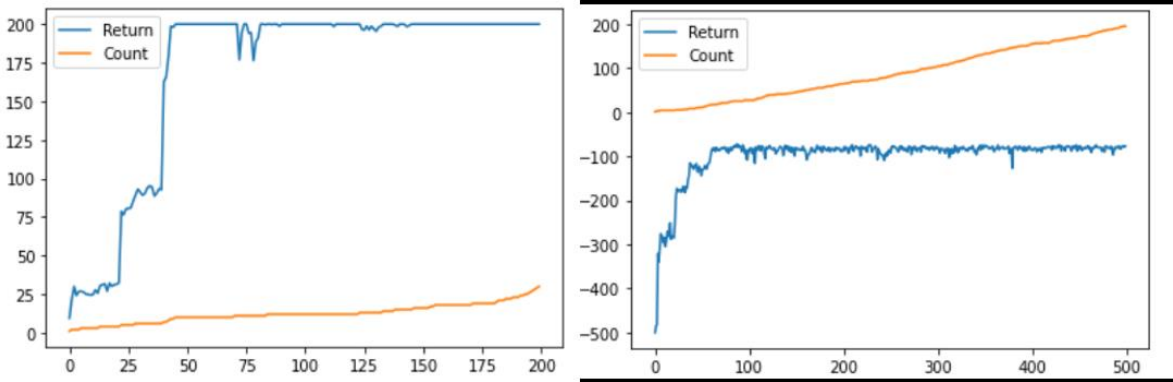
Require: K : the number of iterations, N : number of episodes

Initialise Agent π

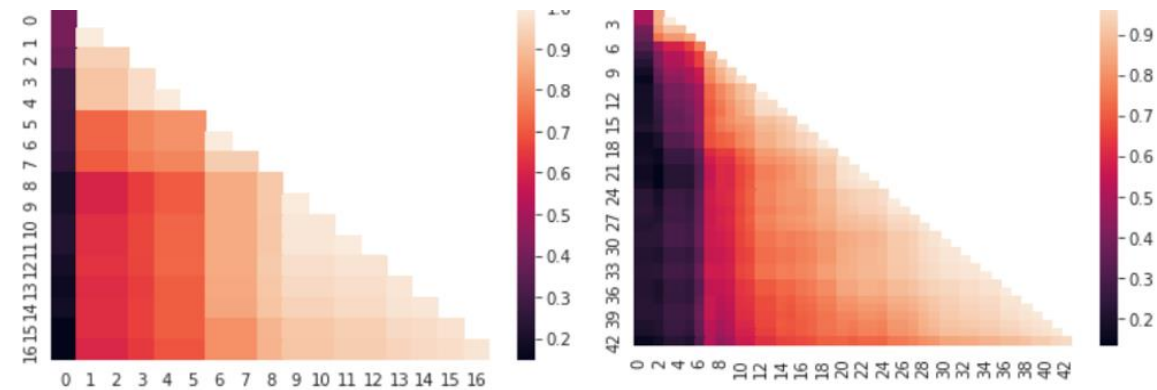
First Results

Environments	Cartpole	Acrobot
Snapshot		
State Space $s_t \in$	\mathbb{R}^4	$[-1, 1]^4 \times [-4\pi, 4\pi] \times [-9\pi, 9\pi]$
Action Space $a_t \in$	$\{0, 1\}$	$\{0, 1, 2\}$
Reward $r_t =$	+1 per time step	-1 per time step

- Continuous State space
- Discrete Action space



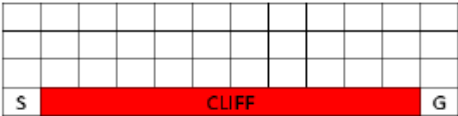
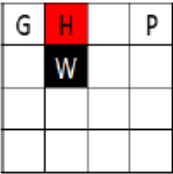
Average return for every new θ (in blue) and incremental count of the number of θ_i retained (in orange) on Cartpole (left) and Acrobot (right) using simple implementation.



Cosine similarity between pairs of retained θ_i on Cartpole (left) and Acrobot (right) using simple implementation.

- Succeeding θ_i are heavily correlated, this hints at the fact that the behaviours may be identical

First Results

Environments	Cliff	Gridworld
Snapshot		
State Space $s_t \in$	$\{1, \dots, 48\}$	$\{1, \dots, 16\}$
Action Space $a_t \in$	$\{0, 1, 2, 3\}$	$\{0, 1, 2, 3\}$
Reward $r_t =$	-1 per move, 10 for the goal and -10 for the pit	-1 per move, 10 for the goal and -10 for the pit

- the simple approach fails when confronted to Gridworld or Cliff. In both cases, the agent remains stuck, always performing the same action

CAMEO (prior and bootstrap)

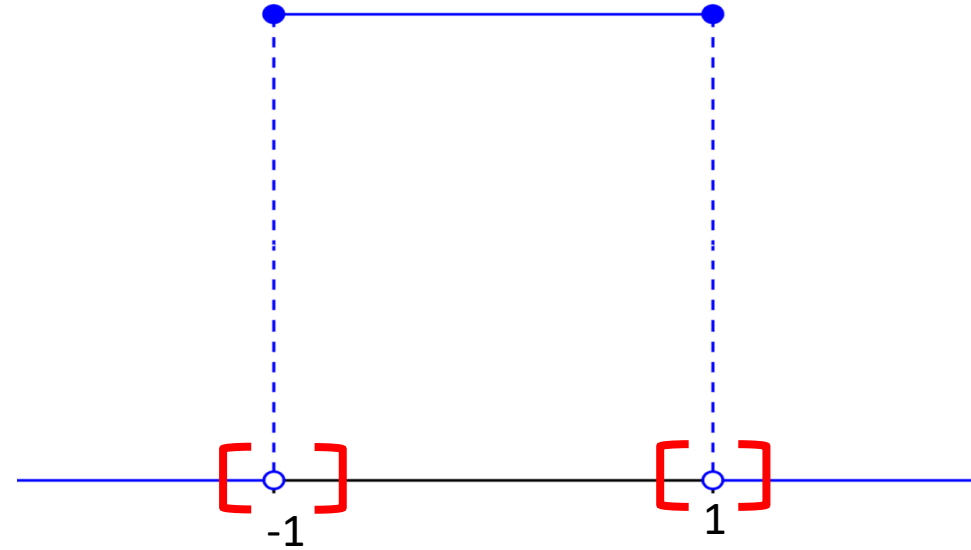
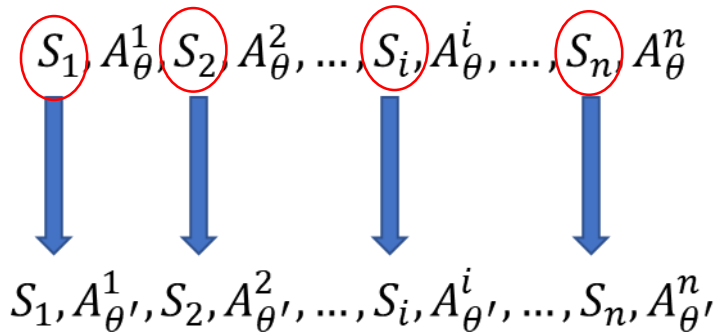
Prior:

We considered that $p(\theta) \sim U[-a, a]$

$$\beta \leftarrow \frac{\cancel{p(\theta')} \bar{U}_N(\theta')}{\cancel{p(\theta_{k-1})} \bar{U}_N(\theta_{k-1})}$$

➔ The priors cancel each other and therefore θ_i are not bounded.

$$\theta_{\{k+1\}} = \lambda \theta_k \text{ With } \lambda \text{ a constant}$$



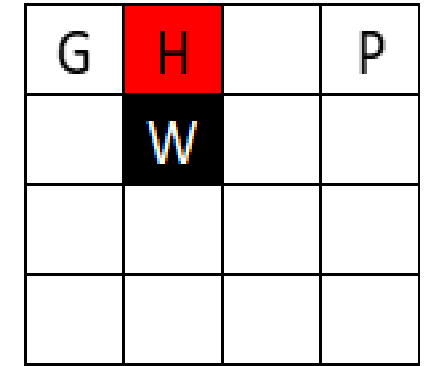
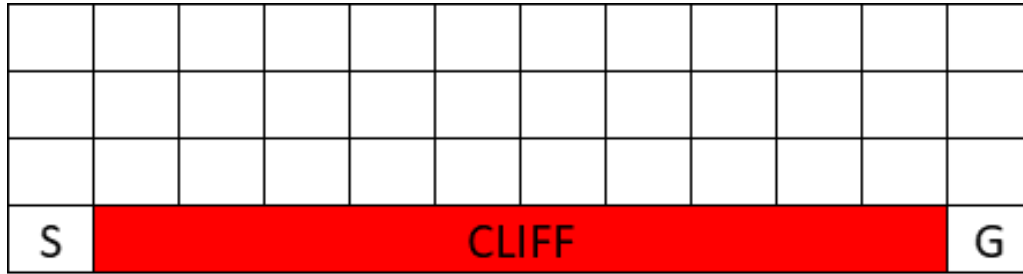
Trajectory Bootstrap:

Goal: Avoid running N episodes twice to estimate the returns

Advantage function: $A_{\theta}(s, a) = Q_{\theta}(s, a) - V_{\theta}(s)$

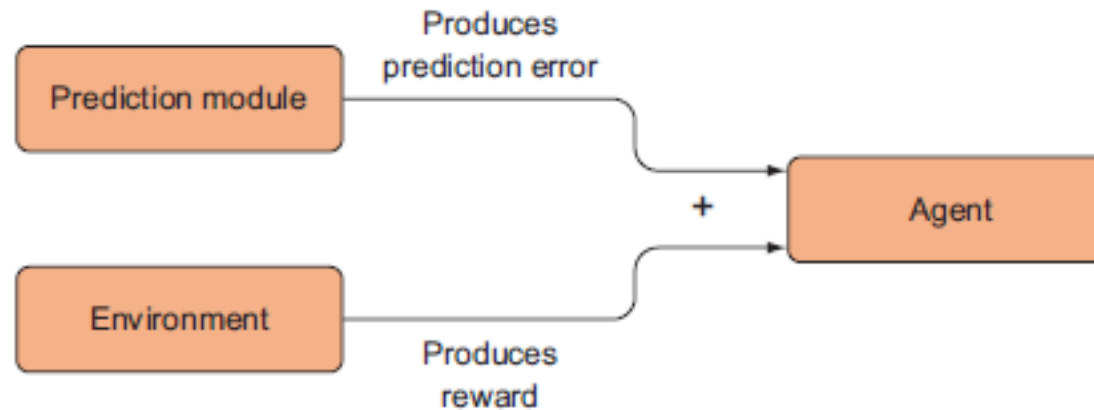
$$\begin{aligned} G(\theta') &= G(\theta_{k-1}) + \mathbb{E}_{\theta'} \left[\sum \gamma^t A_{\theta'}(s_t, a_t) \right] \\ &\approx G(\theta_{k-1}) \\ &\quad + \mathbb{E}_{s, a \sim \rho(\theta_{k-1}), \pi_{\theta_{k-1}}} \left[\underbrace{\Pi(r + \gamma(V_{\pi_{\theta'}}(s') - V_{\pi_{\theta_{k-1}}}(s)))}_{\text{TD Error}} \right], \end{aligned}$$

Sparse Rewards in RL

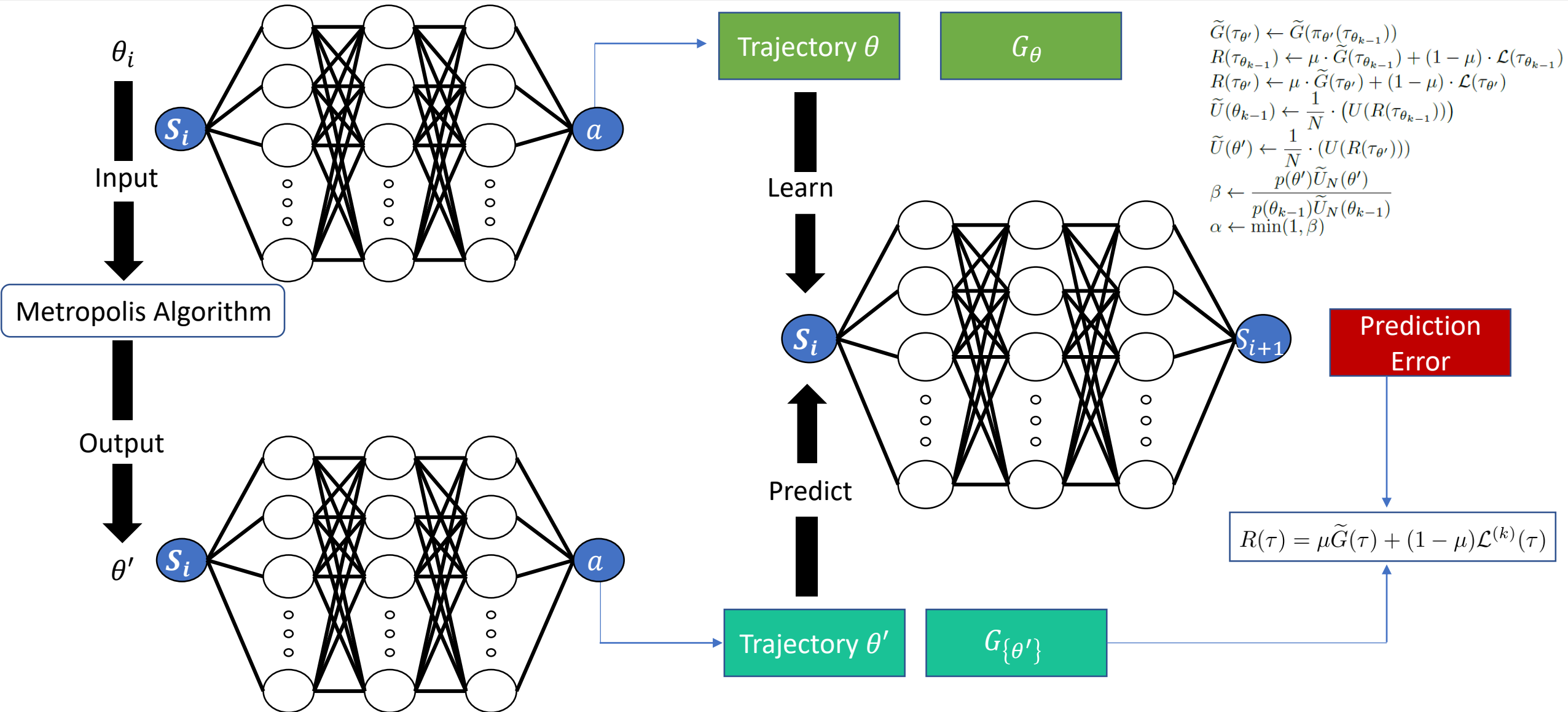


The only areas with positive reward are the final states. All other areas (excluding the holes) have reward -1

➡ The reward is not informative on the agent's progression



CAMEO (Curiosity Module)



CAMEO (Results)

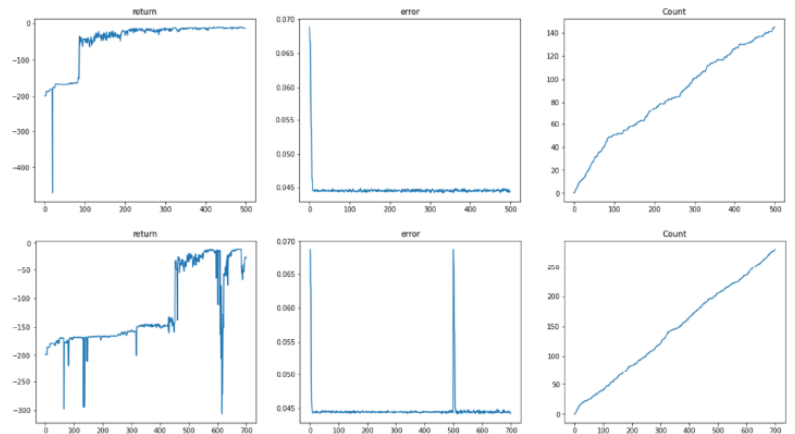


Fig. 3: CAMEO Results on Cliff (above) and Gridworld (below). The figure presents the mean return, the Prediction error and the count of θ_i retained over time steps

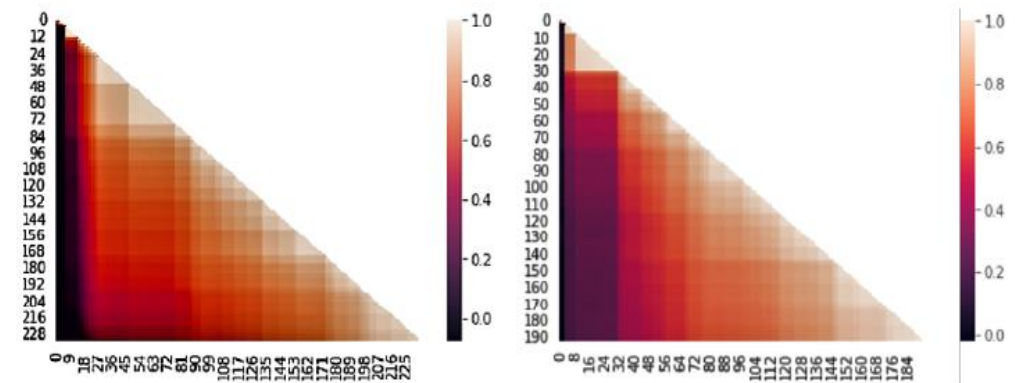


Fig. 4: Cosine similarity between pairs of retained θ_i on Gridworld (left) and Cliff (right) using CAMEO implementation.

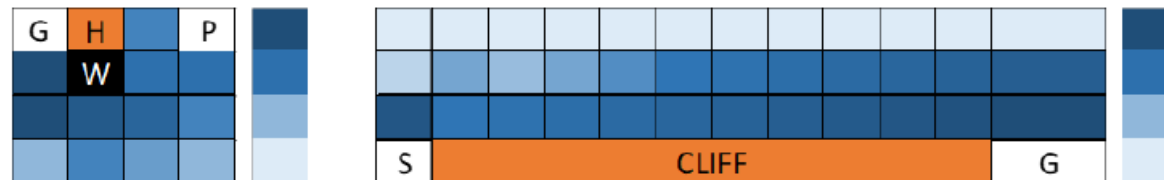


Fig. 5: State visitation frequency aggregated on 100 policies obtained using CAMEO on Gridworld and Cliff. Less visited states are in light blue and most visited ones in dark shade

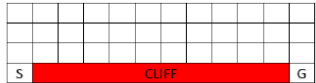


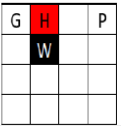
Conclusion and perspectives

Take home messages:

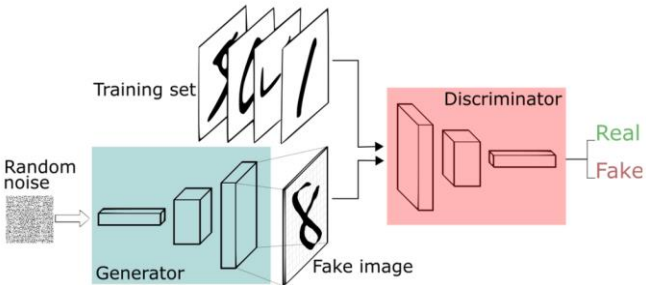
- We were able to generate several optimal policies with different behaviours on the fly
- First mandatory step towards learning the distribution of optimal policies itself

Perspectives:

- But only works for environments with discrete action spaces

Environments	Cliff	Cartpole	Acrobot	Gridworld
Snapshot				
State Space $s_t \in$	$\{1, \dots, 48\}$	\mathbb{R}^4	$[-1, 1]^4 \times [-4\pi, 4\pi] \times [-9\pi, 9\pi]$	$\{1, \dots, 16\}$
Action Space $a_t \in$	$\{0, 1, 2, 3\}$	$\{0, 1\}$	$\{0, 1, 2\}$	$\{0, 1, 2, 3\}$
Reward $r_t =$	-1 per move, 10 for the goal and -10 for the pit	+1 per time step	-1 per time step	-1 per move, 10 for the goal and -10 for the pit

- It is possible to replace the standard proposal distribution with a policy guided proposal that draws educated samples. The new distribution could therefore explore larger spaces by focusing on areas of interest
- Study the theoretic foudations explaining the similarity with GANs in order to use them to learn the distribution and generate optimal policies





THANK YOU FOR YOUR ATTENTION

ACKNOWLEDGMENTS & FUNDINGS



Authors details



Mohamed ALAMI CHEHBOUNE

PhD Candidate

Ecole Polytechnique/IRT SystemX

Mohamed.alami-chehboune@polytechnique.edu



Fernando Llorente

PhD Candidate

Universidad Carlos III de Madrid

felloren@est-econ.uc3m.es



Rim kaddah, PhD

Project Manager

IRT SystemX

Rim.kaddah@irt-systemx.fr



Luca Martino, PhD

Associate Professor

Universidad Rey Juan Carlos

luca.martino@urjc.es



Jesse read, PhD

Professor

Ecole Polytechnique

Jesse.read@polytechnique.edu