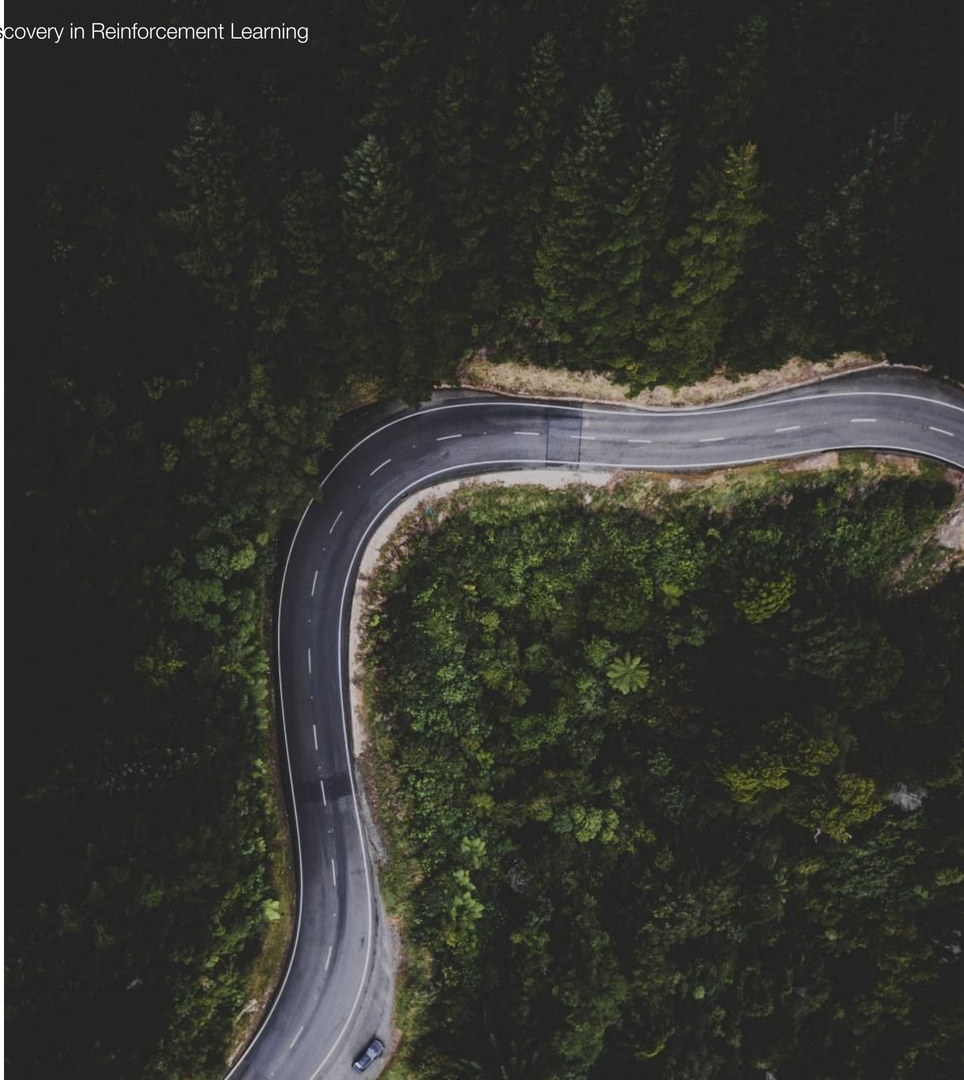
Representation-driven Option Discovery in Reinforcement Learning

Marlos C. Machado

Sharif University of Technology, June 6, 2025

This is a research program that touches upon a lot of what you've seen in the other guest lectures



RL is now commonly deployed in the real-world

RL is now commonly deployed in the real-world

Video compression

[Mandhane et al.,2022]

Matrix multiplication

[Fawzi et al., 2022]

Hardware design

[Mirhoseini et al., 2021]

Cooling systems

[Luo et al., 2022]

Thermal power generators

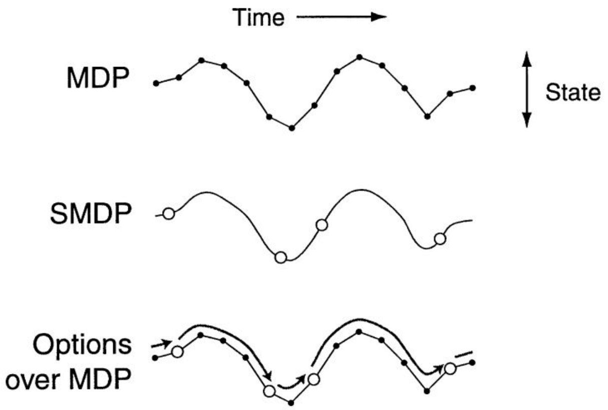
[Zhan et al., 2022]

Managing inventories

[Madekaet al., 2022]

...

Temporal abstraction – Options [Sutton, Precup, & Singh, AIJ 1999]



$$\omega = \langle \mathcal{I}_\omega, \pi_\omega, \beta_\omega \rangle \tag{1}$$

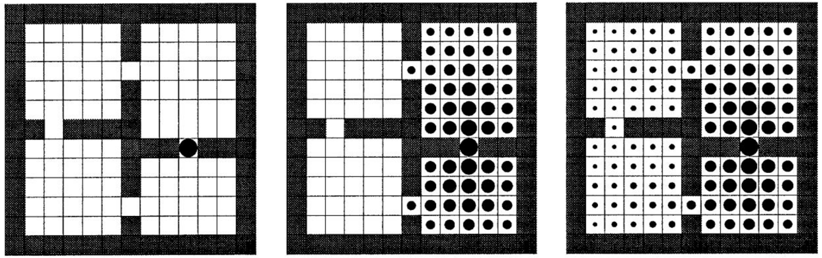
initiation set

policy

termination condition

The many use cases of options

Faster credit assignment / planning:



[Sutton, Precup, & Singh. Artif. Intelligence 1999]

Exploration [Jong, Hester, & Stone, AAMAS 2008]

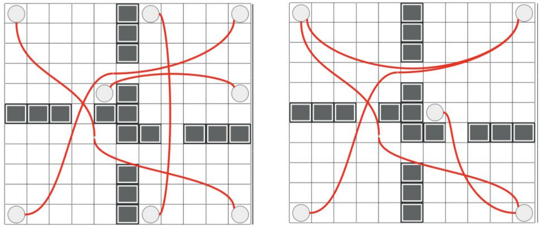
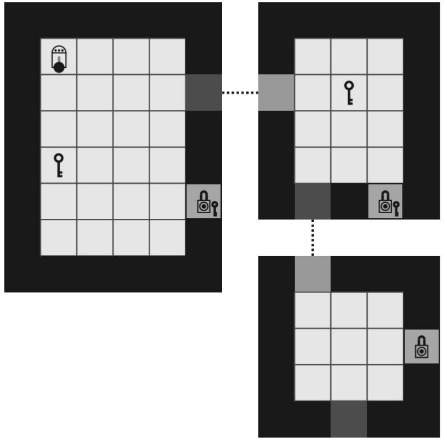


Figure by Jinnai et al. (2019)

Transfer learning:

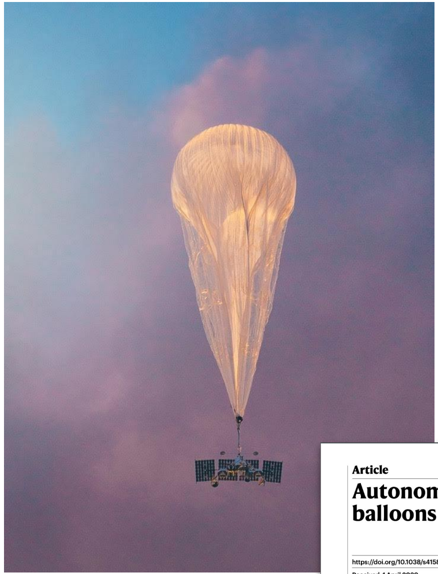


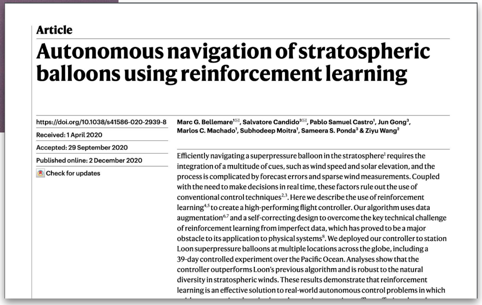
[Konidaris & Barto, IJCAI 2007]

and more…

It works! RL in the real world

https://www.nature.com/articles/s41586-020-2939-8





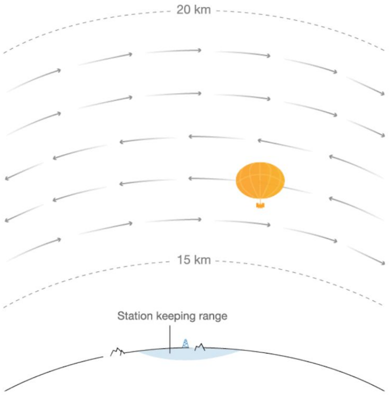
Article

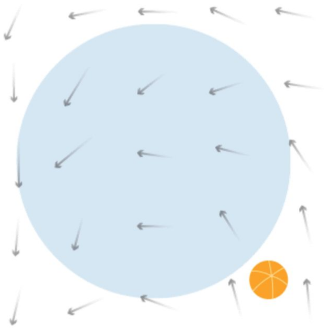
Autonomous navigation of stratospheric balloons using reinforcement learning

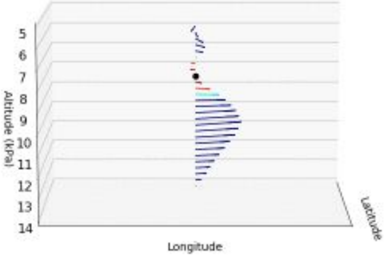
https://doi.org/10.1038/s41586-020-2939-8
Received: 1 April 2020
Accepted: 29 September 2020
Published online: 2 December 2020
Check for updates

Marc G. Bellemare[1], Salvatore Candido[3], Pablo Samuel Castro[1], Jun Gong[3], Marlos C. Machado[1], Subhodeep Moitra[1], Sameera S. Ponda[3] & Ziyu Wang[2]

Efficiently navigating a superpressure balloon in the stratosphere[1] requires the integration of a multitude of cues, such as wind speed and solar elevation, and the process is complicated by forecast errors and sparse wind measurements. Coupled with the need to make decisions in real time, these factors rule out the use of conventional control techniques[2,3]. Here we describe the use of reinforcement learning[4,5] to create a high-performing flight controller. Our algorithm uses data augmentation[6,7] and a self-correcting design to overcome the key technical challenge of reinforcement learning from imperfect data, which has proved to be a major obstacle to its application to physical systems[8]. We deployed our controller to station Loon superpressure balloons at multiple locations across the globe, including a 39-day controlled experiment over the Pacific Ocean. Analyses show that the controller outperforms Loon's previous algorithm and is robust to the natural diversity in stratospheric winds. These results demonstrate that reinforcement learning is an effective solution to real-world autonomous control problems in which

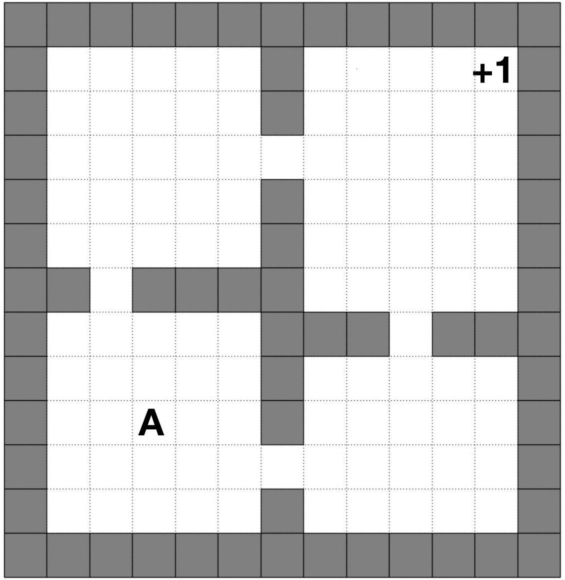




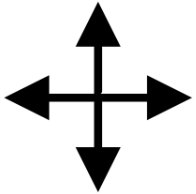


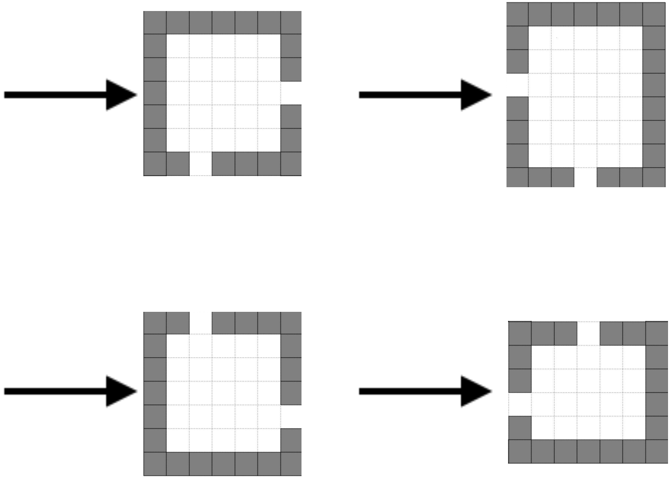
[Bellemare, Candido, Castro, Gong, Machado, Moitra, Ponda, & Wang, Nature 2020]

Exploring at a higher level of abstraction



"actions":





But where do options come from?

Temporal abstraction – Options

[Sutton, Precup, & Singh, AIJ 1999]
[Sutton et al., AAMAS 2011]

$$v^{c,z}_{\pi,\beta}(s) \doteq \mathbb{E}_{\pi,\beta}\left[\sum_{j=1}^{K} \gamma^{j-1} c(S_j) + \gamma^{K-1} z(S_K) \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

[Sutton, Machado, Holland, Szepesvari, Timbers, Tanner, & White, AIJ 2023]

Temporal abstraction – Options

[Sutton, Precup, & Singh, AIJ 1999]
[Sutton et al., AAMAS 2011]

subtask (problem): maximize discounted sum of cumulants plus a stopping value

$$v_{\pi,\beta}^{c,z}(s) \doteq \mathbb{E}_{\pi,\beta}\left[\sum_{j=1}^{K} \gamma^{j-1} \underline{c(S_j)} + \gamma^{K-1} \underline{z(S_K)} \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

[Sutton, Machado, Holland, Szepesvari, Timbers, Tanner, & White, AIJ 2023]

Temporal abstraction – Options

[Sutton, Precup, & Singh, AIJ 1999]
[Sutton et al., AAMAS 2011]

random variable

$$v^{c,z}_{\underline{\pi,\beta}}(s) \doteq \mathbb{E}_{\underline{\pi,\beta}}\left[\sum_{j=1}^{\boxed{K}} \gamma^{j-1} c(S_j) + \gamma^{\boxed{K}-1} z(S_K) \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

option (solution): policy and stopping function

[Sutton, Machado, Holland, Szepesvari, Timbers, Tanner, & White, AIJ 2023]

Temporal abstraction – Options

[Sutton, Precup, & Singh, AIJ 1999]
[Sutton et al., AAMAS 2011]

subtask (problem): maximize discounted sum of cumulants plus a stopping value

random variable

$$v^{\underline{c,z}}_{\underline{\pi,\beta}}(s) \doteq \mathbb{E}_{\underline{\pi,\beta}}\left[\sum_{j=1}^{\boxed{K}} \gamma^{j-1} \underline{c(S_j)} + \gamma^{\boxed{K}-1} \underline{z(S_K)} \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

option (solution): policy and stopping function

[Sutton, Machado, Holland, Szepesvari, Timbers, Tanner, & White, AIJ 2023]

Defining the option discovery problem

$$v_{\pi,\beta}^{c,z}(s) \doteq \mathbb{E}_{\pi,\beta}\left[\sum_{j=1}^{K} \gamma^{j-1} c(S_j) + \gamma^{K-1} z(S_K) \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

Specify subtask:

- *c:* signal to maximize
- *z:* stopping-value function

Defining the option discovery problem

$$v^{c,z}_{\pi,\beta}(s) \doteq \mathbb{E}_{\pi,\beta}\left[\sum_{j=1}^{K} \gamma^{j-1} c(S_j) + \gamma^{K-1} z(S_K) \,\middle|\, S_0 = s\right], \quad \forall s \in \mathcal{S}$$

Specify subtask:

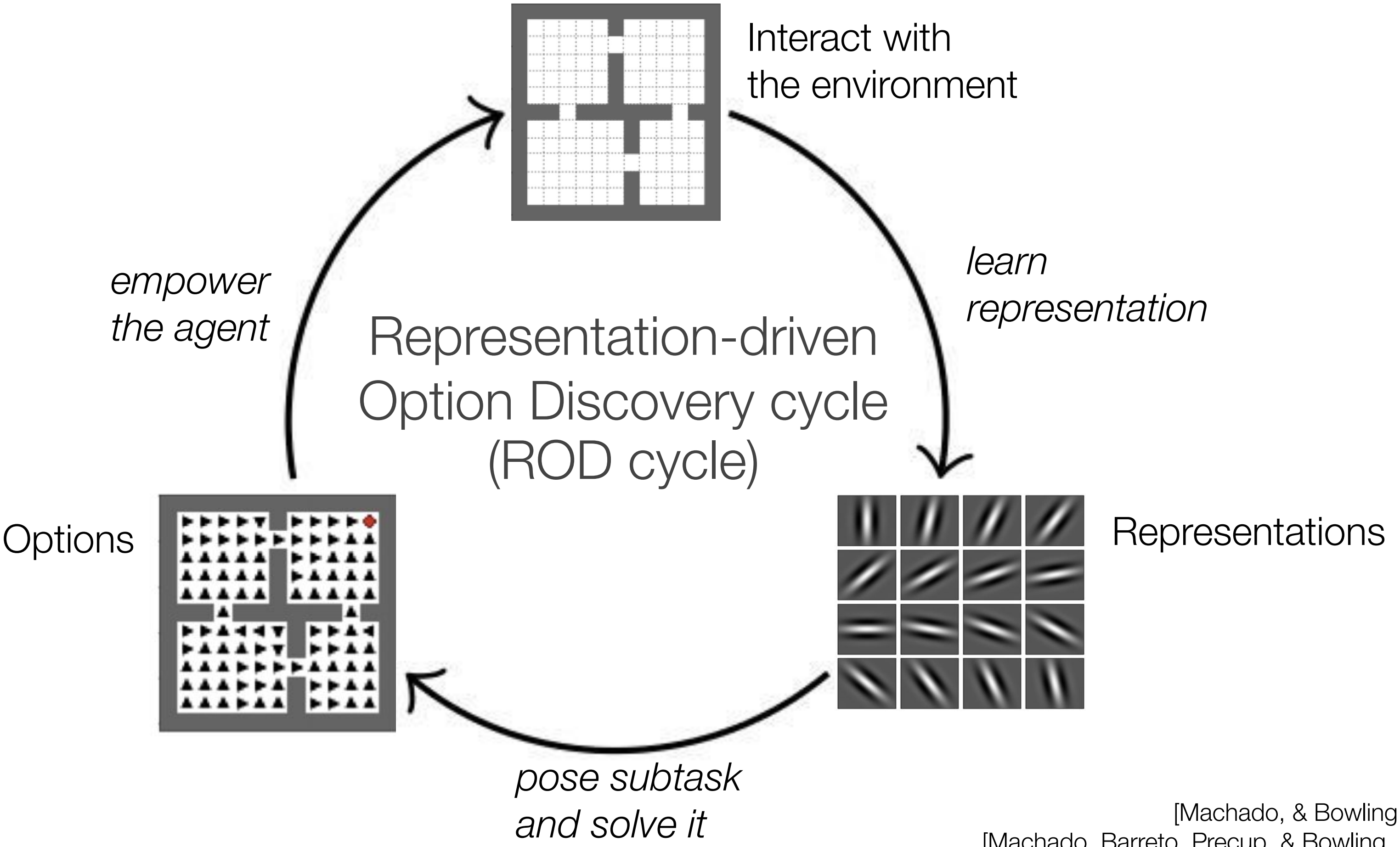
- *c:* signal to maximize
- *z:* stopping-value function

Example:
Shortest-path option to a bottleneck state

- **C_t = -1**
- **z(s) = 0 at subgoal states or z(s) = -∞ o.w.**

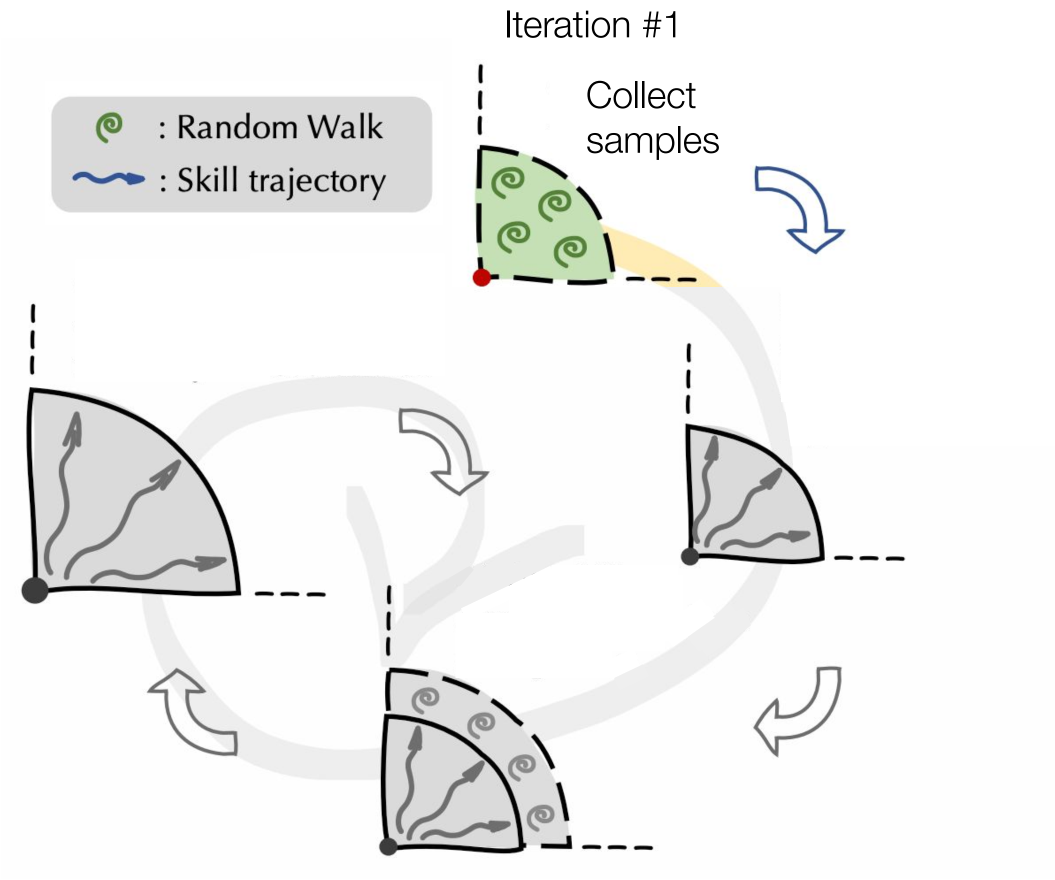
~~Where should options come from?~~

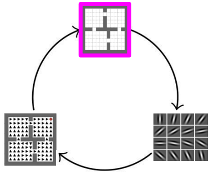
What subtasks should we use?



[Machado, & Bowling, arXiv 2016]
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

Intuition

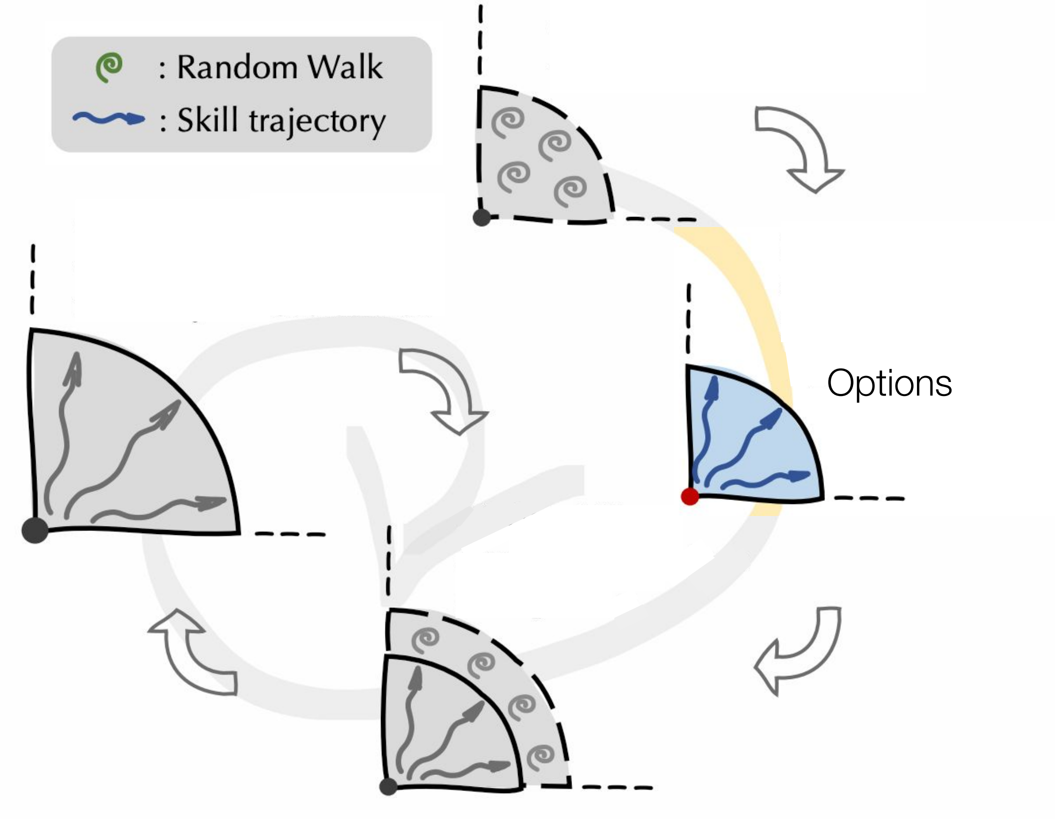


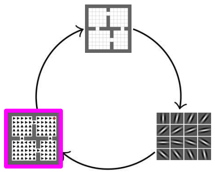


[Erraqabi, Machado, Zhao, Sukhbaatar, Lazaric, Denoyer, & Bengio, UAI 2022]

Intuition

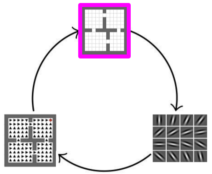
Iteration #1

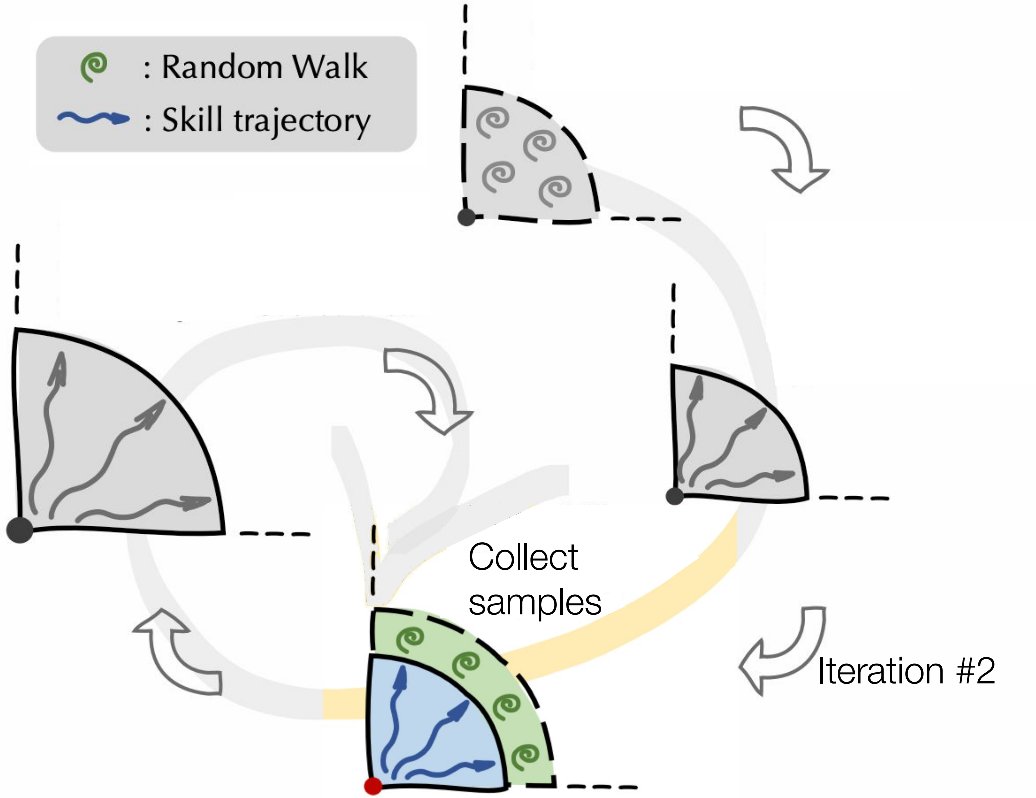




[Erraqabi, Machado, Zhao, Sukhbaatar, Lazaric, Denoyer, & Bengio, UAI 2022]

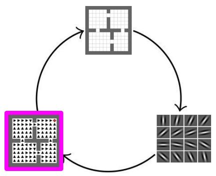
Intuition

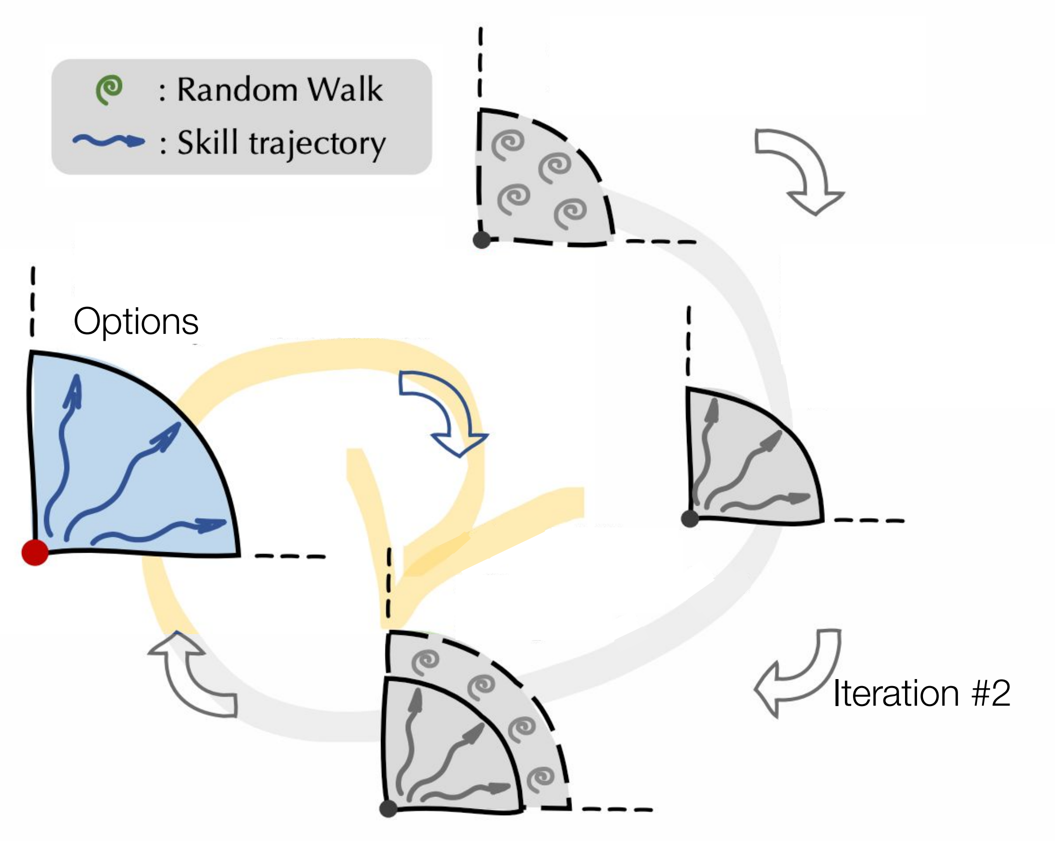




[Erraqabi, Machado, Zhao, Sukhbaatar, Lazaric, Denoyer, & Bengio, UAI 2022]

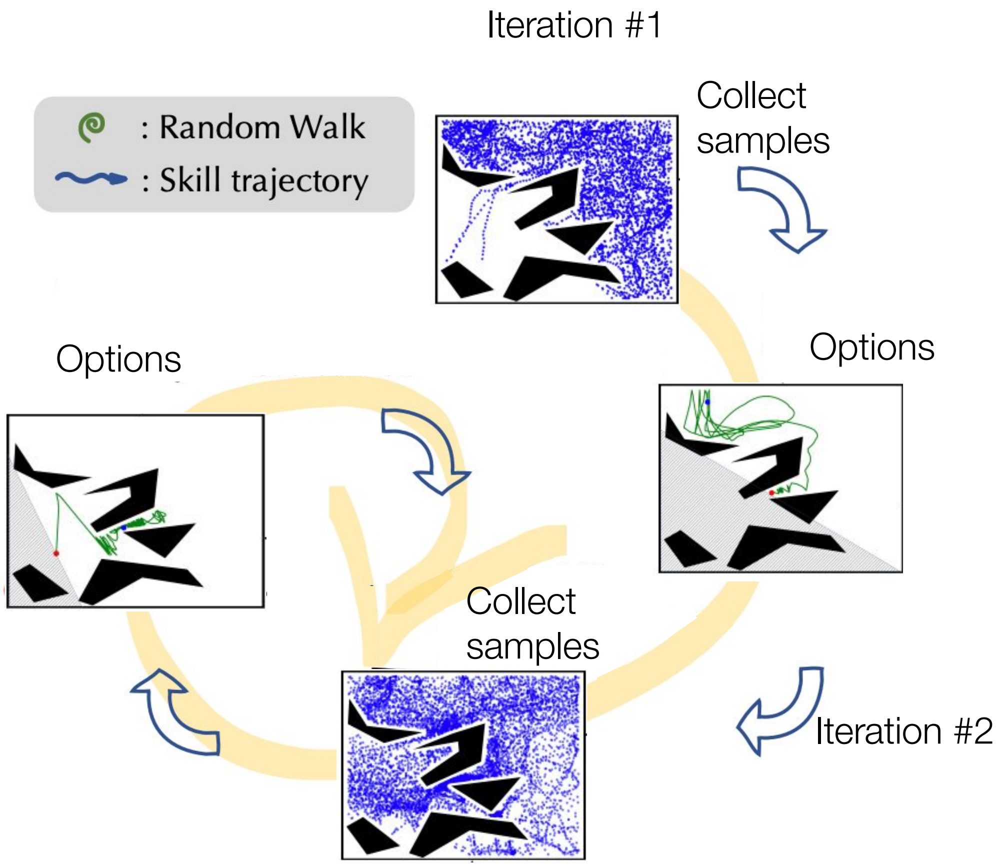
Intuition



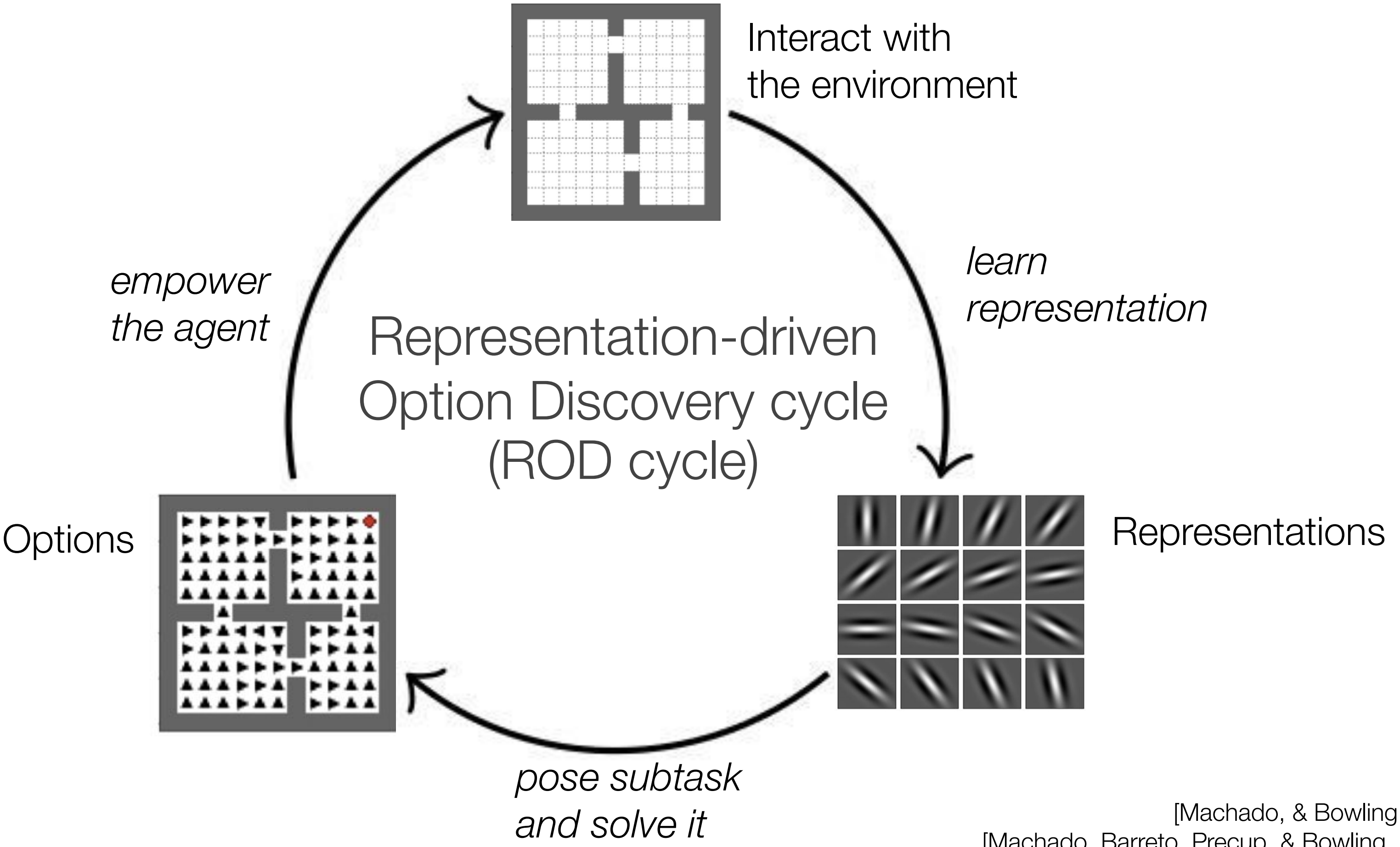


[Erraqabi, Machado, Zhao, Sukhbaatar, Lazaric, Denoyer, & Bengio, UAI 2022]

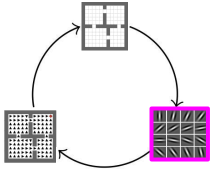
Intuition



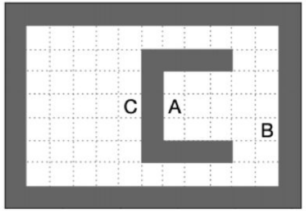
[Jinnai, Park, Machado, & Konidaris, ICLR 2020]

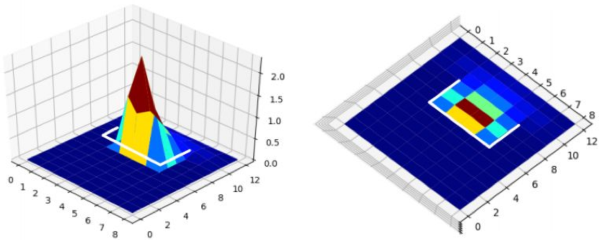


[Machado, & Bowling, arXiv 2016]
[Machado, Barreto, Precup, & Bowling, JMLR 2023]



Successor Representation [Dayan, Neural Computation 1993]



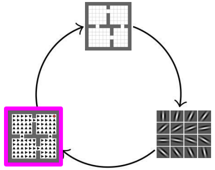


$$\boldsymbol{\Psi}_\pi(s, s') = \mathbb{E}_\pi\left[\sum_t \gamma^t \mathbf{1}_{S_t = s'} \,\middle|\, S_0 = s\right] \quad (1)$$

$$\boldsymbol{\Psi}_\pi = \sum_t (\gamma \mathbf{P}_\pi)^t = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} \quad (2)$$

The SR as a collection of GVFs

- **C_t: indicator function for state visitation**
- **γ: any fixed γ, but the same across all GVFs**
- **π: any policy, but the same across all GVFs**
- **z(s) = 0 for all states**



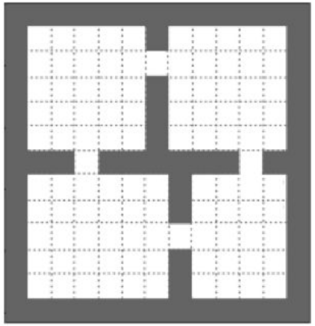
Eigenoptions

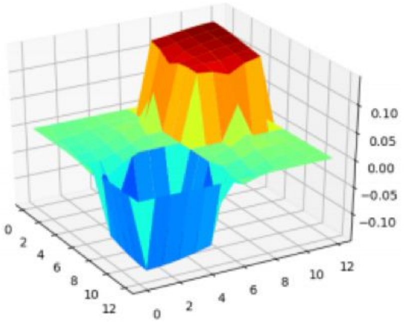
$$\mathbf{\Psi}_\pi \mathbf{e} = \lambda \mathbf{e} \quad (1)$$

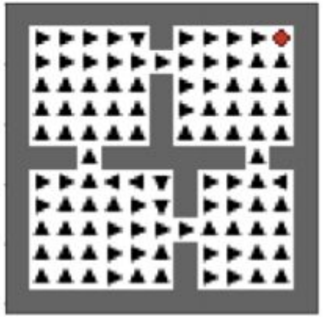
$$C_t \doteq \mathbf{e}_i{}^\top \big(\mathbf{x}(S_t) - \mathbf{x}(S_{t-1})\big) \quad (2)$$

$$z(s) = 0 \qquad \forall s \in \mathcal{S} \quad (3)$$

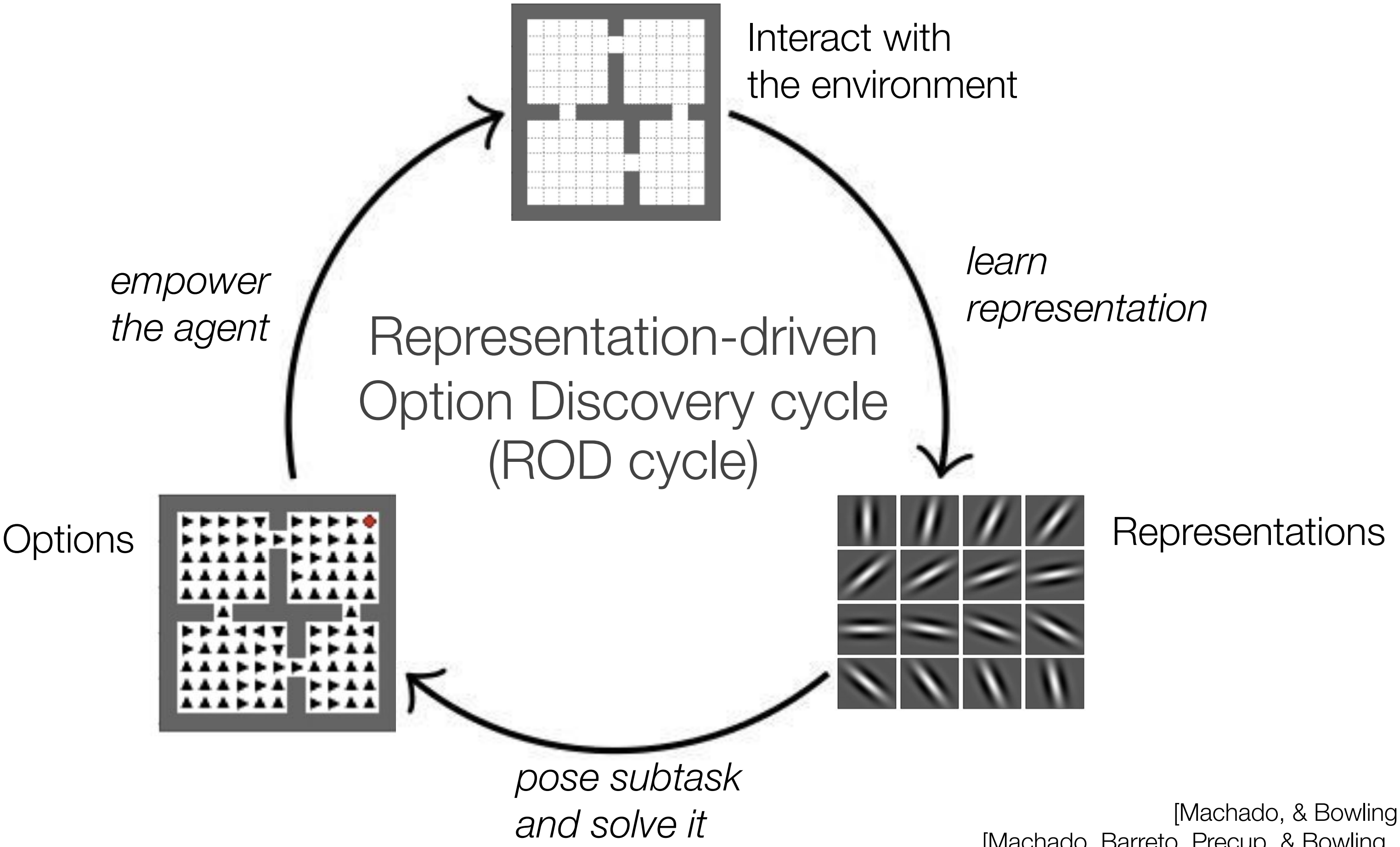
$$q_\pi(\cdot, \perp) = 0 \qquad \forall \pi \quad (4)$$



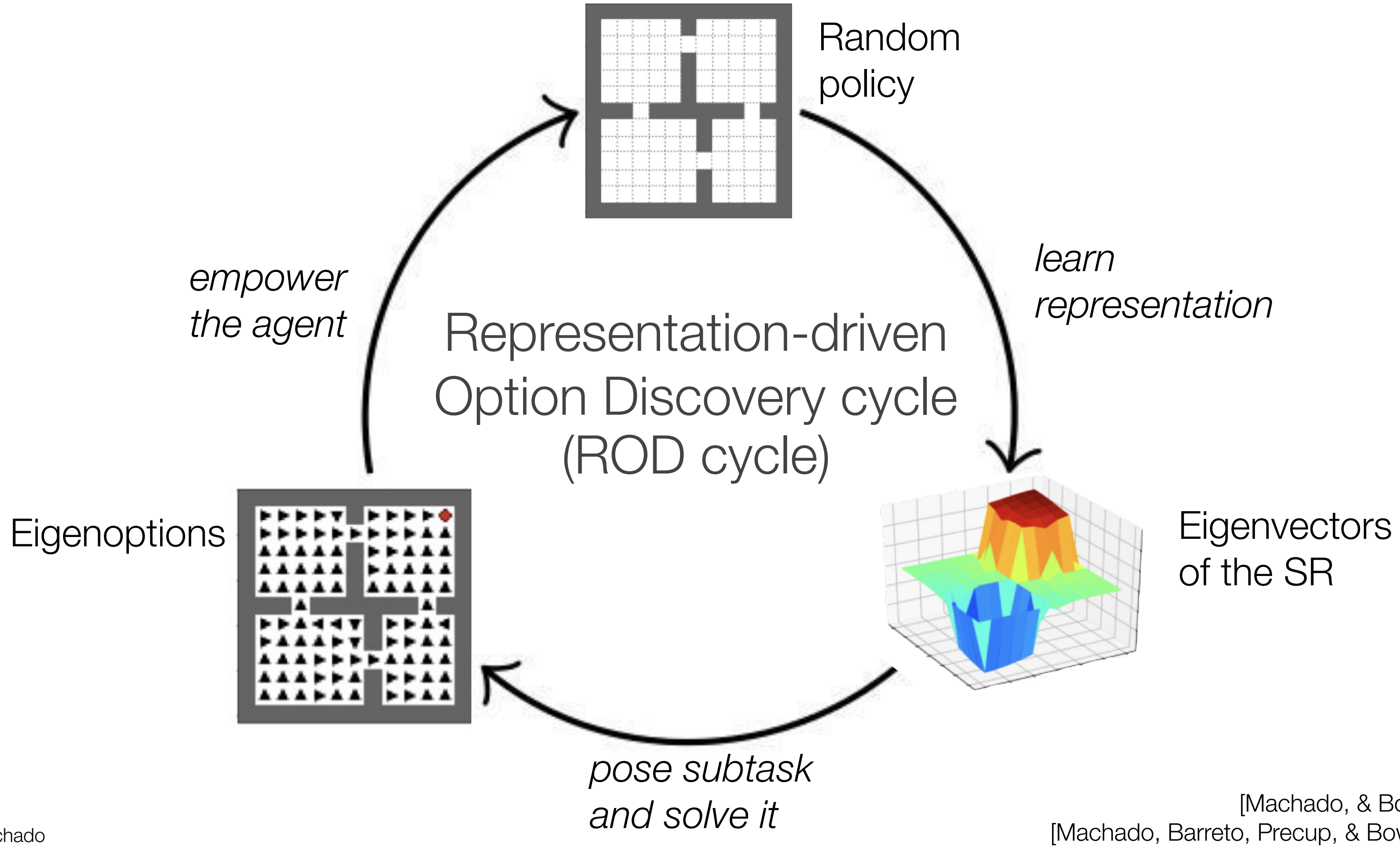




[Machado, Bellemare, & Bowling, ICML 2017]
[Machado, Rosenbaum, Guo, Liu, Tesauro, & Campbell, ICLR 2018]



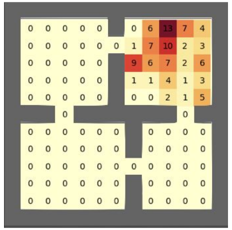
[Machado, & Bowling, arXiv 2016]
[Machado, Barreto, Precup, & Bowling, JMLR 2023]



[Machado, & Bowling, arXiv 2016]
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

An example

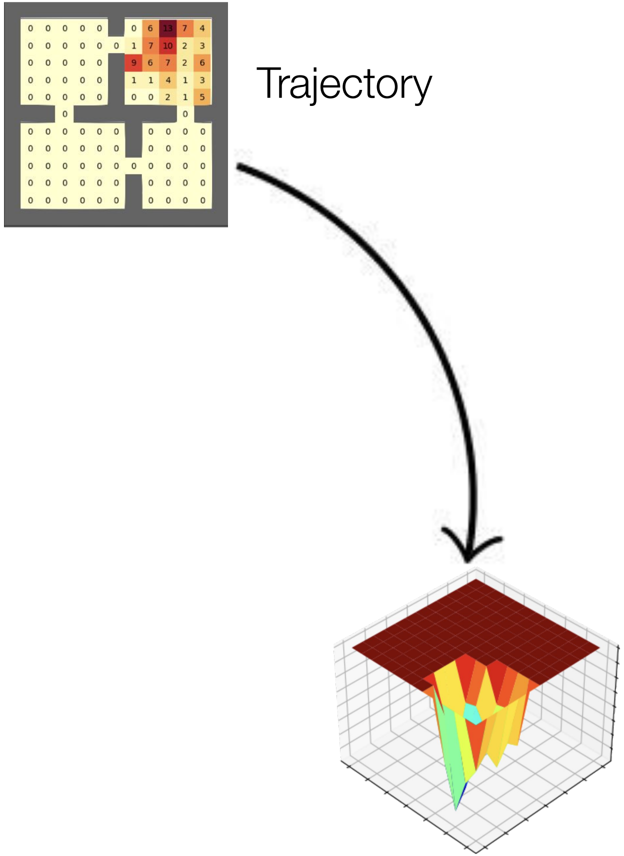
ROD cycle
Iteration 1



Trajectory

[Machado, Barreto, Precup, & Bowling, JMLR 2023]

ROD cycle
Iteration 1

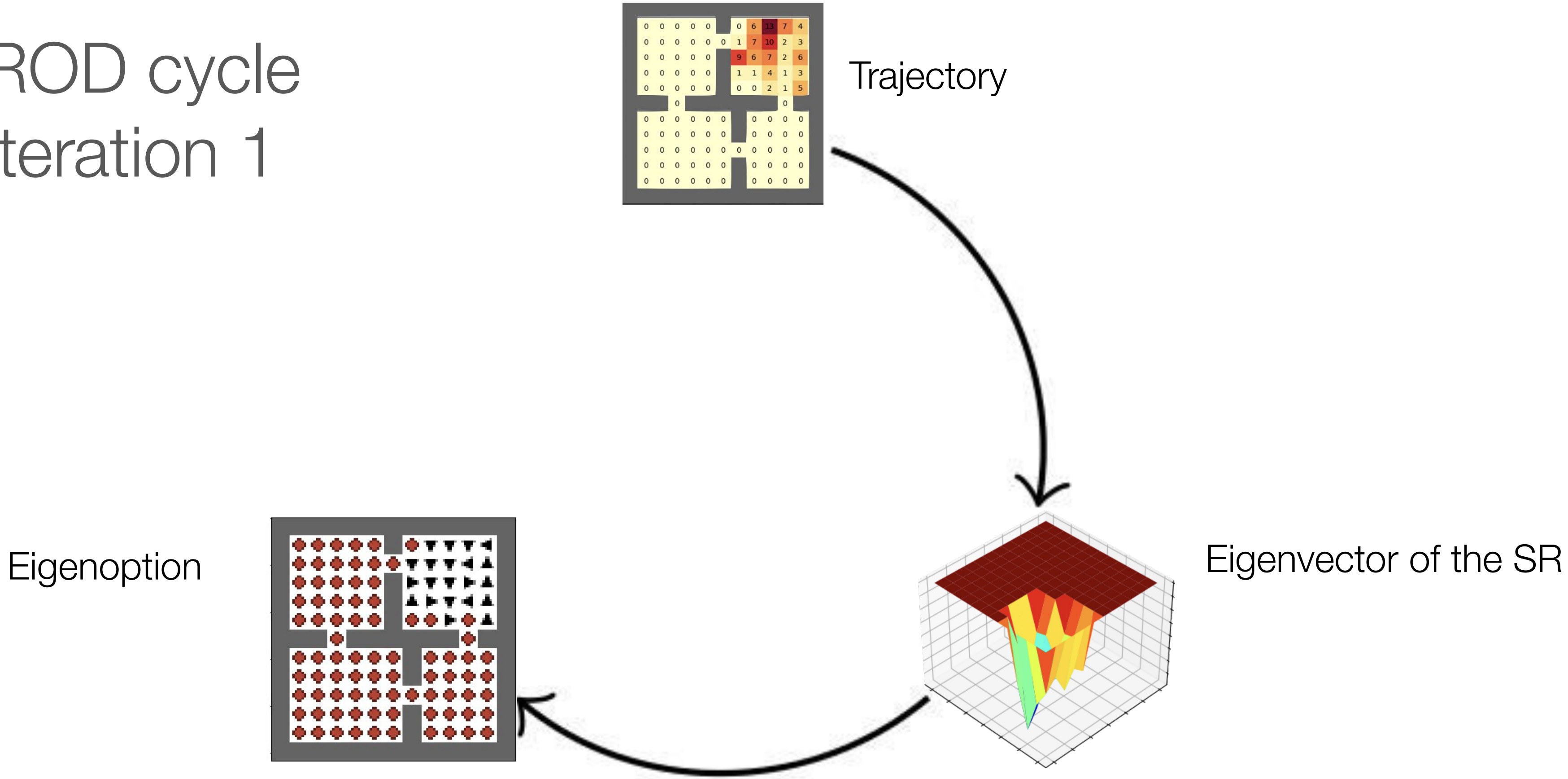


Trajectory

Eigenvector of the SR

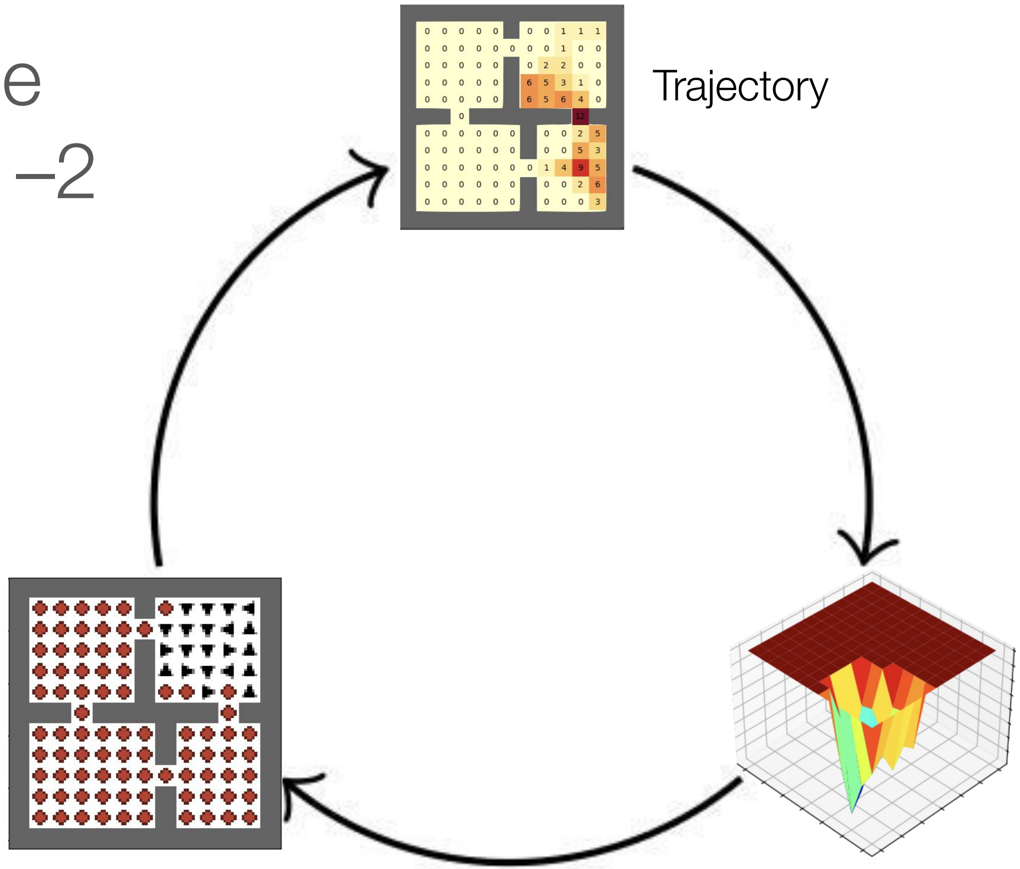
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

ROD cycle
Iteration 1



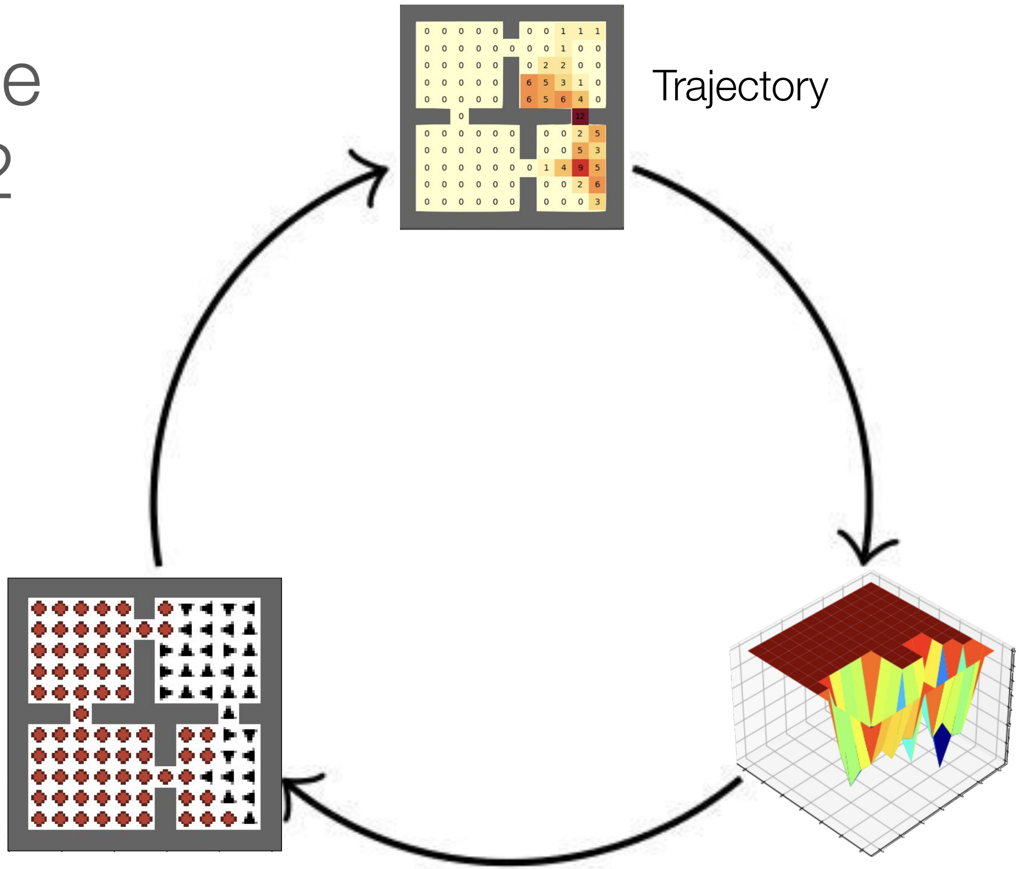
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

ROD cycle
Iteration 1–2



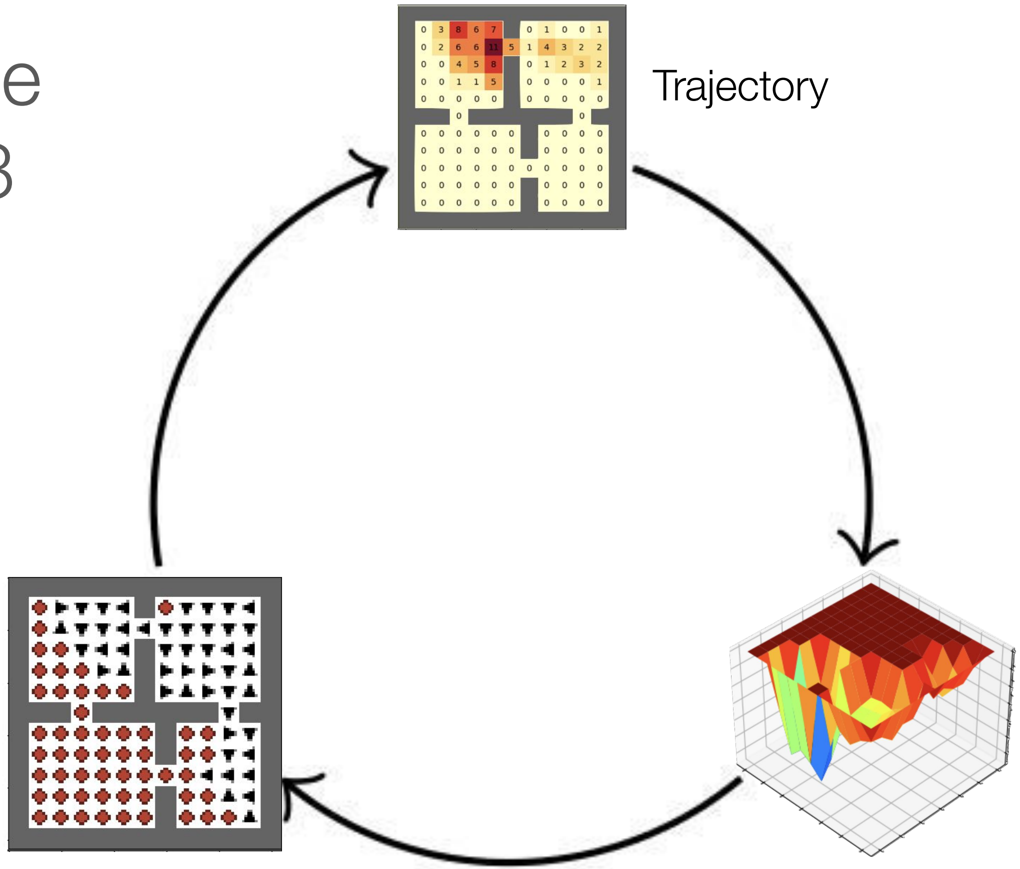
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

ROD cycle
Iteration 2



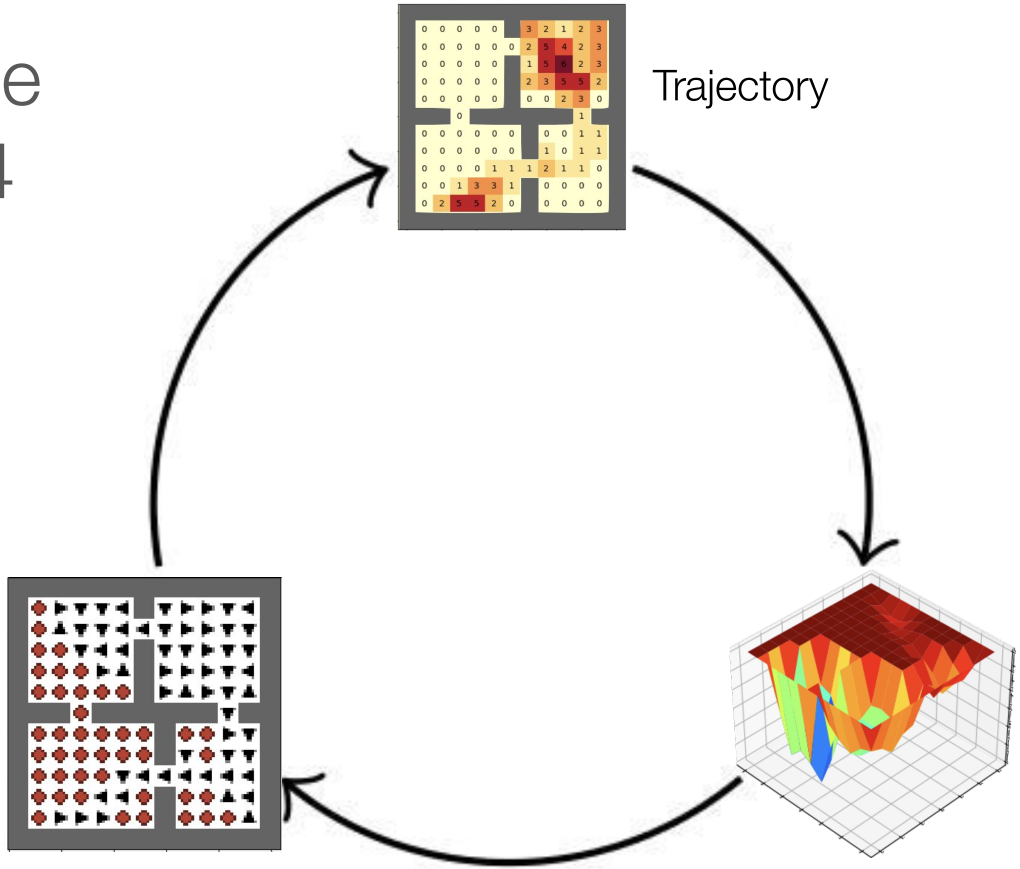
[Machado, Barreto, Precup, & Bowling, JMLR 2023]

ROD cycle
Iteration 3



[Machado, Barreto, Precup, & Bowling, JMLR 2023]

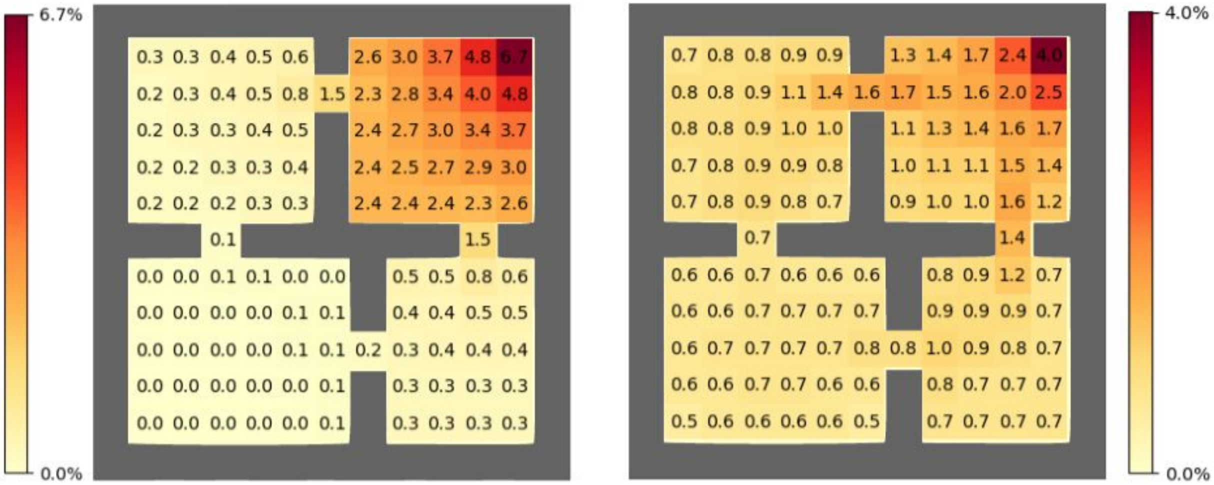
ROD cycle
Iteration 4



[Machado, Barreto, Precup, & Bowling, JMLR 2023]

Covering eigenoptions vs random policy

State visitation:



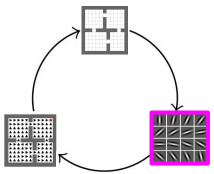
Random policy over primitive actions

Random policy over primitive actions and covering eigenoptions

steps needed to visit all states: ~27,000 ~2,300

[Machado, Barreto, Precup, & Bowling, JMLR 2023]

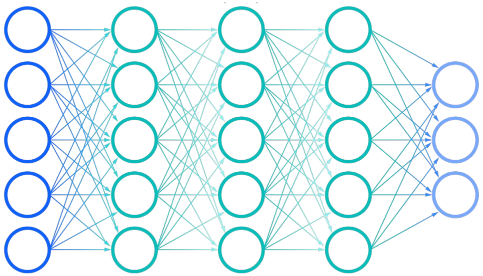
Approximating the eigenfunctions of the graph Laplacian with neural networks



Approximate eigendecomposition

We can approximate the eigenvectors using a neural network and SGD

Observation →



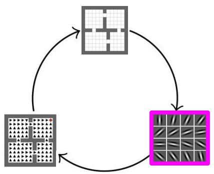
$f_1(s)$
$f_2(s)$
$\vdots$
$f_d(s)$

→ Approximate the top d eigenvectors of the Laplacian.

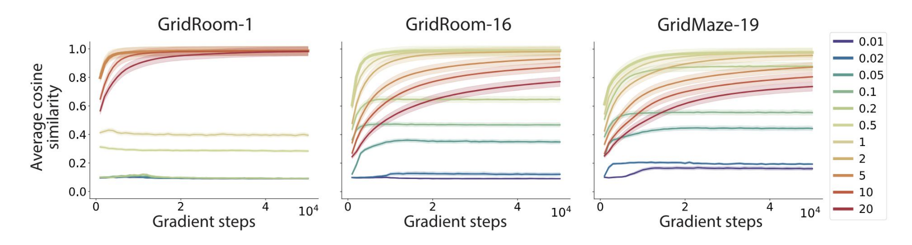
minimizing the *augmented Lagrangian Laplacian objective (ALLO)*:

$$\max_{\boldsymbol{\beta}} \min_{\mathbf{u} \in \mathbb{R}^{d|\mathcal{S}|}} \sum_{i=1}^{d} \langle \mathbf{u}_i, \mathbf{L}\mathbf{u}_i \rangle + \sum_{j=1}^{d} \sum_{k=1}^{j} \beta_{jk} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right) + b \sum_{j=1}^{d} \sum_{k=1}^{j} \left(\langle \mathbf{u}_j, [\![\mathbf{u}_k]\!] \rangle - \delta_{jk} \right)^2$$

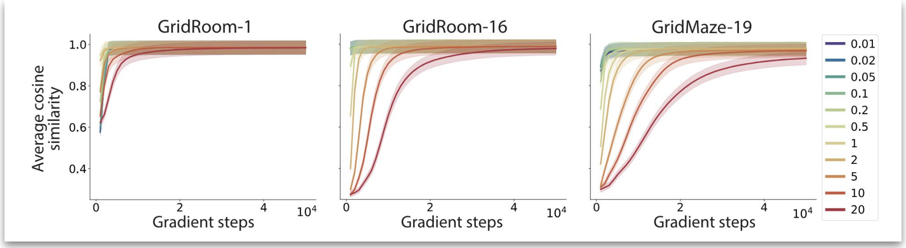
[Gomez, Bowling, & Machado, ICLR 2024]



Proper Laplacian Representation Learning

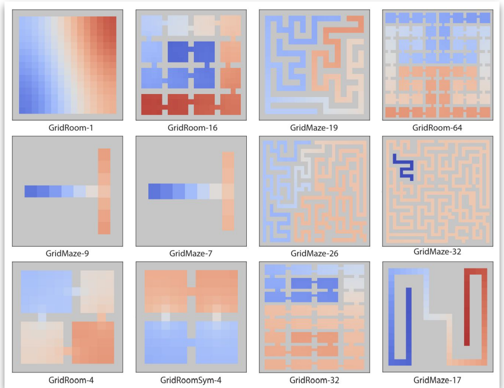


GGDO [Wang et al., ICML 2021]



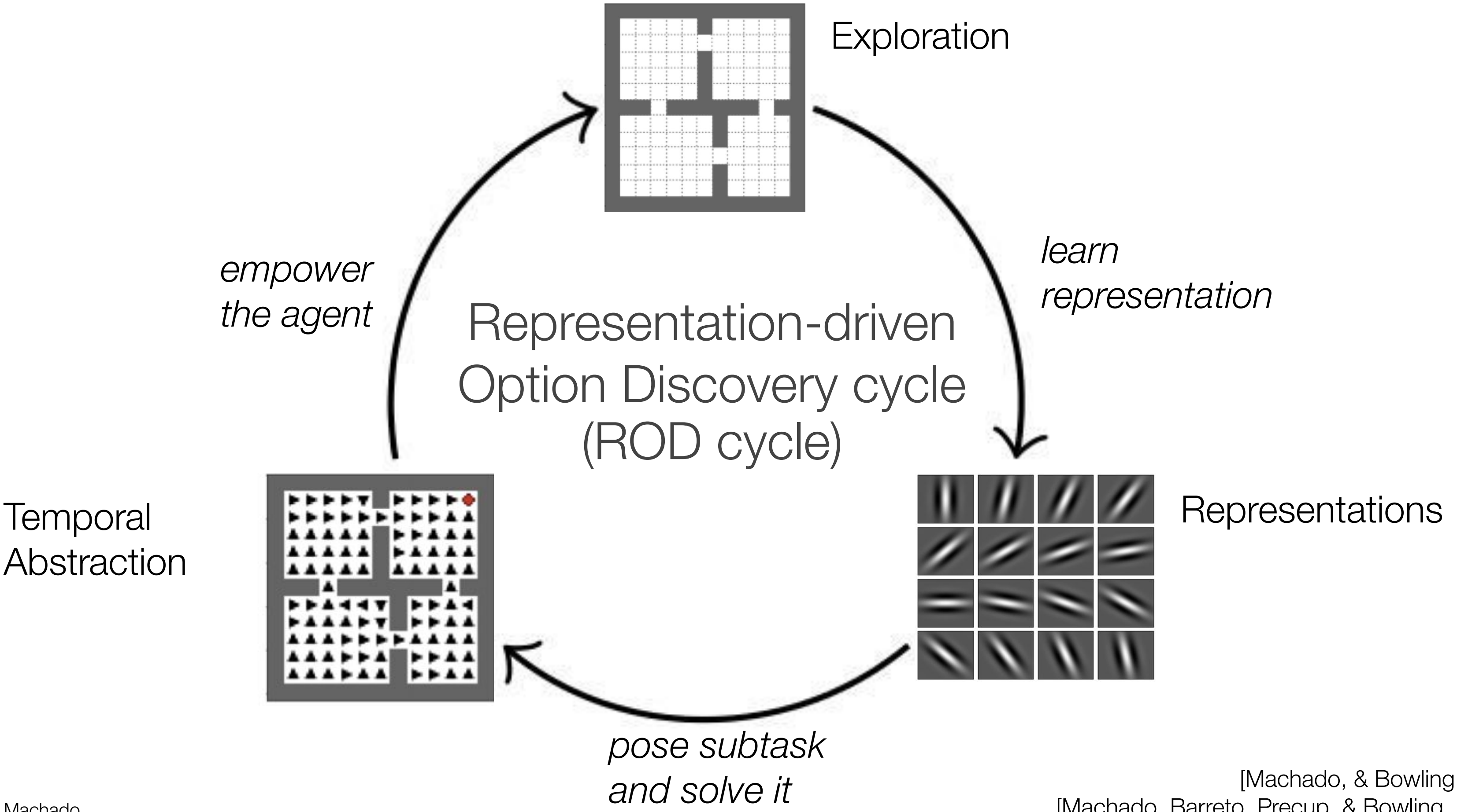
ALLO

Compared to previous approaches it is more robust, accurate, and it works across different data streams

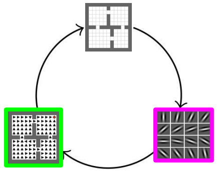


[Gomez, Bowling, & Machado, ICLR 2024]

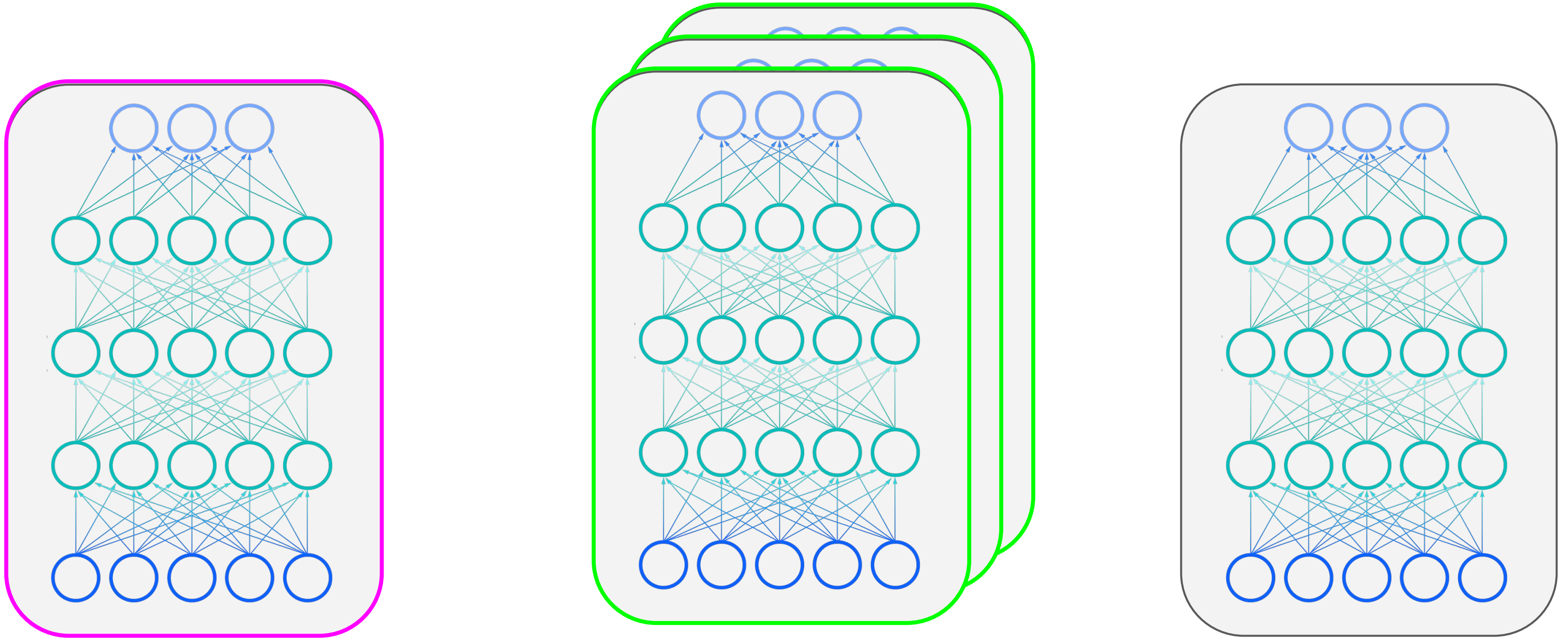
Putting everything together



[Machado, & Bowling, arXiv 2016]
[Machado, Barreto, Precup, & Bowling, JMLR 2023]



Approximations all the way down



Laplacian representation

GGDO [Wang et al., 2021]

Eigenoptions

DDQN + n-step

[van Hasselt et al., 2016]

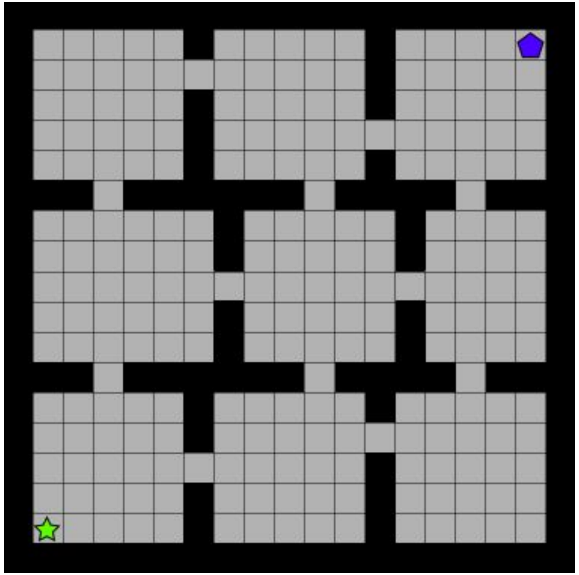
Main Q-learner

DDQN + n-step

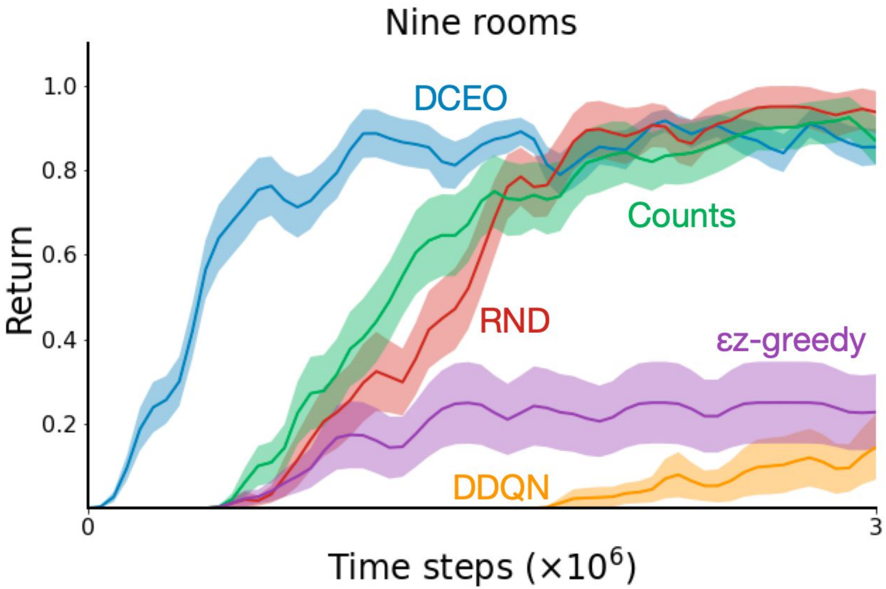
[van Hasselt et al., 2016]

[Klissarov & Machado, ICML 2023]

Online deep covering eigenoptions

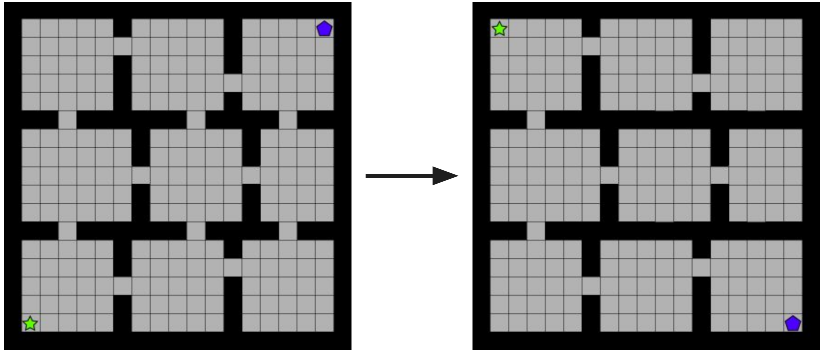


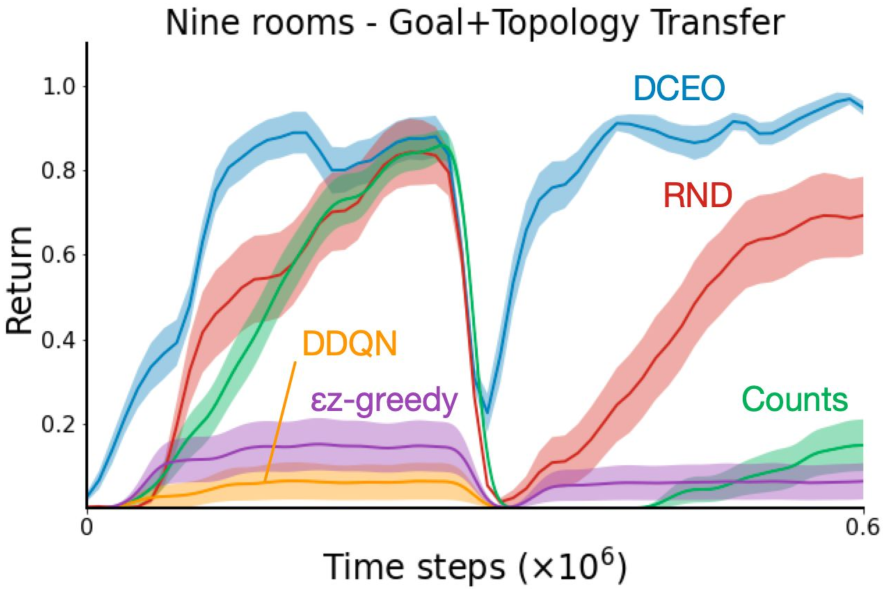
Observation type:
Image



[Klissarov & Machado, ICML 2023]

DCEO in non-stationary environments (continual learning)

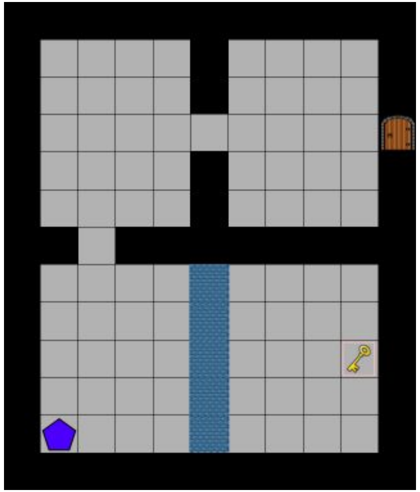




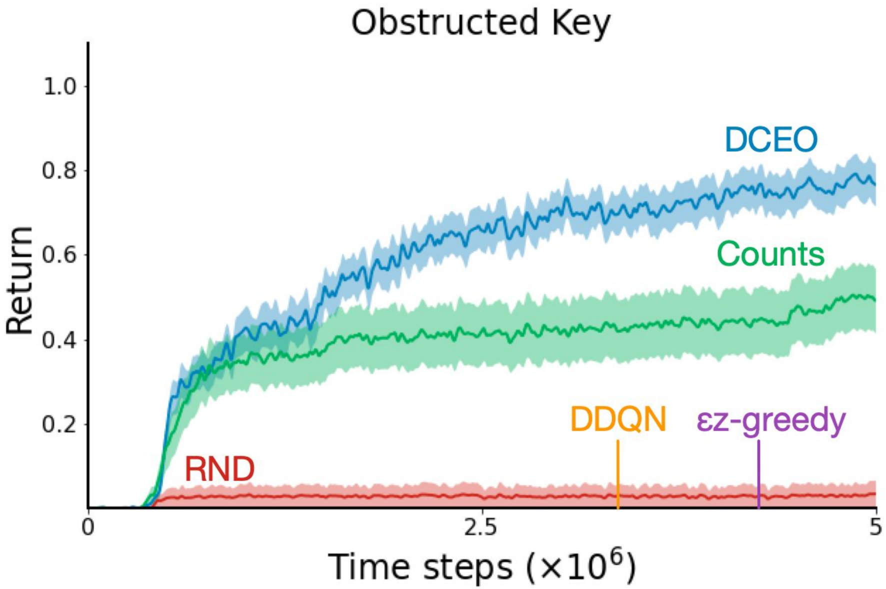
Observation type:
Image

[Klissarov & Machado, ICML 2023]

and more!

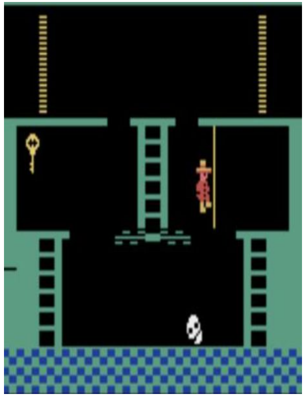


Observation type:
Image

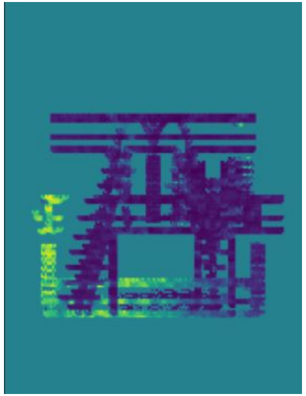


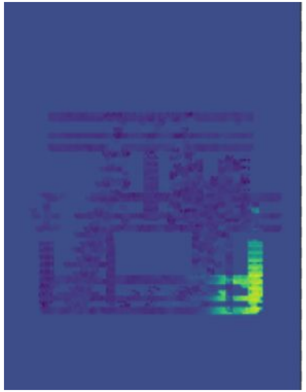
[Klissarov & Machado, ICML 2023]

Subgoals in Montezuma's Revenge



Montezuma's Revenge

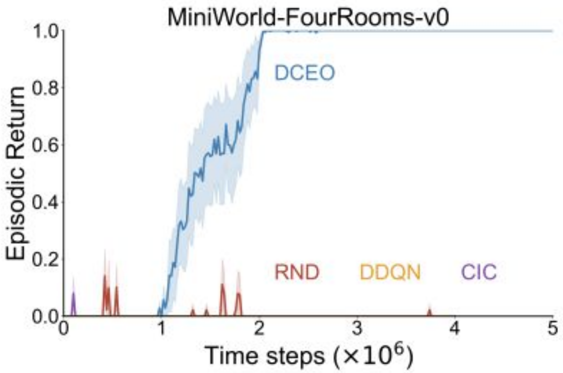


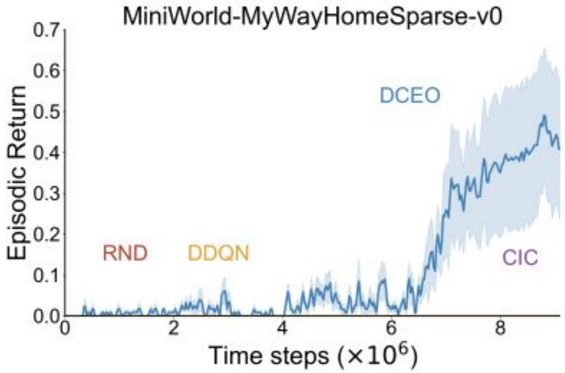


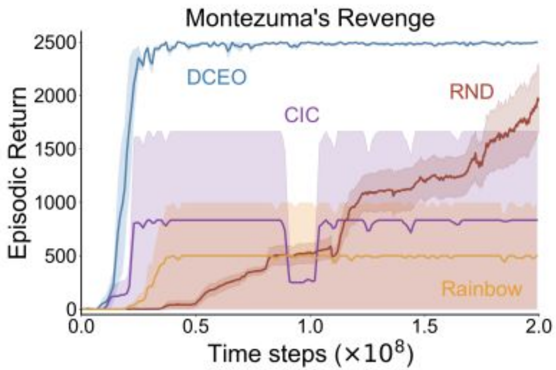
Eigenfunctions discovered by the Laplacian

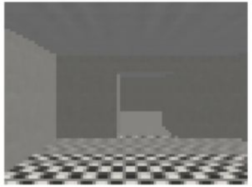
[Klissarov & Machado, ICML 2023]

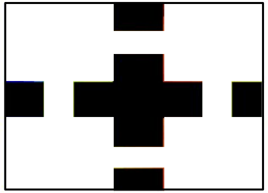
Beyond “navigation, 2d gridworld tasks”

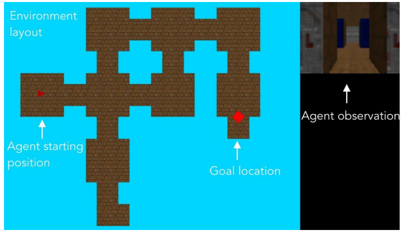












Marlos C. Machado

[Klissarov & Machado, ICML 2023]

Are we done?

Are we done?

- More demonstrations!

- More algorithms!

Are we done?

- More demonstrations! Learning from scratch is slow, almost by definition
  - There are many good reasons to do research when learning from scratch,
    but that impacts the types of demonstrations we can provide
  - We do have results leveraging domain knowledge (a.k.a. LLMs) alongside temporal abstractions,
    but I won't talk about that today (see work by Klissarov et al., 2025)
- More algorithms! Conceptually speaking, there are still some pieces missing
  - We are not using options for credit assignment
  - Should we define state similarity in terms of rewards as well?
  - What about partial observability?
  - Can we combine options without additional learning?
  - What about MBRL and planning?

Are we done?

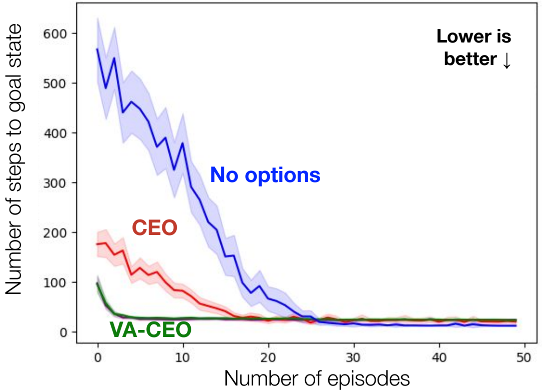
- More demonstrations! Learning from scratch is slow, almost by definition
    - There are many good reasons to do research when learning from scratch, but that impacts the types of demonstrations we can provide
    - We do have results leveraging domain knowledge (a.k.a. LLMs) alongside temporal abstractions, but I won't talk about that today (see work by Klissarov et al., 2025)
- More algorithms! Conceptually speaking, there are still some pieces missing
    - We are not using options for credit assignment  (Kotamreddy & Machado, In preparation)
    - Should we define state similarity in terms of rewards as well?  (Tse, Chandrasekar, & Machado, arXiv 2025)
    - What about partial observability?  (Jose & Machado, In preparation)
    - Can we combine options without additional learning?  (Chandrasekar & Machado, In preparation)
    - What about MBRL and planning? (see Sutton et al., 2023)

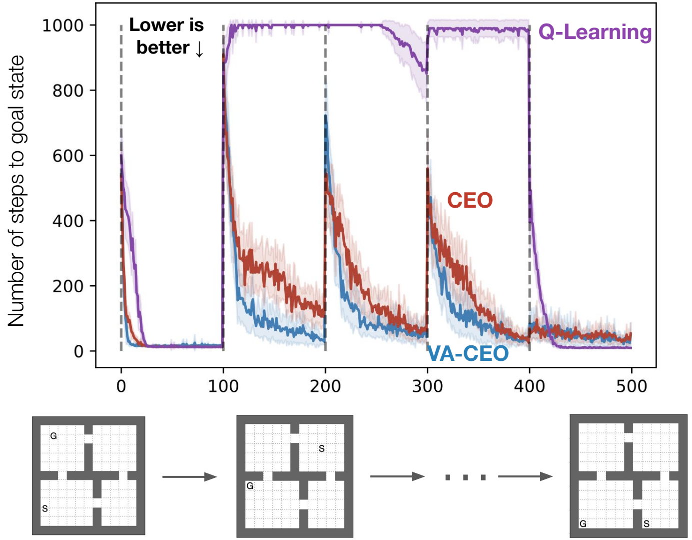
Are we done?

- More demonstrations! Learning from scratch is slow, almost by definition
  - There are many good reasons to do research when learning from scratch, but that impacts the types of demonstrations we can provide
  - We do have results leveraging domain knowledge (a.k.a. LLMs) alongside temporal abstractions, but I won’t talk about that today (see work by Klissarov et al., 2025)
- More algorithms! Conceptually speaking, there are still some pieces missing
  - **We are not using options for credit assignment**  (Kotamreddy & Machado, In preparation)
  - **Should we define state similarity in terms of rewards as well?**  (Tse, Chandrasekar, & Machado, arXiv 2025)
  - What about partial observability?  (Jose & Machado, In preparation)
  - Can we combine options without additional learning?  (Chandrasekar & Machado, In preparation)
  - What about MBRL and planning? (see Sutton et al., 2023)

Model-Free Value-Aware Covering Eigenoptions

- What if we used options not only for exploration but also for credit assignment?
- Pre-computed (tabular) options in the four-room environment:

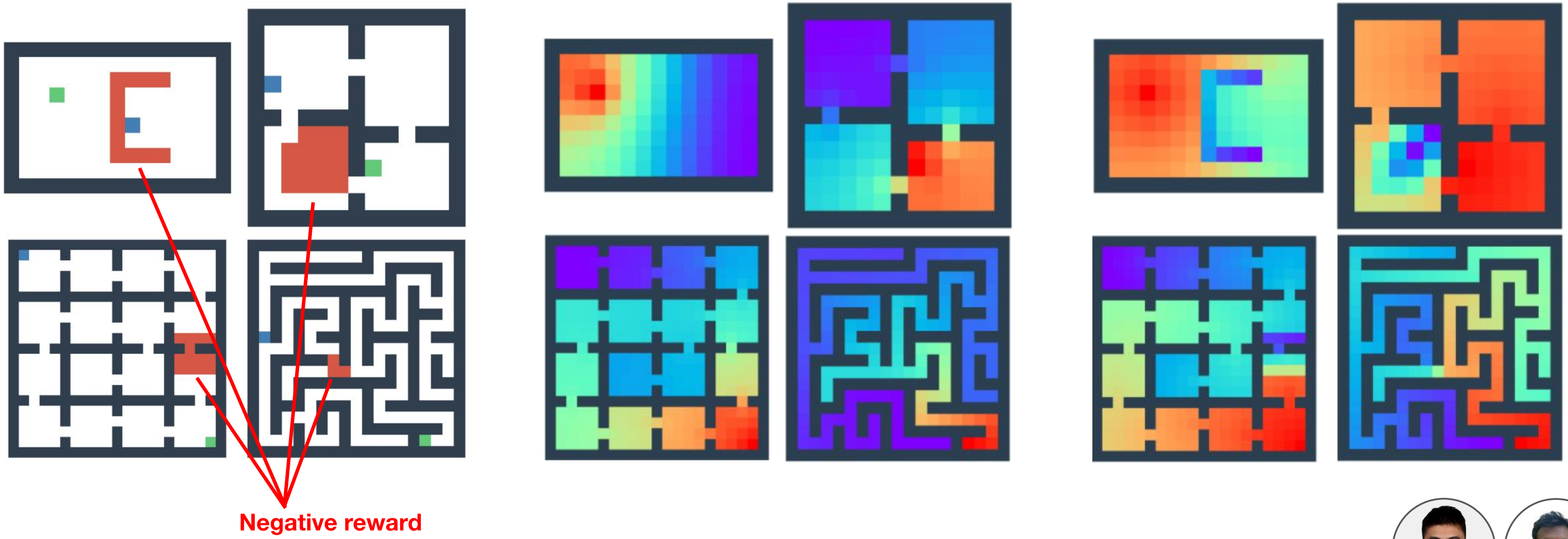




[Kotamreddy & Machado, In Preparation]

Reward-Aware Proto-Representation

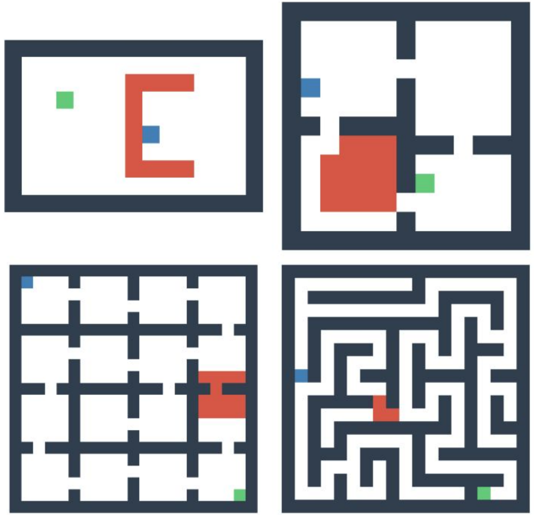
Environments | Successor representation | Reward-aware Successor representation



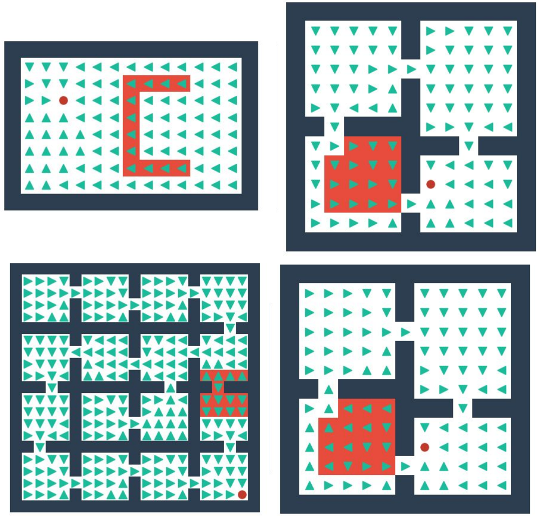
[Tse, Chandrasekar, & Machado, arXiv 2025]

Reward-Aware Eigenoptions

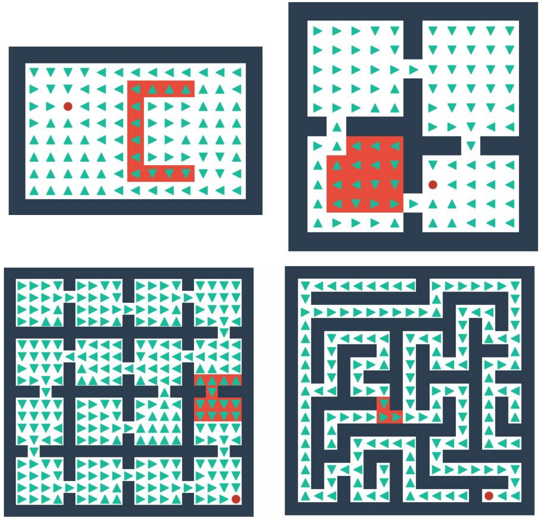
Environments



Successor representation

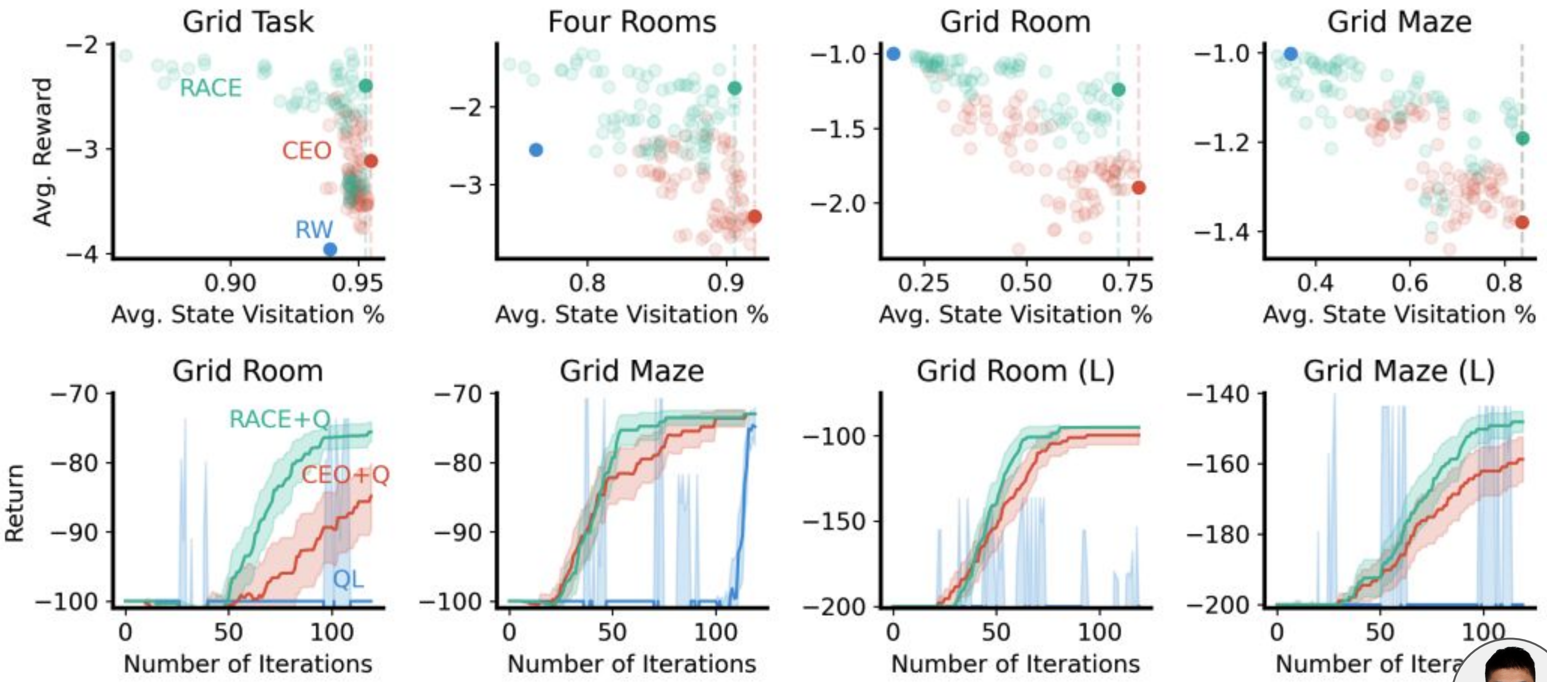


Reward-aware
Successor representation



[Tse, Chandrasekar, & Machado, arXiv 2025]

Reward-Aware Eigenoptions




[Tse, Chandrasekar, & Machado, arXiv 2025]

Wrapping up

Conclusion

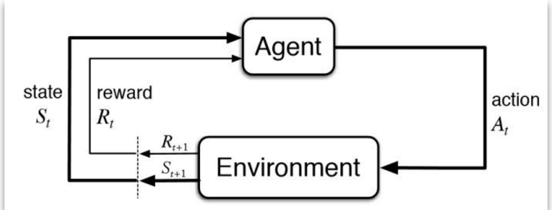
I’m still very excited about options. We now have option discovery methods that are general, that work, and that:

- demonstrate a virtuous cycle, **and**
- are fully experiential, **and**
- are scalable, **and**
- are amenable to function approximation, **and**
- work for different data streams, **and**
- don’t make any assumptions about the topology of the environment, **and**
- is (sort of) biologically plausible, if you are into that kind of thing ¯_(ツ)_/¯

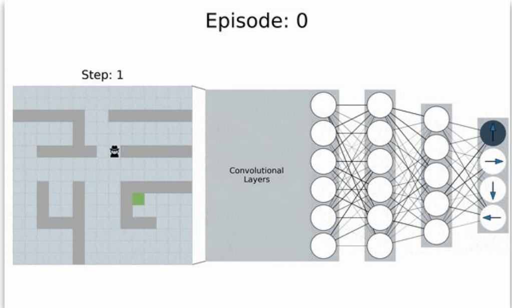
Conclusion

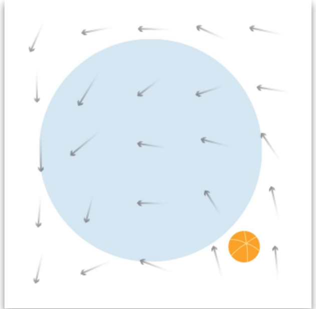
- Temporal abstractions should be a central piece of reinforcement learning
- Where should options come from?
    - Specific representations learned by the agent
- Given the right discovery method, options are scalable.
- Options are particularly helpful for continual learning/in the face of non-stationarity.
    - This is what I believe is the future of our field ↑

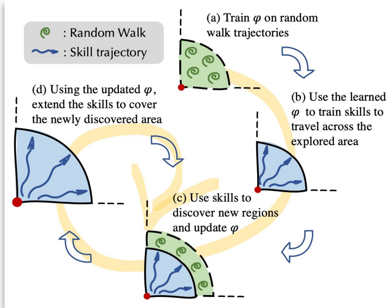
"State representations and temporal abstractions should be deeply intertwined, where representations and options are constantly refined based on each other."

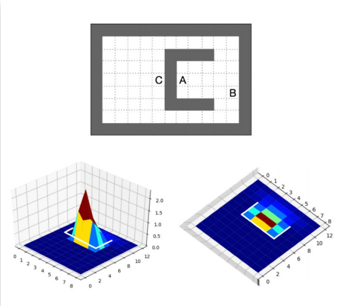


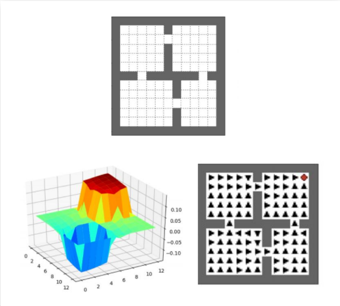
$$\omega = \langle \mathcal{I}_\omega, \pi_\omega, \beta_\omega \rangle$$











Thanks!