

[Home](#)[Articles](#)[Front Matter](#)[News](#)[Podcasts](#)[Authors](#)

NEW RESEARCH IN

Physical Sciences

Social Sciences

PHYSICAL SCIENCES



Fast reinforcement learning with generalized policy updates

 André Barreto,  Shaobo Hou, Diana Borsa, David Silver, and Doina Precup

PNAS first published August 17, 2020 <https://doi.org/10.1073/pnas.1907370117>

Edited by David L. Donoho, Stanford University, Stanford, CA, and approved July 9, 2020 (received for review July 20, 2019)



WHO WE ARE

Artificial intelligence could be one of humanity's most useful inventions. We research and build safe AI systems that learn how to solve problems and advance scientific discovery for all.



Demis Hassabis

Co-Founder & CEO

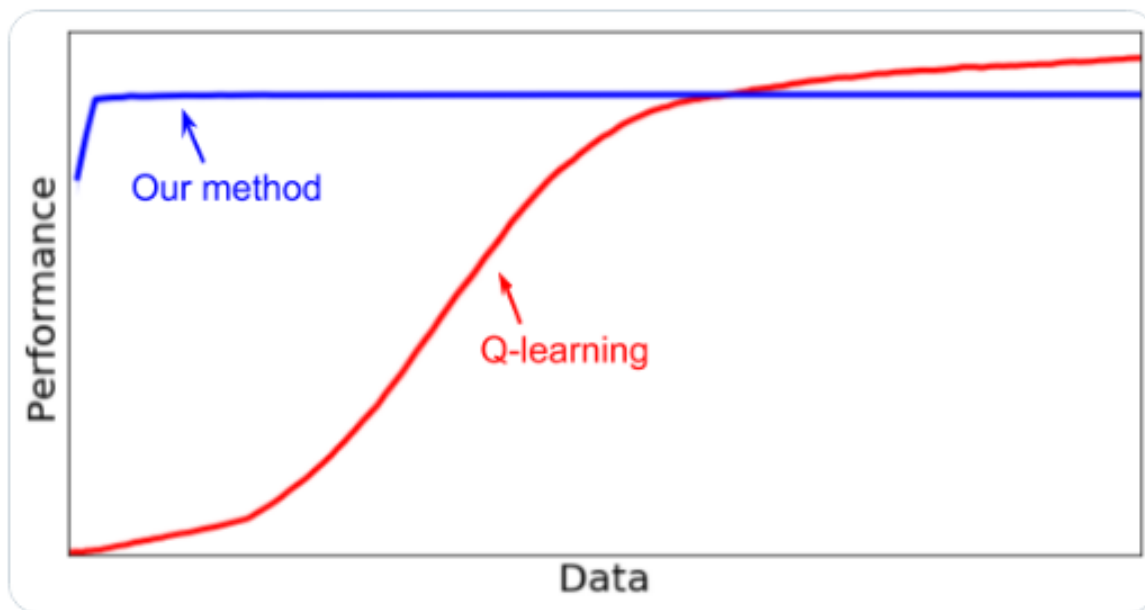
Demis Hassabis co-founded DeepMind in 2010 after successful careers in academia and computer game development. A child chess prodigy, he designed and programmed the multi-million selling, award-winning game Theme Park at the age of 17. After graduating from Cambridge University, he founded pioneering videogames firm Elixir Studios and completed a PhD in cognitive neuroscience at UCL. The journal Science listed his research on imagination and memory as one of 2007's top ten breakthroughs. Demis is a Fellow of the Royal Society, Royal Academy of Engineering and the Royal Society of Arts. In 2017 he featured in the Time 100 list of most influential people, and in 2018 he was awarded a CBE for services to science and technology.

tl;dr



How can we make [#reinforcementlearning](#) agents learn faster?

A new article published in [@PNASNews](#) proposes a divide-and-conquer approach to RL that allows an agent to combine the solution of known tasks to quickly solve new ones: bit.ly/2Q2vXbl



2:46 PM · Aug 18, 2020 · [Twitter Web App](#)

content alert: this is not a lecture on RL

you are encouraged to follow the wealthy materials online, and here I try to cover some of the basics.

TAGGED IN

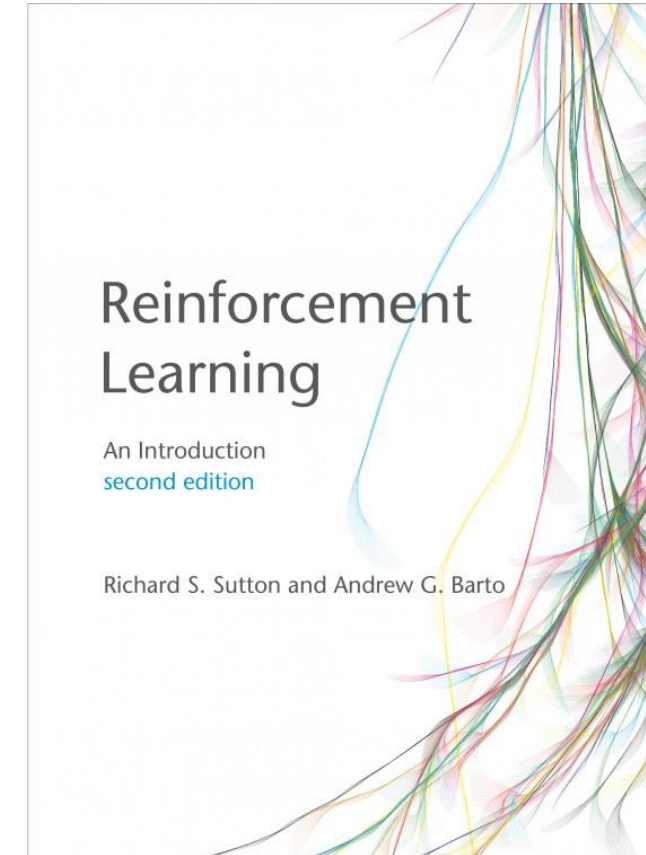
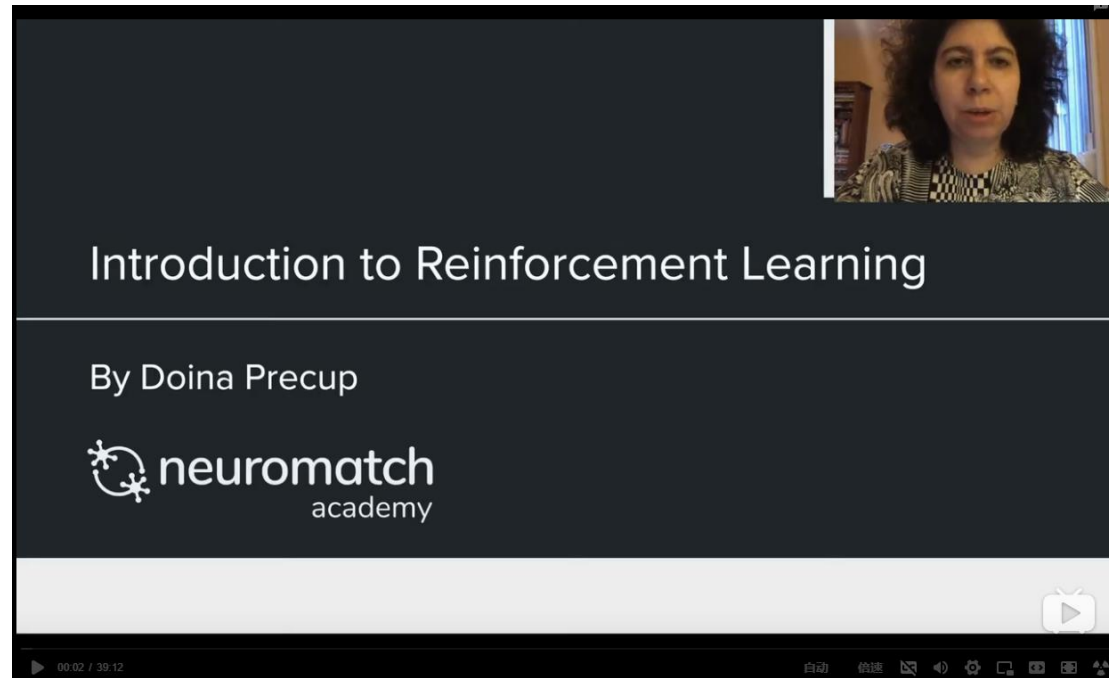
A Journey Into R L



Towards Data Science

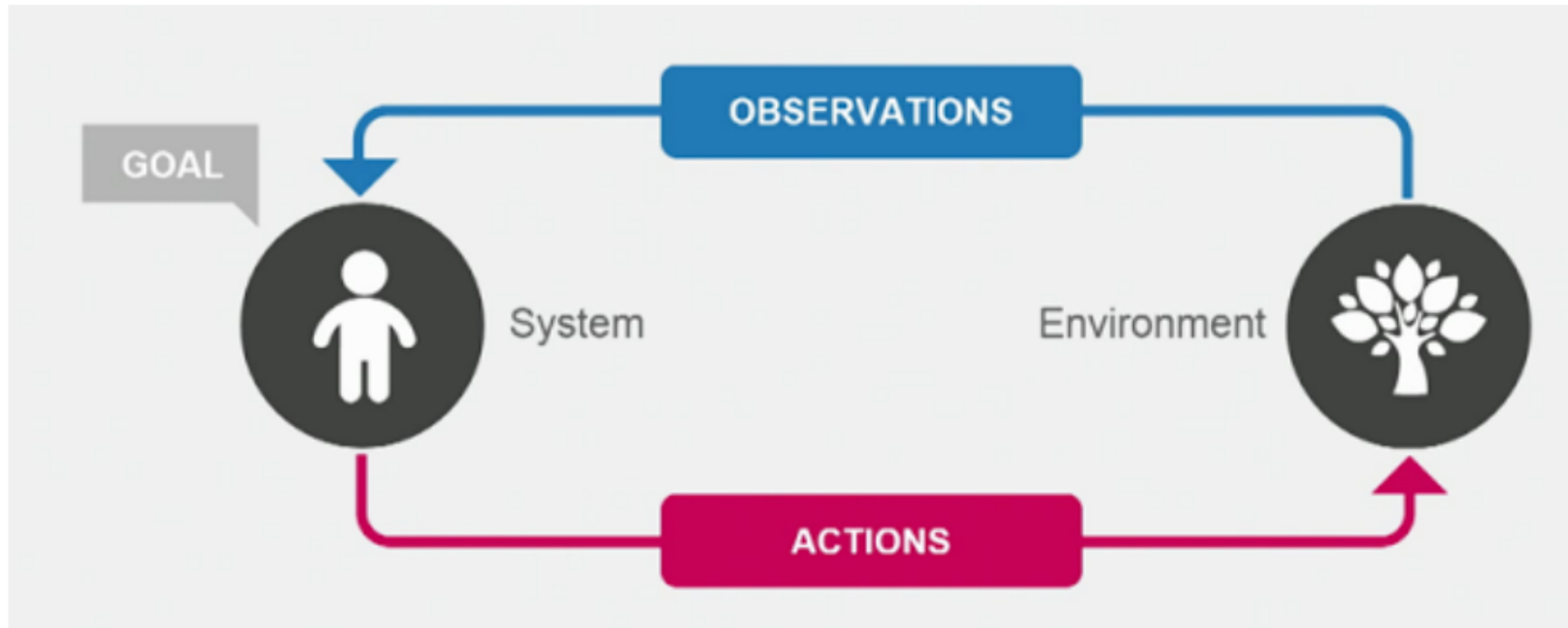
A Medium publication sharing concepts, ideas, and codes.

[More information](#)



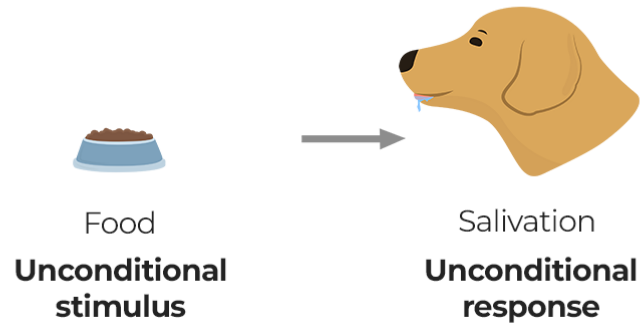
what is the purpose of RL?

- learn how to behave while interacting with the environment
- select actions to get as much reward as possible in the long run

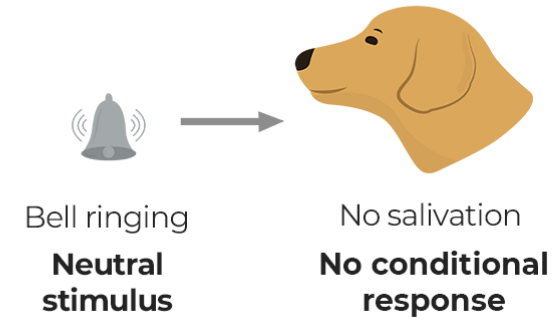


why is it relevant?

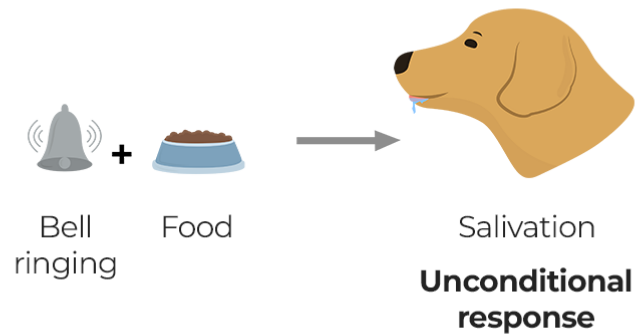
1. Before conditioning



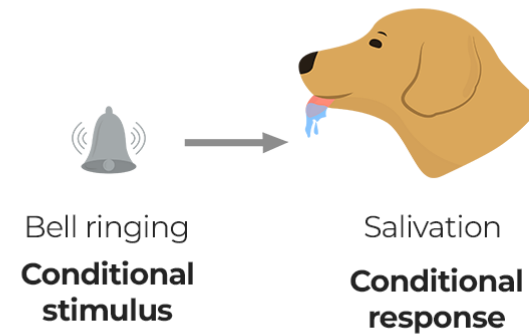
2. Before conditioning



3. During conditioning



4. After conditioning



predicting reward with Markov decision process (MDP)

For a random process, that if we know the value taken by the process at a given time, we won't get any additional information about the future behaviour of the process by gathering more knowledge about the past. Stated in slightly more mathematical terms, for any given time, the conditional distribution of future states of the process given present and past states **depends only on the present state and not at all on the past states.**

$$P(\text{future} \mid \text{present, past}) = P(\text{future} \mid \text{present, } \text{~~past~~})$$

Markov property 

MDP

$$M \equiv (\mathcal{S}, \mathcal{A}, p, r, \gamma)$$

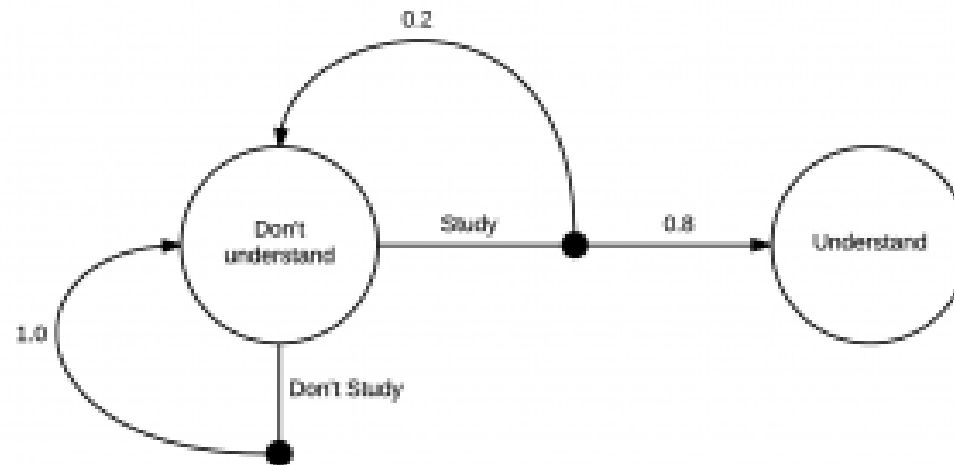
S – State

A – Action

p – transition probability

r – reward

gamma – discounting factor



a grid-world example



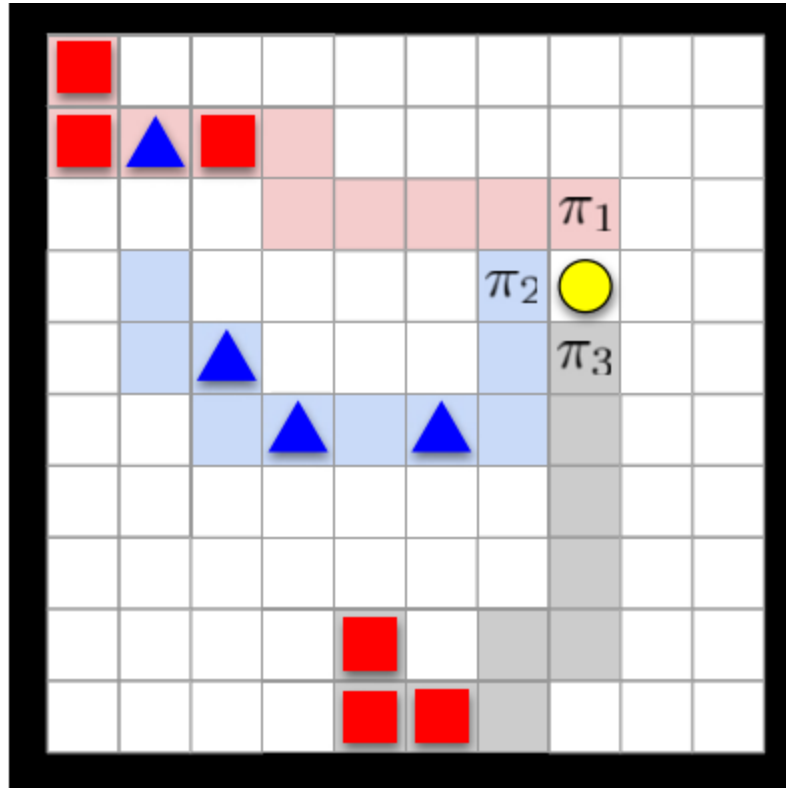
actions

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	

$R_t = -1$
on all transitions

	←	←	↙
↑	↖	↖	↓
↑	↘	↘	↓
↙	→	→	

another grid example



policy (π) and action value (Q)

a policy $\pi : \mathcal{S} \mapsto \mathcal{A}$.

$$Q_r^\pi(s, a) \equiv \mathbb{E}^\pi \left[\sum_{i=0}^{\infty} \gamma^i r(S_{t+i}, A_{t+i}, S_{t+i+1}) \mid S_t = s, A_t = a \right]$$

$$Q_r^\pi(s, a) = \mathbb{E}_{S' \sim p(\cdot | s, a)} \left[\underbrace{r(s, a, S') + \gamma Q_r^\pi(S', \pi(S'))}_{\text{immediate + long-term}} \right].$$

immediate + long-term



Bellman Equation

short recap (in case you already got lost)

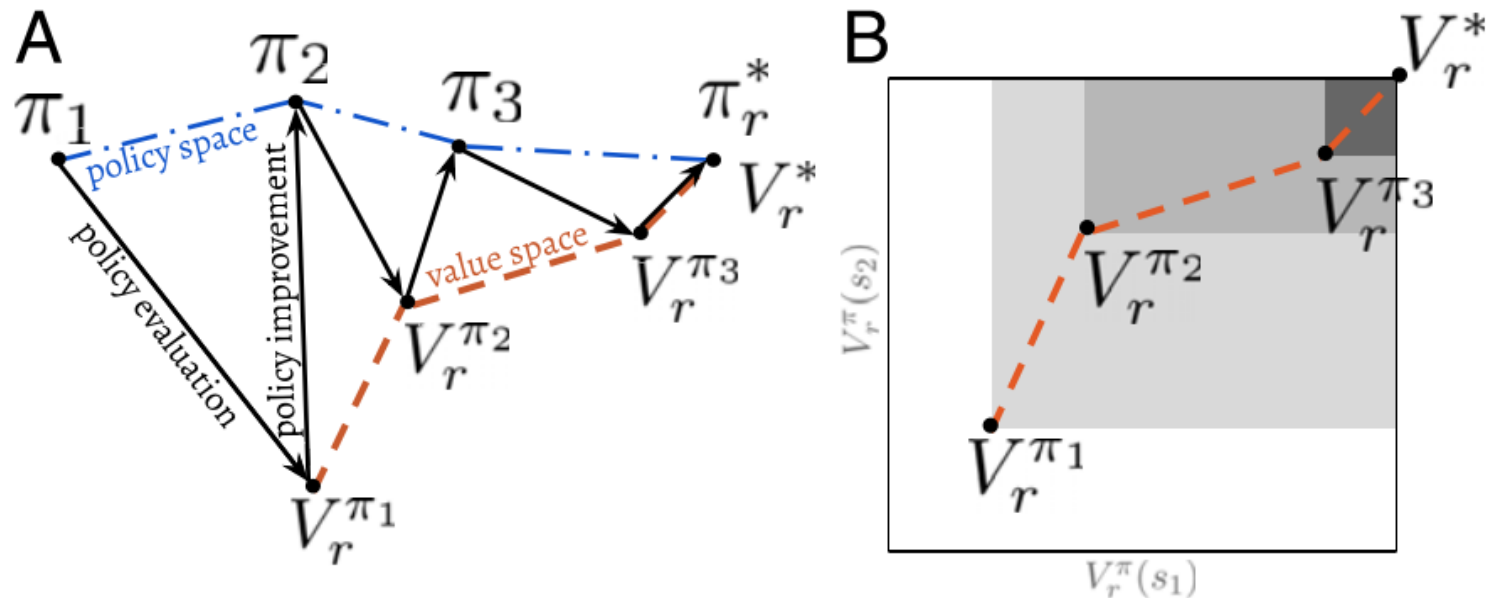
- **Policy:** defines how an agent behave, a mapping from perceived states to available action, stimulus-response associations, policies may be stochastic
- **Reward function:** defines the goal in an RL problem, a mapping from perceived state to a single number indicating the (short-term) desirability of that state, an RL agent sole objective is to maximize the total reward in the long run
- **Value function:** total amount of reward that can be expected in the future starting in that state, indicates the long-term desirability of a state, values are predictions of reward

policy updates: evaluation + improvement

Definition 1. “Policy evaluation” is the computation of Q_r^π , the value function of policy π on task r .

Definition 2. Given a policy π and a task r , “policy improvement” is the definition of a policy π' such that

$$Q_r^{\pi'}(s, a) \geq Q_r^\pi(s, a) \text{ for all } (s, a) \in \mathcal{S} \times \mathcal{A}. \quad [2]$$



is it good enough?

- Challenge identified: tons of data is required

- Solution: **divide-and-conquer**

- generalized policy evaluation (GPE)

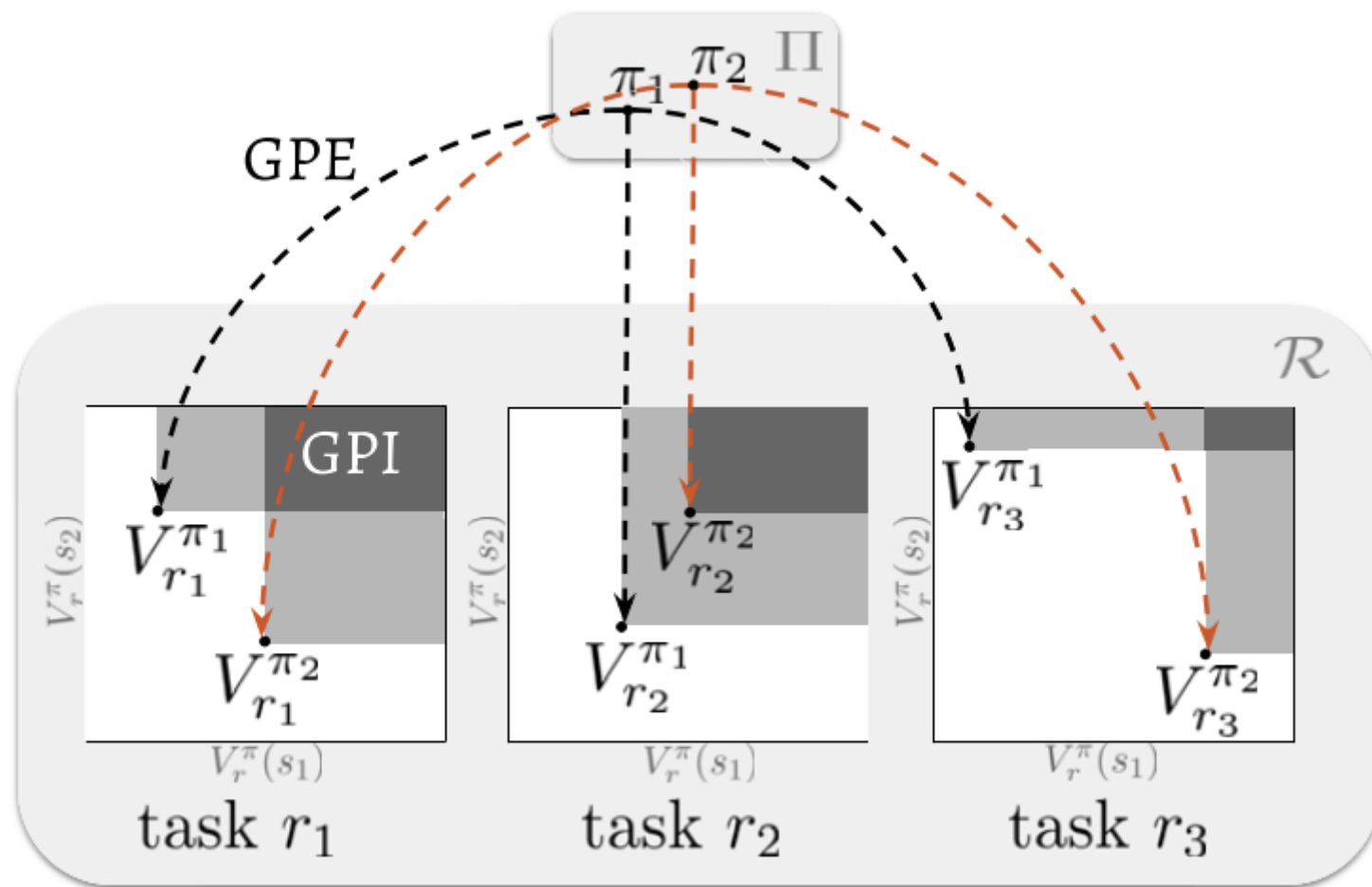
Definition 3. “Generalized policy evaluation” (GPE) is the computation of the value function of a policy π on a set of tasks \mathcal{R} .

- generalized policy improvement (GPI)

Definition 4. Given a set of policies Π and a task r , “generalized policy improvement” (GPI) is the definition of a policy π' such that

$$Q_r^{\pi'}(s, a) \geq \sup_{\pi \in \Pi} Q_r^{\pi}(s, a) \text{ for all } (s, a) \in \mathcal{S} \times \mathcal{A}. \quad [5]$$

GPE & GPI



fast GPE with Successor Features (SF)

IMHO: the most important contribution of the paper

- task \rightarrow GPE \rightarrow value

Let $\phi: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}^d$ be an arbitrary function whose output we will see as “features.” Then, for any $\mathbf{w} \in \mathbb{R}^d$, we have a task defined as

$$r_{\mathbf{w}}(s, a, s') = \phi(s, a, s')^\top \mathbf{w}, \quad [6]$$

Following Barreto et al. (28), we define the “successor features” (SFs) of policy π as

$$\psi^\pi(s, a) \equiv \mathbb{E}^\pi \left[\sum_{i=0}^{\infty} \gamma^i \phi(S_{t+i}, A_{t+i}, S_{t+i+1}) \mid S_t = s, A_t = a \right].$$

$$\mathcal{R}_\phi \equiv \{r_{\mathbf{w}} = \phi^\top \mathbf{w} \mid \mathbf{w} \in \mathbb{R}^d\}$$

$$\begin{aligned} \psi^\pi(s, a)^\top \mathbf{w} &= \mathbb{E}^\pi \left[\sum_{i=0}^{\infty} \gamma^i \phi(S_{t+i}, A_{t+i}, S_{t+i+1})^\top \mathbf{w} \mid S_t = s, A_t = a \right] \\ &= \mathbb{E}^\pi \left[\sum_{i=0}^{\infty} \gamma^i r_{\mathbf{w}}(S_{t+i}, A_{t+i}, S_{t+i+1}) \mid S_t = s, A_t = a \right] \\ &= Q_{r_{\mathbf{w}}}^\pi(s, a) \equiv Q_{\mathbf{w}}^\pi(s, a). \end{aligned} \quad [7]$$

That is, the computation of the value function of policy π on task $r_{\mathbf{w}}$ is reduced to the inner product $\psi^\pi(s, a)^\top \mathbf{w}$. Since this is true for any task $r_{\mathbf{w}}$, SFs provide a mechanism to implement a very efficient form of GPE over the set \mathcal{R}_ϕ (cf. Definition 3).

\mathcal{R}_ϕ is the linear space spanned by the d features ϕ_i .

what is the weight here?

Since each component of \mathbf{w} weighs one of the features $\phi_i(s, a, s')$, changing them can intuitively be seen as setting the agent's current "preferences." For example, the vector $\mathbf{w} = [0, 1, -2]^\top$ indicates that the agent is indifferent to feature ϕ_1 and wants to seek feature ϕ_2 while avoiding feature ϕ_3 with twice the impetus. Specific instantiations of π_Ψ can behave in ways that are very different from its constituent policies $\pi \in \Pi$. We can draw a parallel with nature if we think of features as concepts like water or food and note how much the desire for these items can affect an animal's behavior. Analogies aside, this sort

grid case with fast RL

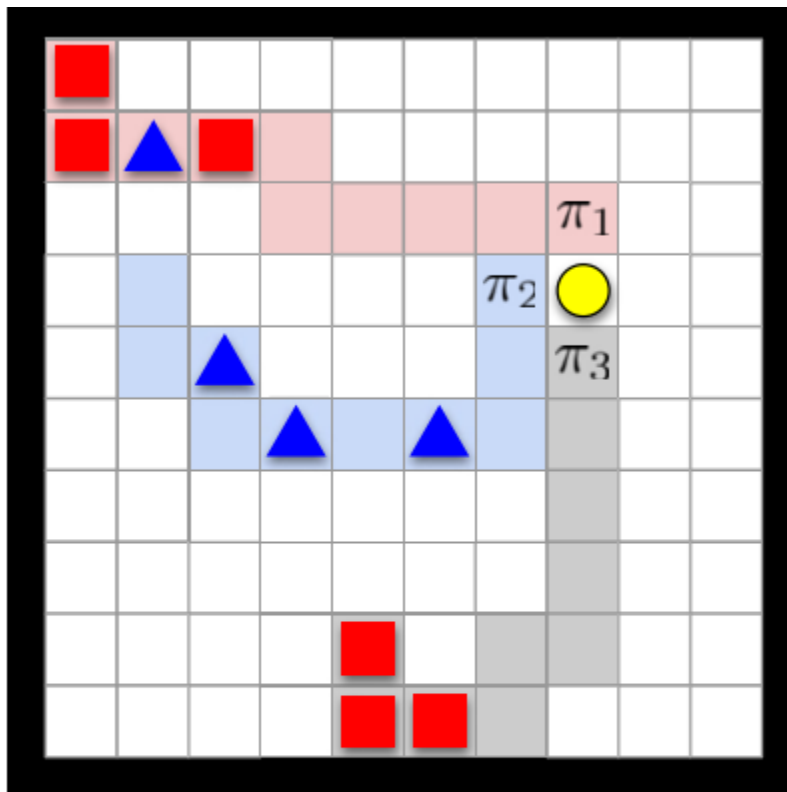


Fig. 4. Depiction of the environment used in the experiments. The shape of the objects (square or triangle) represents their type; the agent is depicted as a circle. We also show the first 10 steps taken by 3 policies, π_1 , π_2 , and π_3 , that would perform optimally on tasks $\mathbf{w}_1 = [1, 0]$, $\mathbf{w}_2 = [0, 1]$, and $\mathbf{w}_3 = [1, -1]$ for any discount factor $\gamma \geq 0.5$.

performance

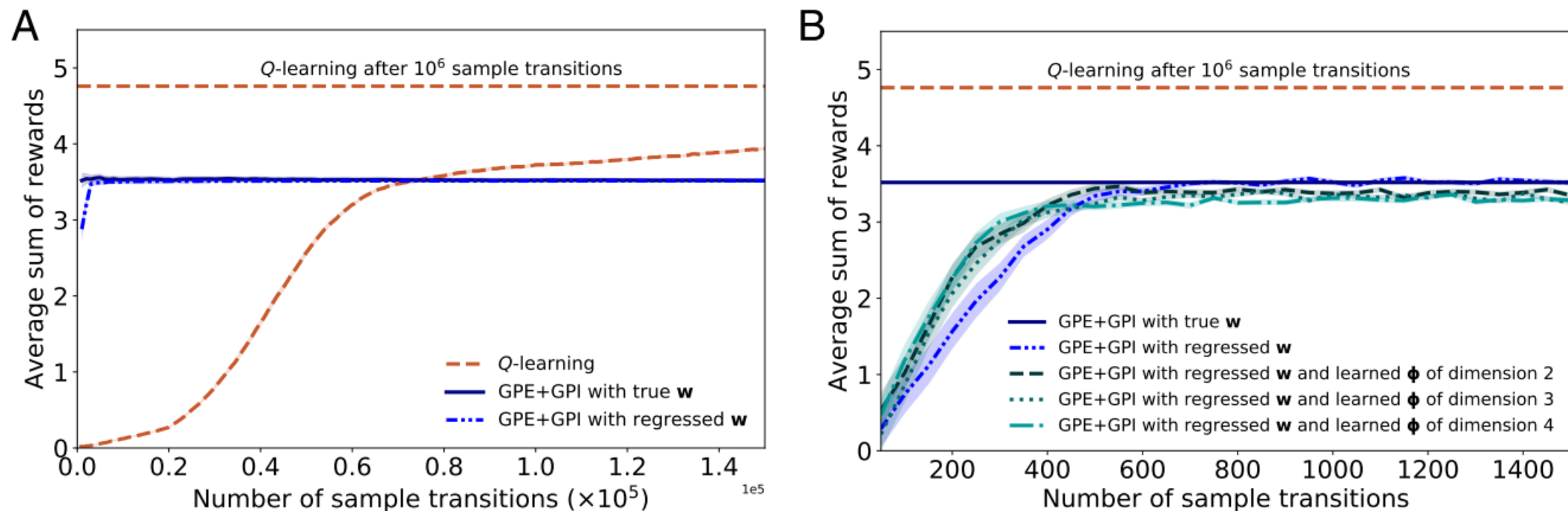
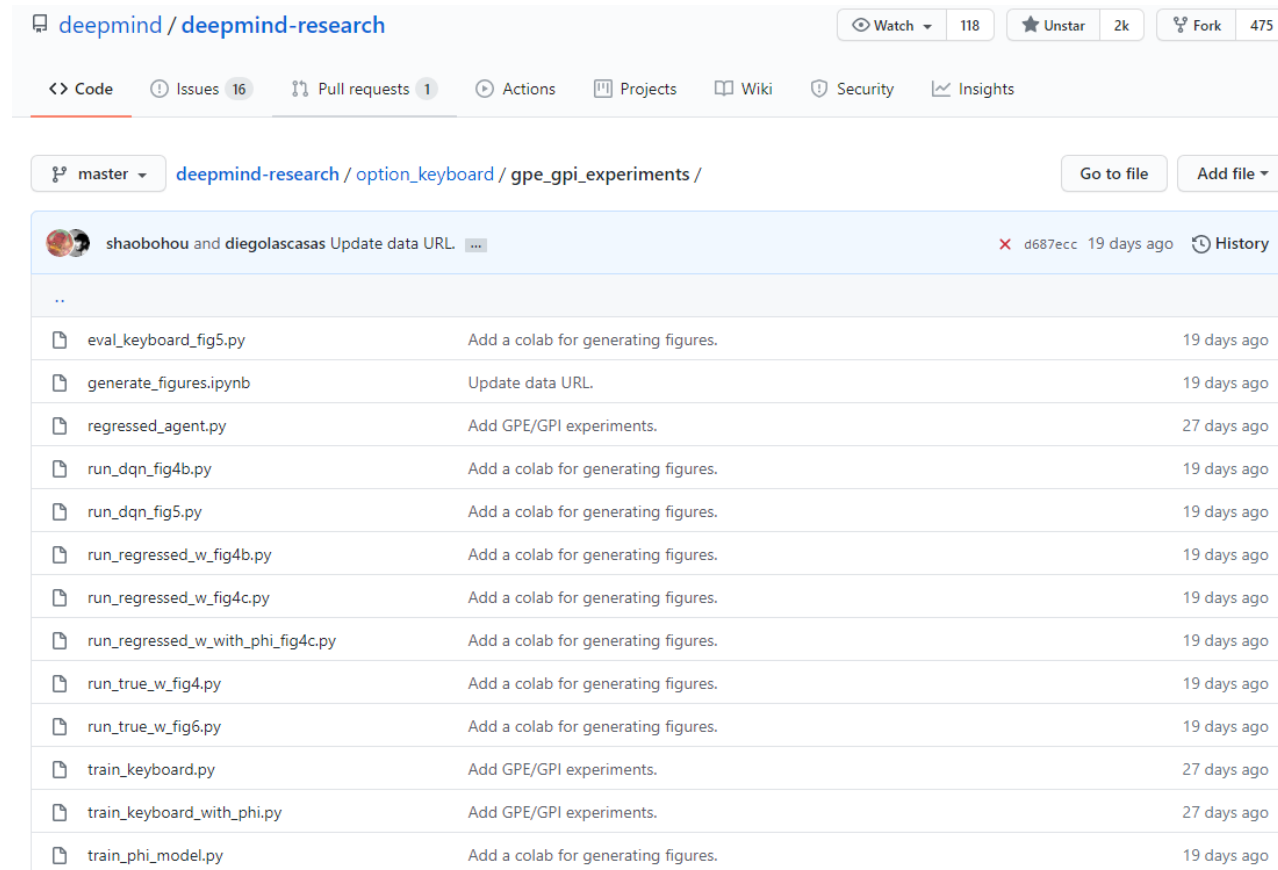


Fig. 5. Average sum of rewards on task $\mathbf{w}_3 = [1, -1]$. GPE and GPI used $\Pi_{12} = \{\pi_1, \pi_2\}$ as the base policies and the corresponding SFs consumed 5×10^5 sample transitions to be trained each. *B* is a zoomed-in version of *A* showing the early performance of GPE and GPI under different setups. The results reflect the best performance of each algorithm over multiple parameter configurations ([SI Appendix](#)). Shaded regions are one standard error over 100 runs.

what if w and ϕ are not available?

- w can be optimized (gradient descent)
- ϕ can be inferred from data



The screenshot shows the GitHub interface for the repository `deepmind / deepmind-research`. The repository has 118 watchers, 2k stars, and 475 forks. The navigation bar includes links for Code, Issues (16), Pull requests (1), Actions, Projects, Wiki, Security, and Insights.

The selected branch is `master`. The current path is `deepmind-research / option_keyboard / gpe_gpi_experiments /`. There are buttons for "Go to file" and "Add file".

A commit history table is displayed, showing the following files and their commit messages:

File	Commit Message	Time
eval_keyboard_fig5.py	Add a colab for generating figures.	19 days ago
generate_figures.ipynb	Update data URL.	19 days ago
regressed_agent.py	Add GPE/GPI experiments.	27 days ago
run_dqn_fig4b.py	Add a colab for generating figures.	19 days ago
run_dqn_fig5.py	Add a colab for generating figures.	19 days ago
run_regressed_w_fig4b.py	Add a colab for generating figures.	19 days ago
run_regressed_w_fig4c.py	Add a colab for generating figures.	19 days ago
run_regressed_w_with_phi_fig4c.py	Add a colab for generating figures.	19 days ago
run_true_w_fig4.py	Add a colab for generating figures.	19 days ago
run_true_w_fig6.py	Add a colab for generating figures.	19 days ago
train_keyboard.py	Add GPE/GPI experiments.	27 days ago
train_keyboard_with_phi.py	Add GPE/GPI experiments.	27 days ago
train_phi_model.py	Add a colab for generating figures.	19 days ago

conclusion

policy improvement and policy evaluation. The generalized version of these operations allow one to leverage the solution of some tasks to speed up the solution of others. If the reward function of a task can be well approximated as a linear combination of the reward functions of tasks previously solved, we can reduce a reinforcement-learning problem to a simpler linear regression. When this is not the case, the agent can still exploit the task solutions by using them to interact with and learn about the environment. Both strategies considerably reduce the amount of data needed to solve a reinforcement-learning problem.

shameless self promotion



lei.zhang@univie.ac.at



<https://lei-zhang.net/>



[@lei_zhang_lz](https://twitter.com/lei_zhang_lz)



[@zhang-lei-44-62](https://wechat.id/@zhang-lei-44-62)



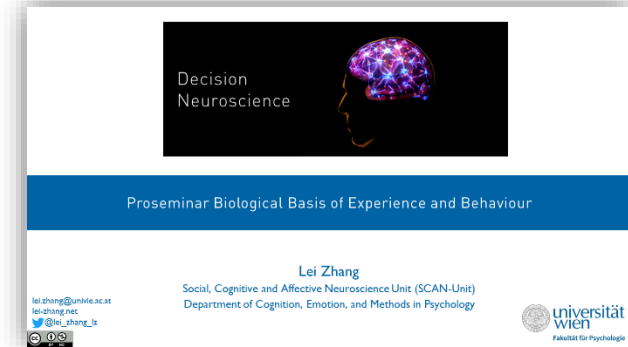
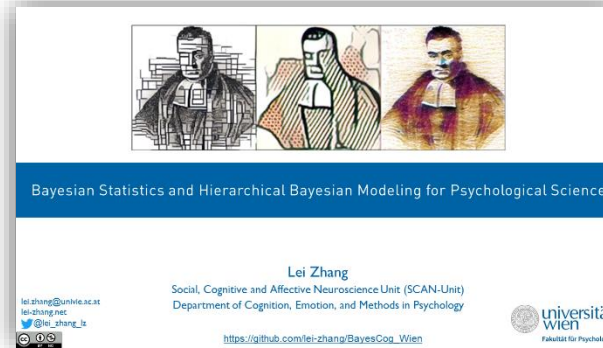
[@leizhang](https://space.bilibili.com/leizhang) 认知神经科学



[@LeiZhang](https://www.youtube.com/@LeiZhang)



[@lei-zhang](https://github.com/lei-zhang)



Using reinforcement learning models in social neuroscience: frameworks, pitfalls and suggestions of best practices

Lei Zhang,^{1,2} Lukas Lengersdorff,^{1,2} Nace Mikus,¹ Jan Gläscher,³ and Claus Lamm^{1,2,4}

¹Neuropsychopharmacology and Biopsychology Unit, Department of Cognition, Emotion, and Methods in Psychology, Faculty of Psychology, University of Vienna, Vienna 1010, Austria, ²Social, Cognitive and Affective Neuroscience Unit, Department of Cognition, Emotion, and Methods in Psychology, Faculty of Psychology, University of Vienna, Vienna 1010, Austria, ³Institute of Systems Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg 20246, Germany and ⁴Vienna Cognitive Science Hub, University of Vienna, Vienna 1010, Austria

<https://academic.oup.com/scan/article/15/6/695/5864690>

RESEARCH

Revealing Neurocomputational Mechanisms of Reinforcement Learning and Decision-Making With the hBayesDM Package

Woo-Young Ahn¹, Nathaniel Haines¹, and Lei Zhang²

¹Department of Psychology, The Ohio State University, Columbus, OH

²Institute for Systems Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

Keywords: reinforcement learning, decision-making, hierarchical Bayesian modeling, model-based fMRI

https://www.mitpressjournals.org/doi/full/10.1162/CPSY_a_00002

Thank you!