

Return Decomposition for Delayed Rewards

Une intuition sur *RUDDER*

Visionner sur *YouTube* 

GLO-7030 Apprentissage par réseaux de neurones profonds [revision 3.4]

Francois-Alexandre Tremblay et Luc Coupal

{francois-alexandre.tremblay.1, luc.coupal.1}@ulaval.ca



Université Laval

2021-04-09

Rudder : Return Decomposition for Delayed Rewards

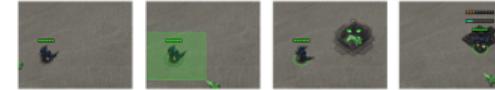
ARJONA-MEDINA et al., 2018

NeurIPS 2019

« We propose Rudder, a novel reinforcement learning approach for delayed rewards in finite Markov decision processes (MDPs). Rudder is based on two main ideas : (i) a backward view approach and (ii) the concept of return-equivalent MDPs. Forward view approaches, like deep Q-networks (DQNs) or Monte Carlo tree search (MCTS), have to average over a large number of probabilistic future state-action paths that increases exponentially with the delay of the reward. Backward view approaches, in contrast, identify actions and states that cause a delayed reward by analyzing already chosen paths. Rudder's backward view transforms tasks of estimating future returns into regression tasks at which deep learning excels. Rudder decomposes the return into new, non-delayed rewards by redistributing the original reward across the episode, thereby creating a new MDP that is return-equivalent to the original MDP. "Return-equivalent MDPs" is a new concept ensuring that both MDPs have the same optimal policies. If the return decomposition is optimal, then the new MDP will be stripped of any delayed rewards. In this case, action-value estimates are unbiased and the future expected return is always zero. On several artificial tasks with delayed rewards Rudder significantly outperforms Monte Carlo, MCTS, temporal difference, $TD(\lambda)$, and reward shaping approaches. Rudder is even exponentially faster than the last three. Rudder on top of a Proximal Policy Optimization (PPO) baseline improves the scores on Atari games and excels for delayed rewards. For long delayed rewards, as in Bowling, Frostbite, PrivateEye, and Venture, Rudder yields exceptional results. »

Deep Reinforcement Learning (DRL) en soixante secondes

AlphaStar



Human Actions	IDLE	Left_Click_Hold (p1)	Press [B] + [B]	IDLE
		Release (p2)	Left_Click (p3)	
Agent Actions	no_op	select_rect(p1, p2)	build_supply(p3)	no_op
Available Actions	no_op rectangle select	Base action Point Point no_op	Base action Point Point no_op rectangle select build supply	no_op rectangle select build supply

Image : DeepMind

- DeepMind and Blizzard open StarCraft II as an AI research environment [▶ Publication](#)
- StarCraft II : A New Challenge for Reinforcement Learning [2] [▶ Publication](#)
- AlphaStar : Mastering the Real-Time Strategy Game StarCraft II [3] [▶ Publication](#)
- AlphaStar : The inside story [▶ Vidéo \(5 min\)](#)

AlphaGo



Image : AlphaGo – The Movie

- *Mastering the game of Go with deep neural networks and tree search* [4] ▶ Publication
- *AlphaGo – The Movie* ▶ Vidéo (90 min)

Stanford University autonomous helicopter



Image : Stanford Autonomous Helicopter

- *An Application of Reinforcement Learning to Aerobatic Helicopter Flight (LQR, 2006) [5]* [▶ Publication](#)
- *Stanford University autonomous helicopter* [▶ Projet](#) [▶ Vidéo](#)

Composable Deep Reinforcement Learning for Robotic Manipulation

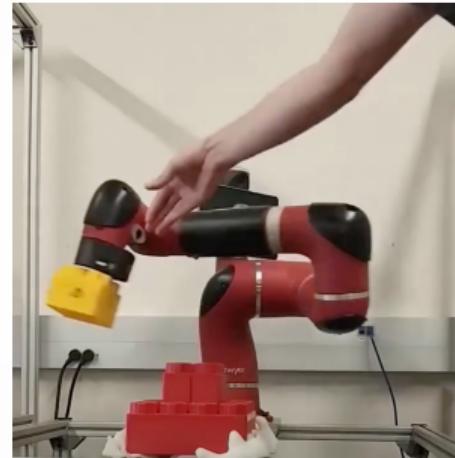
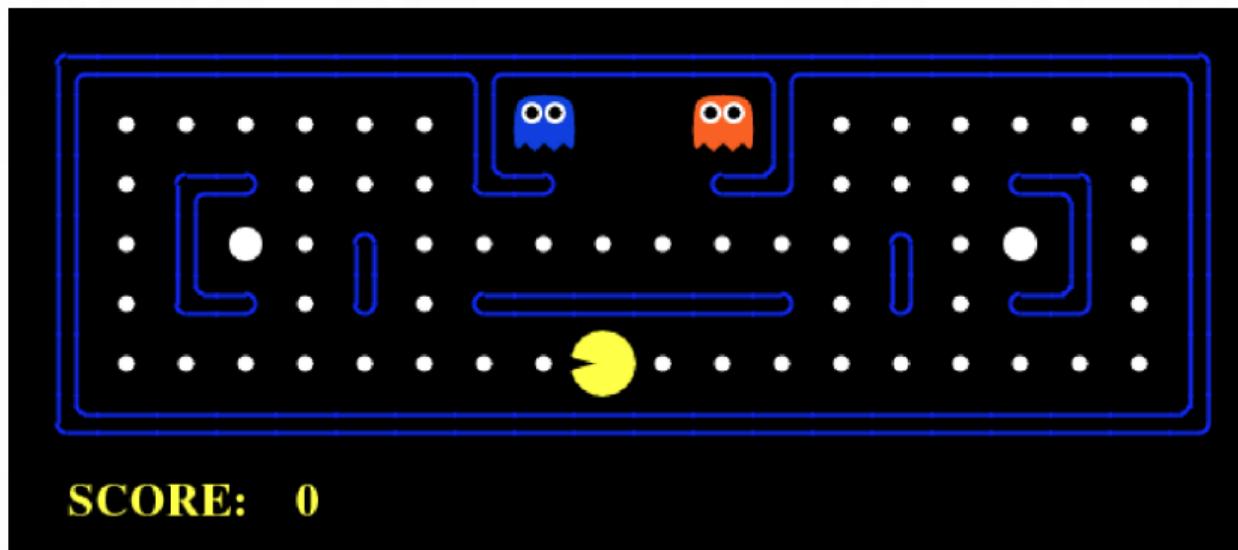


Image : Haarnoja et al.

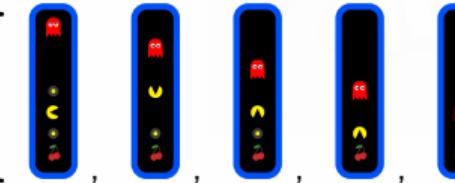
- *Composable Deep Reinforcement Learning for Robotic Manipulation* [6] [▶ Publication](#) [▶ Vidéo](#)
- *Reinforcement learning with deep energy-based policies (SQL)* [7] [▶ Publication](#)



Vidéo : UC Berkeley CS188 Intro to AI – The Pac-Man Projects



Composante clés (environnement *Pac-Man*)

$$\text{L'état } \mathbf{s}_t \in \left\{ \begin{array}{c} \text{, } \\ \dots \end{array} \right\}$$


$$\text{L'action } \mathbf{a}_t \in \left\{ \begin{array}{c} \uparrow \\ \leftarrow \\ \rightarrow \\ \downarrow \end{array} \right\}$$

$$\text{La récompense } r_t \in \left\{ \begin{array}{c} \text{, } \\ \text{, } \\ \text{, } \\ \text{, } \\ \text{, } \end{array} \right\}$$


$$\text{L'agent } \textcircled{\text{P}} := \text{PAC-MAN} \quad \text{« I want more } \text{cherries} \text{ »}$$

Composante clés (environnement *Pac-Man*)

L'état $s_t \in \{$  $\}$

L'action $a_t \in \{ \uparrow, \downarrow, \leftarrow, \rightarrow \}$

La récompense $r_t \in \{ \text{yellow circle}, \text{cherries}, \emptyset, \dots \}$

L'agent  \coloneqq  « *I want more cherries* »

Composante clés (environnement *Pac-Man*)

L'état $s_t \in \{$  $\}$

L'action $a_t \in \{$  $\}$

La récompense $r_t \in \{$  $\}$

L'agent  \coloneqq  « *I want more*  »

Composante clés (environnement *Pac-Man*)

L'état $s_t \in \{$  $\}$

L'action $a_t \in \{$  $\}$

La récompense $r_t \in \{$  $\}$

L'agent  \coloneqq  « *I want more*  »

Composante clés (environnement *Pac-Man*)

$$\text{L'état } \mathbf{s}_t \in \left\{ \begin{array}{c} \text{, } \\ \dots \end{array} \right\}$$

$$\text{L'action } \mathbf{a}_t \in \left\{ \begin{array}{c} \uparrow \\ \downarrow \\ \leftarrow \\ \rightarrow \end{array} \right\}$$

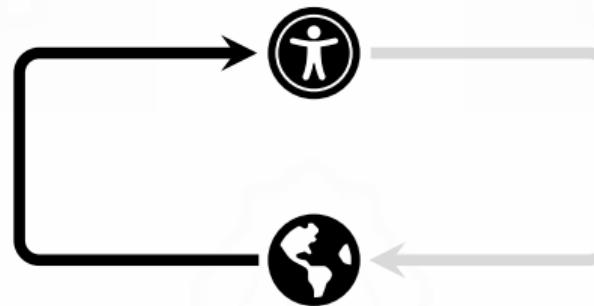
$$\text{La récompense } r_t \in \left\{ \begin{array}{c} \odot \\ \text{cherries} \\ \emptyset \end{array} \right\}$$

L'agent := « *I want more cherries* »

Temps $t = 1$

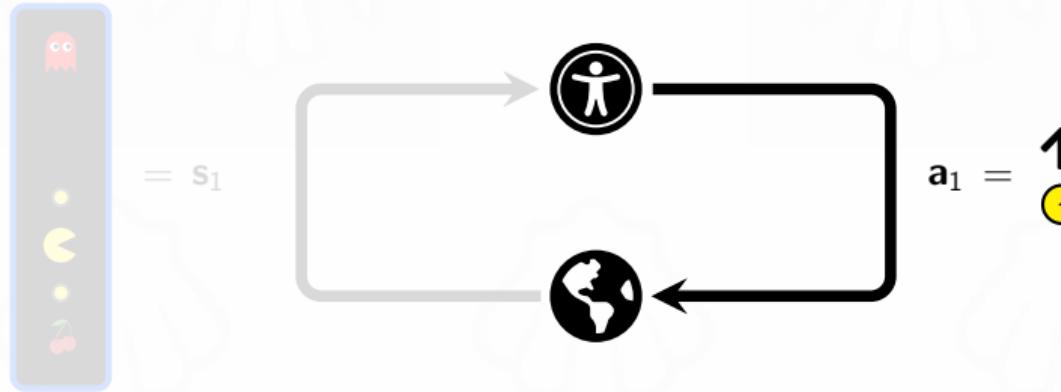


$$= \mathbf{s}_1$$

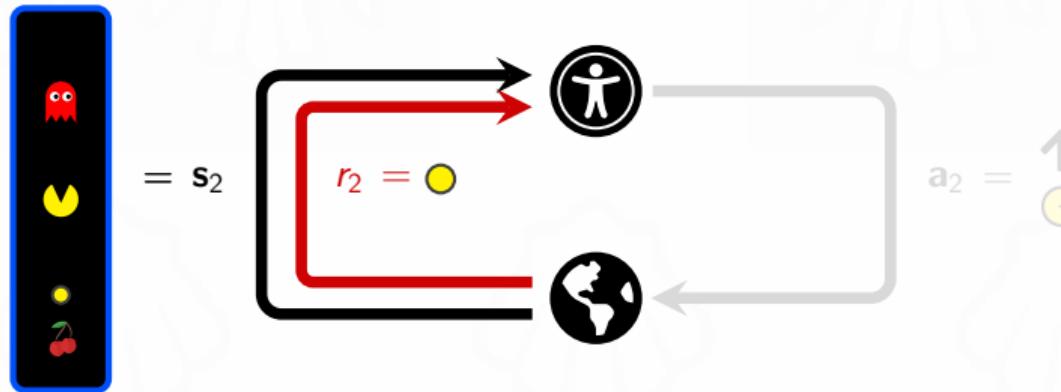


$$\text{trajectoire} = \left(\begin{array}{c} \mathbf{s}_1 \\ \vdots \end{array} \right)$$

Temps $t = 1$

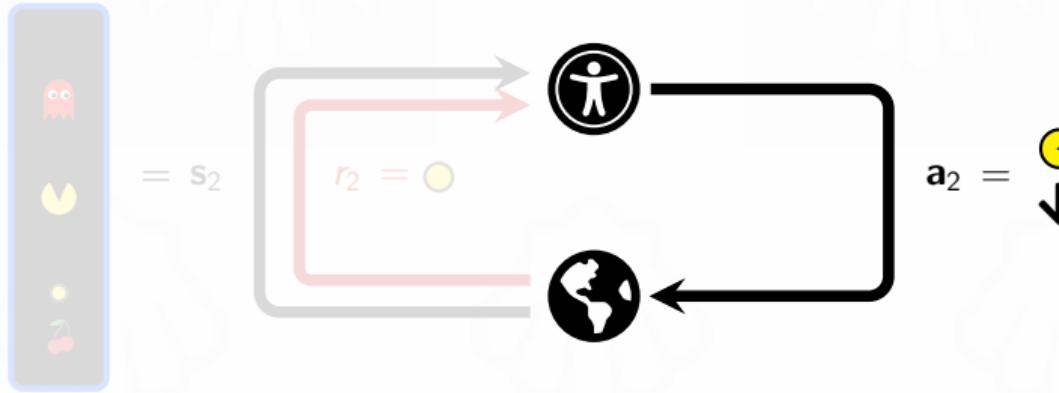


$$\text{trajectoire} = \left(\begin{array}{c} \text{ (vertical bar with icons), } \\ \text{ (yellow circle with up arrow), } \end{array} \right)$$

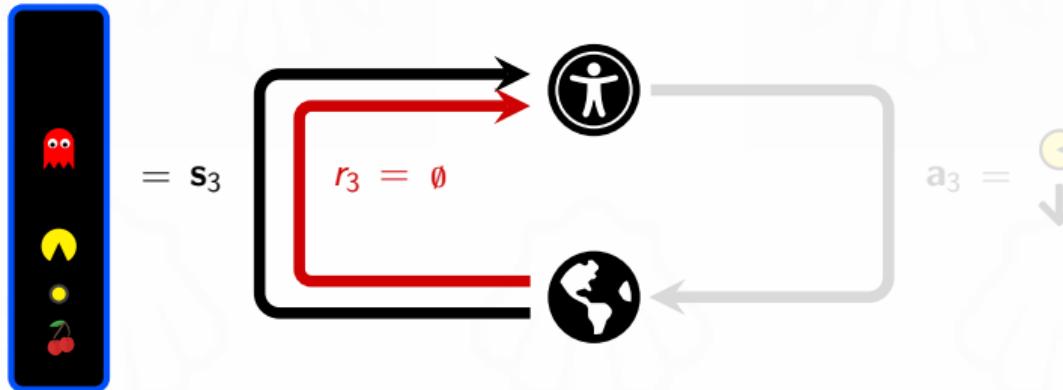
Temps $t = 2$ 

$$\text{trajectoire} = \left(\begin{array}{c} s_1 \\ a_1 \\ r_1 \\ s_2 \\ a_2 \\ r_2 \\ \vdots \end{array} \right)$$

Temps $t = 2$

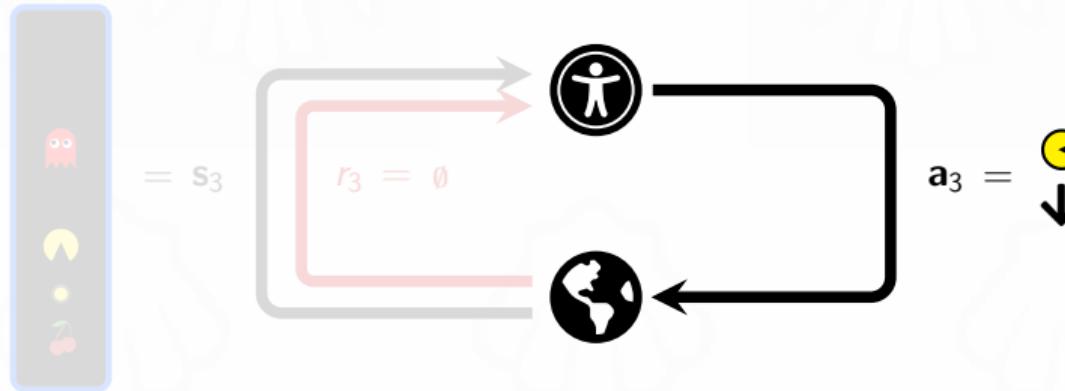


$$\text{trajectoire} = \left(\begin{array}{c} \text{[Pac-Man, Power Pellets, Small Circle, Power Pellet]} \\ \uparrow \\ \text{[Pac-Man, Power Pellets, Small Circle, Power Pellet]} \end{array}, \bullet, \text{[Pac-Man, Power Pellets, Small Circle, Power Pellet]}, \bullet \downarrow \right)$$

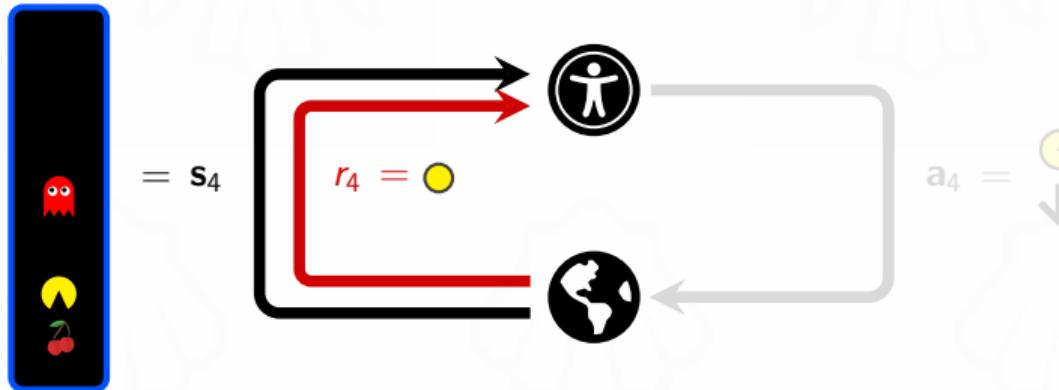
Temps $t = 3$ 

trajectoire = $($, , , , , , , $)$

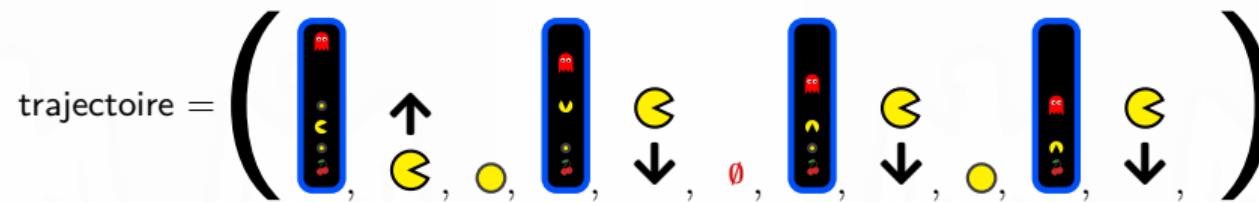
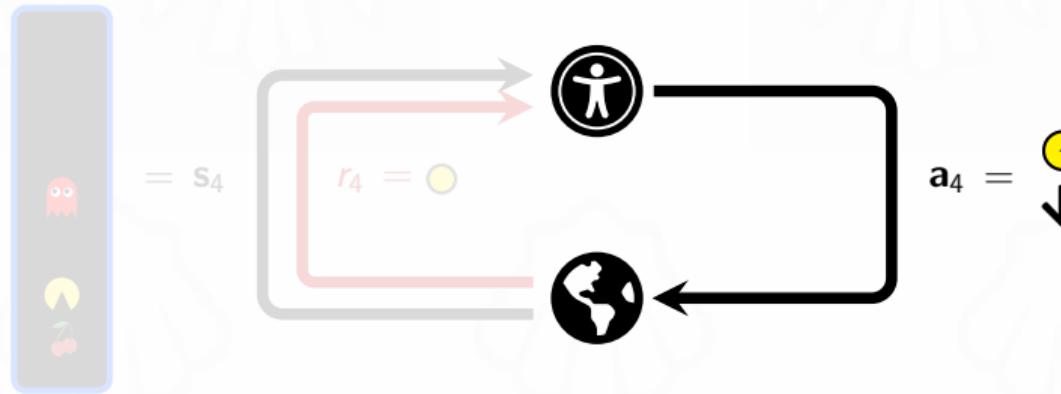
Temps $t = 3$



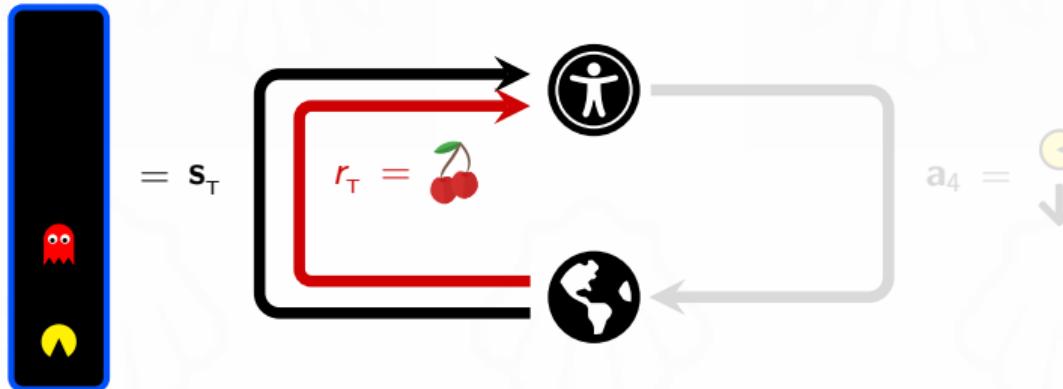
$$\text{trajectoire} = \left(\begin{array}{c} \text{[Pac-Man, ghost, power pellet, flower]}, \\ \uparrow, \text{ [Pac-Man, ghost, power pellet, flower]}, \\ \downarrow, \text{ [Pac-Man, ghost, power pellet, flower]}, \\ \emptyset, \text{ [Pac-Man, ghost, power pellet, flower]}, \\ \downarrow \end{array} \right)$$

Temps $t = 4$ 

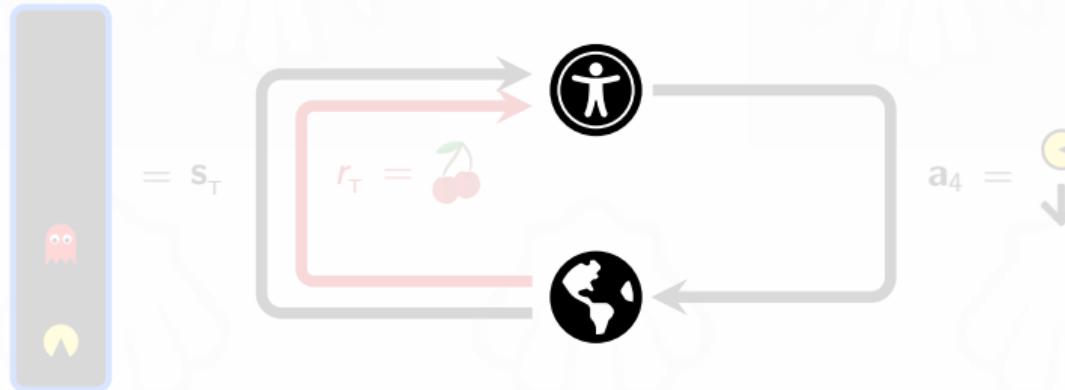
trajectoire = $($
The sequence consists of five vertical Pac-Man screens, each showing the ghost, power pellet, and cherries. Between the screens are three circular arrows indicating actions: an upward arrow above the first screen, a downward arrow below the second screen, and another downward arrow below the third screen. After the third screen, there is a red circle with a black zero symbol, followed by a final downward arrow below the fifth screen.

Temps $t = 4$ 

Temps $t = 5$



Temps $t = 5$



trajectoire = $($, , , , , , , , , , , END $)$

Natural Language Processing (NLP)

$\left[\text{"I"}, \text{"love"}, \text{"whisky"}, \text{"a"}, \text{"lot"} \right]$

Reinforcement Learning (RL)

$s_1, a_1, r_2, s_2, a_2, r_3, \dots, s_T, a_T, r_{T+1}$

Natural Language Processing (NLP)

$$f\left(\left["I", "love", "whisky", "a", "lot" \right]; \theta \right) \mapsto "Good\ for\ you"$$

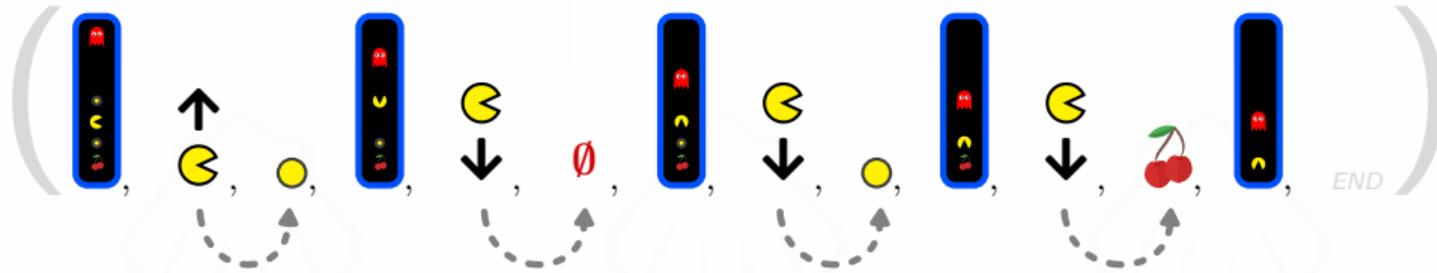
Reinforcement Learning (RL)

$$f\left(s_1, a_1, r_2, s_2, a_2, r_3, \dots, s_T, a_T, r_{T+1}; \theta \right) \mapsto \hat{\pi}_\theta(a_t | s_t)$$

Problématique

Le problème des récompenses non immédiates

Trajectoire *Pac-Man*

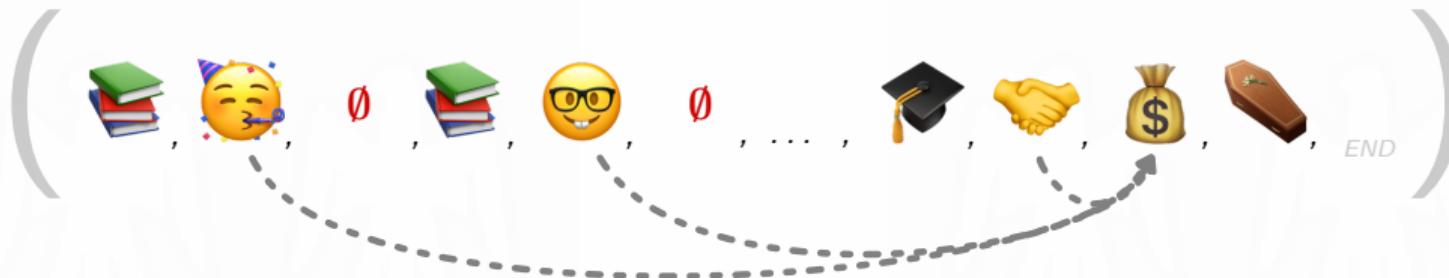


Le problème des récompenses non immédiates

Trajectoire Pac-Man

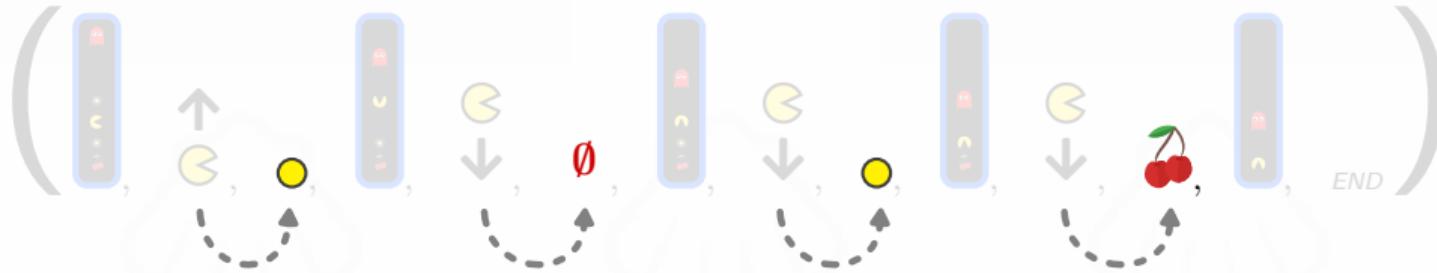


Trajectoire universitaire

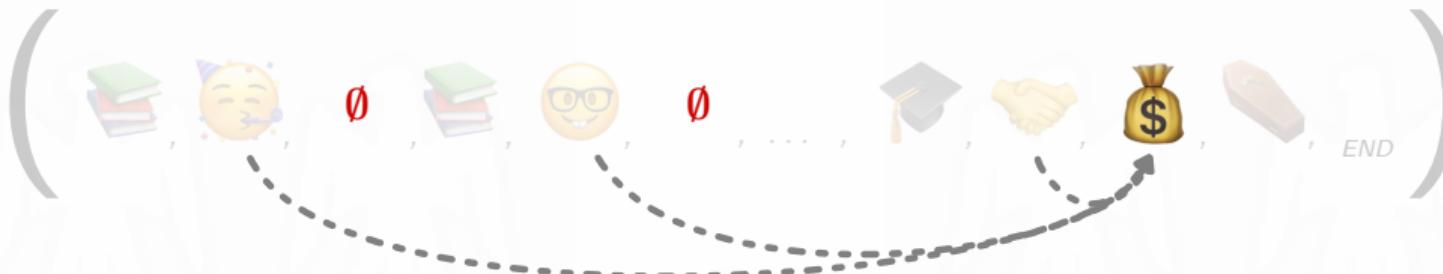


Le problème des récompenses non immédiates

Trajectoire *Pac-Man*



Trajectoire universitaire



Redistribution des récompenses

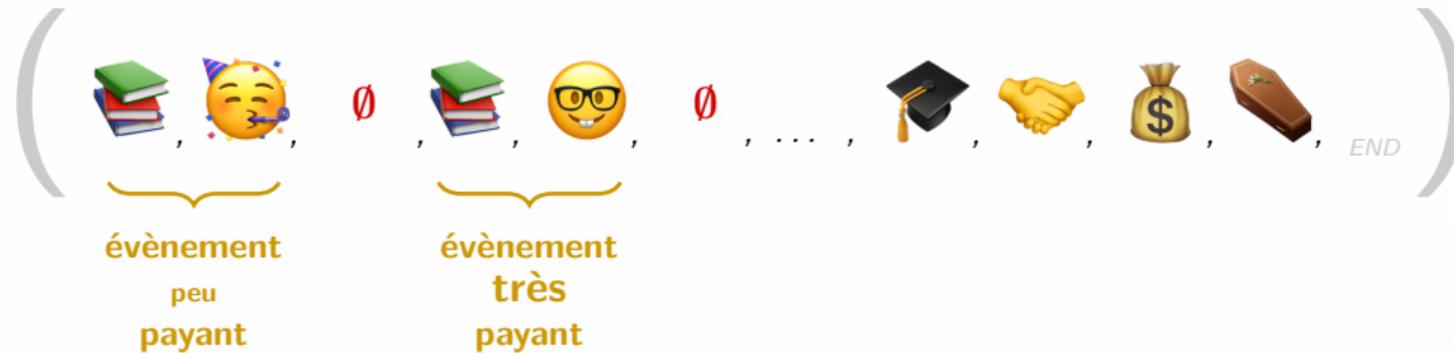
Redistribution des récompenses



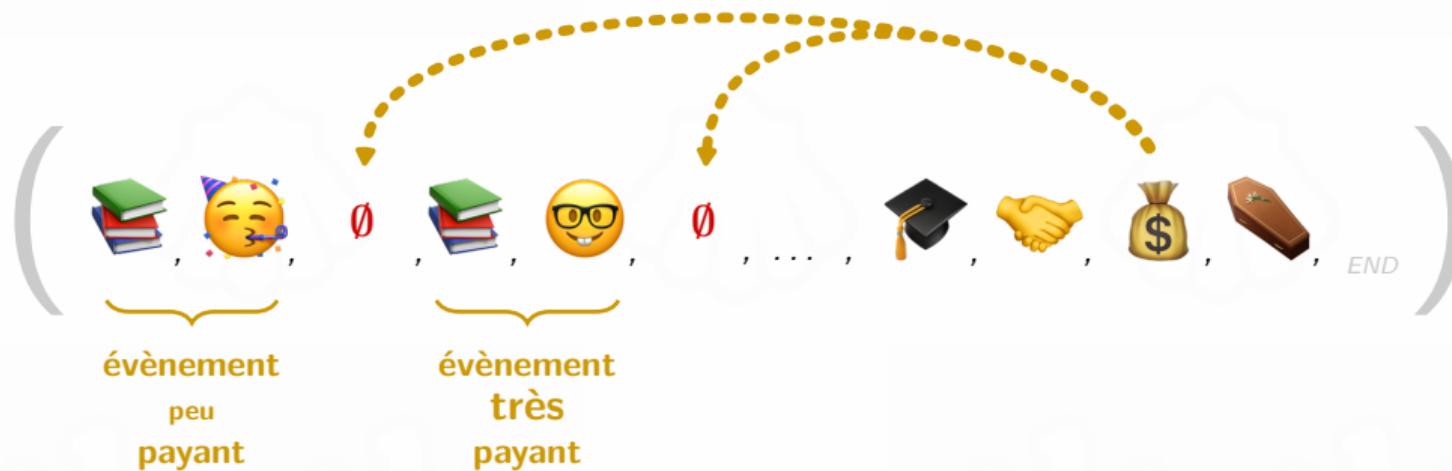
Redistribution des récompenses



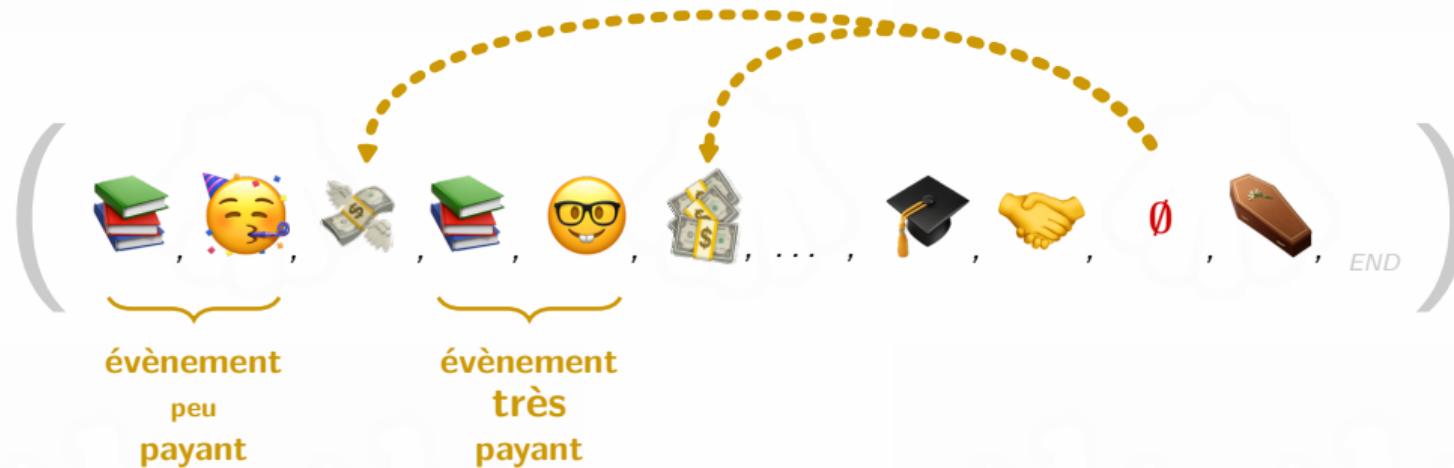
Redistribution des récompenses



Redistribution des récompenses



Redistribution des récompenses



Redistribution des récompenses



Redistribution des récompenses



Redistribution des récompenses

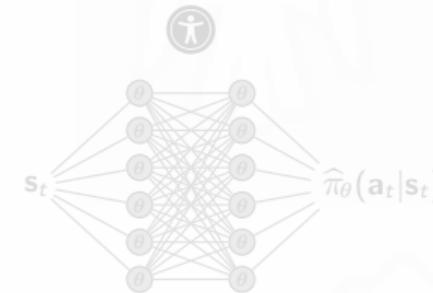
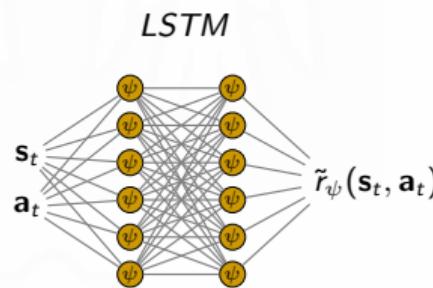


Redistribution des récompenses



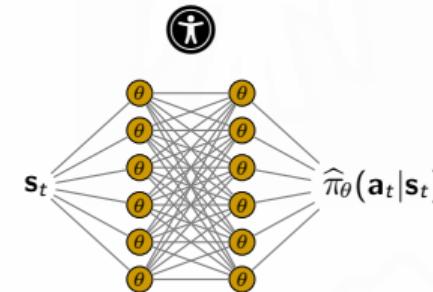
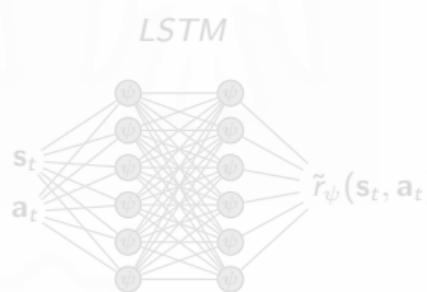
Redistribution des récompenses



**RUDDER**

$$f \left(s_1, a_1, r_2, s_2, a_2, r_3, s_3, \dots, s_T, a_T, r_{T+1}; \psi \right) \mapsto \tilde{r}_\psi(s_t, a_t)$$

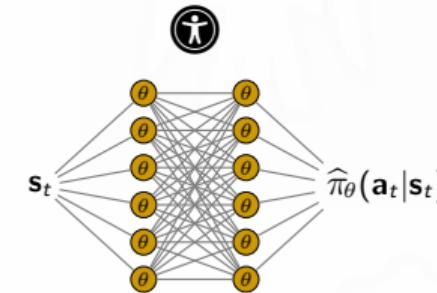
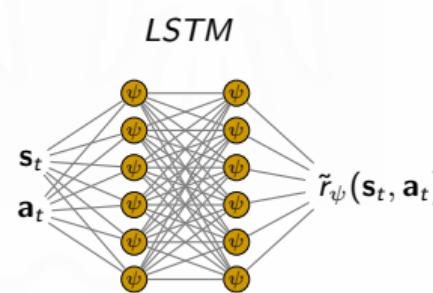
$$f \left(s_1, a_1, r_2, s_2, a_2, r_3, s_3, \dots, s_T, a_T, r_{T+1}; \theta \right) \mapsto \hat{\pi}_\theta(a_t | s_t)$$



Reinforcement Learning (RL) policy gradient version

$$f \left(s_1, a_1, r_2, s_2, a_2, r_3, s_3, \dots, s_T, a_T, r_{T+1}; \psi \right) \mapsto \tilde{r}_\psi(s_t, a_t)$$

$$f \left(s_1, a_1, r_2, s_2, a_2, r_3, s_3, \dots, s_T, a_T, r_{T+1}; \theta \right) \mapsto \hat{\pi}_\theta(a_t | s_t)$$



Reinforcement Learning (RL) + *RUDDER*

$$f \left(s_1, a_1, r_2, s_2, a_2, r_3, s_3, \dots, s_T, a_T, r_{T+1} ; \psi \right) \mapsto \tilde{r}_\psi(s_t, a_t)$$

$$f \left(s_1, a_1, \tilde{r}_\psi(s_1, a_1), s_2, \dots, s_T, a_T, \tilde{r}_\psi(s_T, a_T) ; \theta \right) \mapsto \hat{\pi}_\theta(a_t | s_t)$$



Reinforcement Learning (RL) + **RUDDER**

$$f\left(s_1, a_1, \color{red}r_2, s_2, a_2, \color{red}r_3, s_3, \dots, s_T, a_T, \color{red}r_{T+1}; \psi \right) \mapsto \tilde{r}_\psi(s_t, a_t)$$

$$f\left(s_1, a_1, \color{red}\tilde{r}_\psi(s_1, a_1), s_2, \dots, s_T, a_T, \color{red}\tilde{r}_\psi(s_T, a_T); \theta \right) \mapsto \hat{\pi}_\theta(a_t | s_t)$$

Résultats expérimentaux

Tâche de neutralisation d'explosif

-2	-3	-4	-5	-5	-5
-2	-3	-4	-5	 1000	-5
-2	-3	-4	-5	-5	-5
-2	-3	-4	-4	-4	-4
-2	-3	-3	-3	-3	-3
	-2	-2	-2	-2	-2

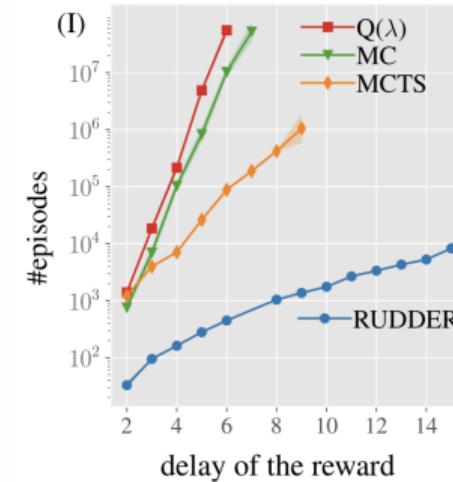
Exemple d'environnement *GridWorld*

Image : *RUDDER - Reinforcement Learning with Delayed Rewards*, ARJONA-MEDINA et al. [1]

Idées clés

⚠ Les problèmes de *RL* avec **récompenses à retardement**
sont **difficiles** à solutionner.

RL + Rudder

- ▶ Trajectoires \longmapsto ***LSTM*** \longmapsto Redistribution de récompenses
- 👍 **Réussit** là où les autres algorithmes de *RL* échouent
- 👍 **Performance** accrue

⚠ Les problèmes de *RL* avec **récompenses à retardement**
sont **difficiles** à solutionner.

RL + Rudder

- ▶ Trajectoires \longmapsto ***LSTM*** \longmapsto Redistribution de récompenses
- 👍 Réussit là où les autres algorithmes de *RL* échouent
- 👍 Performance accrue

Limites du *LSTM*

- 👎 Requiert un jeu de données de trajectoires a priori
- 👎 Performance diminue plus les trajectoires sont longues
- 👎 Temps d'entraînement supplémentaire

Ouvrage de références

ToDo: DONT release

- [1]
- [8]
- [4]
- [3]
- [5]
- [6]

Références I

1. ARJONA-MEDINA, J. A. *et al.* Rudder: Return decomposition for delayed rewards. *arXiv*. ISSN : 23318422. arXiv : 1806.07857 (2018).
2. VINYALS, O. *et al.* StarCraft II: A New Challenge for Reinforcement Learning. arXiv : 1708.04782. <http://arxiv.org/abs/1708.04782> (2017).
3. DEEPMIND. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. DeepMind. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/> (2019).
4. SILVER, D. *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484-489. ISSN : 14764687. <http://dx.doi.org/10.1038/nature16961> (2016).
5. ABBEEL, P., COATES, A., QUIGLEY, M. & NG, A. Y. An Application of Reinforcement Learning to Aerobatic Helicopter Flight. *Advances in Neural Information Processing Systems* (2006).
6. HAARNOJA, T. *et al.* Composable Deep Reinforcement Learning for Robotic Manipulation. in *Proceedings - IEEE International Conference on Robotics and Automation* (2018). ISBN : 9781538630815. arXiv : 1803.06773.

Références II

7. HAARNOJA, T., TANG, H., ABBEEL, P. & LEVINE, S. Reinforcement learning with deep energy-based policies. *34th International Conference on Machine Learning, ICML 2017* 3, 2171-2186. arXiv : 1702.08165 (2017).
8. ARJONA-MEDINA, J. A. *et al.* SUPPLEMENTS to the manuscript “ RUDDER : Return Decomposition for Delayed Rewards ”.