Credit Assignment in Deep Reinforcement Learning

Thomas Mesnard

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAX155

Thèse de doctorat

Attribution de crédit pour l'apprentissage par renforcement dans des réseaux profonds

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Mathématiques et informatique

Thèse présentée et soutenue à Paris, le 21/12/2023, par

MR THOMAS MESNARD

Composition du Jury :

Aurélien Garivier Professeur, École Normale Supérieure de Lyon	Président - Rapporteur
Timothy Lillicrap Professeur, University College London	Rapporteur
Doina Precup Professeur, McGill University	Examinateur
Alessandro Lazaric Chercheur junior CR1, INRIA Lille	Examinateur
Éric Moulines Professeur, École polytechnique	Directeur de thèse
Rémi Munos Chercheur senior DR1, INRIA Lille	Co-directeur de thèse

Contents

Chapter 1

Abstract

L'apprentissage profond par renforcement a été au cœur de nombreux résultats révolutionnaires en intelligence artificielle ces dernières années. Ces agents reposent sur des techniques d'attribution de crédit qui cherchent à établir des corrélations entre actions passées et événements futurs et utilisent ces corrélations pour devenir performants à une tâche. Ce problème est au cœur des limites actuelles de l'apprentissage par renforcement et les techniques d'attribution de crédit utilisées sont encore relativement rudimentaires et incapables de raisonnement inductif. Cette thèse se concentre donc sur l'étude et la formulation de nouvelles méthodes d'attributions de crédit dans le cadre de l'apprentissage par renforcement. De telles techniques pourraient permettre d'accélérer l'apprentissage, de mieux généraliser lorsqu'un agent est entraîné sur de multiples tâches, et peut-être même permettre l'émergence d'abstraction et de raisonnement.

Deep reinforcement learning has been at the heart of many revolutionary results in artificial intelligence in the last few years. These agents are based on credit assignment techniques that try to establish correlations between past actions and future events and use these correlations to become effective in a given task. This problem is at the heart of the current limitations of deep reinforcement learning and credit assignment techniques used today remain relatively rudimentary and incapable of inductive reasoning. This thesis therefore focuses on the study and formulation of new credit assignment methods for deep reinforcement learning. Such techniques could speed up learning, make better generalization when agents are trained on multiple tasks, and perhaps even allow the emergence of abstraction and reasoning.

Chapter 2

Acknowledgements

Je dédis cette thèse à mes parents et à ma sœur. Votre soutien sans faille et votre amour inconditionnel ont forgé la personne que je suis aujourd'hui. Je vous dois tellement et cette thèse est également la vôtre. Pour tout ce que vous m'offrez et ce que vous êtes, merci.

À mes amis du Sud-Ouest, Manon, Élorri, Anaïs, Charlie, Pierre, Marie, Laurie, Alexia, Mathieu, Damien, Martine, Pancho, Pompon et JD, certains compagnons d'aventure depuis toujours, merci d'être ce point de repère dans ma vie et cette source inépuisable de bons moments.

À Maïtane Sebastian, pour ta passion et ta bienveillance qui ont enrichi ma perspective du monde.

À mes professeurs qui m'ont accompagné tout au long de mon parcours académique, en particulier Monsieur Novion, dont les conseils ont changé le cours de ma vie.

À Lucie, Mano, Benoît et Jean-Marc, pour leur humour et leur amitié précieuse.

À mes amis de l'ENS, Gaëtan, Antoine, Marius, Thomas, Faustine, Alex, Étienne, Aurélien, Nicolas, Lucas, Benoît, Chloé, Félix, Louise, Valérie, Pierre, Alice, Hélène, Isa, Guillemette et Clément, avec qui j'ai découvert le monde de la recherche loin de mes racines. Ensemble, nous avons traversé les défis et les succès. Merci pour ces années de voyages, de bons moments et d'aventures.

À Francis Bach pour ses précieux conseils lors de ma scolarité à l'ENS. À Yoshua Bengio, Blake Richards, Johanni Brea et Wulfram Gerstner pour leur bienveillance et leur soutien pendant mes séjours de recherche dans leur laboratoire.

À mes amis de Montréal, Bénédicte, Éloi, Alex A, Étienne, Orhan, Gauthier, Alex B, Asja, César, Adrien, Alex P, Valentin, Tess, Pascal, Guillaume, Fred, Pierre-Luc, Myriam, Kyunghyun, Vincent, Adam, Pierre, Olivier, Nick et Mélanie, qui ont forgé cette période inoubliable de ma vie. Cette ville et ses habitants ont façonné une partie de mon identité.

À Marion et Éloi, merci pour votre amitié et votre soutien indéfectible. J'attends avec impatience nos prochaines aventures, mais pas de ski de rando cette fois !

À Costanza et Francesco, pour leur amitié et leur soutien constants à travers les années.

À ma grande sœur, Claire, pour ces moments précieux partagés à Londres et après !

À Théo, pour ses conseils, sa bienveillance, ces brainstormings et son amitié.

Un grand merci à tous mes collègues et amis de DeepMind, Bilal, Doina, Thibault, Rana, Jean-Bastien, Georg, Arthur, Florent, Mark, Rémi L, Shantanu, Nathalie, Alaa, Kelsey, Harrison, Samrat, Francesco, Claire, Julien, Karl, Rahma, Anna, Florian, Will, Romuald, Théophane, Guillaume, Daniel G, David, Morgane, Armand, Diana, Greg, Tim, Anna, Bernardo, Ramona, Daniel H, Mo, Peter, Marco, Sheila, Daniel M, Andrea, Pierre, Tom, Yunhao, Kay, Sam, Johan, Charlie, Catherine, Olivier, Caglar, Hamza, Toby, Jean-Baptiste, Adam et Amal, qui m'ont accueilli chaleureusement dès mes débuts en 2018. Votre soutien, votre bienveillance et votre esprit d'entraide m'ont permis de grandir dans ma carrière de chercheur et de m'épanouir à DeepMind. Je suis reconnaissant de cette atmosphère de collaboration et d'écoute si crucial pour faire de la bonne recherche.

À Ramona, Daniel et Jade dont la présence a illuminé mon année 2023. J'ai hâte de voyager de nouveau avec vous !

À mes directeurs de thèse, Rémi et Éric, dont l'encadrement attentif et le soutien inconditionnel ont été inestimables. Leur confiance en moi et leur bienveillance m'ont permis d'explorer des sujets qui me passionnent et de grandir en tant que chercheur et individu. Merci pour toutes les précieuses leçons que j'ai apprises à vos côtés.

Je tiens enfin à exprimer ma profonde gratitude envers les membres de mon jury de thèse, pour leur présence et leur investissement malgré leur emploi du temps chargé. C'est un honneur pour moi d'avoir bénéficié de vos expertises, et j'espère pouvoir collaborer avec vous dans un futur proche.

Chapter 3

Introduction

3.1 General context in reinforcement learning

Reinforcement Learning (RL) has witnessed significant advancements in recent years, making it a central area of study within the field of artificial intelligence. RL focuses on training agents to make sequential decisions by learning from interactions with an environment to maximize cumulative rewards. Among the myriad challenges faced by RL agents, two fundamental problems stand out: exploration and credit assignment. Both are intimately linked concepts in RL, and understanding this connection is crucial for understanding the angle taken through this PhD.

Exploration in RL refers to the process by which an agent seeks to discover and learn about the environment it is interacting with. To make informed decisions, an RL agent must explore different actions and states to gather information about the consequences of its choices. Exploration is fundamental because without good exploration, an agent may become stuck in suboptimal policies and fail to discover the best strategies for maximizing cumulative rewards. Exploration is often achieved through a balance between exploiting known actions that have yielded high rewards and exploring new, uncertain actions to discover potentially better ones.

Credit assignment, on the other hand, is the mechanism by which an agent assigns responsibility to its past actions based on their impact on future outcomes. In other words, it's about figuring out which actions were responsible for the rewards or penalties an agent receives. Accurate credit assignment is essential because it allows the agent to update its policy or strategy effectively. If the agent doesn't assign credit correctly, it may reinforce actions that weren't actually responsible for the outcomes.

The link between exploration and credit assignment becomes now clear. When an RL agent explores different states and actions, it gathers data about how each action affects future rewards. This data is then used for credit assignment, helping the agent understand which actions led to the observed outcomes. Effective exploration ensures that the agent collects diverse experiences, which, in turn, improves its ability to assign credit accurately.

In summary, exploration and credit assignment are closely connected in RL. Effective exploration provides the necessary data for the agent to assign credit accurately, and this, in turn, leads to improved decision-making and policy learning. The exploration problem has been vastly explored in the past years. However, credit assignment, on the other hand, was an under explored domain in 2019. This PhD was dedicated to addressing this latter challenge.

3.2 Context on credit assignment

Reinforcement learning agents, while navigating complex environments, need to evaluate the influence of their actions on future outcomes. At its core, credit assignment represents knowledge in the form of associations between actions and outcomes. Traditional RL techniques often employ time as a proxy for credit assignment, assuming that more recent actions are more relevant to the outcome. However, the reality is far more intricate, and time-based credit assignment can be highly imperfect, especially in environments characterized by weak, noisy, or delayed feedback.

The challenges surrounding credit assignment in RL are manifold. RL agents typically operate in environments where their actions may only affect a small part of the eventual outcome, further complicating the assessment of action relevance. Moreover, current deep RL solutions often exhibit a lack a reproducibility in complex environment and suffer from sample inefficiency, partially because of weak credit assignment capacities, hindering their applicability to complex problems. These issues highlight the urgent need for better techniques to address the credit assignment problem.

A central concept in RL, the action-value function, quantifies the impact of choosing a specific action in a given state on future returns. In essence, it answers the question, "how does choosing an action 'a' in a state 'x' affect future return?". Estimating the value function is a crucial step in understanding the credit associated with actions. However, estimating the value function through naive averaging of returns often results in poor and high variance estimates, primarily due to the inherent randomness in the trajectories that RL agents traverse. Temporal Difference (TD) methods, such as Sarsa, offer a means to address this issue by utilizing learned approximations of the value function and bootstrap techniques to reduce variance. However, these methods introduce bias due to approximation and heavily rely on the Markov assumption, which may not hold in scenarios involving partial observability or function approximation.

TD(λ) methods aim to control the bias-variance trade-off by considering the recency of actions as a measure of their relevance. While this approach improves credit assignment, time-based metrics alone often prove to be an imperfect heuristic. This imperfection restricts the agent's ability to assess the broader impact of actions, especially in complex environments.

The aforementioned challenges surrounding credit assignment highlight the complexity of the problem and the pressing need for innovative solutions to enhance the efficiency and accuracy of action value estimation. Addressing these issues is not only crucial for the advancement of RL as a field but also for its broader application in various real-world scenarios, such as robotics, healthcare, finance, and many others.

In light of these challenges and the need for improved credit assignment techniques, this PhD aims to delve deep into the intricacies of credit assignment in RL. It seeks to develop novel methodologies and algorithms that can enhance the data efficiency and stability of current agents. By doing so, this research strives to contribute to the broader goal of advancing RL and making it a more effective tool for solving complex, real-world problems.

In the subsequent chapters, we will explore various aspects of credit assignment in RL, propose innovative solutions, and conduct extensive experiments to demonstrate the effectiveness of these solutions in addressing the complexities of credit assignment.

3.3 Motivations for the PhD

When I just started my PhD, it was a time when the concept of credit assignment in RL had not yet matured into a distinct domain of study. While RL itself was a fast growing field with great potential, the nuanced intricacies of credit assignment were often overshadowed by broader challenges and methodologies. It was a moment in which the importance of understanding how agents assign credit to their actions for making informed decisions was not fully recognized nor explored.

However, throughout the course of our research, we aimed to change this landscape. We have strived to contribute to the transformation of credit assignment from a somewhat overlooked aspect of RL into a dedicated and actively growing field. We envisioned and pursued the goal of nurturing a community of researchers and practitioners who recognize the paramount importance of credit assignment in RL.

Through rigorous investigation, experimentation, and the development of innovative methodologies, we have endeavored to foster a deeper understanding of credit assignment and its far-reaching implications. Our aspiration has been to spark curiosity and enthusiasm among scholars, thereby encouraging a more focused and dedicated exploration of this critical area.

As our work continues to unfold and our contributions take shape, we hope to have played a role in shaping and building the ever-growing interest and attention of the community in the domain of credit assignment. Our aim was to provide a solid foundation upon which future researchers can build and inspiring new insights.

Chapter 4

Hindsight Credit Assignment, NeurIPS 2019

4.1 In short

4.1.1 Motivations

In our 2019 NeurIPS paper, titled "Hindsight Credit Assignment" [4], we delve into the challenge of efficient credit assignment in reinforcement learning. Specifically, we introduce the concept of using hindsight information to enhance an agent's performance. As explained in the introduction, credit assignment in reinforcement learning is a complex task because agents navigate through various states and external influences while taking multiple actions before reaching their goal.

Traditional reinforcement learning algorithms often struggle with credit assignment as they rely solely on foresight. These methods operate under the assumption that we lack knowledge of what occurs beyond a given time step, making accurate credit assignment challenging, especially in intricate environments. Our proposal, on the other hand, centers on utilizing hindsight information, acknowledging that credit assignment and learning typically take place after the agent completes its current trajectory. This approach enables us to leverage this additional data to refine the learning of critical variables necessary for credit assignment.

4.1.2 Approach

Our approach introduces a new family of algorithms known as "Hindsight Credit Assignment" (HCA). HCA algorithms explicitly assign credit to past actions based on the likelihood of those actions leading to the observed outcome. This is achieved by comparing a learned hindsight distribution over actions, conditioned by a future state or return, with the policy that generated the trajectory. The resulting ratio provides a measure of how crucial a particular action was in achieving the outcome. A ratio deviating further from 1 indicates a greater impact (positive or negative) of that action on the outcome.

To compute the hindsight distribution, HCA algorithms employ a technique related to importance sampling. Importance sampling estimates the expected value of a function under one distribution (the hindsight distribution) using samples from another distribution (the policy distribution). In the context of HCA, importance sampling weights are determined based on the likelihood of the agent taking each action in the trajectory, given the hindsight state compared to the likelihood of the policy for that same action.

Once the hindsight distribution is computed, HCA algorithms can be used to update the agent's policy and value function. One approach involves using the hindsight distribution to reweight the agent's experience. This means the agent will learn more from actions that were more likely to have contributed to the observed outcome.

4.1.3 Results

Our research demonstrates several advantages of HCA algorithms over traditional reinforcement learning methods. Firstly, they are more data-efficient, particularly in tasks with complex credit assignment requirements, as they make effective use of hindsight information rather than blindly averaging out future possibilities. Secondly, they exhibit greater robustness in the face of environmental noise and uncertainty.

We evaluate HCA algorithms across a range of illustrative tasks and find that they outperform traditional reinforcement learning algorithms in many of these scenarios.

In summary, [4] represents a significant contribution to the field of reinforcement learning. We introduce a novel algorithmic approach that addresses the challenge of credit assignment more efficiently and robustly than traditional methods. The introduction of the concept of leveraging hindsight information was very impactful and has had a notable impact on the research community, as evidenced by its substantial citation count (73 citations at the time of writing).

Hindsight Credit Assignment

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Nicolas Heess, Mohammad G. Azar, Bilal Piot, Hado van Hasselt, Satinder Singh, Greg Wayne, Doina Precup, Rémi Munos
DeepMind
{harutyunyan, wdabney, munos}@google.com

Abstract

We consider the problem of efficient credit assignment in reinforcement learning. In order to efficiently and meaningfully utilize new data, we propose to explicitly assign credit to past decisions based on the likelihood of them having led to the observed outcome. This approach uses new information in hindsight, rather than employing foresight. Somewhat surprisingly, we show that value functions can be rewritten through this lens, yielding a new family of algorithms. We study the properties of these algorithms, and empirically show that they successfully address important credit assignment challenges, through a set of illustrative tasks.

1 Introduction

A reinforcement learning (RL) agent is tasked with two fundamental, interdependent problems: exploration (how to discover useful data), and credit assignment (how to incorporate it). In this work, we take a careful look at the problem of credit assignment. The instrumental learning object in RL – the value function – quantifies the following question: *"how does choosing an action a in a state x affect future return?"*. This is a challenging question for several reasons.

Issue 1: Variance. The simplest way of estimating the value function is by averaging returns (future discounted sums of rewards) starting from taking a in x. This Monte Carlo style of estimation is inefficient, since there can be a lot of randomness in trajectories.

Issue 2: Partial observability. To amortize the search and reduce variance, temporal difference (TD) methods, like Sarsa and Q-learning, use a learned approximation of the value function and bootstrap. This introduces bias due to the approximation, as well as a reliance on the Markov assumption, which is especially problematic when the agent operates outside of a Markov Decision Process (MDP), for example if the state is partially observed, or if there is function approximation. Bootstrapping may then cause the value function to not converge at all, or to remain permanently biased [19].

Issue 3: Time as a proxy. TD(λ) methods control this bias-variance trade-off, but they rely on *time* as the sole metric for relevance: the more recent the action, the more credit or blame it receives from a future reward [20, 21]. Although time is a reasonable proxy for cause-and-effect (especially in MDPs), in general it is a heuristic, and can hence be improved by learning.

Issue 4: No counterfactuals. The only data used for estimating an action's value are trajectories that contain that action, while ideally we would like to be able to use the same trajectory to update *all* relevant actions, not just the ones that happened to (serendipitously) occur.

Figure 1 illustrates these issues concretely. At the high-level, we wish to achieve credit assignment mechanisms that are both sample-efficient (issues 1 and 4), and expressive (issues 2 and 3). To this end, we propose to reverse the key learning question, and learn estimators that measure: *"given the future outcome (reward or state), how relevant was the choice of a in x to achieve it?*", which is essentially the credit assignment question itself. Although eligibility traces consider the same

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

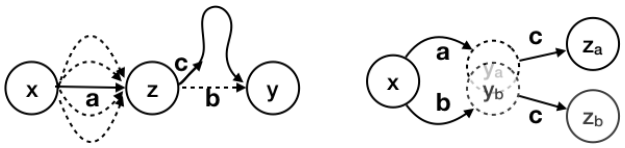


Figure 1: **Left.** Consider the trajectory shown by solid arrows to be the sampled trajectory, τ. An RL algorithm will typically assign credit for the reward obtained in state y to the actions along τ. This is unsatisfying for two reasons: (1) action a was not essential in reaching state z, any other a' would have been just as effective; hence, overemphasizing a is a source of variance; (2) from z, action c was sampled, leading to a multi-step trajectory into y, but action b transitions to y from z directly; so, it should get more of the credit for y. Note that c could have been an exploratory action, but also could have been more likely according to the policy in z, but *given that y was reached*, b was more likely **Right.** The choice between actions a or b at state x causes a transition to either y_a or y_b, but they are perceptually aliased. On the next decision, the same action c transitions the agent to different states, depending on the true underlying y. The state y can be a single state, or could itself be a trajectory. This scenario can happen e.g. when the features are being learned. A TD algorithm that bootstraps in y will not be able to learn the correct values of a and b, since it will average over the rewards of z_a and z_b. When y is a potentially long trajectory with a noisy reward, a Monte Carlo algorithm will incorporate the noise along y into the values of both a and b, despite it being irrelevant to the choice between them. We would like to be able to directly determine the relevance of a to being in z_a.

question, they do so in a way that is (purposefully) equivalent to the forward view [20], and so they have to rely mainly on "vanilla" features, like time, to decide credit assignment. Reasoning in the backward view explicitly opens up a new family of algorithms. Specifically, we propose to use a form of *hindsight conditioning* to determine the relevance of a past action to a particular outcome. We show that the usual value functions can be rewritten in hindsight, yielding a new family of estimators, and derive policy gradient algorithms that use these estimators. We demonstrate empirically the ability of these algorithms to address the highlighted issues through a set of diagnostic tasks, which are not handled well by other means.

2 Background and Notation

A Markov decision process (MDP) [14] is a tuple $(\mathcal{X}, \mathcal{A}, p, r, \gamma)$, with $\mathcal{X}$ being the state space, $\mathcal{A}$ - the action space, $p : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \to [0, 1]$ – the state-transition distribution (with $p(y|x, a)$ denoting the probability of transitioning to state y from x by choosing action a), $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ – the reward function, and $\gamma \in [0, 1)$ – the scalar discount factor. A stochastic policy π maps each state to a distribution over actions: $\pi(a|x)$ denotes the probability of choosing action a in state x. Let $\mathcal{T}(x, \pi)$ and $\mathcal{T}(x, a, \pi)$ be the distributions over trajectories $\tau = (X_k, A_k, R_k)_{k\in\mathbb{N}^+}$ generated by a policy π, given $X_0 = x$ and $(X_0, A_0) = (x, a)$, respectively. Let $Z(\tau) \stackrel{\text{def}}{=} \sum_{k\geq 0} \gamma^k R_k$ be the return obtained along the trajectory τ. The value (or V-) function V^π and the action-value (or Q-) function Q^π denote the expected return under the policy π given $X_0 = x$ and $(X_0, A_0) = (x, a)$, respectively:

$$V^\pi(x) \stackrel{\text{def}}{=} \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[Z(\tau)\Big], \qquad Q^\pi(x, a) \stackrel{\text{def}}{=} \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[Z(\tau)\Big]. \tag{1}$$

The benefit of choosing a given action a over the usual policy π is measured by the advantage function $A^\pi(x, a) \stackrel{\text{def}}{=} Q^\pi(x, a) - V^\pi(x)$. Policy gradient algorithms improve the policy by changing π in the direction of the gradient of the value function [22]. This gradient at some initial state x_0 is

$$\nabla V^\pi(x_0) = \sum_{x,a} d^\pi(x|x_0) Q^\pi(x, a) \nabla\pi(a|x) = \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\sum_a \sum_{k\geq 0} \gamma^k A^\pi(X_k, a) \nabla\pi(a|X_k)\Big],$$

where $d^\pi(x|x_0) \stackrel{\text{def}}{=} \sum_k \gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x_0,\pi)}(X_k = x)$ is the (unnormalized) discounted state-visitation distribution. Practical algorithms such as REINFORCE [25] approximate Q^π or A^π with an n-step truncated return, possibly combined with a bootstrapped approximate value function V, which is also often used as baseline (see [22, 12]) along a trajectory $\tau = (X_k, A_k, R_k)_k \sim \mathcal{T}(x, a, \pi)$:

$$A^\pi(x, a) \approx \sum_{k=0}^{n-1} \gamma^k R_k + \gamma^n V(X_n) - V(x).$$

3 Conditioning on the Future

The classical value function attempts to answer the question: "how does the current action affect future outcomes?" By relying on predictions about these future outcomes, existing approaches often exacerbate problems around variance (issue 1) and partial observability (issue 2). Furthermore, these methods tend to use temporal distance as a proxy for relevance (issue 3) and are unable to assign credit counter-factually (issue 4). We propose to learn estimators that explicitly consider the credit assignment question: *"given an outcome, how relevant were past decisions?"*.

This approach can in fact be linked to some classical methods in statistical estimation. In particular, Monte Carlo simulation is known to be inaccurate when there are rare events that are of interest: the averaging requires an infeasible number of samples to obtain an accurate estimate [16]. One solution is to *change measures*, that is, to use another distribution for which the events are less rare, and correct with importance sampling. The Girsanov theorem is a well-known example of this in processes with Brownian dynamics [4], known to produce lower variance estimates.

This scenario of rare random events is particularly relevant to efficient credit assignment in RL. When a new significant outcome is experienced, the agent ought to quickly update its estimates and policy accordingly. Let $\tau \sim \mathcal{T}(x, \pi)$ be a sampled trajectory, and f some function of it. By changing measures from the policy π with which it was sampled to a future-conditional, or *hindsight* distribution $h(\cdot|x, \pi, f(\tau))$, we hope to improve the efficiency of credit assignment. The importance sampling ratio $\frac{h(a|x,\pi,f(\tau))}{\pi(a|x)}$ then precisely denotes the relevance of an action a to the specific future $f(\tau)$. If the distribution $h(a|x, \pi, f(\tau))$ is accurate, this allows us to quickly assign credit to all actions relevant to achieving $f(\tau)$. In this work, we consider f to be a future *state*, or a future *return*. To highlight the use of the future-conditional distribution, we refer to the resulting family of methods as Hindsight Credit Assignment (HCA).

The remainder of this section formalizes the insight outlined above, and derives the usual value functions and policy gradients in hindsight, while the next one presents new algorithms based on sampling these expressions.

3.1 Conditioning on Future States

The agent composes its estimates of the return from an action a by summing over the rewards obtained from future states X_k. One option of hindsight conditioning is to consider, at each step, the likelihood of an action a *given that the future state* X_k *was reached.*

Definition 1 (*State-conditional hindsight distributions*)**.** *For any action a and any state y, define $h_k(a|x, \pi, y)$ to be the conditional probability over trajectories $\tau \sim \mathcal{T}(x, \pi)$ of the first action A_0 of trajectory τ being equal to a, given that the state y has occurred at step k along trajectory τ:*

$$h_k(a|x, \pi, y) \overset{def}{=} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y). \tag{2}$$

Intuitively, $h_k(a|x, \pi, y)$ quantifies the relevance of action a to the future state X_k. If a is not relevant to reaching X_k, this probability is simply the policy $\pi(a|x)$ (there is no relevant information in X_k). If a is instrumental to reaching X_k, $h_k(a|x, \pi, y) > \pi(a|x)$, and vice versa, if a detracts from reaching X_k, $h_k(a|x, \pi, y) < \pi(a|x)$. In general, h_k is a lower-entropy distribution than π. The relationship of h_k to more familiar quantities can be understood through the following identity obtained by an application of Bayes' rule:

$$\frac{h_k(a|x, \pi, y)}{\pi(a|x)} = \frac{\mathbb{P}(X_k = y|X_0 = x, A_0 = a, \pi)}{\mathbb{P}(X_k = y|X_0 = x, \pi)} = \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(X_k = y)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}.$$

Using this identity and importance sampling, we can rewrite the usual Q-function in terms of h_k. Since there is only one policy π involved here, we will drop the explicit conditioning, but it is implied.

Theorem 1. *Consider an action a and a state x for which $\pi(a|x) > 0$. Then the following holds:*

$$Q^\pi(x, a) = r(x, a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 1} \gamma^k \frac{h_k(a|x, X_k)}{\pi(a|x)} R_k\Big].$$

So, each of the rewards R_k along the way is weighted by the ratio $\frac{h_k(a|x,X_k)}{\pi(a|x)}$, which exactly quantifies how relevant a was in achieving the corresponding state X_k. Following the discussion above, this

ratio is 1 if a is irrelevant, and larger or smaller than 1 in the other cases. The expression for the Q-function is similar to that in Eq. (1), but the new expectation is no longer conditioned on the initial action a – the policy π is followed from the start ($A_0 \sim \pi(\cdot|x)$ instead of $A_0 = a$). This is an important point, as it will allow us to use returns generated by *any* action A_0 to update the values of all actions, to the extent that they are relevant according to $\frac{h_k(a|x,X_k)}{\pi(a|x)}$. Theorem 1 implies the following expression for the advantage:

$$A^\pi(x,a) = r(x,a) - r^\pi(x) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 1}\Big(\frac{h_k(a|x,X_k)}{\pi(a|x)} - 1\Big)\gamma^k R_k\Big], \tag{3}$$

where $r^\pi(x) = \sum_{a\in\mathcal{A}}\pi(a|x)r(x,a)$. This form of the advantage is particularly appealing, since it directly removes irrelevant rewards from consideration. Indeed, whenever $\frac{h_k(a|x,X_k)}{\pi(a|x)} = 1$, the reward R_k does not participate in the advantage for the value of action a. When there is inconsequential noise that is outside of the agent's control, this may greatly reduce the variance of the estimates.

Removing time dependence. For clarity of exposition, here we have considered the hindsight distribution to be additionally conditioned on time. Indeed, h_k depends not only on reaching the state, but also on the number of timesteps k that it takes to do so. In general, this can be limiting, as it introduces a stronger dependence on the particular trajectory, and a harder estimation problem of the hindsight distribution. It turns out we can generalize all of the results presented here to a *time-independent* distribution $h_\beta(a|x,y)$, which gives the probability of a conditioned on reaching y *at some point* in the future. The scalar $\beta \in [0,1)$ is the "probability of survival" at each step. This can either be the discount γ, or a termination probability if the problem is undiscounted. In the discounted reward case Eq. (3) can be written in terms of h_β as follows:

$$A^\pi(x,a) = r(x,a) - r^\pi(x) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 1}\Big(\frac{h_\beta(a|x,X_k)}{\pi(a|x)} - 1\Big)\gamma^k R_k\Big], \tag{4}$$

with the choice of $\beta = \gamma$. The interested reader may find the relevant proofs in the appendix.

Finally, it is possible to obtain a hindsight V-function, analogously to the Q-function from Theorem 1. The next section does this for *return*-conditional HCA. We include other variations in appendix.

3.2 Conditioning on Future Returns

The previous section derived Q-functions that explicitly reweigh the rewards at each step, based on the corresponding states' connection to the action whose value we wish to estimate. Since ultimately we are interested in the return, we could alternatively use it for future conditioning itself.

Definition 2 (*Return-conditional hindsight distributions*)**.** *For any action a and any possible return z, define $h_z(a|x,\pi,z)$ to be the conditional probability over trajectories $\tau \sim \mathcal{T}(x,\pi)$ of the first action A_0 being a, given that z has been observed along τ:*

$$h_z(a|x,\pi,z) \stackrel{def}{=} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}\big(A_0 = a | Z(\tau) = z\big).$$

The distribution $h_z(a|x,\pi,z)$ is intuitively similar to h_k, but instead of future states, it directly quantifies the relevance of a to obtaining the entire return z. This is appealing, since in the end we care about returns. Further, this could be simpler to learn, since instead of the possibly high-dimensional state, we now need to worry only about a scalar outcome. On the other hand, it is no longer "jumpy" in time, so may benefit less from structure in the dynamics. As with h_k, we will drop the explicit conditioning on π, but it is implied. We have the following result.

Theorem 2. *Consider an action a, and assume that for any possible random return $z = Z(\tau)$ for some trajectory $\tau \sim \mathcal{T}(x,\pi)$ we have $h_z(a|x,z) > 0$. Then we have:*

$$V^\pi(x) = \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[Z(\tau)\frac{\pi(a|x)}{h_z(a|x,Z(\tau))}\Big]. \tag{5}$$

The V- (rather than Q-) function form here has interesting properties that we will discuss in the next section. Mathematically, the two forms are analogous to derive, but the ratio is now flipped. Equations (5) and (1) imply the following expression for the advantage:

$$A^\pi(x,a) = \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[\Big(1 - \frac{\pi(a|x)}{h_z(a|x,Z(\tau))}\Big)Z(\tau)\Big]. \tag{6}$$

The factor $c(a|x,Z) = 1 - \frac{\pi(a|x)}{h_z(a|x,Z)}$ expresses how much a single action a contributed to obtaining a return Z. If other actions (drawn from $\pi(\cdot|x)$) would have yielded the same return, $c(a|x,Z) = 0$, and the advantage is 0. If an action a has made achieving Z more likely, then $c(a|x,Z) > 0$, and conversly, if other actions would have contributed to achieving Z more than a, then $c(a|x,Z) < 0$. Hence, $c(a|x,Z)$ expresses the impact an action has on the environment, in terms of the return, if everything else (future decisions as well as randomness of the environment) is unchanged.

Both h_β and h_z can be learned online from sampled trajectories (see Sec. 4 for algorithms, and a discussion in Sec. 4.1). Finally, while we chose to focus on state and return conditioning, one could consider other options. For example, conditioning on the reward (instead of the state) at a future time k, or an embedding of (or part of) the future trajectory, could have interesting properties.

3.3 Policy Gradients

We now give a policy gradient theorem based on the new expressions of the value function.

Theorem 3. *Let π_θ be the policy parameterized by θ, and $\beta = \gamma$. Then, the gradient of the value at some state x_0 is:*

$$\nabla_\theta V^{\pi_\theta}(x_0) = \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_{k\geq 0}\gamma^k \sum_a \nabla\pi_\theta(a|X_k)Q^x(X_k,a)\Big] \tag{7}$$

$$= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_{k\geq 0}\gamma^k \nabla\log\pi_\theta(A_k|X_k)A^z(X_k,A_k)\Big], \tag{8}$$

$$Q^x(X_k,a) \stackrel{def}{=} r(X_k,a) + \sum_{t\geq k+1}\gamma^{t-k}\frac{h_\beta(a|X_k,X_t)}{\pi_\theta(a|X_k)}R_t,$$

$$A^z(x,a) \stackrel{def}{=} \Big(1 - \frac{\pi_\theta(a|x)}{h_z(a|x,Z(\tau_{k:\infty}))}\Big)Z(\tau_{k:\infty}).$$

Note that the expression for state HCA in Eq. (7) is written for all actions, rather than only the sampled one. Interestingly, this form does not require (or benefit from) a baseline. Contrary to the usual all-actions algorithm which must use the critic, the HCA reweighting allows us to use returns sampled from a particular starting action to obtain value estimates for all actions.

4 Algorithms

Using the new policy gradient theorem 3, we will now give novel algorithms based on sampling the expectations (7) and (8). Then, we will discuss the training of the relevant hindsight distributions.

State-Conditional HCA Consider a parametric representation of the policy $\pi(\cdot|x)$ and the future-state-conditional distribution $h_\beta(a|x,y)$, as well as the baseline V and an estimate of the immediate reward $\hat{r}$. Generate T-step trajectories $\tau^T = (X_s, A_s, R_s)_{0\leq s\leq T}$. We can compose an estimate of the return for all actions a (see Theorem 7 in appendix):

$$Q^x(X_s,a) \approx \hat{r}(X_s,a) + \sum_{t=s+1}^{T-1}\gamma^{t-s}\frac{h_\beta(a|X_s,X_t)}{\pi(a|X_s)}R_t + \gamma^{T-s}\frac{h_\beta(a|X_s,X_T)}{\pi(a|X_s)}V(X_T).$$

The algorithm proceeds by training $V(X_s)$ to predict the usual return $Z_s = \sum_{t=s}^{T-1}\gamma^{t-s}R_t + \gamma^{T-s}V(X_T)$ and $\hat{r}(X_s,A_s)$ to predict R_s (square loss), the hindsight distribution $h_\beta(a|X_s,X_t)$ to predict A_s (cross entropy loss), and finally by updating the policy logits with $\sum_a Q^x(X_s,a)\nabla\pi(a\mid X_s)$. See Algorithm 1 in appendix for the detailed pseudocode.

Return-Conditional HCA Consider a parametric representation of the policy $\pi(\cdot|x)$ and the return-conditioned distribution $h_z(a|x,z)$. Generate full trajectories $\tau = (X_s, A_s, R_s)_{s\in\mathbb{N}^+}$ and compute

the sampled advantage at each step:

$$A^z(X_s, A_s) = \Big(1 - \frac{\pi(A_s|X_s)}{h_z(A_s|X_s, Z_s)}\Big) Z_s,$$

where $Z_s = \sum_{t \geq s} \gamma^{t-s} R_t$. The algorithm proceeds by training the hindsight distribution $h_z(a|X_s, Z_s)$ to predict A_s (cross entropy loss), and updating the policy gradient with $\nabla \log \pi(A_s \mid X_s) A^z(X_s, A_s)$. See Algorithm 2 in appendix for the detailed pseudocode.

RL without value functions. The return-conditional version lends itself to a particularly simple algorithm. In particular, we no longer need to learn the value function V – if $h_z(a|X_s, Z_s)$ is estimated well, using complete rollouts is feasible without variance issues. This takes our idea of reversing the direction of the learning question to the extreme, it is now *entirely* in hindsight.

The result is an actor-critic algorithm, where the usual baseline $V(X_s)$ is replaced by $b_s \stackrel{\text{def}}{=} \frac{\pi(A_s|X_s)}{h_z(A_s|X_s, Z_s)} Z_s$. This baseline is strongly correlated to the return Z_s (it is proportional to it), which is desirable since we would like to remove as much of the variance (due to the dynamics of the world, or the agent's own policy) as possible. The following proposition verifies that despite being correlated, this baseline does not introduce bias into the policy gradient.

Proposition 1. *The baseline $b_s = \frac{\pi(A_s|X_s)}{h_z(A_s|X_s, Z_s)} Z_s$ does not introduce any bias in the policy gradient:*

$$\mathbb{E}_{\tau \sim \mathcal{T}(x_0, \pi)} \Big[\sum_s \gamma^s \nabla \log \pi(A_s|X_s) \big(Z_s(\tau) - b_s\big) \Big] = \nabla V(x_0).$$

4.1 Learning Hindsight Distributions

We have given equivalent rewritings of the usual value functions in terms of the proposed hindsight distributions, and have motivated their properties, when they are accurate. Now, the question is if it is feasible to learn good estimates of those distributions from experience, and whether shifting the learning problem in this way is beneficial. The remainder of this section discusses this question, while the next one provides empirical evidence for the affirmative.

There are several conventional objects that could be learned to help with credit assignment: a value function, a forward model, or an inverse model over states. An accurate forward model allows one to compute value functions directly with no variance, and an accurate inverse model – to perform precise credit assignment. However, learning such generative models accurately is difficult and has been a long-standing challenge in RL, especially in high-dimensional state spaces. Interestingly, the hindsight distribution is a discriminative, rather than generative model, and is hence not required to model the full distribution over states. Additionally, the action space is usually much smaller than the state space, and so shifting the focus to actions potentially makes the problem much easier. When certain structure in the dynamics is present, learning hindsight distributions may be significantly easier still – e.g. if the transition model is stochastic or the policy is changing, a particular (x, a) can lead to many possible future states, but a particular future state can be explained by a small number of past actions. In general, learning h_z and h_β are supervised learning problems, so the new algorithms delegate some of the learning difficulty in RL to a supervised setting, for which many efficient approaches exist (e.g. [7, 23]).

5 Experiments

To empirically validate our proposal in a controlled way, we devised a set of diagnostic tasks that highlight issues 1-4, while also being representative of what occurs in practice (Fig. 2). We then systematically verify the intuitions developed throughout the paper. In all cases, we learn the hindsight distributions in tandem with the control policy. For each problem we compare HCA with state and return conditioning to standard baseline policy gradient, that is: n-step advantage actor critic (with $n = \infty$ for Monte Carlo). All the results are an average of 100 independent runs, with the plots depicting means and standard deviations. For simplicity we take $\gamma = 1$ in all of the tasks.

Shortcut. We begin with an example capturing the intuition from Fig. 1 (left). Fig. 2 (left) depicts a chain of length n with a rewarding final state. At each step, one action takes a shortcut and directly

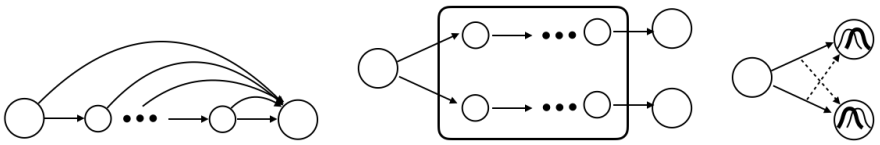


Figure 2: **Left:** Shortcut. Each state has two actions, one transitions directly to the goal, the other to the next state of the chain. **Center:** Delayed effect. Start state presents a choice of two actions, followed by an aliased chain, with the consequence of the initial choice apparent only in the final state. **Right:** Ambiguous bandit. Each action transitions to a particular state with high probability, but to the other action's state with low probability. When the two states have noisy rewards, credit assignment to each action becomes challenging.

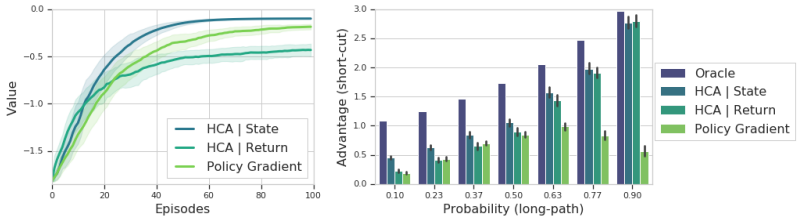


Figure 3: Shortcut. **Left:** learning curves for $n = 5$ with the policy between long and short paths initialized uniformly. Explicitly considering the likelihood of reaching the final state allows state-conditioned HCA to more quickly adjust its policy. **Right:** the advantage of the shortcut action estimated by performing 1000 rollouts from a fixed policy. The x-axis depicts the policy probabilities of the actions on the long path. The oracle is computed analytically without sampling. When the shortcut action is unlikely and rarely encountered, it is difficult to obtain an accurate estimate of the advantage. HCA is consistently able to maintain larger (and more accurate) advantages.

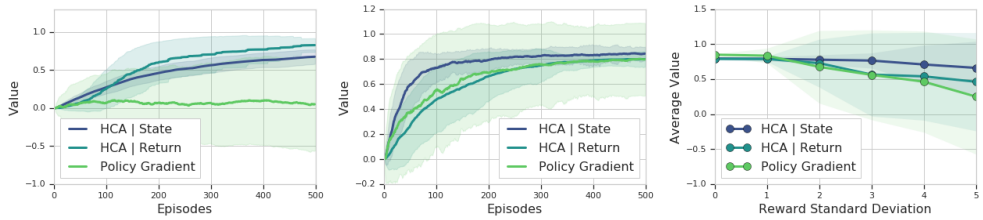


Figure 4: Delayed effect. **Left:** Bootstrapping. The learning curves for $n = 5$, $\sigma = 0$, and a 3-step return, which causes the agent to bootstrap in the partially observed region. As expected, naive bootstrapping is unable to learn a good estimate. **Middle:** Using full Monte Carlo returns (for $n = 3$) overcomes partial observability, but is prone to noise. The plot depicts learning curves for the setting with added white noise of $\sigma = 2$. **Right.** The average performance w.r.t. different noise levels – predictably, state HCA is the most robust.

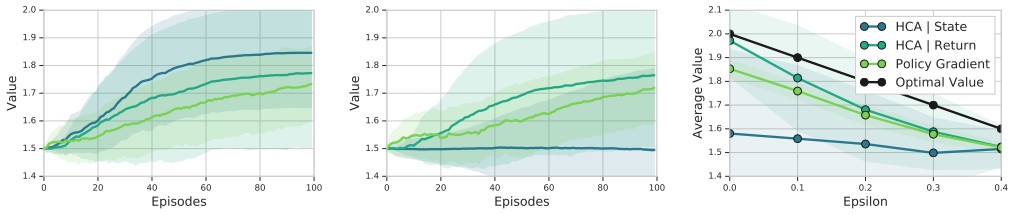


Figure 5: Ambiguous bandit with Gaussian rewards of means 1, 2, and standard deviation 1.5. **Left:** The state identity is observed. Both HCA methods improve on PG. **Middle:** The state identity is hidden, handicapping state HCA, but return HCA continues to improve on PG. **Right:** Average performance w.r.t. different ϵ-s with Gaussian rewards of means 1, 2, and standard deviation 0.5. Note that the optimal value itself decays in this case.

transitions to the final state, while the other continues on the longer path, which may be more likely according to the policy. There is a per-step penalty (of -1), and a final reward of 1. There is also a chance (of 0.1) that the agent transitions to the absorbing state directly.

This problem highlights two issues: (1) the importance of counter-factual credit assignment (issue 4); when the long path is taken more frequently than the shortcut path, counter-factual updates become increasingly effective (see Fig. 3, right) (2) the use of time as a proxy for relevance (issue 3) is shown to be only a heuristic, even in a fully-observable MDP. The relevance for the states along the chain is not accurately reflected in the long temporal distance between them and the goal state. In Fig. 3 we show that HCA is more effective at quickly adjusting the policy towards the shortcut action.

Delayed Effect. The next task instantiates the example from Fig. 1 (right). Fig. 2 (middle) depicts a POMDP, in which after the first decision, there is aliasing until the final state. This is a common case of partial observability, and is especially pertinent if the features are being learned. We show that (1) Bootstrapping naively is inadequate in this case (issue 2), but HCA is able to carry the appropriate information;[1] and (2) While Monte Carlo is able to overcome the partial observability, its performance deteriorates when intermediate reward noise is present (issue 1). HCA on the other hand is able to reduce the variance due to the irrelevant noise in the rewards.

Additionally, in this example the first decision is the most relevant choice, despite being the most temporally remote, once again highlighting that using temporal proximity for credit assignment is a heuristic (issue 3). One of the final states is rewarding (with $r = 1$), the other penalizing (with $r = -1$), and the middle states contain white noise of standard deviation σ. Fig. 4 depicts our results. In this task, the return-conditional HCA has a more difficult learning problem, as it needs to correctly model the noise distribution to condition on, which is as difficult as learning the values naively, and hence performs similarly to the baseline.

Ambiguous Bandit. Finally, to emphasize that credit assignment can be challenging, even when it is not long-term, we consider a problem without a temporal component. Fig. 2 (right) depicts a bandit with two actions, leading to two different states, whose reward functions are similar (here: drawn from overlapping Gaussian distributions), with some probability ϵ of crossover. The challenge here is due to variance (issue 1) and a lack of counter-factual updates (issue 4). It is difficult to tell whether an action was genuinely better, or just happened to be on the tail end of the distribution. This is a common scenario when bootstrapping with similar values. Due to the explicit aim at modeling the distributions, the hindsight algorithms are more efficient (Fig. 5 (left)).

To highlight the differences between the two types of hindsight conditioning, we introduce partial observability (issue 2), see Fig. 5 (right). The return-conditional policy is still able to improve over policy gradient, but state-conditioning now fails to provide informative conditioning (by construction).

6 Related Work

Hindsight experience replay (HER) [1] introduces the idea of off-policy learning about many goals from the same trajectory. The intuition is that regardless of what goal the trajectory was pursuing originally, *in hindsight* it, e.g., successfully found the one corresponding to its final state, and there is something to be learned. Rauber et al. [15] extend the same intuition to policy gradient algorithms, with goal-conditioned policies. Goyal et al. [5] also use goal conditioning and learn a backtracking model, which predicts the state-action pairs occurring on trajectories that end up in goal states. These works share our intuition of in hindsight using the same data to learn about many things, but in the context of goal-conditioned policies, while we essentially contrast conditional and unconditional policies, where the conditioning is on the extra outcome (state or return). Note that we never act w.r.t. the conditional policy, and it is used solely for credit assignment.

The temporal value transport algorithm [11] also aims to propagate credit efficiently backward in time. It uses an attention mechanism over memory to jump over parts of a trajectory that are irrelevant for the rewards obtained. While demonstrated on challenging problems, that method is biased; a promising direction for future research is to apply our unbiased hindsight mechanism with past states chosen by such an attention mechanism. Another line of work with a related intuition is RUDDER [2]. It uses an LSTM to predict future returns and sensitivity analysis to distribute those

[1] See the discussion in Appendix F.

returns as immediate rewards in order to reduce the learning horizon and make long-term credit assignment easier. Instead of aiming to redistribute the return, state HCA up- or dowmnweights individual rewards according to their relevance to the past action.

A large number of variance reduction techniques have been applied in RL, e.g. using learned value functions as critics, and other control variates [e.g. 24]. When a model of the environment is available, it can be used to reduce variance. Rollouts from the same state fill the same role in policy gradients [18]. Differentiable system dynamics allow low-variance estimates of the Q-value gradient by using the pathwise derivative estimator, effectively backpropagating the gradient of the objective along trajectories [e.g. 17, 9, 10]. In stochastic systems this requires knowledge of the environment noise. To bypass this, Heess et al. [9] *infer* the noise given an observed trajectory. Buesing et al. [3] apply this idea to POMDPs, where it can be viewed as reasoning about events in hindsight. They use a structural causal model of the dynamics and infer the posterior over latent causes from empirical trajectories. Using an empirical rather than a learned distribution over latent causes can reduce bias and, together with the (deterministic) model of the system dynamics, allows exploring the effect of alternative action choices for an observed trajectory.

Inverse models similar to the ones we use appear, for instance, in variational intrinsic control [6] (see also e.g. [8]). However, in our work, the inverse model serves as a way of determining the influence of an action on a future outcome, whereas the work in [6, 8] aims to use the inverse model to derive an intrinsic reward for training policies in which actions influence the future observations.

Finally, prioritized sweeping can be viewed as changing the sampling distribution with hindsight knowledge of the TD errors [13].

7 Closing

We proposed a new family of algorithms that explicitly consider the question of credit assignment as a part of, or instead of, estimating the traditional value function. The proposed estimators come with new properties, and as we validate empirically, are able to address some of the key issues in credit assignment. Investigating the scalability of these algorithms in the deep reinforcement learning setting is an exciting problem for future research.

Acknowledgements

The authors thank Joseph Modayil for reviews of earlier manuscripts, Theo Weber for several insightful suggestions, and the anonymous reviewers for their useful feedback.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058, 2017.

[2] Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13544–13555, 2019.

[3] Lars Buesing, Theophane Weber, Yori Zwols, Sébastien Racanière, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *CoRR*, abs/1811.06272, 2018.

[4] Igor Vladimirovich Girsanov. On transforming a certain class of stochastic processes by absolutely continuous substitution of measures. *Theory of Probability & Its Applications*, 5(3):285–301, 1960.

[5] Anirudh Goyal, Philemon Brakel, William Fedus, Soumye Singhal, Timothy Lillicrap, Sergey Levine, Hugo Larochelle, and Yoshua Bengio. Recall traces: Backtracking models for efficient reinforcement learning. In *International Conference on Learning Representations(ICLR)*, 2019.

[6] Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

[7] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[8] Karol Hausman, Jost Tobias Springenberg, Ziyu Wang, Nicolas Heess, and Martin Riedmiller. Learning an embedding space for transferable robot skills. In *International Conference on Learning Representations (ICLR)*, 2018.

[9] Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2944–2952. Curran Associates, Inc., 2015.

[10] Mikael Henaff, William F Whitney, and Yann LeCun. Model-based planning with discrete and continuous actions. *arXiv preprint arXiv:1705.07177*, 2017.

[11] Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *arXiv preprint arXiv:1810.06721*, 2018.

[12] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.

[13] Andrew W Moore and Christopher G Atkeson. Prioritized sweeping: Reinforcement learning with less data and less time. *Machine learning*, 13(1):103–130, 1993.

[14] Martin Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[15] Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Jürgen Schmidhuber. Hindsight policy gradients. In *International Conference on Learning Representations (ICLR)*, 2019.

[16] Gerardo Rubino, Bruno Tuffin, et al. *Rare event simulation using Monte Carlo methods*, volume 73. Wiley Online Library, 2009.

[17] John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. *CoRR*, abs/1506.05254, 2015.

[18] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[19] Satinder Singh, Tommi Jaakkola, and Michael I. Jordan. Learning without state estimation in partially observable environments. In *International Conference on Machine Learning (ICML)*, 1994.

[20] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[21] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition, 2018.

[22] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

A Proofs

A.1 Proof of Theorem 1

Lemma 1. *For any initial state x, a state y that can occur on a trajectory $\tau \sim \mathcal{T}(x,\pi)$, that is: $\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y) \neq 0$ for some k an action a for which $\pi(a|x) \neq 0$, we have:*

$$\frac{h_k(a|x,y)}{\pi(a|x)} = \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}. \tag{9}$$

Proof. From Bayes' rule, we have:

$$\begin{aligned}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) &= \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a)},\\ &= \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)h_k(a|x,y)}{\pi(a|x)}.\end{aligned}$$

□

Proof of Theorem 1. From the definition of the Q-function for a state-action pair (x,a), we have

$$Q^\pi(x,a) = r(x,a) + \sum_{k\geq 1}\sum_{y\in\mathcal{X}} \gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) r^\pi(y), \tag{10}$$

where $r^\pi(y) = \sum_{a\in\mathcal{A}} \pi(a|y) r(y,a)$.

Combining Eq. (9) with Eq. (10) we deduce

$$\begin{aligned}Q^\pi(x,a) &= r(x,a) + \sum_{y\in\mathcal{X}}\sum_{k\geq 1} \gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)\frac{h_k(a|x,y)}{\pi(a|x)} r^\pi(y),\\ &= r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\left[\sum_{k\geq 1}\gamma^k \frac{h_k(a|X_k,x)}{\pi(a|x)} R_k\right].\end{aligned}$$

□

A.2 Proof of Theorem 2

Proof. For any action a, the value function writes as

$$\begin{aligned}V^\pi(x) &= \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\big[Z(\tau)\big],\\ &= \int_z z\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)dz,\\ &= \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)}\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)dz,\\ &= \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z|A_0 = a)}\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)dz,\\ &\overset{(i)}{=} \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|Z(\tau) = z)}\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)dz,\\ &= \int_z z\frac{\pi(a|x)}{h_z(a|x,z)}\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)dz,\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[Z(\tau)\frac{\pi(a|x)}{h_z(a|x,Z(\tau))}\Big],\end{aligned}$$

where (i) follows from Bayes' rule.

□

A.3 Proof of Theorem 3

Proof. Using (3), we have:

$$\begin{aligned}\nabla_\theta V^{\pi_\theta}(x_0) &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_a\sum_{k\geq 0}\gamma^k\nabla\pi_\theta(a|X_k)A^\pi(X_k,a)\Big]\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_a\sum_{k\geq 0}\gamma^k\nabla\pi_\theta(a|X_k)\Big(r(X_k,a)-r^{\pi_\theta}(X_k)+\sum_{t\geq k+1}\gamma^{t-k}\Big(\frac{h_\beta(a|X_k,X_t)}{\pi_\theta(a|X_k)}-1\Big)R_t\Big)\Big]\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_a\sum_{k\geq 0}\gamma^k\nabla\pi_\theta(a|X_k)\Big(r(X_k,a)+\sum_{t\geq k+1}\gamma^{t-k}\frac{h_\beta(a|X_k,X_t)}{\pi_\theta(a|X_k)}R_t\Big)\Big].\end{aligned}$$

where the third equality is due to $\sum_a \nabla\pi_\theta(a|X_k)f(X_k) = f(X_k)\sum_a\nabla\pi_\theta(a|X_k) = 0$, for $f(X_k) = r^{\pi_\theta}(X_k) + \sum_{t\geq k+1}\gamma^{t-k}R_t$.

Similarly, for the return version and any action a, we have:

$$\begin{aligned}\nabla_\theta V^{\pi_\theta}(x_0) &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_a\sum_{k\geq 0}\gamma^k\nabla\pi_\theta(a|X_k)A^\pi(X_k,a)\Big]\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_a\sum_{k\geq 0}\gamma^k\pi(a|X_k)\nabla\log\pi_\theta(a|X_k)A^\pi(X_k,a)\Big]\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_{k\geq 0}\gamma^k\nabla\log\pi_\theta(A_k|X_k)A^\pi(X_k,A_k)\Big]\\ &= \mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi_\theta)}\Big[\sum_{k\geq 0}\gamma^k\nabla\log\pi_\theta(A_k|X_k)\Big(1-\frac{\pi(A_k|X_k)}{h_z(A_k|X_k,Z(\tau_{k:\infty}))}\Big)Z(\tau_{k:\infty})\Big].\end{aligned}$$

□

A.4 Proof of Proposition 1

Proof. We have:

$$\begin{aligned}&\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\sum_s\gamma^s\nabla\log\pi(A_s|X_s)\big(Z_s(\tau)-b_s\big)\Big]\\ =&\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\sum_s\gamma^s\nabla\log\pi(A_s|X_s)Q^\pi(X_s,A_s)\Big]-\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\nabla\log\pi(A_s|X_s)b_s\Big],\\ =&\nabla V(x_0)-\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\nabla\log\pi(A_s|X_s)\frac{\pi(A_s|X_s)}{h_z(A_s|X_s,Z_s(\tau))}Z_s(\tau)\Big],\\ \overset{(i)}{=}&\nabla V(x_0)-\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[\mathbb{E}_{A_s\sim\pi(\cdot|X_s)}\Big[\nabla\log\pi(A_s|X_s)\underbrace{\mathbb{E}_{\tau\sim\mathcal{T}(X_s,A_s,\pi)}\Big[\frac{\pi(A_s|X_s)}{h_z(A_s|X_s,Z_s(\tau))}Z_s(\tau)\Big]}_{V^\pi(X_s)}\Big]\Big],\\ =&\nabla V(x_0)-\mathbb{E}_{\tau\sim\mathcal{T}(x_0,\pi)}\Big[V^\pi(X_s)\sum_{a\in\mathcal{A}}\nabla\pi(a|X_s)\Big],\\ =&\nabla V(x_0).\end{aligned}$$

where (i) follows from Theorem 2. □

B Other variants

Analogously to Theorems 1 and 2, we can obtain the V- and Q-functions for state and return conditioning, respectively. We have:

Theorem 4. *Consider an action a for which $\pi(a|x) > 0$ and $\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) > 0$ for any state X_k sampled on $\tau\sim\mathcal{T}(x,a,\pi)$:*

$$V^\pi(x) = \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[\sum_{k\geq 0}\gamma^k\frac{\pi(a|x)}{h_k(a|x,X_k)}R_k\Big].$$

Proof. We can flip the result of Lemma 1 for actions a for which $\pi(a|x) > 0$ and $\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) > 0$.

$$\frac{\pi(a|x)}{h_k(a|x,y)} = \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)}. \tag{11}$$

Let $r^\pi(y) = \sum_{a\in\mathcal{A}} \pi(a|y)r(y,a)$. We have

$$\begin{aligned}
V^\pi(x) &= \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 0}\gamma^k R_k\Big] \\
&= \sum_{k\geq 0}\sum_{y\in\mathcal{X}}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)r^\pi(y) \\
&= \sum_{k\geq 0}\sum_{y\in\mathcal{X}}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)}r^\pi(y) \\
&= \sum_{k\geq 0}\sum_{y\in\mathcal{X}}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)\frac{\pi(a|x)}{h_k(a|x,y)}r^\pi(y) \\
&= \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\Big[\sum_{k\geq 0}\gamma^k \frac{\pi(a|x)}{h_k(a|x,X_k)}R_k\Big].
\end{aligned}$$

□

Theorem 5. *Consider an action a for which $\pi(a|x) > 0$. We have:*

$$Q^\pi(x,a) = \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[Z(\tau)\frac{h_z(a|x,Z(\tau))}{\pi(a|x)}\Big]. \tag{12}$$

Proof. The Q-function writes:

$$\begin{aligned}
Q^\pi(x,a) &= \mathbb{E}_{\tau\sim\mathcal{T}(x,a,\pi)}\big[Z(\tau)\big], \\
&= \int_z z\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)dz, \\
&= \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,a,\pi)}(Z(\tau) = z)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)dz, \\
&= \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z|A_0 = a)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)dz, \\
&\overset{(i)}{=} \int_z z\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|Z(\tau) = z)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a)}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)dz, \\
&= \int_z z\frac{h_z(a|x,z)}{\pi(a|x)}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(Z(\tau) = z)dz, \\
&= \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[Z(\tau)\frac{h_z(a|x,Z(\tau))}{\pi(a|x)}\Big],
\end{aligned}$$

where (i) follows from Bayes' rule. □

C Time-Independent State-Conditional Case

We begin by introducing a time independent variant of state-conditional distribution. Let $\beta \in [0,1)$ and $\rho(k) = \beta^{k-1}(1-\beta)$ be the geometric distribution on $k \in \mathbb{N}^+$. Then the state-conditional distribution $h_\beta(a|y,x)$ writes as follows for a future state y:

$$h_\beta(a|x,y) \overset{\text{def}}{=} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y, k\sim\rho). \tag{13}$$

We draw the attention of readers to the difference between the new definition of h_β and the original one in Eq. 2: in this case the timestep k is a random event drawn from the distribution ρ, whereas in Eq. 2 the timestep k is a fixed scalar.

We now show that the result of Theorem 1 extends to the case of h_β with the choice of $\beta = \gamma$.

Theorem 6. *Consider an action a and a state x for which $\pi(a|x)$>0. Set the scalar $\beta = \gamma$. Then Q^π writes as*

$$Q^\pi(x,a) = r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 1}\gamma^k \frac{h_\beta(a|x,X_k)}{\pi(a|x)} R_k\Big].$$

Proof. Let us introduce the coefficient $c_\gamma = \frac{\gamma}{1-\gamma}$ such that $c_\gamma \rho(k) = \gamma^k$. By definition of the Q-function for a state-action couple (x,a), we have

$$Q^\pi(x,a) = r(x,a) + \sum_{k\geq 1}\sum_{y\in\mathcal{X}} \gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) r^\pi(y),$$

which can be rewritten:

$$Q^\pi(x,a) = r(x,a) + c_\gamma \sum_{y\in\mathcal{X}}\sum_{k\geq 1} \rho(k) \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a) r^\pi(y). \tag{14}$$

From the law of total probability and the independence between the events $k\sim\rho$ and $A_0 = a$:

$$\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a, k\sim\rho) = \sum_{k\geq 1}\rho(k)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a).$$

Combining this with Eq. (14) we deduce

$$Q^\pi(x,a) = r(x,a) + c_\gamma \sum_{y\in\mathcal{X}} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a, k\sim\rho) r^\pi(y). \tag{15}$$

From applying the Bayes' rule and independence between the events $k\sim\rho$ and $A_0 = a$, we have

$$\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a, k\sim\rho) = \frac{h_\beta(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|k\sim\rho)}{\pi(a|x)}.$$

Combining this with Eq. (15) we deduce

$$\begin{aligned}
Q^\pi(x,a) &= r(x,a) + c_\gamma \sum_{y\in\mathcal{X}} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|k\sim\rho)\frac{h_\beta(a|x,y)}{\pi(a|x)} r^\pi(y),\\
&= r(x,a) + \sum_{y\in\mathcal{X}}\sum_{k\geq 1}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)\frac{h_\beta(a|x,y)}{\pi(a|x)} r^\pi(y),\\
&= r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\left[\sum_{k\geq 1}\gamma^k \frac{h_\beta(a|X_k,x)}{\pi(a|x)} r^\pi(X_k)\right],\\
&= r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\left[\sum_{k\geq 1}\gamma^k \frac{h_\beta(a|X_k,x)}{\pi(a|x)} R_k\right].
\end{aligned}$$

□

We now extend the result of Theorem 6 to the case of T-step bootstrapped return. Let ρ_T be the distribution on the set $\{1,2,\ldots,T\}$ defined as

$$\rho_T(k) \stackrel{\text{def}}{=} \begin{cases} \beta^{k-1}(1-\beta) & 1\leq k < T \\ \beta^{T-1} & k = T \end{cases} \tag{16}$$

We also define the T-step state-conditional distribution $h_{\beta,T}(a|y,x)$ for a future state y:

$$h_{\beta,T}(a|x,y) \stackrel{\text{def}}{=} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y, k\sim\rho_T). \tag{17}$$

Theorem 7. *Consider an action a and a state x for which $\pi(a|x)$>0. Set the scalar $\beta = \gamma$. Then Q^π writes as*

$$Q^\pi(x,a) = r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\Big[\sum_{k\geq 1}^{T-1}\gamma^k \frac{h_{\beta,T}(a|x,X_k)}{\pi(a|x)}R_k + \gamma^T \frac{h_{\beta,T}(a|x,X_T)}{\pi(a|x)}V^\pi(X_T)\Big].$$

Proof. By definition of the Q-function for a state-action couple (x,a), we have

$$Q^\pi(x,a) = r(x,a) + \sum_{k=1}^{T-1}\sum_{y\in\mathcal{X}}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)r^\pi(y) + \sum_{y\in\mathcal{X}}\gamma^T \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_T = y|A_0 = a)V^\pi(y),$$

From the definition of the (normalized) discounted visit distribution $\tilde{d}^\pi(z|y) \stackrel{\text{def}}{=} (1-\gamma)\sum_k \gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(y,\pi)}(X_k = z)$, we have:

$$V^\pi(y) = \frac{1}{1-\gamma}\sum_{z\in\mathcal{X}}\tilde{d}^\pi(z|y)r^\pi(z).$$

Therefore $Q^\pi(x,a)$ can be rewritten:

$$\begin{aligned} Q^\pi(x,a) = r(x,a) &+ \sum_{k=1}^{T-1}\sum_{y\in\mathcal{X}}\gamma^k \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)r^\pi(y) \\ &+ \frac{\gamma^T}{1-\gamma}\sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_T = y|A_0 = a)\tilde{d}^\pi(z|y)r^\pi(z). \end{aligned}$$

Now let us define the following distribution $\mu_k(.|y)$ for each (k,y):

$$\mu_k(z|y) \stackrel{\text{def}}{=} \begin{cases} \mathbf{1}_{z=y} & 1 \leq k < T \\ \tilde{d}^\pi(z|y) & k = T. \end{cases} \tag{18}$$

Thus we can rewrite $Q^\pi(x,a)$ as:

$$Q^\pi(x,a) = r(x,a) + c_\gamma \sum_{k=1}^{T}\sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}}\rho_T(k)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)\mu_k(z|y)r^\pi(z).$$

From the law of total probability, independence between the events $k\sim\rho_T$ and $A_0 = a$ and the Markovian relation between X_k and Z_k (Z_k is a random variable with distribution $\mu_k(.|X_k)$):

$$\begin{aligned} \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|A_0 = a, k\sim\rho_T) &= \sum_{k=1}^{T}\rho_T(k)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|A_0 = a), \\ &= \sum_{k\geq 1}\rho_T(k)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y|A_0 = a)\mu_k(Z_k = z|X_k = y). \end{aligned}$$

Therefore we have:

$$Q^\pi(x,a) = r(x,a) + c_\gamma \sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}}\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|A_0 = a, k\sim\rho_T)r^\pi(z).$$

Then, by applying the Bayes' rule:

$$\frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|A_0 = a, k\sim\rho_T)}{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y, Z_k = z, k\sim\rho_T)} = \frac{\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|k\sim\rho_T)}{\pi(a|x)}.$$

In addition, by the Markov property:

$$\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y, Z_k = z, k\sim\rho_T) = \mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(A_0 = a|X_k = y, k\sim\rho_T),$$

$$= h_{\beta,T}(a|x,y).$$

Therefore:

$$\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|A_0 = a, k \sim \rho_T) = \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|k \sim \rho_T)}{\pi(a|x)}.$$

Thus, we can rewrite $Q^\pi(x,a)$ as:

$$\begin{aligned}
Q^\pi(x,a) &= r(x,a) + c_\gamma \sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y, Z_k = z|k \sim \rho_T)}{\pi(a|x)} r^\pi(z), \\
&= r(x,a) + c_\gamma \sum_{k=1}^{T}\sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\rho_T(k)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)\mu_k(Z = z|X_k = y)}{\pi(a|x)} r^\pi(z), \\
&= r(x,a) + \sum_{k=1}^{T-1}\gamma^k \sum_{y\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\pi(a|x)} r^\pi(y) \\
&\quad + \gamma^T \sum_{y\in\mathcal{X}}\sum_{z\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\pi(a|x)} \tilde{d}^\pi(z|y) r^\pi(z), \\
&= r(x,a) + \sum_{k=1}^{T-1}\gamma^k \sum_{y\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\pi(a|x)} r^\pi(y) \\
&\quad + \gamma^T \sum_{y\in\mathcal{X}} \frac{h_{\beta,T}(a|x,y)\mathbb{P}_{\tau\sim\mathcal{T}(x,\pi)}(X_k = y)}{\pi(a|x)} V^\pi(y), \\
&= r(x,a) + \mathbb{E}_{\tau\sim\mathcal{T}(x,\pi)}\left[\sum_{k=1}^{T-1}\gamma^k \frac{h_{\beta,T}(a|x,X_k)}{\pi(a|x)} r^\pi(X_k) + \gamma^T \frac{h_{\beta,T}(a|x,X_T)}{\pi(a|x)} V^\pi(X_T)\right],
\end{aligned}$$

which concludes the proof. □

D Algorithms

Algorithm 1 State-conditional HCA

Given: Initial π, h_β, V, $\hat{r}$; horizon T

for $k = 1, \ldots$ **do**
  Sample $\tau = X_0, A_0, R_0, \ldots, R_T$ from π
  for $i = 0, \ldots, T-1$ **do** ▷ Train hindsight distribution
    for $j = i, \ldots, T$ **do**
      Train $h_\beta(A_i|X_i, X_j)$ via cross-entropy
    end for
  end for
  for $i = 0, \ldots, T-1$ **do** ▷ Train baseline and reward predictor
    $Z = 0$
    for $j = i, \ldots, T-1$ **do**
      $Z \leftarrow Z + \gamma^{j-i} R_j$
    end for
    $Z \leftarrow Z + \gamma^{T-i} V(X_T)$
    Update $V(X_i)$ towards Z
    Update $\hat{r}$ towards R_i
  end for
  for $i = 0, \ldots, T-1$ **do** ▷ Train policy of all actions with the hindsight-conditioned return
    for all actions a **do**
      $Z_h = \pi(a|X_i, a)\hat{r}(X_i, a)$
      for $j = i+1, \ldots, T-1$ **do**
        $Z_h \leftarrow Z_h + \gamma^{j-i} \frac{h_\beta(a|X_i, X_j)}{\pi(a|X_i)} R_j$
      end for
      $Z_{h,a} \leftarrow Z_h + \gamma^{T-i} \frac{h_\beta(a|X_i, X_T)}{\pi(a|X_i)} V(X_T)$
    end for
    Follow the gradient $\sum_a \nabla \pi(a|X_i) Z_{h,a}$
  end for
end for

Algorithm 2 Return-conditional HCA

Given: Initial π, h_z, V

for $k = 1, \ldots$ **do**
  Sample $\tau = X_0, A_0, R_0, \ldots$ from π
  for $i = 0, 1, \ldots$ **do**
    Compose the return $Z(\tau_{i:\infty})$ starting from X_i
    Train $h_z(A_i|X_i, Z_i)$ via cross-entropy
    $Z_h \leftarrow \left(1 - \frac{\pi(A_i|X_i)}{h_z(A_i|X_i, Z(\tau_{i:\infty}))}\right) Z(\tau_{i:\infty})$
    Follow the gradient $\nabla \log \pi(A_i|X_i) Z_h$
  end for
end for

E Experiment Details

The learning rate α for the baseline was chosen to be the best value from $[0.1, 0.2, 0.3, 0.4]$, while our model hyperparameters (the learning rate α_h for h, and the number of bins n_b for the return version of HCA were selected informally to be $\alpha = 0.3, \alpha_b = 0.4, n_b = 3$ for the results in Fig. 4, and $n_b = 10$ elsewhere. Return HCA is sensitive to n_b, but all variants are robust to the choice of learning rate.

F Bootstrapping with state HCA

Consider the Delayed Effect task from Section 5, in which an action causes an outcome T steps in the future, with everything in between being irrelevant. It is not immediately obvious why state HCA should be beneficial when one bootstraps with $n < T$. Indeed, if h was perfect, the intermediate coefficient would be uninformative. However, we observe the opposite, precisely because V, π and h are being learned at the same time, but with different learning dynamics. In particular, in this case h moves faster than π (independently of the learning rate) as it is updated towards 1 for any observed sample, while π updates are modulated by the return. Now consider some interim $V(y) < 0$. The negative value implies that the policy at the initial state x prefers the bad action a over the good action b: $\pi(a|x) > \pi(b|x)$. But this in turn implies that $h(a|x, y)$ has been observed more frequently, and since h is quicker to update: $h(a|x, y) > \pi(a|x)$. Now, take the policy gradient theorem (7) with π as a baseline. The HCA return becomes $(h(a|x, y) - \pi(a|x))V(y) < 0$ and discourages the bad action. Similarly, $(h(b|x, y) - \pi(b|x))V(y) > 0$ and the good action is encouraged. We tested different learning rates, and initializations, and the effect persisted.

Chapter 5

Counterfactual Credit Assignment, ICML 2022

5.1 In short

5.1.1 Motivations

In this second article [6], published at ICML 2021, we look at the credit assignment problem under a specific angle. To be data efficient, credit assignment methods need to disentangle the effects of a given action of the agent from the effects of external factors and subsequent actions.

External factors in reinforcement learning are any factors that affect the state of the environment or the agent's reward, but are outside of the agent's control. This can include things like the actions of other agents in the environment, changes in the environment state due to natural processes or events, sensor noise, hardware failures, ...

These factors can make credit assignment difficult because they can obscure the relationship between the agent's actions and its rewards. For example, if an agent is trying to learn to walk, and it happens to fall just after taking a step, it is difficult for the agent to determine whether the fall was caused by its own action, or by an external factor such as a slippery surface.

5.1.2 Approach

We propose to draw inspiration from counterfactuals from causality theory to improve credit assignment in model-free reinforcement learning. The key idea is to use value functions conditioned on future events, and learn to extract relevant information from a trajectory.

Relevant information here corresponds to all information that is predictive of the return while being independent of the agent's action at time t. This allows the agent to separate the effect of its own actions from the effect of external factors and subsequent actions which will enable refined credit assignment and therefore

faster and more stable learning.

We propose a family of policy gradient algorithms that use these future-conditional value functions as baselines. We show that these algorithms are provably lower variance that vanilla policy gradient, and we develop valid, practical variants that avoid the potential bias from conditioning on future information. One variant explicitly tries to remove information from the hindsight conditioning that depends on the current action while the second variant avoids the potential bias from conditioning on future information thanks to a technique related to important sampling.

5.1.3 Results

We evaluate our algorithm on a number of illustrative but complex problems. We show that the "Counterfactual Credit Assignment" (CCA) algorithm outperforms standard policy gradient algorithms on a number of tasks that are challenging for credit assignment reasons as the return in those is highly influenced by external factors.

Overall, we propose a novel and effective approach able to disentangle the impact of the agent's actions from the impact of exogenous factors on the overall return. This allows the agent to learn more efficiently. This line of work is a great avenue for research to better generalize to new situations thanks to our expressive and future-conditioned credit assignment approach.

Let's mention a few potential real-world applications of this work. Robots often operate in dynamic and unpredictable environments, where it is difficult to distinguish between the effects of their own actions and the effects of external factors. CCA could help robots to learn more quickly and effectively in these environments, and to become more robust to disturbances.

Furthermore, financial markets are complex and volatile, and it is difficult for investors to predict how their actions will affect their returns. CCA could help investors to better understand the risks and rewards of different investment strategies, and to make more informed decisions when investing in the very stochastic financial market.

Finally, in healthcare, it is important to be able to identify the factors that contribute to patient outcomes. CCA could help doctors to better understand the effects of different treatments and interventions, and to personalize care for their patients.

The work and general idea of conditioning value functions on hindsight information was impactful as evidenced by its substantial citation count (55 citations at the time of writing).

Counterfactual Credit Assignment in Model-Free Reinforcement Learning

Thomas Mesnard [*1] **Théophane Weber** [*1] **Fabio Viola** [1] **Shantanu Thakoor** [1] **Alaa Saade** [1] **Anna Harutyunyan** [1] **Will Dabney** [1] **Tom Stepleton** [1] **Nicolas Heess** [1] **Arthur Guez** [1] **Éric Moulines** [2] **Marcus Hutter** [1] **Lars Buesing** [1] **Rémi Munos** [1]

Abstract

Credit assignment in reinforcement learning is the problem of measuring an action's influence on future rewards. In particular, this requires separating *skill* from *luck*, i.e. disentangling the effect of an action on rewards from that of external factors and subsequent actions. To achieve this, we adapt the notion of counterfactuals from causality theory to a model-free RL setup. The key idea is to condition value functions on *future* events, by learning to extract relevant information from a trajectory. We formulate a family of policy gradient algorithms that use these future-conditional value functions as baselines or critics, and show that they are provably low variance. To avoid the potential bias from conditioning on future information, we constrain the hindsight information to not contain information about the agent's actions. We demonstrate the efficacy and validity of our algorithm on a number of illustrative and challenging problems.

1. Introduction

Reinforcement learning (RL) agents act in their environments and learn to achieve desirable outcomes by maximizing a reward signal. A key difficulty is the problem of *credit assignment* (Minsky, 1961), i.e. to understand the relation between actions and outcomes, and to determine to what extent an outcome was caused by external, uncontrollable factors. In doing so we aim to disentangle the relative aspects of 'skill' and 'luck' in an agent's performance. One possible solution to this problem is for the agent to build a model of the environment, and use it to obtain a more fine-grained understanding of the effects of an action. While this topic has recently generated a lot of interest (Heess et al., 2015; Ha & Schmidhuber, 2018; Hamrick, 2019; Kaiser et al., 2019; Schrittwieser et al., 2019), it remains difficult to model complex, partially observed environments.

In contrast, model-free reinforcement learning algorithms such as policy gradient methods (Williams, 1992; Sutton et al., 2000) perform simple time-based credit assignment, where events and rewards happening after an action are credited to that action, *post hoc ergo propter hoc*. While unbiased in expectation, this coarse-grained credit assignment typically has high variance, and the agent will require a large amount of experience to learn the correct relation between actions and rewards. Another issue is that existing model-free methods are not capable of *counterfactual reasoning*, i.e. reasoning about what would have happened had different actions been taken *with everything else remaining the same*. Given a trajectory, model-free methods can in fact only learn about the actions that were actually taken to produce the data, and this limits the ability of the agent to learn efficiently.

As environments grow in complexity due to partial observability, scale, long time horizons, and increasing number of agents, actions taken by an agent will only affect a vanishing part of the outcome, making it increasingly difficult to learn from classical reinforcement learning algorithms. We need better credit assignment techniques.

In this paper, we investigate a new method of credit assignment for model-free reinforcement learning which we call *Counterfactual Credit Assignment* (CCA). CCA leverages *hindsight* information to implicitly perform counterfactual evaluation–an estimate of the return for actions other than the ones which were chosen. These counterfactual returns can be used to form unbiased and lower variance estimates of the policy gradient by building future-conditional baselines. Unlike classical Q functions, which also provide an estimate of the return for all actions but do so by averaging over all possible futures, our methods provide trajectory-specific counterfactual estimates, i.e. an estimate of the return for different actions, but keeping as many of the ex-

[*]Equal contribution [1]DeepMind [2]INRIA XPOP, CMAP, École Polytechnique, Palaiseau, France. Correspondence to: Théophane Weber <theophane@deepmind.com>, Thomas Mesnard <mesnard@deepmind.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

ternal factors constant between the return and its counterfactual estimate[1]. Such a method would perform finer-grained credit assignment and could greatly improve data efficiency in environments with complex credit assignment structures. Our method is inspired by ideas from causality theory, but does not require learning a model of the environment.

Our main contributions are: a) introducing a family of novel policy gradient estimators that leverage hindsight information and generalizes previous approaches, b) proposing a practical instantiation of this algorithm with sufficiency conditions for unbiasedness and guarantees for lower variance, c) introducing a set of environments which further our understanding of when credit assignment is made difficult due to exogenous noise, long-term effects and task interleaving, and thus leads to poor policy learning, d) demonstrating the improved performance of our algorithm on these environments, e) formally connecting our results to notions of counterfactuals in causality theory, further linking the causal inference and reinforcement learning literatures.

2. Counterfactual Credit Assignment

2.1. Notation

We use capital letters for random variables and lowercase for the value they take. Consider a generic MDP $(\mathcal{X}, \mathcal{A}, p, r, \gamma)$. Given a current state $x \in \mathcal{X}$ and assuming an agent takes action $a \in \mathcal{A}$, the agent receives reward $r(x, a)$ and transitions to a state $y \sim p(\cdot|x, a)$. The state (resp. action, reward) of the agent at step t is denoted X_t (resp. A_t, R_t). The initial state of the agent X_0 is a fixed x_0. The agent acts according to a policy π, i.e. action A_t is sampled from the policy $\pi_\theta(\cdot|X_t)$ where θ are the policy parameters, and aims to optimize the expected discounted return $\mathbb{E}[G] = \mathbb{E}[\sum_t \gamma^t R_t]$. The return G_t from step t is $G_t = \sum_{t' \geq t} \gamma^{t'-t} R_{t'}$. Note $G = G_0$. Finally, we define the score function $s_\theta(\pi_\theta, a, x) = \nabla_\theta \log \pi_\theta(a|x)$; the score function at time t is denoted $S_t = \nabla_\theta \log \pi_\theta(A_t|X_t)$. In the case of a partially observed environment, we assume the agent receives an observation E_t at every time step, and simply define X_t to be the set of all previous observations, actions and rewards $X_t = (O_{\leq t})$, with $O_t = (E_t, A_{t-1}, R_{t-1})$.[2] $\mathbb{P}(X)$ will denote the probability distribution of a random variable X.

2.2. Policy gradient algorithms

We begin by recalling two forms of policy gradient algorithms and the credit assignment assumptions they make. The first is the REINFORCE algorithm introduced by Williams (1992), which we will also call the single-action policy gradient estimator. The gradient of $\mathbb{E}[G]$ is given by:

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\Big[\sum_{t \geq 0} \gamma^t \, S_t \, (G_t - V(X_t))\Big], \qquad (1)$$

where $V(X_t) = \mathbb{E}[G_t|X_t]$. Let's note here that $V(X_t)$ (resp. $Q(X_t, A_t) = \mathbb{E}[G_t|X_t, A_t]$) is the value function (resp. Q-function) for the policy π_θ but for notation simplicity the dependence on the policy will be implicit through the rest of this paper.

The appeal of this estimator lies in its simplicity and generality: to evaluate it, the only requirement is the ability to simulate trajectories, and compute both the score function and the return.
Note that subtracting the value function $V(X_t)$ from the return G_t does not bias the estimator and typically reduces variance, since the resulting estimate makes an action A_t more likely proportionally not to the return, but to which extent the return was higher than what was expected before the action was taken (Williams, 1992). Such a function will be called a *baseline* in the following. In theory, the baseline can be any function of X_t. It is however typically assumed that it does not depend on any variable 'from the future' (including the action about to be taken, A_t), i.e. with time index greater than t, since including variables which are (causally) affected by the action generally results in a biased estimator (Weber et al., 2019).

This estimator updates the policy through the score term; note however the learning signal only updates the policy $\pi_\theta(a|X_t)$ for the taken action $A_t = a$ (other actions are only updated through normalization of action probabilities). The policy gradient theorem from Sutton et al. (2000), which we will also call all-action policy gradient, shows it is possible to provide learning signal to all actions, given we have access to a Q-function, $Q(x, a) = \mathbb{E}[G_t|X_t = x, A_t = a]$, which we will call a *critic* in the following. The gradient of $\mathbb{E}[G]$ is given by:

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\Big[\sum_{t \geq 0} \gamma^t \sum_{a \in \mathcal{A}} \nabla_\theta \pi(a|X_t) Q(X_t, a)\Big]. \qquad (2)$$

2.3. Intuitive example on hindsight reasoning and skill versus luck

Imagine a scenario in which Alice just moved to a new city, is learning to play soccer, and goes to the local soccer field to play a friendly game with a group of other kids she has never met. As the game goes on, Alice does not seem to play at her best and makes some mistakes. It turns out however her partner Megan is a strong player, and eventually scores the goal that makes the game a victory. What should Alice learn from this game?

When using the single-action policy gradient estimate, the

[1] From from a causality standpoint, one-step action-value functions are interventional concepts ("What would happen if") instead of counterfactuals ("What would have happened if").

[2] Previous actions and rewards are provided as part of the observation as it is generally beneficial to do so in partially observable Markov decision processes.

outcome of the game being a victory, and assuming a ± 1 reward scheme, all her actions taken during the game are made more likely; this is in spite of the fact that during this particular game she may not have played well and that the victory is actually due to her strong teammate. From an RL point of view, her actions are wrongly credited for the victory and positively reinforced as a result; effectively, Alice was lucky rather than skillful. Regular baselines do not mitigate this issue, as Alice did not a priori know the skill of Megan, resulting in an assumption that Megan was of average strength and therefore a guess that their team had a 50% chance of winning. This could be fixed by understanding that Megan's strong play were not a consequence of Alice's play, that her skill was a priori unknown but known in hindsight, and that it is therefore valid to retroactively include her skill level in the baseline. A hindsight baseline, conditioned on Megan's estimated skill level, would therefore be closer to 1, driving the advantage estimate (and corresponding learning signal) close to 0.

As pointed out by Buesing et al. (2019), situations in which hindsight information is helpful in understanding a trajectory are frequent. In that work, the authors adopt a model-based framework, where hindsight information is used to ground counterfactual trajectories (i.e. trajectories under *different actions, but same randomness*). Our proposed approach follows a similar intuition, but is model-free: we attempt to *measure*—instead of model— information known in hindsight to compute a *future-conditional baseline*, but in a way that maintains unbiasedness. As we will see later, this corresponds to a constraint that the captured information must not have been caused by the agent.

2.4. Future-conditional (FC-PG) and Counterfactual (CCA-PG) Policy Gradient Estimators

Intuitively, our approach for assigning proper credit to action A_t relies on measuring statistics Φ_t that capture relevant information from the trajectory (e.g. including observations $O_{t'}$ at times t' greater than t). We then learn value functions or critics which are conditioned on the additional hindsight information contained in Φ_t. In general, these future-conditional values and critics would be biased for use in a policy gradient algorithm; we therefore use an importance correction term to eliminate this bias.

Theorem 1 (Future-Conditional Policy Gradient (FC-PG) estimators)**.** *Let Φ_t be an arbitrary random variable. Assuming that $\frac{\pi(a|X_t)}{\mathbb{P}(a|X_t,\Phi_t)} < \infty$ for all a, the following is the single-action unbiased estimator of the gradient of $\mathbb{E}[G]$:*

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\Big[\sum_t \gamma^t \, S_t\Big(G_t - \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t,\Phi_t)\Big)\Big] \tag{3}$$

where $V(x,\phi) = \mathbb{E}[G_t|X_t = x, \Phi_t = \phi]$ is the future Φ-conditional value function .

With no requirements on Φ_t, we also have an all-action unbiased estimator:

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\big[\sum_{t,a} \gamma^t \nabla_\theta \log \pi(a|X_t) \mathbb{P}(a|X_t,\Phi_t) Q(X_t,\Phi_t,a)\big]$$

where $Q(x,\phi,a) = \mathbb{E}[G_t|X_t = x, \Phi_t = \phi, A_t = a]$ is the future-conditional Q function (critic). Furthermore, we have $Q(X_t,a) = \mathbb{E}\left[Q(X_t,\Phi_t,a)\frac{\mathbb{P}(a|X_t,\Phi_t)}{\pi(a|X_t)}\right]$.

Intuitively, the $\frac{\pi(a|X_t)}{\mathbb{P}(a|X_t,\Phi_t)} < \infty$ condition means that knowing Φ_t should not preclude any action a which was possible for π from having potentially produced Φ_t. A counterexample is $\Phi_t = A_t$; knowing Φ_t precludes any action $a \neq A_t$ from having produced Φ_t. Typically, Φ_t will be chosen to a function of the present and future trajectory $(X_s, A_s, R_s)_{s \geq t}$. The estimators above are very general and generalize similar estimators (HCA) introduced by Harutyunyan et al. (2019) (see App. C for a discussion of how HCA can be rederived from FC-PG) and different choices of Φ will have varying properties. Φ may be hand-crafted using domain knowledge, or, as we will see later, learned using appropriate objectives. Note that in general an FC-PG estimator doesn't necessarily have lower variance (a good proxy for fine-grained credit assignment) than the classical policy gradient estimator; this is due to the variance introduced by the importance weighting scheme. It would be natural to study an estimator where this effect is nullified through independence of the action and statistics Φ (resulting in a ratio of 1).

The resulting advantage estimate could thus be interpreted not just as an estimate of 'what outcome should I expect', but also a measure of 'how (un)lucky did I get?' and 'what other outcomes might have been possible in this precise situation, had I acted differently'. It will in turn provide finer-grained credit for action A_t in a sense to be made precise below.

Corollary 1 (Counterfactual Policy Gradient (CCA-PG))**.** *If A_t is independent from Φ_t given X_t, the following is an unbiased* single-action *estimator of the gradient of $\mathbb{E}[G]$:*

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\left[\sum_t \gamma^t \, S_t \left(G_t - V(X_t,\Phi_t)\right)\right]. \tag{4}$$

Furthermore, the hindsight advantage estimate has no higher variance than the forward one:

$$\mathbb{E}\left[\left(G_t - V(X_t,\Phi_t)\right)^2\right] \leq \mathbb{E}\left[\left(G_t - V(X_t)\right)^2\right].$$

Similarly, for the all-action *estimator:*

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\Big[\sum_t \gamma^t \sum_a \nabla_\theta \pi(a|X_t) Q(X_t,\Phi_t,a)\Big]. \tag{5}$$

Also, we have for all a,

$$Q(X_t,a) = \mathbb{E}[Q(X_t,\Phi_t,a)|X_t, A_t = a]$$

The benefit of the first estimator (equation 4) is clear: under the specified condition, and compared to the regular policy gradient estimator, the CCA estimator also has no bias, but the variance of its advantage estimate $G_t - V(X_t, \Phi_t)$ (the critical component behind variance of the overall estimator) is no higher.

For the all-action estimator, the benefits of CCA (equation 5) are less self-evident, since this estimator has *higher* variance than the regular all action estimator (which has variance 0). The interest here lies in bias due to learning imperfect Q functions. Both estimators require learning a Q function from data; any error in Q leads to a bias in π. Learning $Q(X_t, a)$ requires averaging over all possible trajectories initialized with state X_t and action a: in high variance situations, this will require a lot of data. In contrast, $Q(X_t, \Phi_t, a)$ predicts the average of the return G_t *conditional* on (X_t, Φ_t, a); if Φ_t has a high impact on G_t, the variance of that conditional return will be lower, and learning its average will in turn be far easier and data efficient.

2.5. Learning the relevant statistics: practical implementation of CCA-PG

The previous section proposes a sufficient condition on Φ for useful estimators to be derived. A question remains - how to compute such a Φ from the trajectory? While useful Φ could be handcrafted using expert knowledge, we propose to *learn* to extract Φ from the trajectory. The learning signal will be guided by two objectives: first, we will encourage Φ_t to be conditionally independent from A_t, as it is required for the estimator to be valid. Second, corollary 1 highlights that hindsight features which are predictive of the return lead to a decreased variance of the advantage estimate. To summarize, we want Φ to be predictive of the return while being independent of the action being currently credited. The corresponding hindsight conditional baseline would capture the 'luck' part of the outcome while the advantage estimate would capture the 'skill' aspect of it. We detail our agent components and losses below. See also Fig. 1 for a depiction of the resulting architecture and Appendix A for more details.

Agent components:

- **Agent network**: Our algorithm can generally be applied to arbitrary environments (e.g. POMDPs), so we assume the agent constructs an internal state X_t from past observations $(O_{t'})_{t' \leq t}$ using an arbitrary network, for instance an RNN, i.e. $X_t = \text{RNN}_{\theta_{\text{fs}}}(O_t, X_{t-1})$[3]. From X_t the agent computes a policy $\pi_{\theta_{\text{fs}}}(a|X_t)$, where θ_{fs} denotes the parameters of the representation network and policy.
- **Hindsight network**: Additionally, we assume the agent uses a hindsight network φ with parameters θ_{hs} which computes a hindsight statistic $\Phi_t = \varphi((X, A, R))$ which may depend arbitrarily on the vectors of observations, agent states and actions (in particular, it may depend on observations from timesteps $t' \geq t$).
- **Value network**: The third component is a future-conditional value network $V_{\theta_{\text{V}}}(X_t, \Phi_t)$, with parameters θ_{V}.
- **Hindsight predictor**: The last component is a probabilistic predictor h_ω with parameters ω that takes X_t, Φ_t as input and outputs a distribution over A_t which is used to enforce the independence condition.

Learning objectives:

- The first loss is the hindsight baseline loss $\mathcal{L}_{\text{hs}} = \sum_t (G_t - V_{\theta_{\text{V}}}(X_t, \Phi_t))^2$.
- The second loss is the independence loss, which ensures the conditional independence between A_t and Φ_t. There exists multiple ways to measure dependence between random variables; we assume a surrogate *independence maximization* (IM) loss $\mathcal{L}_{\text{IM}}(X_t)$ which is non-negative and zero if and only if A_t and Φ_t are conditionally independent given X_t. An example is to choose the Kullback-Leibler divergence between the distributions $\mathbb{P}(A_t|X_t)$ and $\mathbb{P}(A_t|X_t, \Phi_t)$. In this case, the KL can be estimated by $\sum_a \mathbb{P}(a|X_t)\left(\log \mathbb{P}(a|X_t) - \log \mathbb{P}(a|X_t, \Phi_t)\right)$; $\log \mathbb{P}(a|X_t)$ is simply the policy $\pi(a|X_t)$; the posterior $\mathbb{P}(a|X_t, \Phi_t)$ is generally not known exactly, but we estimate it with the probabilistic predictor $h_\omega(A_t|X_t, \Phi_t)$, which we train with the next loss.
- The third loss is the hindsight predictor loss, which we train by minimizing the supervised learning loss $\mathcal{L}_{\text{sup}} = -\sum_t \mathbb{E}[\log h_\omega(A_t|X_t, \Phi_t)]$ on samples (X_t, A_t, Φ_t) from the trajectory (note that this is a proper scoring rule, i.e. the optimal solution to the loss is the true probability $\mathbb{P}(a|X_t, \Phi_t)$).
- The last loss is the policy gradients surrogate objective, implemented as $\mathcal{L}_{\text{PG}} = \sum_t \log \pi_\theta(A_t|X_t)\overline{(G_t - V(X_t, \Phi_t))}$, where the bar notation indicates that the quantity is treated as a constant from the point of view of gradient computation, as is standard.

The overall loss is therefore $\mathcal{L} = \mathcal{L}_{\text{PG}} + \lambda_{\text{hs}}\mathcal{L}_{\text{hs}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_{\text{IM}}\mathcal{L}_{\text{IM}}$. We again want to highlight the very special role played by ω here: only $\mathcal{L}_{\text{sup}}$ is optimized with respect to ω (the parameters of the probabilistic predictor), while all the other losses are optimized treating ω as a constant.

[3] Obviously, if the environment is fully observed, a feed-forward network suffices.

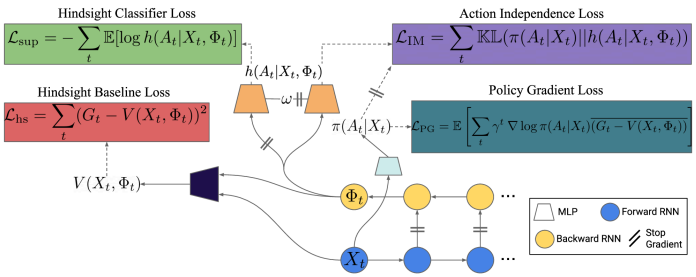


Figure 1: Counterfactual Credit Assignment in a nutshell: (1) The backward RNN which in this example computes the hindsight features is shaped by the hindsight baseline loss. This ensures that it is predictive of the return. (2) However, to have an unbiased baseline, this hindsight feature Φ_t needs to be independent from the action A_t. To that end, we first train a hindsight predictor that tries to predict what action has been taken a time t from X_t and Φ_t. (3) Then the action independence loss helps removing any information about A_t from the hindsight feature Φ_t (This only enforces that the output of the backward RNN Φ_t is independent of the action A_t. However, this could potentially translate in Φ_t being independent from further actions). This loss only impacts the backward RNN and no gradient is being applied to the hindsight predictor MLP. (4) Finally, the policy gradient loss helps improving the policy while no gradient is being sent to the hindsight baseline (i.e as expressed by the bar notation).

3. Connections to causality

In this section we provide a formal connection between the CCA-PG estimator and counterfactuals in causality theory (this connection is investigated in greater depth in appendices F and G).

To this end, we assume that the MDP $(\mathcal{X}, \mathcal{A}, p, r, \gamma)$ in question is generated by an underlying structural causal models (SCM) analogous to (Buesing et al., 2019; Zhang, 2020). In this setting the trajectory $(X_s, A_s, R_s)_{s\geq t}$ and return G_t resulting from the agent-environment interaction is represented as the output of a deterministic function f taking as input the current state X_t, the action A_t, and a set of exogenous random variables $\boldsymbol{\mathcal{E}}$ which do not have any causal ancestors (in the graph). The latter represent the randomness required for sampling all future actions, transitions, and rewards. Such a "reparametrization" of trajectories and return is always possible, i.e. there is always an SCM (possibly non unique) that induces the same joint distribution $\mathbb{P}$ as the original MDP. Intuitively, $\boldsymbol{\mathcal{E}}$ represent all factors external to A_t which affect the outcome[4].

SCMs allow to formally define the notion of counterfactual. Given an observed trajectory $\tau = (X_s, A_s, R_s)_{s\geq t}$, we define the counterfactual trajectory τ' for an alternative action $A'_t = a'_t$ as a the output of the following procedure:

- Abduction: infer the exogenous noise variables ϵ under the factual observation: $\epsilon \sim \mathbb{P}(\mathcal{E}|\tau)$.
- Intervention: Fix the value of A'_t to a'_t (mutilating incoming causal arrows).
- Prediction: Evaluate the counterfactual outcome τ' conditional on the fixed values $\mathcal{E}$ and $A_t = a'_t$ yielding $\tau' = f(x_t, a'_t, \epsilon)$

The counterfactual distribution will be denoted $P(\tau'|\text{observe}(\tau), \text{do}(A'_t = a'_t))$. Note that it typically requires knowledge of the model (SCM) to be computed; samples from the models which do not expose the exogenous variables $\boldsymbol{\mathcal{E}}$ are not typically not sufficient to identify the SCM, as several SCMs may correspond to the same distribution. However, under the CCA assumptions and an additional faithfulness assumption, we can show that the counterfactual return is indeed identifiable and is equal to the future conditional state-action value function:

Theorem 2. *Assume the causal model is faithful (i.e. that conditional independence assumptions are reflected in the graph structure and not only in the parameters). If Φ_t is conditionally independent from A_t given X_t, then the counterfactual distribution, having observed only Φ_t, is identifiable from samples of (X_t, Φ_t, A_t), and we have*

$$\mathbb{E}[G(\tau')|\tau' \sim P(\tau'|X_t = x, \textit{observe}(\Phi_t = \phi), \textit{do}(A'_t = a)\big] = Q(X_t = x, A_t = a, \Phi_t = \phi) \quad (6)$$

4. Numerical experiments

Given its guarantees on lower variance and unbiasedness, we run all our experiments on the single action version of CCA-PG and leave the all-action version for future work. We first investigate a bandit with feedback task, then a task that requires short and long-term credit assignment (i.e. Key-to-Door), and finally an interleaved multi-task setup where each episode is composed of randomly sampled and interleaved tasks. All results for Key-to-Door and interleaved multi-task are reported as median performances over 10 seeds with quartiles represented by a shaded area.

4.1. Bandit with Feedback

We first demonstrate the benefits of hindsight value functions in a toy problem designed to highlight these. We consider a contextual bandit problem with feedback. At each time step, the agent receives a context $-N \leq C \leq N$ (where N is an environment parameter), and based on the context, chooses an action $-N \leq A \leq N$. The agent receives a reward $R = -(C - A)^2 + \epsilon_r$, where the exogenous noise ϵ_r is sampled from $\mathcal{N}(0, \sigma_r)$, as well as a feedback vector F which is a function of C, A and ϵ_r. More details about this problem as well as variants are presented in Appendix B.1.

[4]Note that from this point of view, actions at future time-step are effectively 'chance' from the point of view of computing credit for action A_t

For this problem, the optimal policy is to choose $A = C$, resulting in average reward of 0. However, the reward signal R is corrupted by the exogenous noise ϵ_r, uncorrelated to the action. The higher the standard deviation, the more difficult proper credit assignment becomes, as high rewards are more likely due to a high value of ϵ_r than an appropriate choice of action. On the other hand, the feedback F contains information about C, A and ϵ_r. If the agent can extract information Φ from F in order to capture information about ϵ_r and use it to compute a hindsight value function, the effect of the perturbation ϵ_r may be removed from the advantage estimate, resulting in a significantly lower variance estimator. However, if the agent blindly uses F to compute the hindsight value information, information about the action will 'leak' into the hindsight value, leading to an advantage estimate of 0 and no learning.

We investigate the proposed algorithm with $N = 10$. As can be seen on Fig. 2, increasing the variance of the exogenous noise leads to dramatic decrease of performance for the vanilla PG estimator without the hindsight baseline; in contrast, the CCA-PG estimator is generally unaffected by the exogenous noise. For very low level of exogenous noise however, CCA-PG suffers from a decrease in performance. This is due to the agent computing a hindsight statistic Φ which is not perfectly independent from A, leading to bias in the policy gradient update. To demonstrate this effect, and evaluate the importance of the independence constraint on performance, we run an ablation where we test lower values of the weight λ_{IM} of the independence maximization loss (leading to a larger mutual information between Φ and A) and indeed observed that the performance is dramatically degraded, as seen in Fig. 2.

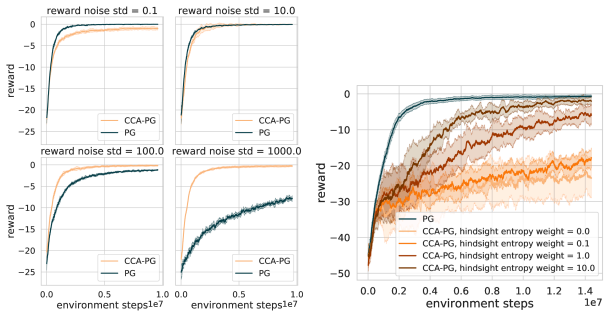


Figure 2: Top: Comparison of CCA-PG and PG in contextual bandits with feedback, for various levels of reward noise σ_r. Results are averaged over 6 independent runs with standard deviation represented by a shaded area. **Bottom:** Performance of CCA-PG on the bandit task, for different values of λ_{IM}. Properly enforcing the independence constraint prevents the degradation of performance.

4.2. Key-to-Door environments

Task Description. We investigate new versions of the Key-To-Door family of environments, initially proposed by Hung et al. (2019), as a testbed of tasks where credit assignment is hard and is necessary for success. In this partially observable grid-world environment (cf. Fig. 7 in the appendix), the agent has to pick up a key in the first room, for which it has *no immediate reward*. In the second room, the agent can pick up 10 apples, that each give immediate rewards. In the final room, the agent may open a door (only if it is carrying a key), and receive a small reward for doing so. In this task, a single action (i.e picking up the key) has a direct impact on the reward it receives in the final room, however this signal is hard to detect as the episode return is largely driven by its performance in the second room (i.e picking up apples).

We now consider two instances of the Key-To-Door family that illustrate the difficulty of credit assignment in the presence of extrinsic variance. In the Low-Variance-Key-To-Door environment, each apple is worth a reward of 1 and opening the final door also gets a reward of 1. Thus, an agent that solves the apple phase perfectly sees very little variance in its episode return and the learning signal for picking up the key and opening the door is relatively strong.

High-Variance-Key-To-Door keeps the overall structure of the Key-To-Door task. The door keeps giving a deterministic reward of 1 when the key was grabbed but now the reward for each apple is randomly sampled to be either 1 or 10, and fixed within the episode. In this setting, even an agent that is skilled at picking up apples sees a large variance in episode returns, and thus the learning signal for picking up the key and opening the door is comparatively weaker. Appendix B.2.1 has some additional discussion illustrating the difficulty of learning in such a setting.

Results We test CCA-PG on these environments, and compare it against Actor-Critic (Williams, 1992), as well as State-conditional HCA and Return-conditional HCA (Harutyunyan et al., 2019) as baselines. An analysis of the relation between HCA and CCA is described in Appendix C. We test using both a backward-LSTM (referred to as CCA-PG RNN) or an attention model (referred to as CCA-PG Attn) for the hindsight function. Details for experimental setup are provided in Appendix B.2.2.

We evaluate agents both on their ability to maximize total reward, as well as to solve the specific credit assignment problem of picking up the key and opening the door. Fig. 3 compares CCA-PG with the baselines on the High-Variance-Key-To-Door task. Both CCA-PG architectures outperform the baselines in terms of total reward, as well as probability of picking up the key and opening the door.

This experiment highlights the capacity of CCA-PG to learn and incorporate trajectory-specific external factors into its baseline, resulting in lower variance estimators. Despite being a difficult task for credit assignment, CCA-PG is capable of solving it quickly and consistently. On the other

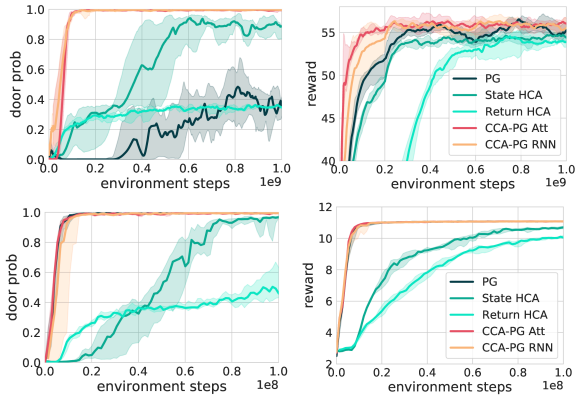


Figure 3: Probability of opening the door and total reward obtained on the **High-Variance-Key-To-Door** task (top two) and the **Low-Variance-Key-To-Door** task (bottom two).

hand, vanilla actor-critic is greatly impacted by this external variance, and needs around 3.10^9 environment steps to have an 80% probability of opening the door. CCA-PG also outperforms State- and Return- Conditional HCA, which do use hindsight information but in a more limited way than CCA-PG.

On the Low-Variance-Key-To-Door task (Fig. 3), due to the lack of extrinsic variance, standard actor-critic is able to perfectly solve the environment. However, it is interesting to note that CCA-PG still matches this perfect performance. On the other hand, the other hindsight methods struggle with both door-opening and apple-gathering. This might be explained by the fact that both these techniques do not guarantee lower variance, and rely strongly on their learned hindsight classifiers for their policy gradient estimators, which can be harmful when these quantities are not perfectly learned. See Appendix B.2.3 for additional experiments and ablations on these environments.

These experiments demonstrate that CCA-PG is capable of efficiently leveraging hindsight information to mitigate the challenge of external variance and learn strong policies that outperform baselines. At the same time, it suffers no drop in performance when used in cases where external variance is minimal.

4.3. Task Interleaving

Motivation. In the real world, human activity can be seen as solving a large number of loosely related problems. At an abstract level, one could see this lifelong learning process as solving problems not in a sequential, but an interleaved fashion instead. These problems are not solved sequentially, as one may temporarily engage with a problem and only continue engaging with it or receive feedback from its earlier actions significantly later. The structure of this interleaving will also typically vary over time.

To better understand the effects of interleaving on agent learning, we introduce a new class of environments capturing the structural properties mentioned above. In contrast to most work on multi-task learning, we do not assume a clear delineation between subtasks, nor focus on skill retention. The agent will encounter multiple tasks in a single episode in an interleaved fashion (switching between tasks will occur before a task gets completed), and will have to detect the implicitly boundaries between them.

Task Description. This task consists of pairs of query-answer rooms with different visual contexts that each indicates a different subtask. In the query room, the agent gets to pick between two colored boxes (out of 10 possible colors). Later, in the answer room, the agents gets to observe which of the two boxes was rewarding in the first room, and receives a reward if it picked the correct box (there is always exactly one rewarding color in the query room). The mapping of colors to whether it is rewarding or not is specific to each subtask and fixed across training. Each subtask would be relatively easy to solve if encountered in an isolated fashion. However, each episode is composed of *randomly sampled subtasks* and color pairs within those subtasks. Furthermore, query rooms and answer rooms of the sampled subtasks are presented in a random (interleaved) order which differs from one episode to another. Each episode are 140 steps long and it takes at least 9 steps for the agent to reach one colored square from its initial position. A visual example of what an episode looks like can be seen in Fig. 4.

There are six tasks, each classified as 'easy' or 'hard'; easy tasks have high reward signals (i.e. easier for agents to pick up on), while hard tasks have low rewards. In the 2 tasks setup (resp. 4 tasks and 6 tasks), there is one (resp. two and two) 'easy' and one (resp. two and four) 'hard' task. More details about the experimental setup can be found in B.3.

In addition to the total reward, we record the probability of picking up the correct square for the easy and hard tasks separately. Performance in the hard tasks will indicate ability to do fine-grained credit assignment.

Results. While CCA-PG is able to perfectly solve both the 'easy' and 'hard' tasks in the three setups in less than 5.10^8 environment steps (Fig. 5), actor-critic is only capable to solve the 'easy' tasks for which the associated rewards are large. Even after 2.10^9 environment steps, actor-critic is still greatly impacted by the variance and remains incapable of solving 'hard' tasks in any of the three settings. CCA-PG also outperforms actor-critic in terms of the total reward obtained in each setting. State-conditional and Return-conditional HCA were also evaluated on this task but results are not reported as almost no learning was taking place on the 'hard' tasks. More results along with an ablation study can be found in Appendix B.3.

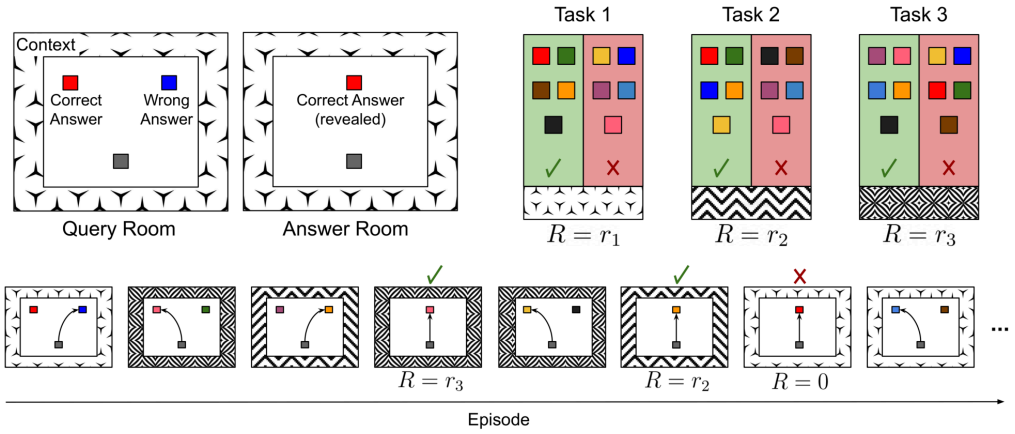


Figure 4: Task Interleaving Description. Top left: Delayed feedback contextual bandit problem. Given a context shown as a surrounding visual pattern, the agent has to decide to pick up one of the two colored squares where only one will be rewarding. The agent is later teleported to the second room where it is provided with the reward associated with its previous choice and a visual cue about which colored square it should have picked up. **Top right:** Different tasks with each a different color mapping, visual context and associated reward. **Bottom:** Example of a generated episode, composed of randomly sampled tasks and color pairs.

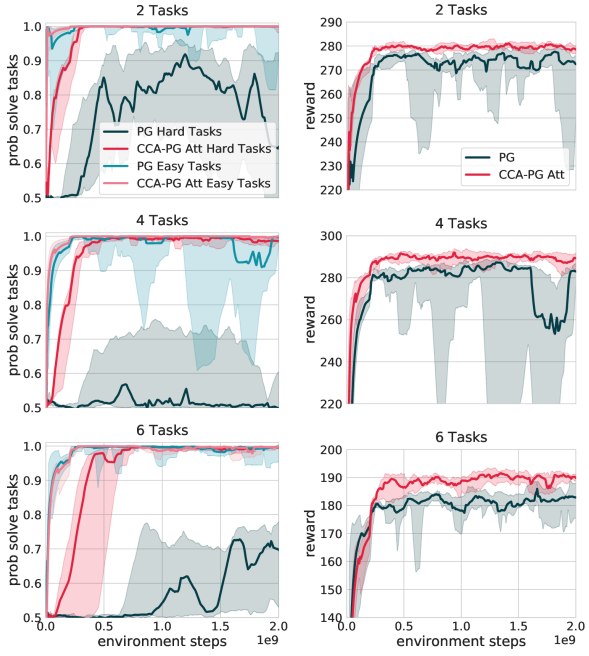


Figure 5: Probability of solving 'easy' and 'hard' tasks (left) and total reward (right) obtained for the **Multi Task Interleaving.** Left plots: Median over 10 seeds after doing a mean over the performances in 'easy' or 'hard' tasks.

Through efficient use of hindsight, CCA-PG is able to take into account trajectory-specific factors such as the kinds of rooms encountered in the episode and their associated rewards.

In the case of the Multi-Task Interleaving environment, an informative hindsight function would capture the reward for different contexts and exposes as Φ_t all rewards obtained in the episode except those associated with the current context. This experiment again highlights the capacity of CCA-PG to solve hard credit assignment problems in a context where the return is affected by multiple distractors, while PG remains highly sensitive to them.

5. Related work

This paper builds on work from Buesing et al. (2019) which shows how causal models and real data can be combined to generate counterfactual trajectories and perform off-policy evaluation for RL. Their results however require an explicit model of the environment. In contrast, our work proposes a model-free approach, and focuses on policy improvement. Oberst & Sontag (2019) also investigate counterfactuals in reinforcement learning, point out the issue of non-identifiability of the correct SCM, and suggest a sufficient condition for identifiability; we discuss this issue in appendix G. Closely related to our work is Hindsight Credit Assignment, a concurrent approach from Harutyunyan et al. (2019). In this paper, the authors also investigate value functions and critics that depend on future information. However, the information the estimators depend on is fixed (future state or return) instead of being an arbitrary functions of the trajectory. Our FC estimators generalizes both the HCA and CCA estimators while CCA further characterizes which statistics of the future provide a useful estimator. Relations between HCA, CCA and FC are discussed in appendix C. The HCA approach is further extended by Young (2019), and Zhang et al. (2019) who minimize a surrogate for the variance of the estimator, but that surrogate cannot be guaranteed to actually lower the variance. Similarly to state-HCA, it treats each reward separately instead of taking

a trajectory-centric view as CCA. Guez et al. (2019) also investigate future-conditional value functions. Similarly to us, they learn statistics of the future Φ from which returns can be accurately predicted, and show that doing so leads to learning better representations (but use regular policy gradient estimators otherwise). Instead of enforcing a information-theoretic constraint, they bottleneck information through the size of the encoding Φ. In domain adaptation (Ganin et al., 2016; Tzeng et al., 2017), robustness to the training domain can be achieved by constraining the agent representation not to be able to discriminate between source and target domains, a mechanism similar to the one constraining hindsight features not being able to discriminate the agent's actions. Also closely related to our paper, Bica et al. (2020) also leverages a similar mechanism to compute counterfactuals, for a different purpose than ours (computing treatment effects vs. policy improvement operators).

Both Andrychowicz et al. (2017) and Rauber et al. (2017) leverage the idea of using hindsight information to learn goal-conditioned policies. Hung et al. (2019) leverages attention-based systems and episode memory to perform long term credit assignment; however, their estimator will in general be biased. Ferret et al. (2019) looks at the question of transfer learning in RL and leverages transformers to derive a heuristic to perform reward shaping. Arjona-Medina et al. (2019) also addresses the problem of long-term credit assignment by redistributing delayed rewards earlier in the episode but their approach still fundamentally uses time as a proxy for credit.

Previous research also leverages the fact that baselines can include information unknown to the agent at time t (but potentially revealed in hindsight) but not affected by action A_t: for instance, when using independent multi-dimensional actions, the baseline for one dimension of the action vector can include the actions in other dimensions (Wu et al., 2018); or when the dynamic of the environment is partially driven by an exogenous and stochastic factor, independent of the agent's actions, which can be included in the baseline (Mao et al., 2018). Similarly, in multi-agent environments, actions of other agents at the same time step (Foerster et al., 2018) can be used; and so can the full state of the simulator when learning control from pixels (Andrychowicz et al., 2020), or the use of opponent observations in Starcraft II (Vinyals et al., 2019). Note however that all of these require privileged information, both in the form of feeding information to the baseline inaccessible to the agent, and in knowing that this information is independent from the agent's action A_t and therefore won't bias the baseline. Our approach seeks to replicate a similar effect, but in a more general fashion and from an agent-centric point of view, where the agent *learns itself* which information from the future can be used to improve its baseline at time t.

6. Conclusion

In this paper we have considered the problem of credit assignment in RL. Building on insights from causality theory and structural causal models, we have investigated the concept of future-conditional value functions. Contrary to common practice these allow baselines and critics to condition on future events thus separating the influence of an agent's actions on future rewards from the effects of other random events thus reducing the variance of policy gradient estimators. A key difficulty lies in the fact that unbiasedness relies on accurate estimation and minimization of mutual information. Learning inaccurate hindsight classifiers will result in miscalibrated estimation of luck, leading to bias in learning. Future research will investigate how to scale these algorithms to more complex environments, and the benefits of the more general FC-PG and all-actions estimators.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. Hindsight experience replay. In *Advances in neural information processing systems*, pp. 5048–5058, 2017.

Andrychowicz, O. M., Baker, B., Chociej, M., Jozefowicz, R., McGrew, B., Pachocki, J., Petron, A., Plappert, M., Powell, G., Ray, A., et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

Arjona-Medina, J. A., Gillhofer, M., Widrich, M., Unterthiner, T., Brandstetter, J., and Hochreiter, S. Rudder: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, pp. 13544–13555, 2019.

Bica, I., Alaa, A. M., Jordon, J., and van der Schaar, M. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. *arXiv preprint arXiv:2002.04083*, 2020.

Buesing, L., Weber, T., Zwols, Y., Racaniere, S., Guez, A., Lespiau, J.-B., and Heess, N. Woulda, coulda, shoulda: Counterfactually-guided policy search. *2019 International Conference for Learning Representations (ICLR)*, 2019.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Gated feedback recurrent neural networks. In *International conference on machine learning*, pp. 2067–2075, 2015.

Ferret, J., Marinier, R., Geist, M., and Pietquin, O. Credit assignment as a proxy for transfer in reinforcement learning. *arXiv preprint arXiv:1907.08027*, 2019.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Glasserman, P. and Yao, D. D. Some guidelines and guarantees for common random numbers. *Management Science*, 38(6):884–908, 1992.

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., and Schölkopf, B. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Guez, A., Viola, F., Weber, T., Buesing, L., Kapturowski, S., Precup, D., Silver, D., and Heess, N. Value-driven hindsight modelling. *https://openreview.net/forum?id=rJxBa1HFvS*, 2019.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Hamrick, J. B. Analogues of mental simulation and imagination in deep learning. *Current Opinion in Behavioral Sciences*, 29:8–16, 2019.

Harutyunyan, A., Dabney, W., Mesnard, T., Azar, M. G., Piot, B., Heess, N., van Hasselt, H. P., Wayne, G., Singh, S., Precup, D., et al. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, pp. 12467–12476, 2019.

Heess, N., Wayne, G., Silver, D., Lillicrap, T., Erez, T., and Tassa, Y. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, pp. 2944–2952, 2015.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

Hung, C.-C., Lillicrap, T., Abramson, J., Wu, Y., Mirza, M., Carnevale, F., Ahuja, A., and Wayne, G. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):1–12, 2019.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mao, H., Venkatakrishnan, S. B., Schwarzkopf, M., and Alizadeh, M. Variance reduction for reinforcement learning in input-driven environments. In *International Conference on Learning Representations*, 2018.

Minsky, M. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

Oberst, M. and Sontag, D. Counterfactual off-policy evaluation with gumbel-max structural causal models. *arXiv preprint arXiv:1905.05824*, 2019.

Parisotto, E., Song, H. F., Rae, J. W., Pascanu, R., Gulcehre, C., Jayakumar, S. M., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. *arXiv preprint arXiv:1910.06764*, 2019.

Pearl, J. *Causality*. Cambridge university press, 2009a.

Pearl, J. Causality: Models, reasoning, and inference. 2009b.

Rauber, P., Ummadisingu, A., Mutz, F., and Schmidhuber, J. Hindsight policy gradients. *arXiv preprint arXiv:1711.06006*, 2017.

Rezende, D. J. and Viola, F. Taming VAEs. *arXiv preprint arXiv:1810.00597*, 2018.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *arXiv preprint arXiv:1911.08265*, 2019.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7167–7176, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Weber, T., Heess, N., Buesing, L., and Silver, D. Credit assignment techniques in stochastic computation graphs. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2650–2660, 2019.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. *2018 International Conference for Learning Representations (ICLR)*, 2018.

Young, K. Variance reduced advantage estimation with δ-hindsight credit assignment. *arXiv preprint arXiv:1911.08362*, 2019.

Zhang, J. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In *International Conference on Machine Learning*, pp. 11012–11022. PMLR, 2020.

Zhang, P., Zhao, L., Liu, G., Bian, J., Huang, M., Qin, T., and Tie-Yan, L. Independence-aware advantage estimation. *https://openreview.net/forum?id=B1eP504YDr*, 2019.

Appendix

A. Algorithmic and implementation details

A.1. Constrained optimization

Corollary 1 requires an independence assumption between A_t and Φ_t, conditional on X_t. We can therefore cast the problem of learning Φ_t as a constrained optimization problem, where the loss $\mathcal{L}_{\text{hs}}$ measures how predictive of the return Φ_t is, and the constraint enforces a maximum (tolerance) value of β_{IM} for the independence maximization loss $\mathcal{L}_{\text{IM}}$ (exact independence is obtained by $\beta_{\text{IM}} = 0$, but this is hard to achieve exactly in practice).

The resulting optimization problem for finding an appropriate counterfactual baseline is given by:

$$\min_{\theta} \mathbb{E}\left[\mathcal{L}_{\text{hs}}\right] \quad \text{subject to: } \forall t \;\; \mathcal{L}_{\text{IM}}(X_t) \leq \beta_{\text{IM}} \tag{7}$$

The resulting hindsight baseline can then be used in the policy gradient estimate. There are two problems remaining to solve. First, the form of the (IM) loss used requires knowing the exact hindsight probability $\mathbb{P}(A_t|X_t, \Phi_t)$. As explained in the main text, we replace it by the classifier h, tracking the optimal classifier by stochastically minimizing the supervised loss (optimizing it only with respect to the parameters of the hindsight classifier). Second, we relax the constraint using a Lagrangian method (the Lagrangian parameter can either be set as a hyperparameter, or optimized using an algorithm like GECO (Rezende & Viola, 2018)).

A.2. Parameter updates

The corresponding parameter updates are as follows:

For each trajectory $(X_t, A_t, R_t)_{t\geq 0}$, compute the parameter updates :

- $\Delta\theta_{\text{fs}} = -\lambda_{\text{PG}} \sum_t \gamma^t \nabla_{\theta_{\text{fs}}} \log \pi(A_t|X_t)(G_t - V(X_t, \Phi_t)) + \lambda_{\text{H}} \sum_t \nabla_{\theta_{\text{fs}}} \mathcal{L}_{\text{H}}(t) + \lambda_{\text{hs}} \sum_t \nabla_{\theta_{\text{fs}}} \mathcal{L}_{\text{hs}}(t)$
  where $\mathcal{L}_{\text{H}}(t) = -\sum_a \pi(a|X_t) \log \pi(a|X_t)$ is an entropy bonus.
- $\Delta\theta_{\text{hs}} = \lambda_{\text{hs}}(t) \sum_t \nabla_{\theta_{\text{hs}}} \mathcal{L}_{\text{hs}} + \lambda_{\text{IM}} \sum_t \nabla_{\theta_{\text{hs}}} (\mathcal{L}_{\text{IM}}(t) - \beta \mathbb{H}[A_t|X_t])$
- $\Delta\omega = \sum_t \nabla_{\omega} \mathcal{L}_{\text{sup}}(t)$
- $\Delta\lambda_{\text{IM}} = -\lambda_{\text{IM}} \sum_t (\mathcal{L}_{\text{IM}}(t) - \beta \mathbb{H}[A_t|X_t])$ (when using GECO)

A.3. Design choices

Here we detail practical choices for two aspects of the general CCA algorithm. These concern a) the form of the hindsight function, b) the form of the independence maximization constraint.

CHOICE OF THE HINDSIGHT FUNCTION φ

In principle, this function can take any form: in practice, we investigated two architectures. The first is a backward RNN, where $(\Phi_t, B_t) = \text{RNN}(X_t, B_{t+1})$, where B_t is the state of the backward RNN. Backward RNNs are justified in that they can extract information from arbitrary length sequences, and allow making the statistics Φ_t a function of the entire trajectory. They also have the inductive bias of focusing more on near-future observations. The second is a transformer (Vaswani et al., 2017; Parisotto et al., 2019). Alternative networks could be used, such as attention-based networks (Hung et al., 2019) or RIMs (Goyal et al., 2019).

INDEPENDENCE MAXIMIZATION CONSTRAINT $\mathcal{L}_{\text{IM}}$

We investigated two IM losses. The first is the conditional mutual information $\mathbb{I}(A_t; \Phi_t|X_t) = \mathbb{E}_{\Phi_t|X_t}[\mathbb{H}[A_t|X_t] - \mathbb{H}[A_t|X_t, \Phi_t]]$, where $\mathbb{H}[A|B]$ denotes the conditional entropy $\mathbb{H}[A|B] = -\sum_a P(A = a|B) \log P(A = a|B)$. The expectation can be stochastically approximated by the trajectory sample value $\mathbb{H}(A_t|X_t) - \mathbb{H}(A_t|X_t, \Phi_t)$. The first term is simply the entropy of the policy $-\sum_a \pi(a|X_t) \log \pi(a|X_t)$. The second term is estimated using the h network. The

second we investigated is the Kullback-Leibler divergence, $\mathbb{KL}(\pi(A_t|X_t)||\mathbb{P}(A_t|X_t,\Phi_t)) = \sum_a \pi(a|X_t)\log\pi(a|X_t) - \sum_a \pi(a|X_t)\log\mathbb{P}(a|X_t,\Phi_t)$. Again, we approximate the second term using h. We did not see significant differences between the two, with the KL slightly outperforming the mutual information.

B. Additional Experimental Details

B.1. Bandits

B.1.1. Environment

Our bandit with feedback environment is defined by two positive integers (N, K), a noise level $\sigma_r > 0$ and three arbitrary matrices U, V, W, where $U, V \in \mathbb{R}^{K\times N}$ and $W \in \mathbb{R}^K$. For each replication of the experiment (i.e. each seed), these matrices are sampled from a standard Gaussian distribution and kept constant throughout all episodes. For each episode (of length 1, since this is a bandit problem tackled without meta-learning), we sample a context $-N \leq C \leq N$. Given C, an agent chooses an action $-N \leq A \leq N$. The agent then receives a reward $R = -(C-A)^2 + \epsilon_r$, where ϵ_r is sampled from $\mathcal{N}(0, \sigma_r)$. The agent additionally receives a K-dimensional feedback vector $F = U_C + V_A + W\epsilon_r$, where U_C (resp. V_A) denotes the C$^{\text{th}}$ (resp. A$^{\text{th}}$) column of U (resp. V).

The choices above were made without any particular intent: we would expect the intuitions to generalize for other noise distributions and feedback functions. In section B.1.3, we investigate a decentralized multiagent variant of this problem where the exogenous noise actually corresponds to other players' actions.

B.1.2. Architecture

For the bandit problems, the agent architecture is as follows:

- The hindsight feature Φ is computed by a backward RNN. We tried multiple cores for the RNN: GRU (Chung et al., 2015) with 32 hidden units, a recurrent adder ($h_t = h_{t-1} + \text{MLP}(x_t)$, where the MLP has two layers of 32 units), or an exponential averager ($h_t = \lambda h_{t-1} + (1-\lambda)\text{MLP}(x_t)$).
- The hindsight classifier h_ω is a simple MLP with two hidden layers with 32 units each.
- The policy and value functions are computed as the output of a simple linear layer with concatenated observation and feedback as input.
- All weights are jointly trained with Adam (Kingma & Ba, 2014).
- Hyperparameters are chosen as follows: learning rate $4e$-4, entropy loss $4e$-3, independence maximization tolerance $\beta_{\text{IM}} = 0.1$, $\lambda_{\text{sup}} = \lambda_{\text{hs}} = 1$, λ_{IM} is set through Lagrangian optimization (with GECO).

B.1.3. Additional results

Multi-agent Bandit Problem: In the multi-agent version, the environment is composed of M replicas of the bandit with feedback task. Each agent $i = 1, \ldots, M$ interacts with its own version of the environment, but feedbacks and rewards are coupled across agents. The multi-agent bandit is obtained by modifying the single agent version as follows:

- The contexts C^i are sampled i.i.d. from $\{-N, \ldots, N\}$. C and A now denote the concatenation of all agents' contexts and actions.
- The feedback tensor is (M, K) dimensional, and is computed as $W_c\mathbf{1}(C) + W_a\mathbf{1}(A) + \epsilon_f$; where the W are now three dimensional tensors. Effectively, the feedback for agent i depends on the context and actions of all other agents.
- The terminal joint reward is $\sum_i -(C^i - A^i)^2$ for all agents.

The multi-agent version does not require the exogenous noise ϵ_r, as other agents play the role of exogenous noise; it is a minimal implementation of the example found in section 2.3.

We report results from the multi-agent version of the environment in Fig. 6. As the number of interacting agents increases, the effective variance of the vanilla PG estimator increases as well, and the performance of each agent decreases. In contrast, CCA-PG agents learn faster and reach higher performance (though they never learn the optimal policy).

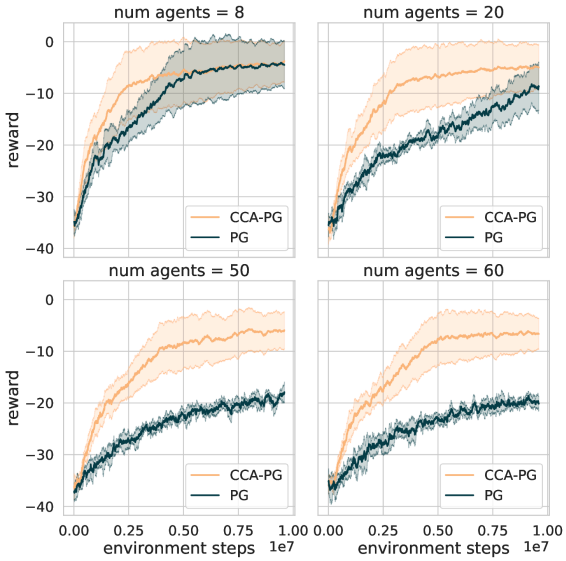


Figure 6: Multi-agent versions of the bandit problem. CCA-PG agents outperform vanilla PG ones.

B.2. Key to Door Tasks

B.2.1. Environment details

Observations returned by the key-to-door family of environments for each of the three phases can be visualized in Fig. 7.

Figure 7: Key-To-Door environments visual. The agent is represented by the beige pixel, key by brown, apples by green, and the final door by blue. The agent has a partial field of view, highlighted in white.

		Lucky (high apple reward)	Unlucky (low apple reward)
Hindsight Advantage Estimate	Skillful (Got key + Door)	1	1
	Unskillful (Did not get key or door)	0	0
Forward Advantage Estimate	Skillful (Got key + Door)	46	-44
	Unskillful (Did not get key or door)	45	-45

Table 1: The advantage estimate of the action of picking up a key in High-Variance-Key-To-Door, as computed by an agent that always picks up every apple, and never picks up the key or the door. We see that an advantage estimate learned using hindsight clearly differentiates between the skillful and unskillful actions; whereas for an advantage estimate learned without using hindsight, this difference is dominated by the extrinsic randomness.

To motivate our approach, Table 1 shows the advantage estimates for either picking up the key or not on High-Variance-Key-To-Door, for an agent that has a perfect apple-phase policy, but never picks up the key or door. Since there are 10 apples which can be worth 1 or 10, the return will be either 10 or 100. Thus the forward baseline in the key phase, i.e. before it has seen how much an apple is worth in the current episode, will be 55. As seen in Table 1, the difference in advantage estimates due to 'luck' is far larger than the difference in advantage estimates due to 'skill' when not using hindsight. This makes learning difficult and leads to the policy never learning to start picking up the key or opening the door. However, when we use a hindsight-conditioned baseline, we are able to learn a Φ (such as the value of a single apple in the current episode) that is completely independent from the actions taken by the agent, but which can provide a perfect hindsight-conditioned baseline of either 10 or 100.

B.2.2. Architecture

The agent architecture is as follows:

- The observations are first fed to 2-layer CNN with $(16, 32)$ output channels, kernel shapes of $(3, 3)$ and strides of $(1, 1)$. The output of the CNN is flattened and fed to a linear layer of size 128.
- The agent state is computed by a forward LSTM with a state size of 128. The input to the LSTM is the output of the previous linear layer, concatenated with the reward at the previous timestep.
- The hindsight feature Φ is computed either by a backward LSTM (i.e CCA-PG RNN) with a state size of 128 or by an attention mechanism (Vaswani et al., 2017) (i.e CCA-PG Att) with value and key sizes of 64, 1 transformer block with 2 attention heads, a 1 hidden layer MLP of size 1024, an output size of 128 and a rate of dropout of 0.1. The input provided is the concatenation of the output of the forward LSTM and the reward at the previous timestep.
- The policy is computed as the output of a single-layer MLP with 64 units where the output of the forward LSTM is provided as input.
- The forward baseline is computed as the output of a 3-layer MLP of 128 units each where the output of the forward LSTM is provided as input.
- The hindsight baseline is computed as the sum of the forward baseline and a hindsight residual baseline; the hindsight residual baseline is the output of a 3-layer MLP of 128 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input. It is trained to learn the residual between the return and the forward baseline.
- For CCA, the hindsight classifier h_ω is computed as the concatenation of the output of an MLP, with four hidden layers with 256 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input, and the log of the policy outputs.
- For State HCA, the hindsight classifier h_ω is computed as the output of an MLP, with four hidden layers with 256 units each, where the concatenation of the outputs of the forward LSTM at two given time steps is provided as input.
- For Return HCA, the hindsight classifier h_ω is computed as the output of an MLP, with four hidden layers with 256 units each, where the concatenation of the output of the forward LSTM and the return is provided as input.
- All weights are jointly trained with RMSprop (Hinton et al., 2012) with epsilon $1e\text{-}4$, momentum 0 and decay 0.99.

For High-Variance-Key-To-Door, the optimal hyperparameters found for each algorithm can be found in Table 2.

For Key-To-Door, the optimal hyperparameters found for each algorithm can be found in Table 3.

The agents are trained on full-episode trajectories, using a discount factor of 0.99.

B.2.3. Additional results

As shown in Fig. 8, in the case of vanilla policy gradient, the baseline loss increases at first. As the reward associated with apples varies from one episode to another, getting more apples also means increasing the forward baseline loss. On the

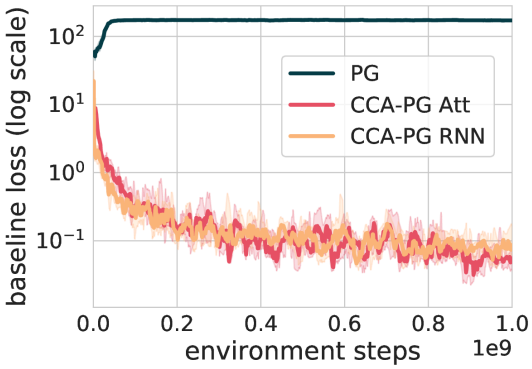


Figure 8: Baseline loss for vanilla PG versus hindsight baseline loss for CCA in **High-Variance-Key-To-Door**.

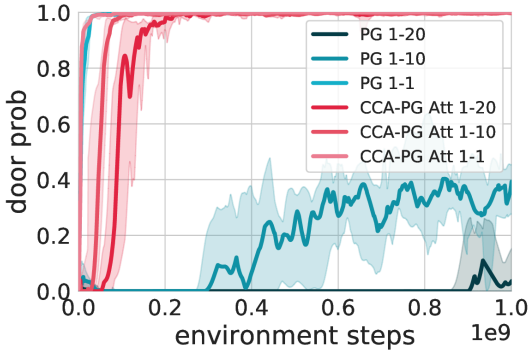


Figure 9: Impact of variance over credit assignment performances. Probability of picking up the key and opening the door as a function of the variance level induced by the apple reward discrepancy between episodes.

other hand, as CCA is able to take into account trajectory specific exogenous factors, the hindsight baseline loss can nicely decrease as learning takes place.

Fig. 9 shows the impact of the variance level induced by the apple reward discrepancy between episodes on the probability of picking up the key and opening the door. Thanks to the use of hindsight in its value function, CCA-PG is almost not impacted by this whereas vanilla PG sees its performances drop dramatically as variance increases.

Fig. 10 shows a qualitative analysis of the attention weights learned by CCA-PG Att on the High-Variance-Key-To-Door task. For this experiment, we used only a single attention head for easier interpretation of the hindsight function, and show both a heatmap of the attention weights over the entire episode, and a histogram of attention weights at the step where the agent picks up the key. As expected, the most attention is paid to timesteps just after the agent picks up an apple - since these are the points at which the apple reward is provided to the Φ computation. In particular, very little attention is paid to the timestep where the agent opens the door. These insights further show that the hindsight function learned is highly predictive of the episode return, while not having mutual information with the action taken by the agent, thus ensuring an unbiased policy gradient estimator.

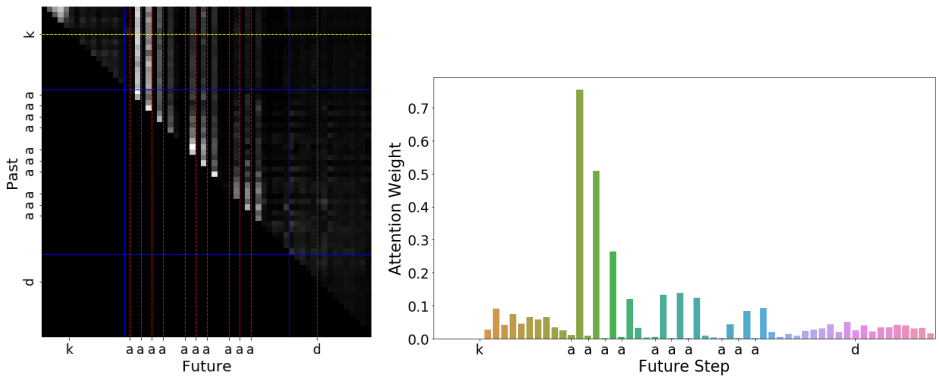


Figure 10: Visualization of attention weights on the High-Variance-Key-To-Door task. **Top:** a 2-dimensional heatmap showing how the hindsight function at each step attends to each step in the future. Red lines indicate the timesteps at which apples are picked up (marked as 'a'); green indicates the door (marked as 'd'); yellow indicates the key (marked as 'k'). **Bottom:** A bar plot of attention over future timesteps, computed at the step where the agent is just about to pick up the key.

	CCA Att	CCA RNN	PG	State HCA	Return HCA
Policy cost	1	1	1	1	1
Entropy cost	5e-3	5e-3	5e-3	5e-3	5e-3
Forward baseline cost	5e-2	5e-2	5e-2	5e-2	5e-2
Hindsight residual baseline cost	5e-2	5e-2	—	———	———
Hindsight classifier cost	1e-2	1e-2	—	1e-2	1e-2
Action independence cost	1e2	1e2	—	———	———
Learning rate	5e-4	5e-4	1e-4	5e-4	1e-3

Table 2: High-Variance-Key-To-Door hyperparameters

	CCA Att	CCA RNN	PG	State HCA	Return HCA
Policy cost	1	1	1	1	1
Entropy cost	5e-3	5e-3	5e-3	5e-3	5e-3
Forward baseline cost	5e-2	5e-2	5e-2	5e-2	5e-2
Hindsight residual baseline cost	5e-2	5e-2	—	———	———
Hindsight classifier cost	1e-2	1e-2	—	1e-2	1e-2
Action independence cost	1e2	1e2	—	———	———
Learning rate	5e-4	5e-4	5e-4	5e-4	5e-4

Table 3: Key-To-Door hyperparameters

B.3. Task Interleaving

B.3.1. Environment details

For each task, a random, but fixed through training, set of 5 out of 10 colored squares are leading to a positive reward. Furthermore, a small reward of 0.5 is provided to the agent when it picks up any colored square. As mentioned previously, each episode are 140 steps long and it takes at least 9 steps for the agent to reach one colored square from its initial position.

The 6 tasks we consider (numbered #1 to #6) are respectively associated with a reward of 80, 4, 100, 6, 2 and 10. Tasks #2, #4, #5 and #6 are referred to as 'hard' while tasks #1 and #3 as 'easy' because of their large associated rewards. The settings 2, 4 and 6-task are respectively considering tasks 1-2, 1-4 and 1-6.

B.3.2. Architecture

We use the same architecture setup as reported in Appendix B.2.2. The agents are also trained on full-episode trajectories, using a discount factor of 0.99.

For Task Interleaving, the optimal hyperparameters found for each algorithm can be found in Table 4.

B.3.3. Additional results

Fig. 11 shows that CCA is able to solve all 6 tasks quickly despite the variance induced by the exogenous factors. Vanilla PG on the other hand despite solving the 'easy' tasks 1 and 3 for which the agent receives big rewards, it is incapable of reliably solve the 4 remaining tasks for which the associated reward is smaller. This helps unpacking Fig. 5.

B.3.4. Ablation Study

Fig.12 shows the impact of the number of back-propagation through time steps performed into the backward RNN of the hindsight function while performing full rollouts. This shows that learning in 'hard' tasks, i.e. where hindsight is crucial for performances, is not much impacted by the number of back-propagation steps performed into the backward RNN. This is great news as this indicates that learning in challenging credit assignment tasks still works when the hindsight function sees the whole future but can only backprop through a limited window.

Fig.13 shows how performances of CCA-RNN are impacted by the unroll length. As expected, the less it is able to look into the future, the harder it becomes to solve hard credit assignment tasks as it is limited in its capacity to take into account

exogenous effects.

The two previous results are promising since CCA seems to only require to have access to as many steps into the future as possible while not needing to do back-propagation through the full sequence.

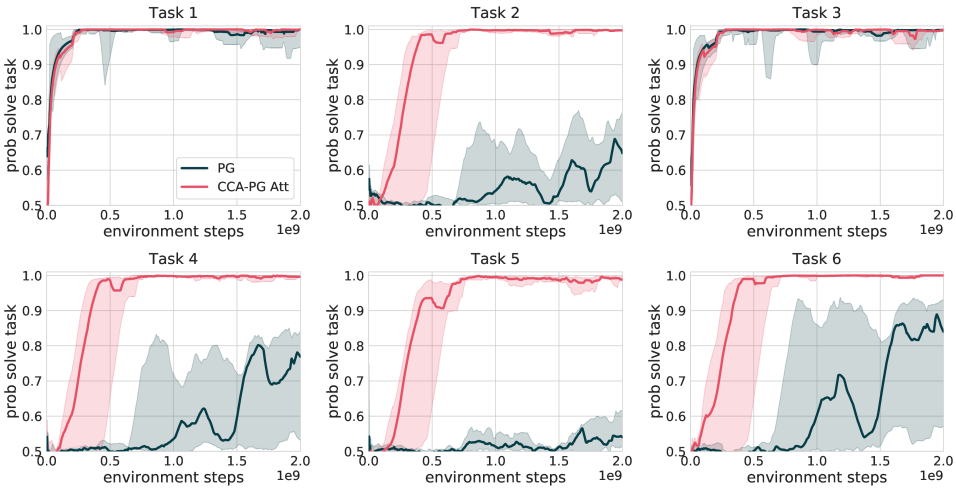


Figure 11: Probability of solving each task in the 6-task setup for **Task Interleaving**.

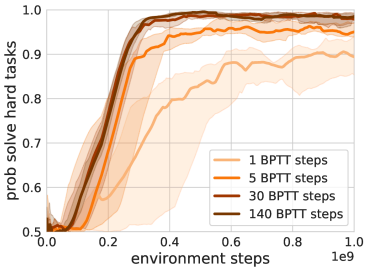


Figure 12: Impact of the number of back-propagation through time steps performed into the hindsight function for CCA-RNN. Probability of solving the 'hard' tasks in the 6-task setup of **Task Interleaving**.

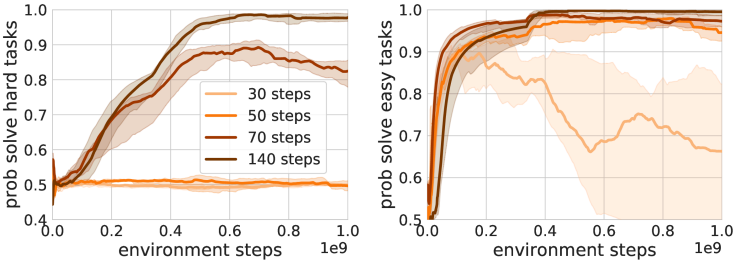


Figure 13: Impact of the unroll length for CCA-RNN. Probability of solving the 'hard' and 'easy' tasks in the 6-task setup of **Task Interleaving**.

	CCA Att	CCA RNN	PG
Policy cost	1	1	1
Entropy cost	5e-2	5e-2	5e-2
Forward baseline cost	1e-2	5e-3	5e-2
Hindsight residual baseline cost	1e-2	5e-3	—
Hindsight classifier cost	1e-2	1e-2	—
Action independence cost	1e1	1e1	—
Learning rate	5e-4	5e-4	1e-3

Table 4: Task Interleaving hyperparameters

C. Relation between HCA, CCA, and FC estimators

The FC estimators generalize both the HCA and CCA estimators. From FC, we can derive CCA by assuming that Φ_t and A_t are conditionally independent (see next section). We can also derive state and return HCA from FC.

For return HCA, we obtain both an all-action and baseline version of return HCA by choosing $\Phi_t = G_t$. For state HCA, we first need to decompose the return into a sum of rewards, and apply the policy gradient estimator to each reward separately. For a pair (X_t, R_{t+k}), and assuming that R_{t+k} is a function of X_{t+k} for simplicity, we choose $\Phi_t = X_{t+k}$. We then sum the different FC estimators for different values of k and obtain both an all-action and single-action version of state HCA.

Note however that HCA and CCA *cannot* be derived from one another. Both estimators leverage different approaches for unbiasedness, one (HCA) leveraging importance sampling, and the other (CCA) eschewing importance sampling in favor of constraint satisfaction (in the context of inference, this is similar to the difference between obtaining samples of the posterior by importance sampling versus directly parametrizing the posterior distribution).

D. Proofs

D.1. Policy gradients

Proof of equation 1. By linearity of expectation, the expected return can be written as $\mathbb{E}[G] = \sum_t \gamma^t \mathbb{E}[R_t]$. Writing the expectation as an integral over trajectories, we have:

$$\mathbb{E}[R_t] = \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\prod_{s \leq t} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) R(x_t, a_t)$$

Taking the gradient with respect to θ:

$$\nabla_\theta \mathbb{E}[R_t] \quad = \quad \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\sum_{s' \leq t} \nabla_\theta \pi_\theta(a_{s'}|x_{s'}) P(x_{s'+1}|x_{s'}, a_{s'}) \left(\prod_{s \leq t, s \neq s'} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) \right) R(x_t, a_t)$$

We then rewrite $\nabla_\theta \pi_\theta(a_{s'}|x_{s'}) = \nabla_\theta \log \pi_\theta(a_{s'}|x_{s'}) \pi_\theta(a_{s'}|x_{s'})$, and obtain

$$\begin{aligned} \nabla_\theta \mathbb{E}[R_t] =& \sum_{\substack{x_0,\dots,x_t \\ a_0,\dots,a_t}} \left(\sum_{s' \leq t} \nabla_\theta \pi_\theta(a_{s'}|x_{s'}) \left(\prod_{s \leq t, s} (\pi_\theta(a_s|x_s) P(x_{s+1}|x_s, a_s)) \right) \right) R(x_t, a_t) \\ =& \mathbb{E}\left[\sum_{s' \leq t} \nabla_\theta \log \pi_\theta(A_{s'}|X_{s'}) R_t \right] \end{aligned}$$

Summing over t, we obtain

$$\nabla_\theta \mathbb{E}[G] = \mathbb{E}\left[\sum_{t \geq 0} \gamma^t \sum_{s' \leq t} \nabla_\theta \log \pi_\theta(A_{s'}|X_{s'}) R_t \right]$$

which can be rewritten (with a change of variables):

$$\begin{aligned} \nabla_\theta \mathbb{E}[G] =& \mathbb{E}\left[\sum_{t \geq 0} \nabla_\theta \log \pi_\theta(A_t|X_t) \sum_{t' \geq t} \gamma^{t'} R_{t'} \right] \\ =& \mathbb{E}\left[\sum_{t \geq 0} \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \sum_{t' \geq t} \gamma^{t'-t} R_{t'} \right] \\ =& \mathbb{E}\left[\sum_{t \geq 0} \gamma^t S_t G_t \right] \end{aligned}$$

To complete the proof, we need to show that $\mathbb{E}[S_t V(X_t)] = 0$. By iterated expectation, $\mathbb{E}[S_t V(X_t)] = \mathbb{E}[\mathbb{E}[S_t V(X_t)|X_t]] = \mathbb{E}[V(X_t)\mathbb{E}[S_t|X_t]]$, and we have $\mathbb{E}[S_t|X_t] = \sum_a \nabla_\theta \pi(a|X_t) = \nabla_\theta(\sum_a \pi(a|X_t)) = \nabla_\theta 1 = 0$. □

Proof of equation 2. We start from the single action policy gradient $\nabla_\theta \mathbb{E}[G] = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t S_t G_t\right]$ and analyse the term for time t, $\mathbb{E}[S_t G_t]$.

$$\begin{aligned}
\mathbb{E}[S_t G_t] =& \mathbb{E}[\mathbb{E}[S_t G_t|X_t, A_t]] \\
=& \mathbb{E}[S_t \mathbb{E}[G_t|X_t, A_t]] \\
=& \mathbb{E}[S_t Q(X_t, A_t)] \\
=& \mathbb{E}\left[\mathbb{E}[S_t Q(X_t, A_t)|X_t]\right] \\
=& \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) Q(X_t, a)\right]
\end{aligned}$$

The first and fourth inequality come from different applications of iterated expectations, the second from the fact S_t is a constant conditional on X_t, A_t, and the third from the definition of $Q(X_t, A_t)$. □

D.2. Proof of FC-PG theorem

Proof of theorem 1 (single action). We need to show that $\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\right] = 0$, so that $\frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)$ is a valid baseline. As previously, we proceed with the law of iterated expectations, by conditioning successively on X_t then Φ_t

$$\begin{aligned}
\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\right] =& \mathbb{E}\left[\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)} V(X_t, \Phi_t)\middle| X_t, \Phi_t\right]\right] \\
=& \mathbb{E}\left[V(X_t, \Phi_t)\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)}\middle| X_t, \Phi_t\right]\right]
\end{aligned}$$

Then we note that

$$\begin{aligned}
\mathbb{E}\left[S_t \frac{\pi(A_t|X_t)}{\mathbb{P}(A_t|X_t,\Phi_t)}\middle| X_t, \Phi_t\right] &= \sum_a \mathbb{P}(a|X_t, \Phi_t) \nabla \log \pi(a|X_t) \frac{\pi(a|X_t)}{\mathbb{P}(a|X_t,\Phi_t)} \\
&= \sum_a \nabla \pi(a|X_t) = 0.
\end{aligned}$$

□

Proof of theorem 1 (all-action). We start from the definition of the Q function:

$$\begin{aligned}
Q(X_t, a) &= \mathbb{E}\left[G_t|X_t, A_t = a\right] \\
&= \mathbb{E}_{\Phi_t}\left[\mathbb{E}\left[G_t|X_t, \Phi_t, A_t = a\right]|X_t, A_t = a\right] \\
&= \int_\phi \mathbb{P}(\Phi = \varphi|X_t, A_t = a) Q(X_t, \Phi_t = \varphi, a)
\end{aligned}$$

We also have

$$\mathbb{P}(\Phi = \varphi|X_t, A_t) = \frac{\mathbb{P}(\Phi = \varphi|X_t)\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\mathbb{P}(A_t = a|X_t)},$$

which combined with the above, results in:

$$\begin{aligned}
Q(X_t, a) =& \int_\phi \mathbb{P}(\Phi = \varphi|X_t) \frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a) \\
=& \mathbb{E}\left[\frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]
\end{aligned}$$

For the compatibility with policy gradient, we start from:

$$\mathbb{E}[S_t G_t] = \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) Q(X_t, a)\right]$$

We replace $Q(X_t, a)$ by the expression above and obtain

$$\begin{aligned}
\mathbb{E}[S_t G_t] =& \mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) \mathbb{E}\left[\frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[\sum_a \nabla_\theta \pi_\theta(a|X_t) \frac{\mathbb{P}(A_t = a|X_t, \Phi_t = \phi)}{\pi_\theta(a|X_t)} Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\mathbb{E}\left[\sum_a \nabla_\theta \log \pi_\theta(a|X_t) \mathbb{P}(A_t = a|X_t, \Phi_t = \phi) Q(X_t, \Phi_t, a)\middle| X_t\right]\right] \\
=& \mathbb{E}\left[\sum_a \nabla_\theta \log \pi_\theta(a|X_t) \mathbb{P}(A_t = a|X_t, \Phi_t = \phi) Q(X_t, \Phi_t, a)\right]
\end{aligned}$$

Note that in the case of a large number of actions, the above can be estimated by

$$\frac{\nabla_\theta \log \pi_\theta(A'_t|X_t) \mathbb{P}(A'_t|X_t, \Phi_t = \phi)}{\pi_\theta(A'_t|X_t)} Q(X_t, \Phi_t, A'_t),$$

where A'_t is an independent sample from $\pi(.|X_t)$; note in particular that A'_t shall NOT be the action A_t that gave rise to Φ_t, which would result in a biased estimator.

D.3. Proof of CCA-PG theorem

Assume that Φ_t and A_t are conditionally independent on X_t. Then, $\frac{\mathbb{P}(A_t=a|X_t,\Phi_t=\phi)}{\mathbb{P}(A_t=a|X_t)} = 1$. In particular, it is true when evaluating at the random value A_t. From this simple observation, both CCA-PG theorems follow from the FC-PG theorems.

To prove the lower variance of the hindsight advantage estimate, note that

$$\begin{aligned}
\mathbb{V}[G_t - V(X_t, \Phi)] =& \mathbb{E}[(G_t - V(X_t, \Phi_t))^2] \\
=& \mathbb{E}[G_t^2] - \mathbb{E}[V(X_t, \Phi_t)^2] \\
\mathbb{V}[G_t - V(X_t)] =& \mathbb{E}[(G_t - V(X_t))^2] \\
=& \mathbb{E}[G_t^2] - \mathbb{E}[V(X_t)^2]
\end{aligned}$$

To prove the first statement, we have $(G_t - V(X_t, \Phi_t))^2 = G_t^2 + V(X_t, \Phi_t)^2 - 2G_t V(X_t, \Phi_t)$, and apply the law of iterated expectations to the last term:

$$\begin{aligned}
\mathbb{E}[G_t V(X_t, \Phi_t)] =& \mathbb{E}[\mathbb{E}[G_t V(X_t, \Phi_t)|X_t, \Phi_t]] \\
=& \mathbb{E}[V(X_t, \Phi_t)\mathbb{E}[G_t|X_t, \Phi_t]] \\
=& \mathbb{E}[V(X_t, \Phi_t)^2]
\end{aligned}$$

The proof for the second statement is identical. Finally, we note that by Jensen's inequality, we have $\mathbb{E}[V(X_t, \Phi_t)^2] \leq \mathbb{E}[V(X_t)^2]$, from which we conclude that $\mathbb{V}[G_t - V(X_t, \Phi_t)] \leq \mathbb{V}[G_t - V(X_t)]$.

□

E. Variance analysis

E.1. Relation between variance of advantage and variance of policy gradient

Consider an advantage estimate Y_t, i.e. a variable such that $\mathbb{E}[Y_t|X_t = x, A_t = a] = Q(x, a) - V(x)$. Possible choices for Y_t include the CCA estimate $G_t - V(X_t, \Phi_t)$ as well as the actual advantage $\mathcal{A}(x, a) = Q(x, a) - V(x)$. Note that

$\nabla_\theta \mathbb{E}[G_t] = \mathbb{E}[\sum_t \gamma^t S_t Y_t]$. We aim to analyze the variance of a single term $S_t Y_t$ (understanding the variance of the sum is more involved). More precisely, we compare the variance $\mathbb{V}[S_t Y_t | X_t]$ of the policy gradient term $S_t Y_t$ given X_t when using Y_t to that of $S_t \mathcal{A}_t$.

We use the conditional variance formula:

$$\mathbb{V}[S_t Y_t | X_t] = \mathbb{E}[\mathbb{V}[S_t Y_t | X_t, A_t] | X_t] + \mathbb{V}[\mathbb{E}[S_t Y_t | X_t, A_t] | X_t],$$

where S_t is constant given X_t, A_t. Therefore the first term becomes $\mathbb{E}[S_t^2 \mathbb{V}[Y_t | X_t, A_t] | X_t]$, and the second one $\mathbb{V}[S_t \mathbb{E}[Y_t | X_t, A_t] | X_t] = \mathbb{V}[S_t \mathcal{A}_t | X_t]$; this term does not depend on the actual advantage estimate used - it is equal to the variance of the policy gradient estimate when using the exact advantage $\mathcal{A}_t$. The additional variance incurred by using an unbiased advantage estimate Y_t instead of the exact advantage $\mathcal{A}_t$ is therefore:

$$\mathbb{V}[S_t Y_t | X_t] - \mathbb{V}[S_t \mathcal{A}_t | X_t] = \mathbb{E}[S_t^2 \mathbb{V}[Y_t | X_t, A_t] | X_t].$$

We see that the (conditional) advantage variance $\mathbb{V}[Y_t | X_t, A_t]$ (as well as the variance of the score function, and their correlation) drives the variance of the policy gradient estimator. We can further find a loose upper bound purely in terms of the unconditional variance of the advantage. First, suppose that the actions are discrete, and that the action distribution is parametrized by a softmax over logits $l_1, \ldots, l_k$, where k is the number of actions. Note that the score is $S_t = \frac{\partial \log \pi(A_t)}{\partial \theta} = \sum_{a'} \frac{\partial \log \pi(A_t)}{\partial l_{a'}} \frac{\partial l_{a'}}{\partial \theta}$, so

$$\begin{aligned}
|S_t| =& |\sum_{a'} \frac{\partial \log \pi(A_t)}{\partial l_{a'}} \frac{\partial l_{a'}}{\partial \theta}| \\
\leq& \sum_{a'} |\frac{\partial \log \pi(A_t)}{\partial l_{a'}}| |\frac{\partial l_{a'}}{\partial \theta}| \\
\leq& \sum_{a'} |\frac{\partial l_{a'}}{\partial \theta}| \leq ||J||_1
\end{aligned}$$

where J is the jacobian of the function mapping parameters θ to logits. The second inequality is due to $|\frac{\partial \log \pi(a)}{\partial l_{a'}}| = |\delta_{a,a'} - \pi(a')| \leq 1$. It follows that:

$$\begin{aligned}
\mathbb{V}[S_t Y_t | X_t] - \mathbb{V}[S_t \mathcal{A}_t | X_t] \leq& ||J||_1^2 \, \mathbb{E}[\mathbb{V}[Y_t | X_t, \mathcal{A}_t] | X_t] \\
\leq& ||J||_1^2 \, (\mathbb{V}[Y_t | X_t] - \mathbb{V}[\mathcal{A}_t | X_t]))
\end{aligned}$$

again using the law of conditional variance $\mathbb{V}[Y_t | X_t] = \mathbb{V}[\mathbb{E}[Y_t | X_t, \mathcal{A}_t] | X_t] + \mathbb{E}[\mathbb{V}[Y_t | X_t, \mathcal{A}_t] | X_t]$. We thus see that the excess variance incurred by using Y_t in the policy gradient estimate can be upper bounded by a constant times the excess variance of the advantage estimate.

E.2. Variance analysis in the bandit problem

Here we provide a back-of-the-envelope variance analysis of the bandit problem. For simplicity (but reasoning can easily be extended), we assume no context and only two actions $\{0, 1\}$, and three vectors $W, V_0, V_1 \in \mathbb{R}^K$ (randomly sampled from a Gaussian and kept constant across all episodes). ϵ_r and ϵ_f are the reward and observation noise respectively, with standard deviations $\sigma_r \gg \sigma_f$. The feedback vector for action a is $W\epsilon_r + V_a + \epsilon_f$.

A forward (in this case, constant) baseline for this problem will have square advantage roughly scale as a σ_r^2.

Let's consider linear hindsight baseline $\alpha^T F$, which is equal to $\epsilon_r(\alpha^T W) + \alpha^T V_a + \alpha^T \epsilon_f$. The expected square advantage $\mathbb{E}[(G - \alpha^T F)^2]$ is therefore

$$\begin{aligned}
\mathbb{E}[(G - \alpha^T F)^2] =& \pi_0 \mathbb{E}[\left(\epsilon_r(\alpha^T W - 1) + \alpha^T V_0 + \alpha^T \epsilon_f\right)^2] + \pi_1 \mathbb{E}[\left(\epsilon_r(\alpha^T W - 1) + (\alpha^T V_1 - 1) + \alpha^T \epsilon_f\right)^2] \\
=& (\alpha^T W - 1)^2 \sigma_r^2 + (\alpha^T \alpha)\sigma_f^2 + \pi_0(\alpha^T V_0)^2 + \pi_1(\alpha^T V_1 - 1)^2
\end{aligned}$$

The vectors W, V_0 and V_1 are independent with probability one (in fact they are nearly orthogonal), one can find a hindsight baseline such that $\alpha^T W - 1 = \alpha^T V_0 = \alpha^T V_1 - 1 = 0$, which leaves an expected squared advantage of $\sigma_f^2 \alpha^T \alpha$ which

is small (for random vectors the matrices will be well-conditioned, the resulting α will have small norm); however that advantage leads to a biased update since the advantage is independent of the action. However, choosing $\alpha^T W = 1$ but $\alpha^T V_0 = \alpha^T V_1 = 0$ leads to a hindsight baseline which is equal to $\epsilon_r + \alpha^T \epsilon_f$, independent from the action; the effect of the noise ϵ_r will be removed entirely from the squared advantage, leading to an unbiased gradient estimator with a considerably lower variance (of order σ_f^2).

F. RL algorithms, common randomness, structural causal models

In this section, we provide an alternative view and intuition behind the CCA-PG algorithm by investigating credit assignment through the lens of causality theory, in particular *structural causal models* (SCMs) (Pearl, 2009a). These ideas are very related to the use of common random numbers (CRN), a standard technique in optimization with simulators (Glasserman & Yao, 1992).

F.1. Structural causal model of the MDP

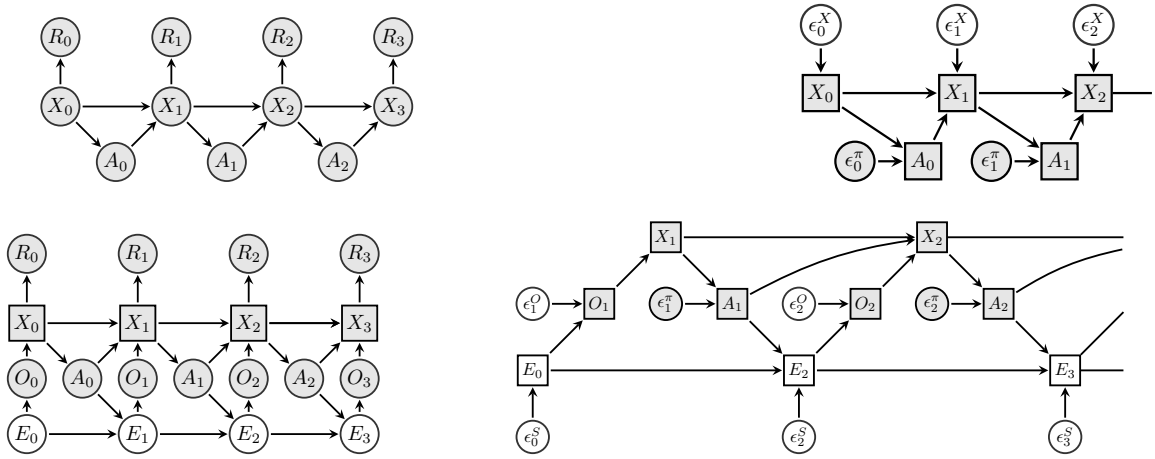


Figure 14: Graphical models and corresponding SCMs for RL problems. Top: MDP, bottom: POMDP; left: graphical model, right: structural causal model. Squares represent deterministic nodes, while circles represent stochastic nodes. Observed nodes are shaded in gray.

Structural causal models (SCM) (Pearl, 2009a) are, informally, models where all randomness is exogenous, and where all variables of interest are modeled as deterministic functions of other variables and of the exogenous randomness. They are of particular interest in causal inference as they enable reasoning about interventions, i.e. how would the *distribution* of a variable change under external influence (such as forcing a variable to take a given value, or changing the process that defines a variable), and about counterfactual interventions, i.e. how would a particular observed outcome (sample) of a variable have changed under external influence. Formally, a SCM is a collection of model variables $\{V \in \boldsymbol{V}\}$, exogenous random variables $\{\mathcal{E} \in \boldsymbol{\mathcal{E}}\}$, and distributions $\{p_{\mathcal{E}}(\varepsilon), \mathcal{E} \in \boldsymbol{\mathcal{E}}\}$, one per exogenous variable, and where the exogenous random variables are all assumed to be independent. Each variable V is defined by a function $V = f_V(\text{pa}(V), \boldsymbol{\mathcal{E}})$, where $\text{pa}(V)$ is a subset of $\boldsymbol{V}$ called the parents of V. The model can be represented by a directed graph in which every node has an incoming edge from each of its parents. For the SCM to be valid, the induced graph has to be a directed acyclic graph (DAG), i.e. there exists a topological ordering of the variables such that for any variable V_i, $\text{pa}(V_i) \subset \{V_1, \dots, V_{i-1}\}$; in the following we will assume such an ordering. This provides a simple sampling mechanism for the model, where the exogenous random variables are first sampled according to their distribution, and each node is then computed in topological order. Note that any probabilistic model can be represented as a SCM by virtue of reparametrization (Kingma & Ba, 2014; Buesing et al., 2019). However, such a representation is not unique, i.e. different SCMs can induce the same distribution.

In the following we give an SCM representation of a MDP (see Fig.14 for the causal graphical model and corresponding SCM for MDPs and POMDPs). The transition from X_t to X_{t+1} under A_t is given by the transition function f^X: $X_{t+1} = f^X(X_t, A_t, \mathcal{E}_t^X)$ with exogenous variable / random number $\mathcal{E}_t^X$. The policy function f^π maps a random number $\mathcal{E}_t^\pi$, policy parameters θ, and current state X_t to the action $A_t = f^\pi(X_t, \mathcal{E}_t^\pi, \theta)$. Together, f^π and $\mathcal{E}_t^\pi$ induce the policy, a distribution $\pi_\theta(A_t|X_t)$ over actions. Without loss of generality we assume that the reward is a deterministic function

of the state and action: $R_t = f^R(X_t, A_t)$. $\mathcal{E}^X$ and $\mathcal{E}^\pi$ are random variables with a fixed distribution; all changes to the policy are absorbed by changes to the deterministic function f^π. Denoting $\mathcal{E}_t = (\mathcal{E}_t^X, \mathcal{E}_t^\pi)$, note the next reward and state (X_{t+1}, R_t) are deterministic functions of X_t and $\mathcal{E}_t$, since we have $X_{t+1} = f^X(X_t, f^\pi(X_t, \mathcal{E}_t^\pi, \theta), \mathcal{E}_t^X)$ and similarly $R_t = R(X_t, f^\pi(X_t, \mathcal{E}_t^\pi, \theta)$. Let $X_{t^+} = (X_{t'})_{t'>t}$ and similarly, $\mathcal{E}_{t^+} = (\mathcal{E}_t^X, \mathcal{E}_{t'})_{t'>t}$ Through the composition of the functions f^X, f^π and R, the return G_t (under policy π) is a deterministic function f^G of X_t, A_t and $\mathcal{E}_{t^+}$.

F.2. Proof of theorem 2

For notation purposes, in the rest of this section, we will focus on credit assignment for action A_t (since policy gradient terms are additive with respect to time), and will denote $X = X_t$, $A = A_t$, $\varepsilon = \mathcal{E}_{t^+}$, and $\tau = (X_s, R_r, A_s)_{s \geq t}$. Furthermore, we will denote $\Phi = \Phi_t$.

From the arguments in the section above, one can write $\tau = f^\tau(X, A, \varepsilon)$, $G = f^G(\tau)$, and $\Phi = f^\phi(\tau)$. We may integrate out τ, in which case the graph only contains X, A, ε and G. In that graph, by the faithfulness assumption, there can be no causal path from A to Φ, as this would violate the conditional independence assumption. It follows that there are functions g_G and g_Φ such that $G = g_G(X, A, \varepsilon)$ and $\Phi = g_\Phi(X, \varepsilon)$.

The resulting structural causal models can be seen in Fig. 15.

The conditional expectation $Q(x, a, \phi)$ is given by $Q(x, a, \phi) = \int_\varepsilon p(\varepsilon|x, \phi, a) G(x, a, \varepsilon)$ The counterfactual return for action a, having observed ϕ is given by $\mathbb{E}[G(\tau')|\tau' \sim P(\tau'|X = x, \text{observe}(\Phi = \phi))]$ is equal to $\int_\varepsilon p(\varepsilon|x, \phi) G(x, a, \phi)$.

Finally, note that from d-separation $p(\varepsilon|x, \phi) = p(\varepsilon|x, a, \phi)$, and the result follows.

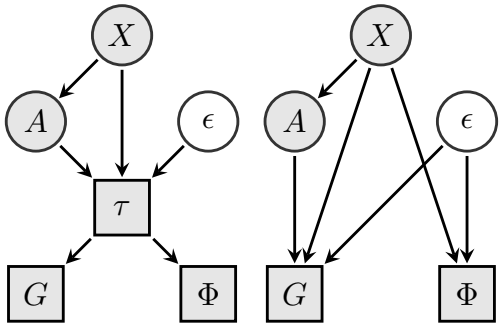


Figure 15: SCMs for the reduced action selection problem; left: including the trajectory; right: trajectory is integrated out. There is no arrow from A to Φ on the right since the graph is assumed to be faithful and A and Φ are conditionally independent given X.

G. Individual Treatment Effects, (Conditional) Average Treatment Effects, Counterfactuals and Counterfactual identifiability

In this section, we will further link the ideas developed in this report to causality theory. In particular we will connect them to two notions of causality theory known as individual treatment effect (ITE) and average treatment effect (ATE). In the previous section, we extensively leveraged the framework of structural causal models. It is however known that distinct SCMs may correspond to the same distribution; learning a model from data, we may learn a model with correct distribution but with with incorrect structural parametrization and counterfactuals. We may therefore wonder whether counterfactual-based approaches may be flawed when using such a model. We investigate this question, and analyze our algorithm in very simple settings for which closed-form computations can be worked out.

G.1. Individual and Average Treatment Effects

Consider a simple medical example which we model with an SCM as illustrated in Fig. 16. We assume population of patients, each with a full medical state denoted S, which summarizes all factors, known or unknown, which affect a patient's future health such as genotype, phenotype etc. While S is never known perfectly, some of the patient's medical history H may be known, including current symptoms. On the basis of H, a treatment decision T is taken; as is often done, for

simplicity we consider T to be a binary variable taking values in {1='treatment', 0='no treatment'}. Finally, health state S and treatment T result in a observed medical outcome O, a binary variable taking values in {1='cured', 0='not cured'}. For a given value $S = s$ and $T = t$, the outcome is a function (also denoted O for simplicity) $O(s, t)$. Additional medical information F may be observed, e.g. further symptoms or information obtained after the treatment, from tests such as X-rays, blood tests, or autopsy.

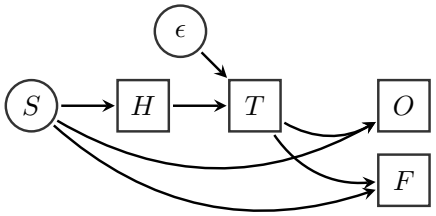


Figure 16: The medical treatment example as a structured causal model.

In this simple setting, we can charactertize the effectiveness of the treatment for an individual a patient with profile S by the Individual Treatment Effect (ITE) which is defined as the difference between the outcome under treatment and no treatment.

Definition 1 (Individual Treatment Effect)**.**

$$\begin{aligned} \textit{ITE(s)} =& \mathbb{E}[O|S = s, \mathrm{do}(T = 1)] - \mathbb{E}[O|S = s, \mathrm{do}(T = 0)] \\ =& O(s, T = 1) - O(s, T = 0) \end{aligned} \tag{8}$$

The conditional average treatment effect is the difference in outcome between the choice of $T = 1$ and $T = 0$ when averaging over all patients with the same set of symptoms $H = h$

Definition 2 (Conditional Average Treatment Effect)**.**

$$\begin{aligned} \textit{ATE}(h) =& \mathbb{E}[O|H = h, \mathrm{do}(T = 1)] - \mathbb{E}[O|H = h, \mathrm{do}(T = 0)] \\ =& \int_s p(S = s|H = h)(O(s, T = 1) - O(s, T = 0)) \end{aligned} \tag{9}$$

Since the exogenous noise (here, S) is generally not known, the ITE is typically an unknowable quantity. For a particular patient (with hidden state S), we will only observe the outcome under $T = 0$ or $T = 1$, depending on which treatment option was chosen; the counterfactual outcome will typically be unknown. Nevertheless, for a given SCM, it can be counterfactually estimated from the outcome and feedback.

Definition 3 (Counterfactually Estimated Individual Treatment Effect)**.**

$$\textit{CF-ITE}[H = h, F = f, T = 1] = \delta(o = 1) - \int_{s'} P(S = s'|H = h, F = f, T = 1)O(s', T = 0) \tag{10}$$

$$\textit{CF-ITE}[H = h, F = f, T = 0] = \int_{s'} P(S = s'|H = h, F = f, T = 1)O(s', T = 0) - \delta(o = 1) \tag{11}$$

In general the counterfactually estimated ITE will not be exactly the ITE, since there may be remaining uncertainty on s. However, the following statements relate CF-ITE, ITE and ATE:

- If S is identifiable from O and F with probability one, then the counterfactually-estimated ITE is equal to the ITE.
- The average (over S, conditional on H) of the ITE is equal to the ATE.
- The average (over S and F, conditional on H) of CF-ITE is equal to the ATE.

Assimilating O to a reward, the above illustrates that the ATE (equation 9) essentially corresponds to a difference of Q functions, the ITE (equation 8) to a difference of returns under common randomness, and the counterfactual ITE to CCA-like

advantage estimates. In contrast, the advantage estimate $G_t - V(H_t)$ is a difference between a return (a sample-level quantity) and a value function (a population-level quantity, which averages over all individuals with the same medical history H); this discrepancy explains why the return-based advantage estimate can have very high variance.

As mentioned previously, for a given joint distribution over observations, rewards and actions, there may exist distinct SCMs that capture that distribution. Those SCMs will all have the same ATE, which measures the effectiveness of a policy on average. But they will generally have different ITE and counterfactual ITE, which, when using model-based counterfactual policy gradient estimators, will lead to different estimators. Choosing the 'wrong' SCM will lead to the wrong counterfactual, and so we may wonder if this is a cause for concern for our methods.

We argue that in terms of learning optimal behaviors (in expectation), estimating inaccurate counterfactual is not a cause for concern. Since all estimators have the same expectation, they would all lead to the correct estimates for the effect of switching a policy for another, and therefore, will all lead to the optimal policy given the information available to the agent. In fact, one could go further and argue that for the purpose of finding good policies in expectations, we should only care about the counterfactual for a precise patient inasmuch as it enables us to quickly and correctly taking better actions for future patients for whom the information available to make the decision (H) is very similar. This would encourage us to choose the SCM for which the CF-ITE has minimal variance, regardless of the value of the true counterfactual. In the next section, we elaborate on an example to highlight the difference in variance between different SCMs with the same distribution and optimal policy.

G.2. Betting against a fair coin

We begin from a simple example, borrowed from (Pearl, 2009b), to show that two SCMs that induce the same interventional and observational distributions can imply different counterfactual distributions. The example consists of a game to guess the outcome of a fair coin toss. The action A and state S both take their values in $\{h, t\}$. Under model $\mathbf{I}$, the outcome O is 1 if $A = S$ and 0 otherwise. Under model $\mathbf{II}$, the guess is ignored, and the outcome is simply $O = 1$ if $S = h$. For both models, the average treatment effect $E[O|A = h] - E[O|A = t]$ is 0 implying that in both models, one cannot do better than random guessing. Under model $\mathbf{I}$, the counterfactual for having observed outcome $O = 1$ and changing the action, is always $O = 0$, and vice-versa (intuitively, changing the guess changes the outcome). Therefore, the ITE is ± 1. Under model $\mathbf{II}$, all counterfactual outcomes are equal to the observed outcomes, since the action has in fact no effect on the outcome. The ITE is always 0.

In the next section, we will next adapt the medical example into a problem in which the choice of action does affect the outcome. Using the CF-ITE as an estimator for the ATE, we will find how the choice of the SCM affects the variance of that estimator (and therefore how the choice of the SCM should affect the speed at which we can learn which is the optimal treatment decision).

G.3. Medical example

Take the simplified medical example from Fig.16, where a population of patients with the same symptoms come to the doctor, and the doctor has a potential treatment T to administer. The state S represents the genetic profile of the patient, which can be one of three $\{\text{GENE}_\text{A}, \text{GENE}_\text{B}, \text{GENE}_\text{C}\}$ (each with probability $1/3$). We assume that genetic testing is not available and that we do not know the value of S for each patient. The doctor has to make a decision whether to administer drugs to this population or not, based on repeated experiments; in other words, they have to find out whether the average treatment effect is positive or not. We consider the two following models:

- In model $\mathbf{I}$, patients of type GENE_A always recover, patients of type GENE_C never do, and patients of type GENE_B recover if they get the treatment, and not otherwise; in particular, in this model, administering the drug never hurts.
- In model $\mathbf{II}$, patients of type GENE_A and GENE_B recover when given the drug, but not patients of type GENE_C; the situation is reversed (GENE_A and GENE_B patients do not recover, GENE_C do) when not taking the drug.

In both models - the true value of giving the drug is $2/3$, and not giving the drug $1/3$, which leads to an ATE of $1/3$. For each model, we will evaluate the variance of the CF-ITE, under one of the four possible treatment-outcome pair. The results are summarized in table 5. Under model $\mathbf{A}$, the variance of the CF-ITE estimate (which is the variance of the advantage estimate used in CCA-PG gradient) is $1/6$, while it is 1 under model $\mathbf{B}$, which would imply $\mathbf{A}$ is a better model to leverage counterfactuals into policy decisions.

Treatment	Outcome	Type	CF-Prob.		CF-O		ITE		CF-V		CF-ITE		Var
Drug	Cured	GENE_A	1/2	1/2	1	0	0	+1					
		GENE_B	1/2	1/2	0	0	+1	+1	1/2	0	1/2	1	
		GENE_C	0	0									1/6
	Not cured	GENE_A	0	0									1
		GENE_B	0	0					0	1	0	-1	
		GENE_C	1	1	0	1	0	-1					
No Drug	Cured	GENE_A	1	0	1	1	0	0					
		GENE_B	0	0					1	0	0	1	
		GENE_C	0	1	0	0	1	1					1/6
	Not cured	GENE_A	0	1/2	1	1	-1	-1					1
		GENE_B	1/2	1/2	1	1	-1	-1	1/2	1	-1/2	-1	
		GENE_C	1/2	0	0	0	0	0					

Table 5: CCA-PG variance estimates in the medical example. CF-Probs. Red value are estimates for model **I**, blue ones are for model **II**. CF-Prob denotes posterior probabilities of the genetic state S given the treatment T and outcome O. CF-O is the counterfactual outcome. The ITE is the individual treatment effect (difference between outcome and counterfactual outcome). CF-V is the counterfactual value function, computed as the average of CF-O under the posterior probabilities for S. CF-ITE is the counterfactual advantage estimate (difference between O and CF-V). Var is the variance of CF-ITE under the prior probabilities for the outcome.

Chapter 6

Quantile Credit Assignment, ICML 2023

6.1 In short

6.1.1 Motivations

In this third and last article [7], published at ICML 2023, we propose a new method for credit assignment called "Quantile Credit Assignment" (QCA). Similarly to the motivations of CCA, we focus here on the inherent randomness in the environment and its impact on the credit assignment problem.

Similarly to "Counterfactual Credit Assignment", both algorithms proposed in this article ("Quantile Credit Assignment" (QCA) and "Hindsight Quantile Credit Assignment" (HQCA)) learn to distinguish between the effects of an action and the effects of the exogenous factors. However, QCA and HQCA have shown to be more robust to noise and outliers than CCA thanks to its simplicity and to the fact that it learns to estimate the quantile of the reward distribution, rather than the expected value.

6.1.2 Approach

QCA works by estimating the quantiles of the return distribution for each state-action pair, whereas HQCA additionally incorporates information about the future. Both QCA and HQCA have the appealing interpretation of leveraging an estimate of the quantile level of the return (interpreted as the level of "luck") in order to derive a "luck-dependent" baseline for policy gradient methods.

We show theoretically that this approach gives an unbiased policy gradient estimator that can yield significant variance reductions over a standard value estimate baseline. This leads to more stable and reliable learning.

6.1.3 Results

We evaluate QCA and HQCA on a variety of challenging credit assignment tasks such as the High-Variance Key-To-Door, the Random Key-To-Door and the Combinatorial RL with Post-Decision Noise Feedback tasks. On all tasks, QCA and

HQCA significantly outperforms prior state-of-the-art methods. Leveraging hindsight information leads to even better results similarly to what was found in CCA.

"Quantile Credit Assignment" is a new method for credit assignment in reinforcement learning that is more robust to noise and outliers than prior methods. QCA has been shown to outperform state-of-the-art methods on a variety of challenging credit assignment tasks.

Quantile Credit Assignment

Thomas Mesnard [*1] **Wenqi Chen** [*2] **Alaa Saade** [*1] **Yunhao Tang** [1] **Mark Rowland** [1] **Theophane Weber** [1] **Clare Lyle** [1] **Audrunas Gruslys** [1] **Michal Valko** [1] **Will Dabney** [1] **Georg Ostrovski** [1] **Eric Moulines** [3] **Remi Munos** [1]

Abstract

In reinforcement learning, the *credit assignment* problem is to distinguish luck from skill, that is, separate the inherent randomness in the environment from the controllable effects of the agent's actions. This paper proposes two novel algorithms, Quantile Credit Assignment (QCA) and Hindsight QCA (HQCA), which incorporate distributional value estimation to perform credit assignment. QCA uses a network that predicts the quantiles of the return distribution, whereas HQCA additionally incorporates information about the future. Both QCA and HQCA have the appealing interpretation of leveraging an estimate of the quantile level of the return (interpreted as the level of "luck") in order to derive a "luck-dependent" baseline for policy gradient methods. We show theoretically that this approach gives an unbiased policy gradient estimator that can yield significant variance reductions over a standard value estimate baseline. QCA and HQCA significantly outperform prior state-of-the-art methods on a range of extremely difficult credit assignment problems.

1. Introduction

Credit assignment (Minsky, 1961) is a critical aspect of sequential decision making. On a high level, the central problem of credit assignment is to understand the relationship between actions and outcomes, and equivalently, to determine to what extent an outcome is caused by external factors instead of actions. For example, a player may have won a football game not because of the actions they took, but because the competitors were weak or they happened to score a low-quality shot on goal. Therefore, when attributing credits, it is important to separate "action" from "luck", i.e., external factors that actions cannot control.

Model-free reinforcement learning (RL) algorithms such as policy gradient methods (Williams, 1992; Sutton et al., 1999) use time as a proxy to perform credit assignment, where actions are credited based upon temporal proximity to subsequent rewards. In spite of the simplicity of such a credit assignment mechanism, such methods tend to introduce high variance due to the uncertainty from the environment. As a result, the agent often requires a larger number of samples to learn good policies, i.e., associating actions with desired outcomes in the optimal way.

A number of prior works have sought to improve the credit assignment mechanisms in model free RL for this type of environment. One important line of work proposed to incorporate hindsight information from the future to perform more efficient credit assignment (Andrychowicz et al., 2017; Harutyunyan et al., 2019). Recently, this has entailed efficient algorithms (e..g, (Mesnard et al., 2020)) that significantly improve over baseline methods in environments where it is important to carry out precise credit assignment for the agent to perform well.

A ubiquitous and indispensable notion to model-free credit assignment is *random return*, i.e., the cumulative sum of reward received by the agent. In this work, we investigate *Quantile Credit Assignment* (QCA), an credit assignment approach that fully exploits the rich structure of random returns in a generic way. QCA leverages the full distribution of the random return, to formalize the notion of "luck" in credit assignment. By performing credit assignment with QCA, model-free agents can disentangle the effect of actions from random external factors more efficiently, leading to much faster policy improvement. Furthermore, we show that QCA can also be flexibly combined with other credit assignment methods, such as incorporating hindsight information. Overall, QCA greatly improves data efficiency in high-variance environments and enable the agents to have robust performance.

Our main contributions are as follows: (1) We formalize the notion of luck with distributions of random return, and how this relates to theoretically grounded variance reduction of

[*]Equal contribution [1]DeepMind [2]Harvard University [3]Ecole polytechnique. Correspondence to: Thomas Mesnard <mesnard@deepmind.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

policy gradient (PG) estimator (Section 3); (2) We propose a scalable implementation of QCA in model-free agents (Section 4); (3) We demonstrate how QCA can combine with orthogonal credit assignment mechanisms such as hindsight information, leading to hindsight QCA (HQCA, Section 5); (4) Finally, we show that QCA and HQCA improves over prior approaches in benchmark tasks (Section 6).

2. Background

We use capital letters for random variables and lower-case for the value they may take. Consider a generic MDP $(\mathcal{X}, \mathcal{A}, R, P, \gamma)$. At each time step t, given a current state $X_t \in \mathcal{X}$ and selected action $A_t \in \mathcal{A}$, the agent receives a reward $R_t = R(X_t, A_t)$ and makes a transition to the next state $X_{t+1} \sim P(\cdot|X_t, A_t)$. Without loss of generality, we assume a fixed initial state $X_0 = x_0 \in \mathcal{X}$. In the case of a partially observed environment, we assume the agent receives an observation O_t at every time step, and simply define state X_t to be the history of previous observations and actions $X_t = (A_{s-1}, R_{s-1}, O_s)_{s \le t}$.

Starting from state action pair $X_t = x, A_t = a$ and following policy π, the agent receives a cumulative sum of reward (also called the return) $Z^\pi_{x,a} \coloneqq \sum_{s=t}^{\infty} \gamma^{s-t} R_s$. Similarly, we define the return distribution η^π_x corresponding to the return obtained when following π from state x. In general, the return is a random variable and we define its distribution as $\eta^\pi_{x,a}$. The value function and Q-function are the expectations of these random returns: $Q^\pi(x,a) = \mathbb{E}_{Z^\pi_{x,a} \sim \eta^\pi_{x,a}}\left[Z^\pi_{x,a}\right]$ and $V^\pi(x) = \mathbb{E}_{Z^\pi_x \sim \eta^\pi_x}\left[Z^\pi_x\right]$.

In general, the agent acts according to a stochastic policy $\pi_\theta(\cdot|X_t)$ where θ are policy parameters.

Notation. In what follows, whenever clear from context, we will omit the explicit dependence on π to simplify notation, writing $\eta_{x,a}$, η_x, $Z_{x,a}$, Z_x, Q, V instead of $\eta^\pi_{x,a}$, η^π_x, $Z^\pi_{x,a}$, Z^π_x, Q^π, V^π.

2.1. Policy Gradient Estimators and Baselines

We begin by recalling the form of policy gradient (PG) algorithms and the intuition behind their credit assignment mechanisms. As a baseline, consider a form of commonly used policy gradient estimator (Williams, 1992; Sutton et al., 1999), which we will also call the single-action policy gradient estimator. Given a trajectory $(X_t, A_t, R_t)_{t=0}^\infty$ generated under π_θ, the policy gradient estimator is defined as follows:

$$\sum_{t \ge 0} \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \left(Z_{X_t,A_t} - V_\psi(X_t)\right), \quad (1)$$

where $V_\psi(x)$ is a state-dependent baseline function. It has been shown that the above estimator is an unbiased estimator of the gradient of value function $\nabla_\theta V^\pi(x_0)$ (Williams, 1992). The baseline function is usually trained to be an approximation to the value function $V_\psi(X_t) \approx V(X_t)$, such that it provides an average estimate of the random return Z_{X_t} from X_t when following policy π. As a result, the difference $Z_{X_t,A_t} - V_\psi(X_t)$ is an estimation to the advantage function $Q(X_t, A_t) - V(X_t)$, which helps identify directions in which the policy can improve.

3. Quantile Credit Assignment PG Estimators

The single-action policy gradient estimator (Equation (1)) uses $Z_{X_t,A_t} - V_\psi(X_t)$ to measure the relative advantage of taking action A_t at state X_t compared to other actions. However, when the random return Z_{X_t,A_t} contains high variance (such as the level of "luck", capturing the intrinsic randomness of the environment as well as the randomness coming from the agent's own future actions), the policy gradient estimator can easily mistake the sign of the advantage estimate, resulting in highly sub-optimal credit assignment. Intuitively, we'd like to remove the amount of 'luck' contained in the random return Z_{X_t,A_t} in order to identify the contribution that the individual action A_t had on that return. For that purpose we design a policy gradient estimator that identifies the luck level $\tau \in [0,1]$ (τ is defined as the quantile level) of the random variable return Z_{X_t,A_t} and assigns credit to the action A_t chosen in X_t at this level of randomness, by subtracting in the policy gradient estimator a baseline function $Q(x, \pi, \tau)$ that depend on this levels of luck. This defines the *quantile credit assignment* (QCA) policy gradient estimator and the corresponding QCA baseline is described next.

QCA Baseline. For any given policy π_θ, recall that $\eta_{x,a}$ is the random return distribution at (x,a). We define $F_{\eta_{x,a}} : \mathbb{R} \to [0,1]$ as its CDF and let

$$Q(x,a,\tau) \coloneqq F^{-1}_{\eta_{x,a}}(\tau) \quad (2)$$

be the inverse CDF evaluated at quantile level $\tau \in [0,1]$ ($Q(x,a,\tau)$ is also called the quantile function). Sampling from $\eta_{x,a}$ is equivalent to generating uniformly $\tau \sim U(0,1)$ and pushing it through the quantile function $Q(x,a,\tau)$. Formally, let $Z_{x,a} \sim \eta_{x,a}$ be a sample of random return, we have

$$Z_{x,a} =_{\mathcal{D}} Q(x,a,\tau), \quad \tau \sim U(0,1)$$

where $=_{\mathcal{D}}$ denotes equality in distribution. Our key insight is to identify the quantile level τ as the luck level in generating the random return Z. When τ is small (corresponding to an unlucky situation), the return is small; when τ is large (a lucky situation), the return is high. Based on $Q(x,a,\tau)$, we define the QCA baseline

$$Q(x,\pi,\tau) \coloneqq \sum_a \pi(a|x) Q(x,a,\tau). \quad (3)$$

For any given quantile (or luck) level $\tau \in [0,1]$, the QCA baseline takes an average of $Q(x,a,\tau)$ over actions under policy π. By taking the average over actions, the QCA baseline $Q(x,\pi,\tau)$ removes the randomness due to sampling immediate actions $a \sim \pi(\cdot|x)$ and retains the other sources of randomness captured in τ. Given a random return Z_{X_t,A_t}, we can understand it as being generated by certain luck level $\widehat{\tau}_t$ via $Z_{X_t,A_t} = Q(X_t,A_t,\widehat{\tau}_t)$. Assume that we can identify the luck level $\widehat{\tau}_t$, a natural advantage estimate would be the difference between the random return and the QCA baseline evaluated at the same luck level $\widehat{\tau}_t$. This gives the QCA advantage estimate

$$Z_{X_t,A_t} - Q(X_t,\pi,\widehat{\tau}_t) = Z_{X_t,A_t} - \sum_a \pi(a|x) F^{-1}_{\eta_{X_t,a}}(F_{\eta_{X_t,A_t}}(Z_{X_t,A_t})),$$

(which is analogous to $Z_{X_t,A_t} - V(X_t)$ in the usual advantage estimator). Finally, we define the QCA PG estimator

$$\sum_{t\geq 0} \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \left(Z_{X_t,A_t} - Q(X_t,\pi,\widehat{\tau}_t)\right). \quad (4)$$

Compared to the single-action PG estimator in Equation (1), the QCA PG estimator applies a baseline $Q(X_t,\pi,\widehat{\tau}_t)$ that depends both on the state x and the level of luck $\widehat{\tau}_t$ inferred from the random return Z_{X_t,A_t}. Intuitively, the QCA advantage estimate $Z_{X_t,A_t} - Q(X_t,\pi,\widehat{\tau}_t)$ represents the advantage of having chosen action A_t instead of a random action drawn from π in X_t, for the same amount of luck (i.e., quantile level $\widehat{\tau}_t$) as in the observed return Z_{X_t,A_t}. An equivalent yet alternative view is that by correlating the random return $Z_{X_t,A_t} = Q(X_t,A_t,\widehat{\tau}_t)$ and the QCA baseline $Q(X_t,\pi,\widehat{\tau}_t)$ by the common luck level $\widehat{\tau}_t$, the QCA PG baseline achieves provable variance reduction compared to the regular PG estimator, as we will formally show below.

Discussion about alternative baselines. In addition to the QCA baseline $Q(X_t,\pi,\widehat{\tau}_t)$, an alternative approach is to define the baseline as $V(X_t,\widehat{\tau}_t)$ where $V(x,\tau) \coloneqq F^{-1}_{\eta_x}(\tau)$ is the quantile function for the return distribution η_x from state x. We refer to this as the value QCA (VQCA) baseline. A conceptual drawback of the VQCA baseline $V(x,\tau)$ is that since the return distribution η_x also contains randomness in the action sampling $a \sim \pi(\cdot|x)$, its quantile level might differ significantly from the quantile levels of the QCA baseline. As a simple example, when the returns from $\eta_{x,a}$ with fixed (x,a) are deterministic, $Q(x,a,\tau)$ is constant for all $\tau \in [0,1]$. However, the return distribution η_x can still have non-zero variance due to stochasticity in choosing actions. Since now the quantile level τ reflects the randomness in the choice of actions instead of the external randomness, we do not advise using $V(x,\tau)$ as a baseline in policy gradient estimators; we provide additional theoretical results regarding VQCA in Appendix B, including an example in which it increases variance relative to a standard value baseline.

3.1. Theoretical Analysis

We now provide several key statistical properties that establish QCA as a principled method for credit assignment, and also give intuition as to when we should expect the benefits of QCA to particularly be pronounced. Our first result proves the unbiasedness of the QCA policy gradient estimator.

Proposition 3.1. The QCA baseline results in unbiased policy gradient estimators when using exact return quantile functions Q^π, as shown in A. That is, with $Z_{X_t,A_t} = Q^\pi(X_t,A_t,\widehat{\tau}_t)$, we have

$$\nabla_\theta V^\pi(x_0) = \mathbb{E}\left[\sum_t \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \left(Z_{X_t,A_t} - Q^\pi(X_t,\pi,\widehat{\tau}_t)\right)\right].$$

Next, we establish that the component of the QCA that estimates the advantage is never worse than the classical state-value function baseline, as measured by variance, and is generally strictly better as shown in A. Traditionally, this has motivated a number of improved baseline functions for policy gradient estimators (see, e.g., (Gu et al., 2016; Liu et al., 2017; Wu et al., 2018; Grathwohl et al., 2017)). Though this does not always guarantee that the full gradient estimator has smaller variance, it is often the case as empirically validated in aforementioned prior work and in our experiments.

Proposition 3.2. The QCA baseline provides an advantage estimate which has no greater variance than that associated with the value baseline when using exact quantile functions Q^π. More precisely, considering a random return $Z_{x,A}$ generated from a state-action pair (x,A), with $A \sim \pi(\cdot|x)$, and writing $Z_{x,A} = Q^\pi(x,A,\widehat{\tau})$ we have

$$\mathrm{Var}\left(Z_{x,A} - Q^\pi(x,\pi,\widehat{\tau})\right) = \mathrm{Var}\left(Z_{x,A} - V^\pi(x)\right) - \mathbb{E}_{\tau'\sim\mathcal{U}([0,1])}\left[\left(Q^\pi(x,\pi,\tau') - V^\pi(x)\right)^2\right].$$

Proposition 3.2 tells us that whenever there is an action a with positive probability $\pi(a|x) > 0$, and a corresponding non-deterministic distribution over returns, we see benefits from the QCA baseline. Further, we should expect large variance reduction when using the QCA baseline (compared to the classical value baseline) precisely when there is high variance in the distribution over returns. With the unbiased property established in Proposition 3.1, it can be guaranteed that quantiles can be used in the baseline of policy gradient without biasing the agent; we note that the assumption that the quantile function is exact is crucial to the unbiasedness property established in Proposition 3.1, in contrast to classical state-based baselines in policy gradients, which are guaranteed to be unbiased even when the value function is

inexact. In summary, while the idea of learning quantiles of the distribution of the return is not new, the great novelty here is that we use it to define a baseline for policy gradient, which further decreases the variance.

4. Implementing QCA

In this section, we introduce the core architectural and algorithmic components of the QCA algorithm.

4.1. Learning the Quantile Function

Central to the QCA baseline is the quantile function $Q(x, a, \tau)$. In practice, since we do not have access to the ground truth quantile function, we parameterize a quantile network $Q_\psi(x, a, \tau) \approx Q(x, a, \tau)$ as an approximation. The network outputs m quantile predictions $Q(x, a, \tau_i)$ with $\tau_i = \frac{2i-1}{2m}$. The i-th quantile prediction is trained using the quantile regression loss (Koenker & Bassett Jr, 1978),

$$\tau_i \left(Z_{x,a} - Q(x, a, \tau_i)\right)_+ + (1 - \tau_i) \left(Z_{x,a} - Q(x, a, \tau_i)\right)_- \tag{5}$$

where $Z_{x,a}$ is a random return generated at (x, a) under policy π. By minimizing the above loss function, $Q_\psi(x, a, \tau_i)$ is guaranteed to form a close approximation to the true quantile function. Intuitively, the more quantile level m we use, the more accurate the approximation is (see, e.g., (Bellemare et al., 2023) for characterizations of the approximation error). Since learning accurate quantile function is of major significance to QCA, we propose two architectural and algorithmic improvements on top of the vanilla quantile network (Dabney et al., 2018), this includes (1) a parameterization that ensures quantile predictions are monotonic $Q_\psi(x, a, \tau_i) \leq Q_\psi(x, a, \tau_{i+1})$, which introduces a useful inductive bias for learning quantiles in general; (2) a novel combination of dueling architecture (Wang et al., 2016) with quantile network, which accelerates learning quantiles through the shared parameterization. Due to space limits, we introduce details in Appendix C.

4.2. Finding the Quantile Level

Recall that in order to define the QCA baseline, we need to identify the quantile level $\widehat{\tau}_t$ for return Z_{X_t,A_t} by solving the equality $Q(X_t, A_t, \widehat{\tau}_t) = Z_{X_t,A_t}$. With the quantile predictions, a challenge with solving the plug-in equality $Q_\psi(X_t, A_t, \widehat{\tau}_t) = Z_{X_t,A_t}$ is that there is a finite number m of predicted quantiles, there might not exist a solution with $\widehat{\tau}_t \in \{\tau_i, 1 \leq i \leq m\}$.

To remedy the above issue, given the set of quantile predictions $Q_\psi(x, a, \tau_i)$ output by the network and given a return sample Z_{X_t,A_t} we select the quantile level $\widehat{\tau}_t$ such that $Z_{X_t,A_t} = Q_\psi(X_t, A_t, \widehat{\tau}_t)$. This is done by considering that our quantile estimate $Q_\psi(X_t, A_t, \tau)$ interpolates piecewise-linearly the quantiles $Q_\psi(X_t, A_t, \tau_i)$ and $Q_\psi(X_t, A_t, \tau_{i+1})$ within each interval $\tau \in [\tau_i, \tau_{i+1}]$, for $1 \leq i < m$. Thus for the index I_t such that $Z_{X_t,A_t} \in [Q_\psi(X_t, A_t, \tau_{I_t}), Q_\psi(X_t, A_t, \tau_{I_t+1})]$, we define

$$\widehat{\tau}_t = (1 - \alpha)\tau_{I_t} + \alpha\tau_{I_t+1} \tag{6}$$
$$\text{with } \alpha = \frac{Z_{X_t,A_t} - Q_\psi(X_t, A_t, \tau_{I_t})}{Q_\psi(X_t, A_t, \tau_{I_t+1}) - Q_\psi(X_t, A_t, \tau_{I_t})}.$$

In the specific extreme cases where $Z_{X_t,A_t} < Q_\psi(X_t, A_t, \tau_1)$ (or $Z_{X_t,A_t} > Q_\psi(X_t, A_t, \tau_m)$) we select the extreme quantile levels: $\widehat{\tau}_t = \tau_1$ (resp. $\widehat{\tau}_t = \tau_m$). By doing this, we ensure that a meaningful quantile level $\widehat{\tau}_t$ can be recovered from the learned quantile function, despite a finite quantile approximation to the full quantile function.

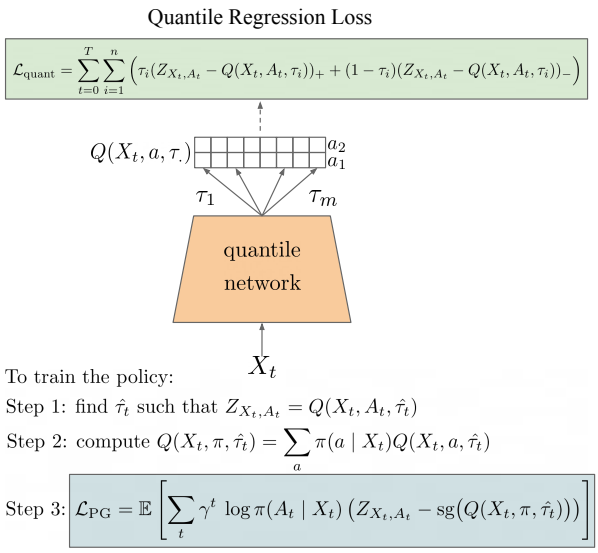


Figure 1. Architecture and pseudocode of the QCA algorithm. QCA trains the quantile function $Q_\psi(x, a, \tau_i)$ with the quantile regression loss; then uses the quantile function to construct QCA baseline and QCA PG estimator.

4.3. The QCA Algorithm

Putting all elements together, we describe the full-fledged QCA algorithm implementation. Throughout, the agent follows the policy network π_θ. At time t, from state X_t, the agent takes action $A_t \sim \pi_\theta(\cdot|X_t)$ and observes return Z_{X_t,A_t} along the trajectory thereafter,

- Train the quantile network $Q_\psi(X_t, A_t, \tau_i)$ (for all $1 \leq i \leq m$) using the quantile regression loss (Eqn 5);
- Identify $\widehat{\tau}_t$ such that $z_{X_t,A_t} = Q_\psi(X_t, A_t, \widehat{\tau}_t)$ using the interpolation strategy in Eqn 6;
- Compute the QCA baseline $Q(X_t, \pi, \widehat{\tau}_t)$ using Eqn 3;

- Update the policy network by the QCA PG estimator

$$\nabla \log \pi_\theta(A_t|X_t) \left(Z_{X_t,A_t} - Q_\psi(X_t, \pi, \widehat{\tau}_t)\right). \quad (7)$$

5. Hindsight QCA

The basic QCA algorithm infers the quantile level $\widehat{\tau}_t$ from the information about the return Z_{X_t,A_t} only. However, it could be the case that information collected after action A_t has been chosen in state X_t can give additional information about the "luck" level of the return. Consider as an example the situation where one has to drive home from work and there are several possible routes. However the length of the travel (the return Z_{X_t,A_t}) may be affected by the weather (impacting the traffic globally) for any route. So observing the weather may directly inform the agent about the level of luck $\widehat{\tau}_t$ without having to learn the quantile function from the returns only. One should be able to generalize the identification of the luck $\widehat{\tau}_t$ level by leveraging information observed along the trajectory $(X_s, A_s, R_s)_{s\geq t}$ instead of relying on the return only.

Motivated by the above example, we introduce Hindsight QCA (HQCA) as a generalization of QCA which uses hindsight information.

5.1. Finding the Hindsight Quantile Level

Inspired by prior work on counterfactual credit assignment (CCA; Mesnard et al., 2020)), we let ϕ_t be a feature vector that summarizes future (hindsight) information collected along the trajectory (observations, actions and rewards $(X_s, A_s, R_s)_{s>t}$) after time t. As a concrete example to represent ϕ_t, , we can compute the feature using a backward RNN. Note that the return $Z_{X_t,A_t} = \sum_{s\geq t} \gamma^{s-t} R_s$ is a special case of hindsight information that can be captured in ϕ_t.

HQCA makes use of an additional network (called the hindsight τ-network) $P(\tau|X_t, A_t, \phi_t)$ from which we predict the quantile level $\widehat{\tau}_t$ of the return Z_{X_t,A_t} from the state of the agent X_t, the selected action A_t and the feature ϕ_t. We now describe how to train the network $P(\tau|X_t, A_t, \phi_t)$ such that it can accurately identify the quantile level based on hindsight features ϕ_t.

Training the hindsight τ-network. In order to train the hindsight τ-network one could think of concatenating the τ network and the quantile network using a variational auto-encoder approach where the encoder (the hindsight τ-network) would produce a distribution over the quantile levels τ (the latent variable), which injected into the decoder (the quantile network) would output the quantiles $Q(x, a, \tau)$, and both networks would be train by regressed these quantiles toward the observed returns. However this training would not produce a latent representation (the quantile level τ) that is independent of the action A_t, given X_t (since the hindsight feature ϕ_t may reveal information about this action), thus possibly biasing the PG estimate.

Instead, we use two separate losses to train these networks. The quantile network is still trained using quantile regression like in the previous section. And the hindsight τ-network $P(\cdot|X_t, A_t, \phi_t)$ is trained (using cross entropy) to predict the quantile level $\hat{\tau}_t$ estimated by the quantile network using Eqn 6. Since the hindsight feature ϕ_t is injected as input to the hindsight τ-network, the corresponding recurrent network $\phi_t = \Phi((X_s, A_s, R_s)_{s>t})$ is trained using the same loss as the hindsight τ-network.

Once all networks have been learned perfectly, we have the property that the quantile level $\hat{\tau}_t$ output by the hindsight τ-network is independent of the action A_t selected in X_t.

5.2. The Hindsight QCA Algorithm

The Hindsight QCA algorithm simply consists in selecting the quantile level $\hat{\tau}_t$ as the output of the hindsight τ-network instead of using the one defined by the quantile network. Once $\hat{\tau}_t$ has been selected, everything else is the same as in QCA: we compute the QCA baseline $Q(X_t, \pi, \widehat{\tau}_t)$ using Eqn 3 and improve the policy network by following the PG estimate using Eqn 7.

The benefit of this approach is that the hindsight τ-network has access to information (through ϕ_t) about the future observations $(X_s, A_s)_{s>t}$ (which is not the case of the quantile network) in addition to the return. This information can be useful to better identify the 'luck level' $\hat{\tau}_t$ because the mapping from the full trajectory to the luck level may generalize better than the same prediction from the return only. For illustration, in the example mentioned above, we expect that the hindsight feature will learn to pay attention to the weather and that this feature will be leveraged by the hindsight τ-network to predict a specific luck level as a function of the observed weather regardless of the route chosen.

Putting it all together. The training process of Hindsight QCA is the following. At time t, from state X_t, the agent takes action $A_t \sim \pi_\theta(\cdot|X_t)$ and observes a trajectory $(X_s, A_s, R_s)_{s\geq t}$,

- Train the quantile network $Q_\psi(X_t, A_t, \tau_i)$ (for all $1 \leq i \leq m$) using the quantile regression loss (Eqn 5),
- Define the target distribution $\hat{\eta}_t = (1-\alpha)\delta_{\tau_{I_t}} + \alpha\delta_{\tau_{I_t+1}}$, where α and I_t are defined as in Eqn 6,
- Train the hindsight τ-network by minimizing the cross entropy loss between $\hat{\eta}_t$ and $P(\tau|X_t, A_t, \phi_t)$, whose gradient further propagates through the hindsight network $\phi_t = \Phi((X_s, A_s, R_s)_{s\geq t})$ for training,

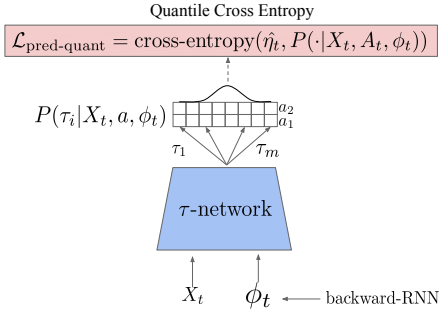
- Identify the hindsight quantile level

$$\hat{\tau}_t = \underset{\{\tau_i\}_{1\leq i\leq m}}{\arg\max}\ P(\tau_i|X_t, A_t, \phi_t),$$
$$\text{or } \hat{\tau}_t = \sum_{i=1}^{m} \tau_i P(\tau_i|X_t, A_t, \phi_t),$$

- Compute the QCA baseline $Q(X_t, \pi, \widehat{\tau}_t)$ using Eqn 3,
- Update the policy network by the QCA PG estimator using Eqn 7.

Notice that we describe two ways to define the quantile level $\hat{\tau}_t$ from the hindsight τ-network $P(\tau_i|x, a, \phi)$. Experimentally these two methods give very similar performances.

Comparison to CCA (Mesnard et al., 2020). HQCA shares similarities with the Counterfactual Credit Assignment (CCA) algorithm in that hindsight information from the future is captured by a feature ϕ_t and used in a policy gradient algorithm. In CCA, ϕ_t is learnt to predict the future return while enforcing the property of conditional independence with the action A_t given X_t. This property is required to avoid biasing the PG estimate. In HQCA, we do not enforce this independence property between ϕ_t and A_t (and in general we expect ϕ_t and A_t to be dependent). However, because we use quantile regression to train the quantile network, the target quantile levels used to train the hindsight τ-network are uniformly distributed and independent of X_t, A_t. This enforces the property that the hindsight quantile level $\hat{\tau}_t$ is independent of A_t given X_t (it is actually independent of A_t and X_t), which guarantees unbiasedness of the PG estimate once the networks have been trained.



To train the policy:

Step 1: Define $\hat{\tau}_t = \text{argmax}_\tau(P(\tau|X_t, A_t, \phi_t))$

Step 2 and 3: Similar to QCA

Figure 2. Architecture and pseudocode of the HQCA algorithm. On top of QCA, HQCA uses a backward RNN to compute the hindsight feature ϕ_t that can be used by the hindsight τ-network P to identify the quantile level. This is then used for computing the baseline and PG estimators.

6. Experiments

In order to validate the hypothesis that QCA and HQCA perform well in environments where disentangling "skill" from "luck" is difficult, we investigate the performances of the proposed algorithms in three high variances environments that are described below. In parallel, we also run three strong baselines: (i) a straightforward policy gradient with baseline (PG); (ii) CCA, to see how well a previous state-of-the-art credit assignment method performs; and (iii) a PG agent with a distributional critic (as in QCA), but using only the *mean* estimated by the critic as a baseline, to disentangle improvements to credit assignment in (H)QCA from improvements in representation learning that are often observed with distributional critics (Barth-Maron et al., 2018; Hoffman et al., 2020; Duan et al., 2021; Nam et al., 2021; Shahriari et al., 2022). All results are reported as median performances over 20 seeds with interquartile range represented by a shaded area. Note that the same amount of time or less was spent to tuned QCA and HQCA in comparison to the time spent to tune the baselines.

6.1. High-Variance Key-To-Door

First, we propose to look at a new version of the Key-To-Door family, initially proposed by Hung et al. (2019), as a testbed for credit assignment in noisy environments. In this partially observable grid-world (Figure 7), the agent has to pick up a key in the first room for which it gets no immediate reward. In a second room, the agent can pick up 10 apples that each give an immediate reward. This is what we call the distraction phase. In the final room, the agent may open a door only if it is carrying a key, and receive a small reward for doing so. In this task, a single and early action (i.e picking up the key) impacts the reward it receives at the end of the episode. This signal is hard to detect as the episode return is largely driven by the agent's performance at picking up apples in the second room.

High-variance Key-to-door (HVKTD) keeps the overall structure of the Key-To-Door task proposed by Mesnard et al. (2020), however the reward for each apple is randomly sampled from the distribution of Uniform$\{-8, 8.2\}$. In this setting, even an agent that is skilled at picking up apples observes a large variance in its episode returns which makes the learning signal for picking up the key and opening the door weak and noisy. A perfectly trained agent will be able to get a total return of 2, half of this return coming from picking up the apples and half coming from opening the door.

Results. As shown in Figure 3, the PG algorithm is unable to learn to pick up the key and open the door. Distributional RL itself allows some learning progress to be made, and leveraging these quantiles to do credit assignment results in a strong improvement as shown by the QCA results. Fi-

nally, using hindsight information to inform the quantile level enables HQCA to match CCA performances which was specifically designed for this environment. Note first that the variance between seeds for HQCA is much smaller than the one for CCA. Furthermore, HQCA is capable of matching the performance of CCA with relatively little hyperparameter tuning. In our experience, we found QCA and HQCA are more robust and less finicky to tune than CCA.

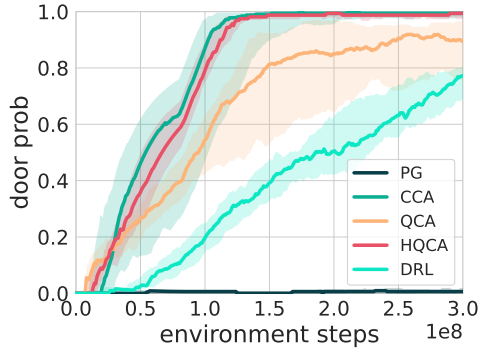


Figure 3. Results for High-Variance Key-To-Door.

6.2. Random Key-To-Door

We now consider a second variation of Key-To-Door. Instead of having immediate rewards from picking up apples during the distraction phase, Random Key-To-Door (RKTD) provides random rewards to the agent sampled from the distribution $N(0, 1)$ for each time step in this phase. The agent always receives noisy rewards in this phase as the noise level is now independent of the agent's behavior.

Results. RKTD is a very challenging task both for policy gradient and CCA as they are not able to not learn to open the door even after $3e6$ environment steps Figure 4 (only $1.5e6$ shown here). On the contrary, DRL, QCA and HQCA solve the environment rapidly and reliably. The quantile loss used in DRL, QCA, and HQCA greatly helps performance. Once again, leveraging these quantiles to do credit assignment improves data efficiency in particular when using hindsight, and we find that QCA and HQCA perform robustly, leading to efficient learning in highly noisy environments. One explanation why QCA and HQCA outperform CCA is that QCA approaches only need to reconstruct the quantile function of a Gaussian while CCA needs to learn to reconstruct the return.

6.3. Combinatorial RL with Post-Decision Noise Feedback

Finally, we consider a task where, in a first "query room", the agent is faced with two colored squares. When the agent picks up a colored square, it immediately receives the reward $R = r + \sigma$ where $\sigma \sim \text{Uniform}[0, 5]$ and $r = 1$ if the

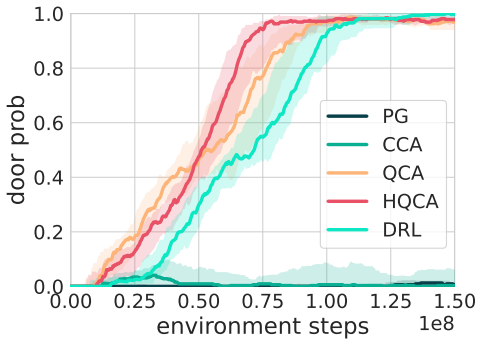


Figure 4. Results for Random Key-To-Door.

agent has picked the "good" object and $r = 0$ otherwise. The agent is then teleported to a second room, the "answer room", where it can observe another colored square whose color is deterministically mapped to the noise level σ that has been sampled for the computation of R.

The task consists in a succession of these two rooms, with randomly sampled pairs of colored squares for the "query" room and a new noise level sampled each time. Note that the sampling makes sure that the agent is always faced with one good and one bad colored square. Finally, the color mapping of "good" and "bad" objects is kept fixed through training. Performances are reported as the number of rooms where the agent has picked the "good" colored object. Note that a small exploration bonus is given to the agent when it picks up any colored square to make the exploration problem simpler. A visual description of the task can be seen in Figure 5.

This task can be abstracted as a contextual bandit problem where with a large number of contexts (i.e the pairs of colored squares) and two actions (i.e left or right), though the agent is of course not *a priori* aware of this structure. However, the reward has a specific form. It is the sum of a deterministic part corresponding to the action taken and a stochastic part independent of the action. The dependency between the good action and the context is unstructured (no regularity). After the agent has taken its action, the stochastic part of the reward is revealed in a visual way. Classical methods such as plain policy gradient are not expected to perform well here because they cannot efficiently learn the reward structure, as it lacks access to hindsight information. It will have to use many samples to distinguish the deterministic from the stochastic part for each state and action. However, as HQCA (and CCA) has access to hindsight information, it should be able to leverage this to learn the reward structure of the task rapidly. Indeed, when given the stochastic part of the reward, inferring the correct action is trivial.

Results. As shown in Figure 6, HQCA and CCA both solve the task thanks to their use of hindsight information. They get close to the optimal score (which is 9) because a very

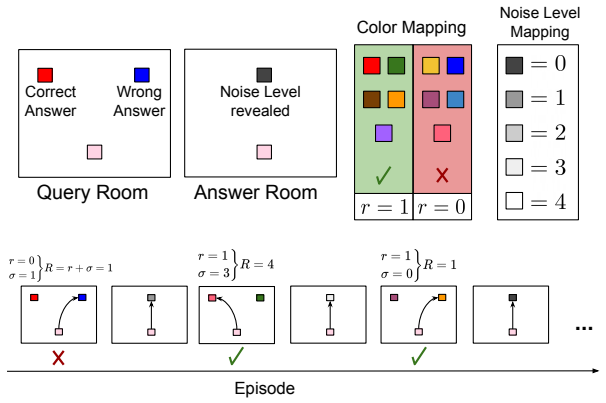


Figure 5. Description of the "Combinatorial RL with post-decision noise feedback" task.

short timing to visit all the rooms before the environment times out. On the contrary, PG and DRL find a local optimal which consists of simply picking up any square, regardless of their color, to get the small exploration bonus for all query rooms. As they pick the "good" object with 50% chance, they get a score of 4.5 out of 9 query rooms that can be visited in an episode. Finally, QCA seems to perform slightly better than PG and DRL but still does not solve the tasks reliably as it also lacks access to hindsight information to inform its "level".

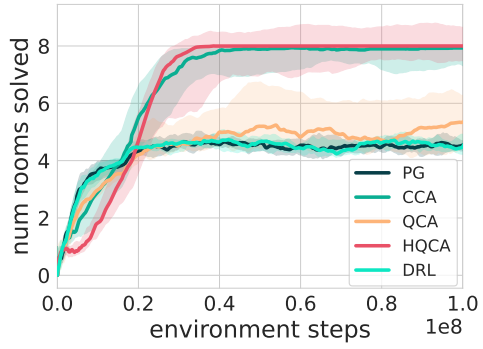


Figure 6. Results for the "Combinatorial RL with post-decision noise feedback" task.

7. Related Work

There are two tracks of motivations for our paper based on the previous work. On the one hand, there is past research that sheds light on proposing a better baseline through incorporating more information from the future to decrease the variance. Counterfactual Credit Assignment (CCA) Mesnard et al. (2020) leverages *hindsight* information to implicitly perform counterfactual evaluation-an estimate of the return for actions other than the ones which were chosen. The counterfactual reasoning will enable the agent to reason about what would have happened had different actions been taken *with everything else remaining the same*. In this way, it can form unbiased and lower variance estimates of the policy gradient by building future-conditional baselines. However, in order to make the baseline independent with actions to prevent biases, CCA has to introduce additional action-removal loss to force the information from actions can be disentangled from baseline. As a result, the algorithm will be unintentionally unstable and will lack interpretability. On the contrary, QCA does not include extra losses during training and will be easily interpreted with the estimation of quantiles.

On the other hand, our work also follows the research that focuses on distributional reinforcement learning in which the distribution over returns is modeled explicitly instead of estimating the mean, which is to examine methods of learning the return distribution instead of value function. Unlike traditional value-based reinforcement learning algorithms like DQN (Mnih et al., 2015) average over randomness to estimate the value, distributional reinforcement learning methods model this distribution over returns explicitly instead of only estimating the mean. This can lead to more insights and knowledge for the agent with a much faster and more stable learning. In Categorical DQN (C51; Bellemare et al., 2017), the possible returns are limited to a discrete set of fixed values (51), and the probability of each value is learned through interacting with environments. Based on it, QR-DQN computes the return quantiles on fixed , uniform quantile fractions using quantile regression and minimizes the quantile Huber loss between the Bellman updated distribution and current return distribution. However, the past research on distributional RL focuses more on representation learning. QCA instead follows the track of policy gradient and innovatively uses the quantiles directly as baseline, which will lower the variance in a straightforward way.

8. Conclusion

In this paper we introduced an approach based on quantile estimation to improve credit assignment in RL by disentangling "luck" from "skill". QCA builds an estimate of the quantile function of the return and HQCA additionally estimates the quantile level (interpreted as the level of "luck") from a full trajectory. These methods produce a "luck-dependent" baseline for policy gradient methods, which does not introduce bias and potentially significantly reduce the variance of the PG estimate (compared to a standard value estimate baseline). Experimentally, QCA and HQCA significantly outperform prior state-of-the-art methods on a range of difficult credit assignment problems.

Future research will investigate the performance of the algorithms and how to scale them in more complex environments which are closer to real-world problems.

References

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Tb, D., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. In *Proceedings of the International Conference on Learning Representations*, 2018.

Bellemare, M. G., Dabney, W., and Munos, R. A distributional perspective on reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2017.

Bellemare, M. G., Dabney, W., and Rowland, M. *Distributional Reinforcement Learning*. MIT Press, 2023.

Dabney, W., Rowland, M., Bellemare, M., and Munos, R. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Duan, J., Guan, Y., Li, S. E., Ren, Y., Sun, Q., and Cheng, B. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6584–6598, 2021.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

Harutyunyan, A., Dabney, W., Mesnard, T., Gheshlaghi Azar, M., Piot, B., Heess, N., van Hasselt, H. P., Wayne, G., Singh, S., Precup, D., and Munos, R. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, 2019.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

Hoffman, M., Shahriari, B., Aslanides, J., Barth-Maron, G., Behbahani, F., Norman, T., Abdolmaleki, A., Cassirer, A., Yang, F., Baumli, K., Henderson, S., Friesen, A., Haroun, R., Novikov, A., Gómez Colmenarejo, S., Cabi, S., Gulcehre, C., Le Paine, T., Srinivasan, S., Cowie, A., Wang, Z., Piot, B., and de Freitas, N. Acme: A research framework for distributed reinforcement learning. *arXiv*, 2020.

Hung, C.-C., Lillicrap, T., Abramson, J., Wu, Y., Mirza, M., Carnevale, F., Ahuja, A., and Wayne, G. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):1–12, 2019.

Koenker, R. and Bassett Jr, G. Regression quantiles. *Econometrica: journal of the Econometric Society*, pp. 33–50, 1978.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. Action-depedent control variates for policy optimization via stein's identity. *arXiv preprint arXiv:1710.11198*, 2017.

Luo, Y., Liu, G., Duan, H., Schulte, O., and Poupart, P. Distributional reinforcement learning with monotonic splines. In *Proceedings of the International Conference on Learning Representations*, 2021.

Mesnard, T., Weber, T., Viola, F., Thakoor, S., Saade, A., Harutyunyan, A., Dabney, W., Stepleton, T., Heess, N., Guez, A., Moulines, É., Hutter, M., Buesing, L., and Munos, R. Counterfactual credit assignment in model-free reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2020.

Minsky, M. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Nam, D. W., Kim, Y., and Park, C. Y. GMAC: A distributional perspective on actor-critic framework. In *Proceedings of the International Conference on Machine Learning*, 2021.

Shahriari, B., Abdolmaleki, A., Byravan, A., Friesen, A., Liu, S., Springenberg, J. T., Heess, N., Hoffman, M., and Riedmiller, M. Revisiting Gaussian mixture critic in off-policy reinforcement learning: A sample-based approach. *arXiv preprint arXiv:2204.10256*, 2022.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999.

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, (3):229–256, 1992.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. *arXiv preprint arXiv:1803.07246*, 2018.

Zhou, F., Wang, J., and Feng, X. Non-crossing quantile regression for distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 2020.

APPENDICES

A. Proofs

Proposition A.1. The QCA baseline results in unbiased policy gradient estimators when using exact return quantile functions Q^π, as shown in A. That is, with $Z_{X_t,A_t} = Q^\pi(X_t, A_t, \widehat{\tau}_t)$, we have

$$\nabla_\theta V^\pi(x_0) = \mathbb{E}\Big[\sum_t \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) \\ (Z_{X_t,A_t} - Q^\pi(X_t, \pi, \widehat{\tau}_t))\Big] .$$

Proof. In analogy with the proof of unbiasedness for the policy gradient estimator with a state-based baseline (see e.g. Sutton & Barto, 2018), it is sufficient to prove that for each $t \geq 0$, the baseline $Q^\pi(X_t, \pi, \widehat{\tau}_t)$ is conditionally independent of the action A_t given X_t, so that

$$\begin{aligned}
&\mathbb{E}\Big[\sum_t \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t)(Z_{X_t,A_t} - Q^\pi(X_t, \pi, \widehat{\tau}_t))\Big] \\
&\qquad = \mathbb{E}\Big[\sum_t \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t) Z_{X_t,A_t}\Big] - \sum_t \gamma^t \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(A_t|X_t) Q^\pi(X_t, \pi, \widehat{\tau}_t))\Big] \\
&\qquad = \nabla V^\pi(x_0) - \sum_t \gamma^t \mathbb{E}_{X_t}\Big[\mathbb{E}_{A_t,\widehat{\tau}_t}[\nabla_\theta \log \pi_\theta(A_t|X_t) Q^\pi(X_t, \pi, \widehat{\tau}_t))|X_t]\Big] \\
&\qquad = \nabla V^\pi(x_0) - \sum_t \gamma^t \mathbb{E}_{X_t}\Big[\mathbb{E}_{A_t}[\nabla_\theta \log \pi_\theta(A_t|X_t) \mid X_t]\mathbb{E}_{\widehat{\tau}_t}[Q^\pi(X_t, \pi, \widehat{\tau}_t))|X_t]\Big] \\
&\qquad = \nabla V^\pi(x_0) ,
\end{aligned}$$

where the second term evaluates to 0 since

$$\mathbb{E}_{A_t}[\nabla_\theta \log \pi_\theta(A_t|X_t)] = \sum_{a_t} \pi(a_t \mid X_t) \nabla \log \pi(a_t \mid X_t) = \nabla \sum_{a_t} \pi(a_t \mid X_t) = 0 .$$

To see the required conditional independence property, note that conditional on X_t and A_t, $\widehat{\tau}_t \sim$ Uniform$([0, 1])$ by construction, which does not depend on A_t and hence $Q^\pi(X_t, \pi, \widehat{\tau}_t)$ is conditionally independent of A_t given X_t. □

Proposition A.2. The QCA baseline provides an advantage estimate which has no greater variance than that associated with the value baseline when using exact quantile functions Q^π. More precisely, considering a random return $Z_{x,A}$ generated from a state-action pair (x, A), with $A \sim \pi(\cdot|x)$, and writing $Z_{x,A} = Q^\pi(x, A, \widehat{\tau})$ we have

$$\begin{aligned}
\mathrm{Var}\left(Z_{x,A} - Q^\pi(x, \pi, \widehat{\tau})\right) = \mathrm{Var}\left(Z_{x,A} - V^\pi(x)\right) - \\
\mathbb{E}_{\tau' \sim \mathcal{U}([0,1])}\left[\left(Q^\pi(x, \pi, \tau') - V^\pi(x)\right)^2\right] .
\end{aligned}$$

Proof. Since both $Z - Q^\pi(x, \pi, \widehat{\tau})$ and $Z - V^\pi(x)$ are unbiased estimators of the advantage $A^\pi(x, a)$, it suffices to compare their second moments. We calculate directly:

$$\begin{aligned}
\mathbb{E}[(Z - V^\pi(x)^2] &= \mathbb{E}[(Z - Q^\pi(x, \pi, \widehat{\tau}) + Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))^2] \\
&= \mathbb{E}[(Z - Q^\pi(x, \pi, \widehat{\tau}))^2 + 2(Z - Q^\pi(x, \pi, \widehat{\tau}))(Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x)) + (Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))^2] \\
&= \mathbb{E}[(Z - Q^\pi(x, \pi, \widehat{\tau}))^2] + \mathbb{E}[(Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))^2] ,
\end{aligned}$$

as required, with the final equality following since

$$\begin{aligned}
\mathbb{E}[(Z - Q^\pi(x, \pi, \widehat{\tau}))(Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))] &= \mathbb{E}[\mathbb{E}_A[Z - Q^\pi(x, \pi, \widehat{\tau})](Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))] \\
&= \mathbb{E}[(Q^\pi(x, \pi, \widehat{\tau}) - Q^\pi(x, \pi, \widehat{\tau}))(Q^\pi(x, \pi, \widehat{\tau}) - V^\pi(x))] \\
&= 0 .
\end{aligned}$$

□

B. Additional analysis

In this section, we provide additional analysis of the value-quantile baseline described in the main paper. First, we show that, as with the QCA baseline, using an exact VQCA baseline results in an unbiased policy gradient estimator.

Proposition B.1. The VQCA baseline results in unbiased policy gradient estimators when using exact return CDFs. That is, with $Z_t = V^\pi(X_t, \widehat{\tau}_t)$, we have

$$\nabla_\theta V^\pi(x_0) = \mathbb{E}\left[\sum_t \gamma^t \nabla_\theta \log \pi_\theta(A_t|X_t)(Z_t - V^\pi(X_t, \widehat{\tau}_t))\right].$$

Proof. We may follow the same approach as the proof of Proposition 3.1; it is sufficient to show that $\mathbb{E}[\nabla_\theta \log \pi_\theta(A_t|X_t)V^\pi(X_t, \widehat{\tau}_t)] = 0$. As in the proof of Proposition 3.1, this follows since $\mathbb{E}[\nabla_\theta \log \pi_\theta(A_t|X_t)] = 0$, and the fact that by construction, $\nabla_\theta \log \pi_\theta(A_t|X_t)$ and τ_t are conditionally independent given X_t. □

Next, we demonstrate that there are scenarios in which using a VQCA baseline can result in higher variance estimators than would be obtained with a standard expected-value baseline. For this reason, we do not recommend VQCA as a policy gradient baseline, instead preferring QCA, with the variance improvement guarantee established in Proposition 3.2.

Example B.2. Consider a single-state environment with two actions, a and b, which are equally likely under the policy π. Suppose that the return when taking action a is distributed as Unif$([-z-\varepsilon, -z+\varepsilon])$, and the return when taking action b is distributed as Unif$([z-\varepsilon, z+\varepsilon])$, for $0 < \varepsilon \ll z$. The expected-value baseline in this case is 0, and so the variance of the return minus this estimator is

$$\mathbb{E}[Z^2] = z^2 + O(\varepsilon z + \varepsilon^2).$$

In contrast, the VQCA baseline, at level τ, is

$$V(\tau) = \begin{cases} -z + 4(\tau - 1/4)\varepsilon & 0 < \tau < 1/2 \\ z + 4(\tau - 3/4)\varepsilon & 0 < \tau < 1/2 \end{cases}.$$

The resulting variance of the return minus this estimator is therefore

$$\mathbb{E}[(Z - V(\tau))^2] = 2z^2 + O(\varepsilon),$$

and hence the variance is greater than with the expected-value baseline.

C. Architectural and algorithmic improvements over quantile networks

Below we introduce a number of architectural improvements over the vanilla quantile network used in prior work on distributional RL (Dabney et al., 2018). The vanilla quantile network $Q_\psi(x, a, \tau_i)$ produces m quantile predictions for $\tau_i = \frac{2i-1}{2m}$ with $1 \leq i \leq m$. Furthermore, in practice, we use a Huber loss variant of the quantile regression loss to learn quantile predictions (Dabney et al., 2018).

Monotonicity of quantile parameterizations. From the definition of quantile functions, we know that they increase monotonically as a function of the quantile levels $Q(x, a, \tau_i) \leq Q(x, a, \tau_j)$ for $i < j$. To leverage this property in the network design, we construct quantile predictions $Q_\psi(x, a, \tau_i)$ as a sum of non-negative increments. To utilize this attribute, we carry out the parameterization $Q_\psi(x, a, \tau_i) = \sum_{j=1}^{i} Q_\psi(x, a, j)$ where $Q_\psi(x, a, j)$ is parameterized to be non-negative via the softplus activation for the output layer $\log(1+\exp(x))$. We can understand $Q_\psi(x, a, 1)$ as the first quantile prediction and $Q_\psi(x, a, j), j \geq 2$ as the difference between two consecutive quantile predictions. This is a useful inductive bias and helps learn quantiles. Existing alternative architectures aiming exploit this structure include those of Zhou et al. (2020) and Luo et al. (2021).

Dueling architecture. Dueling network (Wang et al., 2016) proposes to have two streams to separately estimate state-value and the advantages for each action. Empirically, this has proved particularly useful in accelerating learning accurate Q-functions for value-based learning and distributional RL (Hessel et al., 2018). While prior work has adapted the dueling

architecture for C51 (Bellemare et al., 2017), an alternative distributional RL agent that learns CDF approximation instead of quantile approximation to the return, we propose a novel adaptation for the quantile network. Concretely, we carry out the parameterization

$$Q_\psi(x, a, \tau) = V(x) + A_\psi(x, a, \tau),$$

where $V(x)$ (which we call forward baseline below) is regressed (using a ℓ_2-loss) toward the Monte-Carlo return $Z_{x,A}$, where $A \sim \pi(\cdot|x)$, and $A_\psi(x, a, \tau)$ are actually the output of our quantile-network (which thus learns the quantile function of the return minus the estimated value function).

D. Implementation details

D.1. High-Variance Key-to-Door

D.1.1. Environment details

Observations returned by the Key-to-Door family of environments for each of the three phases can be visualized in Fig. 7. Agents have 10 apples in the second phase to pick.

Figure 7. High-Variance Key-To-Door environments visual. The agent is represented by the beige pixel, key by brown, apples by green, and the final door by blue. The agent has a partial field of view, highlighted in white

D.1.2. Architecture

The agent architecture is as follows:

- The observations are first fed to 2-layer CNN with $(16, 32)$ output channels, kernel shapes of $(3, 3)$ and strides of $(1, 1)$. The output of the CNN is flattened and fed to a linear layer of size 128.
- The agent state is computed by a forward LSTM with a state size of 128. The input to the LSTM is the output of the previous linear layer, concatenated with the reward at the previous timestep.
- The hindsight feature Φ is computed by a backward LSTM with a state size of 128. The input provided is the concatenation of the output of the forward LSTM and the reward at the previous timestep.
- The policy is computed as the output of a 2-layer MLP of 256 units each where the output of the forward LSTM is provided as input. This MLP is shared with the policy. The policy is then linearly decoded from its outputs.
- The forward baseline is computed linearly decoded from the MLP shared with the policy.
- The quantile network is computed as the output of a 3-layer MLP of 128 units each where the output of the forward LSTM is provided as input.
- The $\tau - network$ is the output of a 4-layer MLP of 128 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input.

- For CCA, the baseline is computed as the sum of the forward baseline and a hindsight residual baseline; the hindsight residual baseline is the output of a 3-layer MLP of 128 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input. It is trained to learn the residual between the return and the forward baseline.
- For CCA, the hindsight classifier h_ω is computed as the sum of the log of the policy outputs and the output of an MLP, with four hidden layers with 256 units each where the concatenation of the output of the forward LSTM and the hindsight feature Φ is provided as input.
- All weights are jointly trained with RMSprop (Hinton et al., 2012) with epsilon $1e\text{-}8$, momentum 0 and decay 0.99.

For High-Variance Key-To-Door, the optimal hyperparameters found for each algorithm can be found in Table 1.

The agents are trained on full-episode trajectories, using a discount factor of 0.9999.

	PG	CCA	DRL	QCA	HQCA
Learning rate	5e-4	5e-4	5e-4	5e-4	5e-4
Policy cost	1	1	1	1	1
Entropy cost	1e-2	1e-2	1e-2	1e-2	1e-2
Forward baseline cost	1e-1	1e-2	1e-1	1e-1	1e-1
Number of discrete quantiles	____	____	5	5	10
Huber loss param	____	____	1.	1.	1.
Quantile regression cost	____	____	1e-1	1e-1	1e-1
Hindsight quantile prediction cost	____	____	____	____	1e-2
Hindsight residual baseline cost Mesnard et al. (2020)	____	1e-2	____	____	_____
Hindsight classifier cost Mesnard et al. (2020)	____	5e-3	____	____	_____
Action independence cost Mesnard et al. (2020)	____	1e2	____	____	_____

Table 1. List of hyperparameters used for all experiments.

D.2. Random Key-to-Door

D.2.1. Environment details

Observations returned by the Random Key-To-Door family of environments for each of the three phases can be visualized in Fig. 8. Agents get immediate random rewards during the second phase, distracting them from opening the door.

Figure 8. Random Key-To-Door environments visual. The agent is represented by the beige pixel, key by brown, and the final door by blue. The agent has a partial field of view, highlighted in white

For each task, a random, but fixed through training, set of 5 out of 10 colored squares are leading to a positive reward. Furthermore, a small reward of 0.5 is provided to the agent when it picks up any colored square. Each episode are 130 steps long and it takes at least 9 steps for the agent to reach one colored square in the query rooms from its initial position and 6 in the answer rooms.

D.2.2. ARCHITECTURE

We use the same architecture setup as reported in Appendix D.1.2. The agents are also trained on full-episode trajectories, using a discount factor of 0.9999. For Random Key-to-Door, the optimal hyperparameters found for each algorithm are the same as in Table 1.

Chapter 7

Summary of contributions

Let's summarize the key contributions done in this PhD and discuss the impact of this work on the reinforcement learning community, particularly in helping create the new active domain of credit assignment.

The work presented in these articles addresses the fundamental challenge of credit assignment in reinforcement learning, which is essential for enabling agents to learn efficiently and effectively in complex and uncertain environments.

First, "Hindsight Credit Assignment" [4] introduced the concept of leveraging hindsight information to enhance credit assignment. It proposes a family of algorithms that explicitly assign credit to past actions based on their likelihood of contributing to the observed outcome. By computing a hindsight distribution using importance sampling, these algorithms improve data efficiency and robustness in various tasks.

"Counterfactual Credit Assignment" [6] took a novel approach by using counterfactuals from causality theory to improve credit assignment in model-free reinforcement learning. The key idea was to condition value functions on future events and extract relevant information from a trajectory, enabling the disentanglement of an agent's actions from the effects of external factors. CCA algorithms, which used future-conditional value functions as baselines, exhibited lower variance and outperformed standard policy gradient methods on challenging tasks.

Finally, "Quantile Credit Assignment" [7] introduced a new method for credit assignment that estimates the quantiles of the return distribution for each state-action pair. QCA and "Hindsight Quantile Credit Assignment" were shown to be more robust to noise and outliers compared to CCA. These algorithms leveraged the quantile level of the return as input to a "luck-dependent" baseline for policy gradient methods, leading to unbiased estimators with reduced variance and more stable learning.

The impact of this work on the RL community has been substantial. The introduction of HCA laid the foundation for considering hindsight information as a valuable resource for credit assignment. This approach has been widely adopted and cited (73 citations at the time of writing), influencing subsequent research

and algorithm development.

CCA, extended the understanding of credit assignment by incorporating concepts from causality theory. It provided a valuable perspective on disentangling the effects of agent actions from external factors. CCA's success in improving credit assignment in challenging tasks has inspired further exploration of these ideas and their applications in various domains.

QCA, introduced a new method that embraces randomness and inherent uncertainty in the environment, contributing to a more comprehensive understanding of credit assignment. QCA and HQCA's performance on challenging tasks demonstrated the potential for quantile-based approaches to further enhance credit assignment in RL.

Collectively, this PhD thesis has made significant contributions to the RL community by advancing the field of credit assignment through innovative and effective approaches. These methods are not only sound from a theoretical standpoint but also effective in practice. We hope that the combination of the theoretical approach and a strong experimental component is a strength of this thesis. We believe that these contributions will have a lasting impact on the field and will inspire further research in credit assignment and beyond.

To just name a few, this PhD has greatly influence the following papers [1, 11, 3, 12, 8, 14, 10, 2, 13, 9].

Let's also mention that the same ideas mention in this thesis can also be used for exploration purposes. As mentioned at the beginning of this manuscript, credit assignment and exploration in RL are closely related. For example, exploration techniques are also impacted by stochastic and exogenous factors from the environment. This can dramatically impact exploration and therefore learning. Following the same ideas introduced in [6], we developed [5], published at ICML 2023. This paper uses similar methods to [6] but applied to exploration. In the curiosity-driven paradigm, the agent is rewarded for how much each realized outcome differs from their predicted outcome. But using predictive error as intrinsic motivation is fragile in stochastic environments, as the agent may become trapped by high-entropy areas of the state-action space. The key idea here is to learn representations of the future that capture precisely the unpredictable aspects of each outcome – which we use as additional input for predictions, such that intrinsic rewards only reflect the predictable aspects of world dynamics. By incorporating such hindsight representations into models to disentangle "noise" from "novelty" we get a simple and scalable generalization of curiosity that is robust to stochasticity.

As we can see here, the methods and ideas developed in this PhD are general and not only applicable to credit assignment. They are also making their ways into exploration techniques and hopefully this is just the beginning.

Chapter 8

Perspectives

Credit assignment is a fundamental challenge in reinforcement learning, and it remains a major obstacle to the development of RL agents that can learn to solve efficiently complex real-world problems. Recent advances in credit assignment research have led to significant progress, but there are still many open challenges and opportunities for future work.

One of the most exciting areas of future research is the application of credit assignment algorithms to real-world problems such as robotic, finance, healthcare and transportation. In all of these cases, the ability to accurately assign credit to individual actions is essential for learning effective policies.

Another important area of future research is the development of new benchmarks that reflect the credit assignment capacities of RL agents. Current benchmarks tend to focus on tasks where credit assignment is not a cornerstone to success, which makes it difficult to assess the credit assignment capacities of RL agents through their performance on those. New benchmarks are needed that challenge RL agents to solve problems with delayed rewards, multiple interacting agents and stochastic and exogenous factors.

Credit assignment algorithms can also be used to improve the interpretability of RL agents. By understanding the causal structure of the world and how credit is assigned to individual actions, it is possible to identify the key factors that contribute to the agent's success or failure. This information can be used to debug RL agents and to develop more human-friendly explanations of their behavior. This is essential for developing RL agents that can be deployed in real-world settings.

Finally, credit assignment can also be used to improve the transferability and adaptability of RL agents. By understanding the causal relationships between actions and rewards, RL agents can learn to transfer their knowledge to new problems and to adapt quickly to changes in their environment. This learned causal structure of the world could also revolutionized model-based RL by providing a meaningful yet tractable latent space that could serve as a basis for reasoning and planning.

Overall, the future of credit assignment research in RL is very promising. By developing new credit assignment algorithms and benchmarks, and by applying credit assignment to real-world problems, researchers can help to make RL an even more powerful and interpretable tool for solving complex problems.

Chapter 9

French Summary

Introduction

Cette thèse de doctorat se concentre sur l'apprentissage profond par renforcement, une discipline qui a connu de nombreux développements révolutionnaires ces dernières années. Les agents d'apprentissage par renforcement reposent sur des techniques d'attribution de crédit, visant à établir des corrélations entre les actions passées et les événements futurs, afin d'optimiser les décisions séquentielles. Cependant, les techniques d'attribution de crédit actuelles sont encore relativement rudimentaires et incapables de raisonnement inductif. La thèse se fixe donc pour objectif d'étudier et de formuler de nouvelles méthodes d'attribution de crédit dans le cadre de l'apprentissage par renforcement, avec pour ambition d'accélérer l'apprentissage, de mieux généraliser à plusieurs tâches et peut-être même de permettre l'émergence d'abstraction et de raisonnement.

Dans l'introduction, l'auteur souligne l'importance de l'apprentissage par renforcement et met en lumière deux problèmes fondamentaux : l'exploration et l'attribution de crédit. L'exploration désigne le processus par lequel un agent cherche à découvrir et à apprendre de son environnement, tandis que l'attribution de crédit consiste à évaluer l'influence des actions passées sur les résultats futurs. Ces deux concepts sont étroitement liés, l'exploration fournissant les données nécessaires à l'attribution de crédit. Cependant, l'attribution de crédit reste un domaine sous-exploré dans la recherche en apprentissage par renforcement.

Les techniques traditionnelles d'apprentissage par renforcement utilisent souvent le temps comme proxy pour l'attribution de crédit, mais cette approche est imparfaite dans des environnements caractérisés par des retours faibles, bruités ou retardés. De plus, les solutions actuelles d'apprentissage par renforcement souffrent souvent d'un manque de reproductibilité et d'une inefficacité d'échantillonnage, en partie à cause de capacités d'attribution de crédit limitées. Cela met en évidence la nécessité urgente de meilleures techniques pour aborder ce problème.

Motivations

Les motivations de la thèse soulignent l'importance de combler les lacunes dans la compréhension de l'attribution de crédit en apprentissage par renforcement. L'auteur exprime le désir de transformer l'attribution de crédit en un domaine de recherche distinct et actif, tout en contribuant à établir une communauté de

chercheurs et de praticiens engagés dans ce domaine. À travers des investigations rigoureuses et le développement de méthodologies innovantes, l'objectif est de favoriser une meilleure compréhension de l'attribution de crédit et de susciter un intérêt accru pour cette problématique au sein de la communauté scientifique.

En résumé, cette thèse vise à étudier et à développer de nouvelles méthodes d'attribution de crédit en apprentissage par renforcement, avec l'objectif ultime d'améliorer l'efficacité et la précision des algorithmes d'apprentissage par renforcement lors de prise de décisions séquentielles.

Contributions

Hindsight Credit Assignment

Dans le premier article, intitulé "Hindsight Credit Assignment" (HCA), les auteurs proposent une méthode qui tire parti d'information obtenue a posteriori pour améliorer les performances. Contrairement aux algorithmes traditionnels qui se basent uniquement le retour, HCA utilise des informations plus riches, a posteriori, pour attribuer le crédit. Cette approche se révèle plus efficace et robuste dans des tâches complexes d'attribution de crédit.

Counterfactual Credit Assignment

Le deuxième article, intitulé "Counterfactual Credit Assignment" (CCA), explore une approche qui utilise des fonctions valeurs conditionnées à des événements futurs. Cette méthode vise à séparer l'effet des actions de l'agent de l'effet des facteurs externes ainsi que des actions subséquentes. Les résultats montrent que l'approche CCA surpasse les algorithmes traditionnels sur des tâches complexes où l'attribution de crédit est difficile en raison de l'influence de facteurs externes.

Quantile Credit Assignment

Enfin, le troisième article présente une méthode appelée "Quantile Credit Assignment" (QCA), qui est conçue pour être plus robuste dans des environnements complexes. En estimant les quantiles de la distribution des retours pour chaque paire état-action, QCA offre une alternative efficace pour résoudre le problème de l'attribution de crédit dans des environnements imprévisibles et stochastiques.

Ces trois approches représentent des contributions significatives à la recherche en apprentissage par renforcement, offrant des solutions innovantes et efficaces pour améliorer l'attribution de crédit dans des environnements complexes et imprévisibles. Ces méthodes ont le potentiel d'avoir un impact important dans divers domaines, notamment la robotique, la finance et la santé, en permettant de mieux comprendre et réagir aux influences de leur environnement.

Perspectives

Dans l'ensemble, cette thèse de doctorat a apporté d'importantes contributions à la communauté de l'apprentissage par renforcement en faisant progresser le domaine de l'attribution de crédit grâce à des approches innovantes et efficaces. Ces

méthodes sont non seulement solides d'un point de vue théorique, mais également efficaces en pratique. L'auteur espère que la combinaison de l'approche théorique et d'une composante expérimentale solide est une des grandes forces de cette thèse. L'auteur espère que ces contributions auront un impact durable sur le domaine et inspireront de nouvelles recherches dans l'attribution de crédit et au-delà.

Notons également que les idées mentionnées dans cette thèse peuvent également être utilisées à des fins d'exploration. Comme mentionné au début de ce manuscrit, l'attribution de crédit et l'exploration sont étroitement liées. Par exemple, les techniques d'exploration sont également impactées par des facteurs stochastiques et exogènes de l'environnement. Cela peut avoir un impact important sur l'exploration et donc l'apprentissage. Suivant les mêmes idées introduites dans [6], les auteurs ont développé [5], publié à ICML 2023. Cet article utilise des méthodes similaires à [6] mais appliquées à l'exploration. L'idée clé ici est d'apprendre des représentations du futur qui capturent précisément les aspects imprévisibles de sorte que les récompenses intrinsèques ne reflètent que les aspects prévisibles de la dynamique du monde.

Bibliography

[1] Vyacheslav Alipov, Riley Simmons-Edler, Nikita Putintsev, Pavel Kalinin, and Dmitry Vetrov. Towards practical credit assignment for deep reinforcement learning. *arXiv preprint arXiv:2106.04499*, 2021.

[2] Veronica Chelu, Doina Precup, and Hado P van Hasselt. Forethought and hindsight in credit assignment. *Advances in Neural Information Processing Systems*, 33:2270–2281, 2020.

[3] Veronica Chelu, Diana Borsa, Doina Precup, and Hado van Hasselt. Selective credit assignment. *arXiv preprint arXiv:2202.09699*, 2022.

[4] Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

[5] Daniel Jarrett, Corentin Tallec, Florent Altché, Thomas Mesnard, Remi Munos, and Michal Valko. Curiosity in hindsight: Intrinsic exploration in stochastic environments. In *International Conference on Machine Learning*. PMLR, 2023.

[6] Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S Stepleton, Nicolas Heess, Arthur Guez, Eric Moulines, Marcus Hutter, Lars Buesing, and Remi Munos. Counterfactual credit assignment in model-free reinforcement learning. In *International Conference on Machine Learning*, pages 7654–7664. PMLR, 2021.

[7] Thomas Mesnard, Wenqi Chen, Alaa Saade, Yunhao Tang, Mark Rowland, Theophane Weber, Clare Lyle, Audrunas Gruslys, Michal Valko, Will Dabney, Georg Ostrovski, Eric Moulines, and Remi Munos. Quantile credit assignment. In *International Conference on Machine Learning*. PMLR, 2023.

[8] Alexander Meulemans, Simon Schug, Seijin Kobayashi, Nathaniel Daw, and Gregory Wayne. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. *arXiv preprint arXiv:2306.16803*, 2023.

[9] Hsiao-Ru Pan and Bernhard Schölkopf. Skill or luck? return decomposition via advantage functions. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

[10] Li Zhao Pushi Zhang, Guoqing Liu, Jiang Bian, Minlie Huang, Tao Qin, and Tie-Yan Liu. Independence-aware advantage estimation.

[11] Akash Velu, Skanda Vaidyanath, and Dilip Arumugam. Hindsight-dice: Stable credit assignment for deep reinforcement learning. *arXiv preprint arXiv:2307.11897*, 2023.

[12] Kenny Young. Hindsight network credit assignment: Efficient credit assignment in networks of discrete stochastic units. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8919–8926, 2022.

[13] Zhang Yudi, Du Yali, Huang Biwei, Wang Ziyan, Wang Jun, Meng Fang, and Pechenizkiy Mykola. Interpretable reward redistribution in reinforcement learning: A causal approach. 2023.

[14] Zeyu Zheng, Risto Vuorio, Richard Lewis, and Satinder Singh. Adaptive pairwise weights for temporal credit assignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9225–9232, 2022.

Titre : Attribution de crédit pour l'apprentissage par renforcement dans des réseaux profonds

Mots clés : Apprentissage par renforcement; Attribution de crédit; Raisonnement

Résumé : L'apprentissage profond par renforcement a été au cœur de nombreux résultats révolutionnaires en intelligence artificielle ces dernières années. Ces agents reposent sur des techniques d'attribution de crédit qui cherchent à établir des corrélations entre actions passées et événements futurs et utilisent ces corrélations pour devenir performants à une tâche. Ce problème est au cœur des limites actuelles de l'apprentissage par renforcement et les techniques d'attribution de crédit utilisées sont encore relativement rudimentaires et incapables de raisonnement inductif. Cette thèse se concentre donc sur l'étude et la formulation de nouvelles méthodes d'attributions de crédit dans le cadre de l'apprentissage par renforcement. De telles techniques pourraient permettre d'accélérer l'apprentissage, de mieux généraliser lorsqu'un agent est entraîné sur de multiples tâches, et peut-être même permettre l'émergence d'abstraction et de raisonnement.

Title : Credit Assignment in Deep Reinforcement Learning

Keywords : Deep Reinforcement Learning; Credit Assignment; Reasoning

Abstract : Deep reinforcement learning has been at the heart of many revolutionary results in artificial intelligence in the last few years. These agents are based on credit assignment techniques that try to establish correlations between past actions and future events and use these correlations to become effective in a given task. This problem is at the heart of the current limitations of deep reinforcement learning and credit assignment techniques used today remain relatively rudimentary and incapable of inductive reasoning. This thesis therefore focuses on the study and formulation of new credit assignment methods for deep reinforcement learning. Such techniques could speed up learning, make better generalization when agents are trained on multiple tasks, and perhaps even allow the emergence of abstraction and reasoning.