Beyond Average Reward in Markov Decision Processes

Alexandre Marthe
UMPA
ENS de Lyon
Lyon, France
alexandre.marthe@ens-lyon.fr

Aurelien Garivier
UMPA UMR 5669 and LIP UMR 5668
Univ. Lyon, ENS de Lyon
46 allée d'Italie F-69364 Lyon cedex 07, France
aurelien.garivier@ens-lyon.fr

Claire Vernade
University of Tuebingen
Tuebingen, Germany
claire.vernade@uni-tuebingen.de

Abstract

What are the functionals of the reward that can be computed and optimized exactly in Markov Decision Processes? In the finite-horizon, undiscounted setting, Dynamic Programming (DP) can only handle these operations efficiently for certain classes of statistics. We summarize the characterization of these classes for policy evaluation, and give a new answer for the planning problem. Interestingly, we prove that only generalized means can be optimized exactly. It is possible, however, to evaluate other functionals approximately using Distributional Reinforcement Learning. We prove error bounds on the resulting estimators and discuss the potential and limitations of this approach. These results contribute to advancing the theory of Markov Decision Processes by examining overall characteristics of the return, and particularly risk-conscious strategies.

1 Introduction

Reinforcement Learning (RL) has emerged as a flourishing field of study, delivering significant practical applications ranging from robot control and game solving to drug discovery or hardware design [Lazic et al., 2018, Popova et al., 2018, Volk et al., 2023, Mirhoseini et al., 2020]. The cornerstone of RL is the "return" value, a sum of successive rewards. Conventionally, the focus is on computing and optimizing its expected value on Markov Decision Process (MDP). The remarkable efficiency of MDPs comes from their ability to be solved through dynamic programming with the Bellman equations [Sutton and Barto, 2018, Szepesvári, 2010]. RL theory has seen considerable expansion, with a renewed interest for the consideration of more rich descriptions of a policy's behavior than the sole average return. At the other end of the spectrum, the so-called *Distributional Reinforcement Learning* (DistRL) approach aims at studying and optimizing the entire return distribution, leading to impressive practical results [Bellemare et al., 2017, Hessel et al., 2018][Wurman et al., 2022, Fawzi et al., 2022]. Between the expectation and the entire distribution, the efficient handling of other statistical functionals of the reward appears also particularly relevant for risk-sensitive contexts [Bernhard et al., 2019, Mowbray et al., 2022].

Despite recent progress, the full understanding of the abilities and limitations of DistRL to compute other functionals remains incomplete, with the underlying theory yet to be fully understood. Historically, the theory of RL has been established for discounted MDPs (e.g. [Sutton and Barto, 2018, Watkins and Dayan, 1992, Szepesvári, 2010, Bellemare et al., 2023] but recently more attention was drawn to the undiscounted, finite-horizon setting [Auer, 2002, Osband et al., 2013, Jin et al., 2018, Ghavamzadeh et al., 2020] , for which fundamental questions on the theory of MDPs remain open. In

16th European Workshop on Reinforcement Learning (EWRL 2023).

this paper, we explore policy evaluation, planning and exact learning algorithms for undiscounted MDPs for the optimization problem of general functionals of the reward. We explicitly delimit the possibilities offered by dynamic programming as well as DistRL. Our paper specifically addresses two questions:

(i) How accurately can we evaluate statistical functionals by using DistRL?

(ii) Which functionals can be exactly optimized through dynamic programming?

We first recall the fundamental results in dynamic programming and Distributional RL. Addressing question (i), we refer to Rowland et al. [2019]'s results on Bellman closedness and provide their adaptation to undiscounted MDPs. We then prove upper bounds on the approximation error of Policy Evaluation with DistRL and corroborate these bounds with practical experiments. For question (ii), we draw a connection between Bellman closedness and planning. We utilize the DistRL framework to identify two key properties held by *optimizable* functionals.

Our main contribution is a full characterization of the families of utilities that verify these two properties (Theorem 2). This result gives a comprehensive answer to question (ii) and closes an important open question in MDP theory. For the sake of completeness, we finally recall how these functionals can be optimized with reinforcement learning algorithms (when the parameters of the MDP are unknown).

2 Background

We introduce the classical RL framework in finite-horizon tabular Markov Decisions Processes (MDPs). We write $\mathscr{P}(\mathbb{R})$ the space of probability distributions on $\mathbb{R}$. A finite-horizon tabular MDP is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, p, R, H)$, where $\mathcal{X}$ is a finite state space, $\mathcal{A}$ is a finite action space, H is the horizon, for each $h \in [H]$, $p_h^a(x, \cdot)$ is a transition probability kernel and $R_h(x, a)$ is a reward random variable with distribution ρ_h. The parameters (p_h) and (R_h) define the *model* of dynamics. A deterministic policy on $\mathcal{M}$ is a sequence $\pi = (\pi_1, \dots, \pi_H)$ of functions $\pi_h : \mathcal{X} \to \mathcal{A}$.

Reinforcement Learning traditionally focuses on learning policies optimizing the expected return. For a given policy π, the Q-function maps a state-action pair to its expected return under π:

$$Q_h^\pi(x, a) = \mathbb{E}_{\rho_h}\left[R_h(x, a)\right] + \sum_{s'} p_h^a(x, x') Q_{h+1}^\pi(x', \pi_{h+1}(x')), \qquad Q_{H+1}^\pi(x, a) = 0\,. \quad (1)$$

When the model is known, the Q-function of a policy π can be computed by doing a backward recursion, also called dynamic programming. This is referred to as *Policy Evaluation*. Similarly, an optimal policy can be found by solving the optimal Bellman equation:

$$Q_h^*(x, a) = \mathbb{E}_{\rho_h}\left[R_h(x, a)\right] + \sum_{x'} p_h^a(x, x') \max_{a'} Q_{h+1}^*(x', a'), \qquad Q_{H+1}^*(x, a) = 0\,. \quad (2)$$

Solving this equation when the model is known is also called *Planning*. When it is unknown, *reinforcement learning* aims at finding the optimal policy from sample runs of the MDP. But evaluating and optimizing the *expectation* of the return in the definition of the Q-function above is just one choice of statistical functional. We now introduce Distributional RL and then discuss other statistical functionals that generalize the expected setting discussed so far.

2.1 Distributional RL

Distributional RL (DistRL) refers to the approach that tracks not just a statistic of the return for each state but its *entire distribution*. We introduce here the most important basic concepts and refer the reader to the recent comprehensive survey by Bellemare et al. [2023] for more details. The main idea is to use the full distributions to estimate and optimize various metrics over the returns ranging from the mere expectation [Bellemare et al., 2017] to more complex metrics [Rowland et al., 2019, Dabney et al., 2018a, Liang and Luo, 2022].

At state-action (x, a), let $Z_h^\pi(x, a)$ denote the future sum of rewards when following policy π and starting at step h, also called *return*. It verifies the simple recursive formula $Z_h^\pi(x, a) = R_h(x, a) +$

$Z^\pi_{h+1}(X', \pi_{h+1}(X'))$ where $X' \sim p^a_{h+1}(x, \cdot)$. Its distribution is $\eta = (\eta^{(x,a)}_{\pi,h})_{(x,a,h)\in\mathcal{X}\times\mathcal{A}\times[H]}$ and is often referred to as the *Q-value distribution*. One can easily derive the recursive law of the return as a convolution: for any two measures $\nu_1, \nu_2 \in \mathscr{P}(\mathbb{R})$, we denote their convolution by $\nu_1 * \nu_2(t) = \int_{\mathbb{R}} \nu_1(\tau)\nu_2(t-\tau)d\tau$. For any two independent random variables X and Y, the distribution of the sum $Z = X + Y$ is the convolution of their distributions: $\nu_Z = \nu_X * \nu_Y$. Thus, the law of $Z^\pi_h(x, a)$ is

$$\forall x, a, h, \quad \eta^{(x,a)}_{\pi,h} = \rho^{(x,a)}_h * \sum_{x'} p^a_h(x, x')\eta^{(x,\pi_{h+1}(x))}_{\pi,h+1} \ . \tag{3}$$

This equation is a distributional equivalent to Eq. (1) and thus defines a *distributional Bellman operator* $\eta_{\pi,h} = \mathcal{T}^\pi_h \eta_{\pi,h+1}$.

Obviously, from a practical point of view, distributions form a non-parametric family that is not computationally tractable. It is necessary to choose a parametric (thus incomplete) family to represent them. Even the restriction to discrete reward distributions is not tractable, since the number of atoms in the distributions may grow exponentially with the number of steps[1] [Achab and Neu, 2021]: approximations are unavoidable. The most natural solution is to use projections of the obtained distribution on the parametric family, at each step of the Bellman operator. This process is called *parametrization*. The practical equivalent to Eq. (1) in DistRL hence writes

$$\forall x, a, h, \quad \hat{\eta}^{(x,a)}_{\pi,h} = \Pi \left(\rho_h(x, a) * \sum_{x'} p^a_h(x, x')\hat{\eta}^{(x',\pi_{h+1}(x'))}_{\pi,h+1} \right) \ , \tag{4}$$

where Π is the projector operator on the parametric family. The full policy evaluation algorithm in DistRL is summarized in Alg.1.

Algorithm 1 Policy Evaluation (Dynamic Programming) for Distributional RL

Input: model p, reward distributions ρ_h, policy π to evaluated, Π projection.
Data: $\eta \in \mathbb{R}^{H|\mathcal{X}||\mathcal{A}|N}$
$\forall x, a \in \mathcal{X} \times \mathcal{A}, \quad \eta^{(x,a)}_H = \delta_0$
for $h = H - 1 \to 0$ **do**
  $\eta^{(x,a)}_h = \rho_h(x, a) * \sum_{x'} p^a_h(x, x')\eta^{(x',\pi_{h+1}(x'))}_{h+1} \quad \forall x, a \in \mathcal{X} \times \mathcal{A}$
  $\eta^{(x,a)}_h = \Pi\left(\eta^{(x,a)}_h\right) \quad \forall x, a \in \mathcal{X} \times \mathcal{A}$
end for
Output: $\eta^{(x,a)}_h \forall x, a, h$

Distribution Parametrization The most commonly used parametrization is the so-called *quantile projection*. It put Diracs (atoms) with fixed weights at locations that correspond to the quantiles of the source distribution. One main benefit is that it does not require a previous knowledge of the support of the distribution, and allows for unbounded distributions. The quantile projection is defined as

$$\Pi_Q \mu = \frac{1}{N} \sum_{i=0}^{N-1} \delta_{z_i}, \quad \text{with } (z_i)_i \text{ chosen such as } F_\mu(z_i) = \frac{2i+1}{2N} \, , \tag{5}$$

which corresponds to a minimal W_1 distance: $\Pi_Q \mu \in \arg\min_{\hat{\mu}=\sum_i \delta_i/N} W_1(\mu, \hat{\mu})$, where $W_1(.,.)$ is the Wasserstein distance defined for any distributions ν_1, ν_2 as $W_1(\nu_1, \nu_2) = \int_0^1 \left|F^{-1}_{\nu_1}(u) - F^{-1}_{\nu_2}(u)\right| \, \mathrm{d}u$. Note that this parametrization might admit several solutions and thus the projection may not be unique. For simplicity, we overload the notation to $\Pi_Q \eta = (\Pi_Q \eta^{(x,a)})_{(x,a)\in\mathcal{X}\times\mathcal{A}}$

For a Q-value distribution η with support of length Δ_η, and parametrization of resolution N, Rowland et al. [2019] prove that the projection error is bounded by

$$\sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} W_1(\Pi_Q \eta^{(x,a)}, \eta^{(x,a)}) \leq \frac{\Delta_\eta}{2N} \ . \tag{6}$$

[1]The support of the return (sum of the rewards) is incremented at each step by a number of atoms that depend on the current support.

In Section 3, we extend this result to the full iterative Policy Evaluation process and bound the error on the returned statistical functional in the finite-horizon setting. Note that other parametrizations exist but are less practical. For completeness, we discuss the Categorical Projection [Bellemare et al., 2017][Rowland et al., 2018] in Appendix B.

2.2 Beyond expected rewards

The expected value is an important functional of a probability distribution, but it is not the only one of interest in decision theory – especially when a control of the risk is important. We discuss two concepts that have received considerable attention: *utilities*, defined as expected values of functions of the return, and *distorted means* which place emphasis on certain quantiles.

Utilities are of the form $\mathbb{E}[f(Z)]$, or $\int f \, \mathrm{d}\nu$, where Z is the return of distribution ν and f is an increasing function. For instance, when f is a power function, we obtain the different moments of the return. The case of exponential functions plays a particularly important role: the resulting utility is referred to as *exponential utility*, *exponential risk measure*, or *generalized mean* according to the context:

$$U_{\exp}(\nu) = \frac{1}{\lambda} \log \mathbb{E}\big[\exp(\lambda X)\big] \quad X \sim \nu \text{ and } \lambda \in \mathbb{R} \,. \tag{7}$$

This family of utilities has a variety of applications in finance, economics, and decision making under uncertainty[Föllmer and Schied, 2016]. They can be considered as a risk-aware generalization of the expectation, with benefits such as accommodating a wide range of behaviors [Shen et al., 2014] from risk-seeking when $\lambda > 0$, to risk-averse when $\lambda < 0$ (the limit $\lambda \to 0$ is exactly the expectation). To fix ideas, $U_{\exp}\big(\mathcal{N}(\mu, \sigma^2)\big) = \mu + \lambda\sigma^2$: each λ captures a certain quantile of the Gaussian distribution.

Importantly, utilities satisfy the axiom of *independence* of the distribution: $U_f(\nu_1) \geq U_f(\nu_2)$, then for any new ν_3 and mixture coefficient p, a return of distribution $U_f(p\nu_1 + (1-p)\nu_3) \geq U_f(p\nu_2 + (1-p)\nu_3)$ [von Neumann et al., 1944].

Distorted means, on the other hand, involve taking the mean of a random variable, but with a different weighting scheme [Dabney et al., 2018a]. The goal is to place more emphasis on certain quantiles, which can be achieved by considering the quantile function F^{-1} of the random variable and a continuous increasing function $\beta : [0,1] \to [0,1]$. By applying β to a uniform variable τ on $[0,1]$ and evaluating F^{-1} at the resulting value $\beta(\tau)$, we obtain a new random variable that takes the same values as the original variable, but with different probabilities. The distorted mean is then calculated as the mean of this new random variable, given by the formula $\int \beta'(\tau) F^{-1}(\tau) d\tau$. If β is the identity function, the result is the classical mean. When β is $\tau \mapsto \min(\tau/\alpha, 1)$, we get the α-Conditional Value at Risk (CVaR(α)) of the return, a risk measure widely used in risk evaluation [Rockafellar et al., 2000]. When β is of the form $\tau \mapsto \frac{\tau^\alpha}{(\tau^\alpha + (1-\tau)^\alpha)^{1/\alpha}}$, we obtain the Cumulative Probability Weighting, the value of interest in cumulative prospect theory [Tversky and Kahneman, 1992].

3 Policy Evaluation

The theory of MDPs is particularly developed for estimating and optimizing the mean of the return of a policy. But other values associated to the return can be computed the same way, by dynamic programming. This includes for instance the variance of the return, or more generally, any moment of order $p \geq 2$, as was already noticed in the 1980's [Sobel, 1982]. Recently, Rowland et al. [2019] showed that for utilities in discounted MDPs, this is essentially all that can be done. More precisely, they introduce the notion of *Bellman closedness* (recalled below for completeness) that characterizes a finite set of statistics that can efficiently be computed by dynamic programming.

Definition 1 (Bellman closedness)**.** *A set of statistical functionals $\{s_1, \dots s_K\}$ is said to be* Bellman closed *if for each $(x,a) \in \mathcal{X} \times \mathcal{A}$, the statistics $s_{1:K}(\eta_h^{(x,a)})$ can be expressed in closed form in terms of the random variable $R_h(x,a)$, of the probabilities $p_h^a(x,\cdot)$ and of the statistics of the next step $(s_{1:K}(\eta_{h+1}^{(x',a')})$, independently of the MDP.*

Importantly, in the undiscounted setting, Rowland et al. [2019](Appendix B, Theorem 4.3) show that the only families of utilities that are Bellman closed are of the form $\{x \mapsto x^\ell \exp(\lambda x) | 0 \leq \ell \leq L\}$

for some $L < \infty$. Thus, all utilities and statistics of the form of (or linear combinations of) moments and exponential utilities can easily be computed by classic linear dynamic programming and do not require distributional RL (see Appendix A.3).

Some important metrics such as the CVaR or the quantiles do not belong to any Bellman-closed set and hence cannot be easily computed. For this kind of function of the return, the knowledge of the transitions and the values in following steps is insufficient to compute the value on a specific step. In general, it requires the knowledge of the whole distribution of each reward in each state. Hence, techniques developed in distributional RL come in handy: for a choice of parametrization, one can use the projected dynamic programming step Eq. (4) to propagate a finite set of values along the MDP and approximate the distribution of the return. In the episodic setting, following the line of Rowland et al. [2019] (see Eq.(6)), we prove that the Wasserstein distance error between the exact and approximate distribution of the Q-values of a policy is bounded.

Proposition 1. *Let π be a policy and η_π the associated Q-value distributions. Assume the return is bounded on a interval of length $\Delta_\eta \leq H\Delta_R$, where Δ_R is the support size of the reward distribution. Let $\hat{\eta}_\pi$ be the Q-value distributions obtained by dynamic programming (Algorithm 1) using the quantile projection Π_Q with resolution N. Then,*

$$\sup_{(x,a,h)\in(\mathcal{X},\mathcal{A},[H])} W_1(\hat{\eta}_{\pi,h}^{(x,a)}, \eta_{\pi,h}^{(x,a)}) \leq H\frac{\Delta_\eta}{2N} \leq H^2\frac{\Delta_R}{2N} \ .$$

This result shows that the loss of information due to the parametrization only grows quadraticly with the horizon. The proof consists of summing the projection bound in (6) at each projection step combined together with the non-expansion property of the Bellman operator [Bellemare et al., 2017]

The key question is then to understand how such error translates into our estimation problem when we apply the function of interest to the approximate distribution. We provide a first bound on this error for the family of statistics that are either utilities or distorted means.

First, we prove that the utility is Lipschitz on the set of return distributions.

Lemma 1. *Let s be either an utility or a distorted mean and let L be the Lipschitz coefficient of its characteristic function. Let ν_1, ν_2 be return distributions. Then:*

$$|s(\nu_1) - s(\nu_2)| \leq LW_1(\nu_1, \nu_2) \ .$$

Both family of functionals are treated separately, but lays a similar bound. The utility bound is the direct application of the Kantorovitch-Rubenstein duality, while the distorted mean one is a direct majoration in the integral. Again, the details are provided in the Appendix.

This property allows us to prove a maximal upper bound on the estimation error for those two families.

Theorem 1. *Let π be a policy. Let η_π be the Q-value return distribution associated to π with the return bounded on a interval of length $\Delta_\eta \leq H\Delta_R$ where Δ_R is the support size of the reward distribution. Let $\hat{\eta}_\pi$ be the approximated return distribution computed with Algorithm 1, for the projection Π_Q with resolution N. Let s be either an utility or a distorted mean, and L the Lipschitz coefficient of its characteristic function. Then:*

$$\sup_{x,a,h} |s(\hat{\eta}_{\pi,h}^{(x,a)}) - s(\eta_{\pi,h}^{(x,a)})| \leq LH\frac{\Delta_\eta}{2N} \leq LH^2\frac{\Delta_R}{2N} \ .$$

Note that depending on the choice of utilities, the Lipschitz coefficient L may also depend on H and ΔR. For instance, in a stationary MDP, the Lipschitz constant of the exponential utility depends exponentially on Δ_η. For the CVaR(α), however, L is constant and only depends on $\alpha \in (0, 1)$.

Experiment: empirical validation of the bounds on a simple MDP We consider a simple Chain MDP environment of length $H = 70$ equal to the horizon (see Figure 1 (right)) [Rowland et al., 2019], with a single action leading to the same discrete reward distribution for every step. We designed a simple discrete reward distribution with 2 atoms in $\{0, 1\}$ (see Figure 2 (left)) to ease computations as the number of atoms for the return only grows linearly[2] with the number of steps so the exact distribution can be computed easily.

[2]At round $h \in [H]$, the support of the return is $\{0, 1, ..., h\}$, so h atoms.

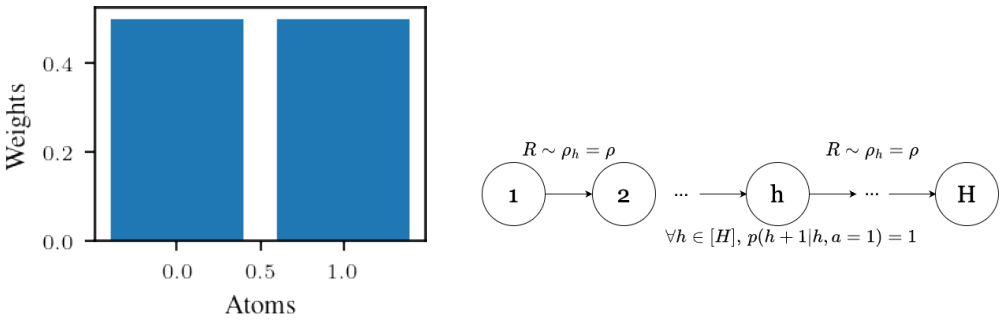

Figure 1: Left: Discrete reward distribution. Right: A Chain MDP of length H with deterministic transition and identical reward distribution for each state.

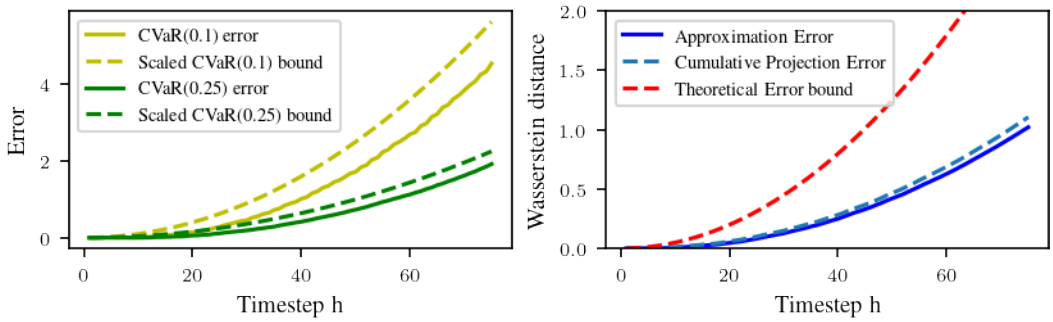

Figure 2: Left: Validation of Theorem 1 on CVaR(α) together with the scaled upper bound (see main text for discussion): the quadratic dependence in H is verified. Right: Validation of Proposition 1: The cumulative projection error (dashed blue) is the sum of the projection errors at every time step, and matches the true approximation error (solid blue). The theoretical upper bound (dashed red) differs only by a factor 2.

We compare the distributions obtained with exact dynamic programming and the approximate distribution obtained by Alg 1, with a quantile projection with resolution $N = 1000$. Note that even at early stages, when the true distribution has less atoms than the resolution, the exact and approximate distributions differ due to the weights of the atoms in the quantile projection. Figure 2 (Right) reports the Wasserstein distance between the two distributions: the cumulative projection approximation error (dashed blue), the true error between the current exact and approximate distributions (solid blue) and the theoretical bound (red). Fundamentally, the proof of Prop. 1 upper bounds the distance between distributions by the cumulative projection error so we plot this quantity to help validating it.

We also empirically validate Theorem 1 by computing the CVaR(α)) for $\alpha \in \{0.1, 0.25\}$), corresponding respectively to utility functionals with Lipschitz constants $L = \{10, 4\}$. We compute these statistics for both distributions and report the maximal error together with the theoretical bound, re-scaled[3] by a factor 5. Figure 2 (Left) shows an impressive correspondence of the theory and the empirical results despite a constant multiplicative gap.

4 Planning

Planning refers to the problem of returning a policy that optimizes our objective for a given model and reward function (or distribution in DistRL). It shares with policy evaluation the property to be grounded on a Bellman equation: see Eq. (2) for the classical expected return, which leads to efficient computations by dynamic programming.

For other statistical functionals of the cumulated reward, however, can the optimal policy be computed efficiently? The main result of this section fully characterizes the family of functionals that can be

[3]Scaling by a constant factor allows us to show the corresponding quadratic trends.

exactly and efficiently optimized by dynamic programming. We then explore how DistRL can be used beyond exact planning.

4.1 Exact Planning

Algorithm 2 Pseudo-Algorithm: Exact Planning with Distributional RL

1: **Input:** model p, reward R, statistical functional s
2: **Data:** $\eta \in \mathbb{R}^{H|\mathcal{X}||\mathcal{A}|N}, \nu \in \mathbb{R}^{H|\mathcal{X}|N}$
3: $\forall x \in \mathcal{X}, \quad \nu^x_{H+1} = \delta_0$
4: **for** $h = H \to 1$ **do**
5: $\quad \eta_h^{(x,a)} = \rho_h^{(x,a)} * \sum_{x'} p_h^a(x,x')\nu_{h+1}^{x'} \quad \forall x, a \in \mathcal{X} \times \mathcal{A}$
6: $\quad \nu_h^x = \eta_h^{(x,a^*)}, \quad a^* \in \arg\max_a s(\eta_h^{(x,a)}) \quad \forall x \in \mathcal{X}$
7: **end for**
8: **Output:** $\eta_h^{(x,a)}\ \forall x, a, h$

In the previous section, we recalled that only exponential and linear utilities are Bellman closed, which means that they satisfy Bellman equations and can be efficiently computed by dynamic programming, for a given policy. In fact, for the exponential utilities, it has been shown that the planning problem can also be solved efficiently [Howard and Matheson, 1972]:

$$Q_h^\lambda(x,a) = U^\lambda_{\exp}(R_h(x,a)) + \frac{1}{\lambda}\log\left[\sum_{s'} p_h^a(x,x')\exp\left(\lambda \max_{a'} Q^\lambda_{h+1}(x',a')\right)\right], \quad Q^\lambda_{H+1}(x,a) = 0\ . \tag{8}$$

The question remains open, however, for non-utility functionals (e.g. quantiles) or non Bellman-closed utilities. In order to fully address it, we consider the most general framework, DistRL, and recall in the Pseudo[4]-Algorithm 2 the theoretical dynamic programming equations for any statistical functional s. DistRL offers the most comprehensive, or 'lossless', approach, so if a statistical functional cannot be optimized with Alg. 2, then there cannot exist a Bellman Operator to perform exact planning. We formalize this idea with the new concept of *Bellman Optimizability*:

Definition 2 (Bellman Optimizability)**.** *A statistical functional s is called* Bellman optimizable *if the Pseudo-Algorithm 2 outputs an optimal return distribution $\eta = \eta^*$ that verifies:*

$$\forall x, a, h, \quad s(\eta_h^{*,(x,a)}) = \sup_\pi s(\eta_{\pi,h}^{(x,a)})\ . \tag{9}$$

We can now state our main results that characterizes all the *Bellman optimizable* statistical functionals. First, we prove that such a functional must satisfy two important properties.

Lemma 2. *A* Bellman optimizable *functional s satifies the two following properties:*

- *Independence Property: If $\nu_1, \nu_2 \in \mathscr{P}(\mathbb{R})$ are such that $s(\nu_1) \geq s(\nu_2)$, then*

$$\forall \nu_3 \in \mathscr{P}(\mathbb{R}), \forall \lambda \in [0,1], \quad s(\lambda\nu_1 + (1-\lambda)\nu_3) \geq s(\lambda\nu_2 + (1-\lambda)\nu_3))\ .$$

- *Translation Property: Let τ_c denote the translation on the set of distributions: $\tau_c\delta_x = \delta_{x+c}$. If $\nu_1, \nu_2 \in \mathscr{P}(\mathbb{R})$ are such that $s(\nu_1) \geq s(\nu_2)$, then*

$$\forall c \in \mathbb{R}, \quad s(\tau_c\nu_1) \geq s(\tau_c\nu_2)\ .$$

Indeed, the expectation and the exponential utility both satisfy these properties (see Appendix A.2). Each property is implied by an aspect of the Distributional Bellman Equation (Alg. 2, line 5) and the proof (in Appendix D) unveils these important consequences of the recursion identities. Fundamentally, they follow from the Markovian nature of policies optimized this way. Specifically, as the Distributional Bellman operator implies a translation of the support of future return, we must have that the optimal policy at the next state does not depend on the current state.

[4]This theoretical algorithm handles the full distribution of the return at each step, which cannot de done in practice.

The Independence property states that, for Bellman optimizable functionals, the value of each next state should not depend on that of any other value in the convex combination in the rightmost term of the convolution. In turn, The Translation property is associated to the leftmost term, the reward, and it imposes that, for Bellman optimizable functionals, the decision on the best action is independent of the previously accumulated reward.

As mentioned in Section 2, utilities are defined to satisfy the Independence property. Interestingly, the Von Neumann-Morgenstein theorem establishes that if a statistical functional s satisfies the Independence property, then there exists a corresponding utility function U such that $\forall \nu_1, \nu_2 \in \mathscr{P}(\mathbb{R})$, we have $s(\nu_1) > s(\nu_2) \iff U(\nu_1) > U(\nu_2)$ [von Neumann et al., 1944].

This result directly narrows down the family of Bellman optimizable functionals to utilities. Indeed, although other functionals might potentially be optimized using the Bellman equation, addressing the problem on utilities is adequate to characterize all possible behaviors. For instance, moment-optimal policies that can be found through dynamic programming, can also be found by optimizing an exponential utility function. The next task is therefore to identify all the utilities that satisfy the second property. We demonstrate that, apart from the mean and exponential utilities, no other functional satisfies this property.

Theorem 2. *Let ρ be a return distribution. The only Bellman Optimizable statistical functionals of the cumulated return are exponential utilities $U_{\exp}(\rho) = \frac{1}{\lambda} \log \mathbb{E}_\rho [\exp(\lambda R)]$ for $\lambda \in \mathbb{R}$, with the special case of the expectation $\mathbb{E}_\rho [R]$ when $\lambda = 0$.*

The full proof is provided in Appendix D. It essentially shows that the only utilities that verify the Translation Property are either linear or exponential. The result is obtained by noticing that this property implies strong regularity constraints on the family of functions. We then exhibit the only family of solutions that verify satisfy these constraints.

We make a few important observations. First, our result shows that algorithms using Bellman updates to optimize any functionals other than the exponential utility cannot guarantee optimality. Conversely, for any other family of utility, there is no point searching for Bellman equations for exact planning because of they are not Bellman Closed.

Most importantly, while in theory, DistRL provides the most general framework for optimizing policies via dynamic programming, our result shows that in fact, the only utilities that can be exactly and efficiently optimized do not require to resort to DistRL. This certainly does not question the very purpose of DistRL, which has been shown to play important roles in practice to regularize or stabilize policies and to perform deeper exploration [Bellemare et al., 2017, Hessel et al., 2018]. Some advantages of learning the distribution lies in the enhanced *robustness* offered in the richer information learned [Rowland et al., 2023], particularly when utilizing neural networks for function approximation [Dabney et al., 2018b, Barth-Maron et al., 2018, Lyle et al., 2019].

5 Q-Learning Exponential Utilities

Algorithm 3 Q-Learning for Linear and Exponential Utilities

Input: $(\alpha_t)_{t\in\mathbb{N}}$, transition and reward generator. $Q_h(x,a) \leftarrow H$, $\forall (x,a,h) \in \mathcal{X} \times \mathcal{A} \times [H]$
Utilities: Linear ($Z \mapsto \lambda Z + b$) or Exponential ($Z \mapsto \log(\mathbb{E}\exp(\lambda Z))/\lambda$)
for episode $K = 1, \dots, K$ **do**
    Observe $x_1 \in \mathcal{X}$
    for step $h = 1, \dots, H$ **do**
        Choose action $a_h \in \arg\max_{a\in\mathcal{A}} Q_h(x_h, a)$
        Observe reward r_h and transition x_{h+1} and update for chosen objective:
        Linear Util.: $Q_h(x_h,a_h) \leftarrow (1-\alpha_k) Q_h(x_h,a_h) + \alpha_k[\lambda(r_h + \max_{a'} Q_{h+1}(x_{h+1},a')) + b]$
        Exponential Util.: $Q_h(x_h,a_h) \leftarrow \frac{1}{\lambda} \log \left[(1-\alpha_k) e^{\lambda \cdot Q_h(x_h,a_h)} + \alpha_k e^{\lambda[r_h + \max_{a'} Q_h(x_{h+1},a')]} \right]$
    end for
end for
Output: $Q_h(x,a) \forall x, a$

The previous sections address the situation when the model, i.e. the reward and transition functions, is perfectly known. Yet in most practical situations, those are either approximated or learnt[5]. This section completes the theory by showing how to learn utilities online (linear or exponential) with Q-learning, a celebrated model-free RL algorithm [Watkins and Dayan, 1992]. We provide the pseudo-code with the according utility-based updates in Alg. 3. The linear utility updates (line 8) only slightly differ from the classical ones for expected return optimization, which have been shown to lead to the optimal value asymptotically [Watkins and Dayan, 1992]. The exponential utility updates (line 9) are quite similar in spirit and have been proposed by Borkar [2002, 2010] together with a convergence proof.

6 Discussions and Related Work

The Discounted Framework We focused in this article on undiscounted MDPs, and it is important to note that the results differ for discounted scenarios. The crucial difference is that the family of exponential utilities no longer retains Bellman Closed or Bellman Optimizable properties due to the introduction of the discount factor γ [Rowland et al., 2019]. When it comes to Bellman Optimization, the necessary translation property becomes an affine property : $\forall c, \gamma,\ s(\tau_c^\gamma \nu_1) > s(\tau_c^\gamma \nu_2)$ where τ_c^γ is the affine operator such that $\tau_c^\gamma \delta_x = \delta_{\gamma x + c}$. This property is not upheld by the exponential utility. Nonetheless, there exists a method to optimize the exponential utility through dynamic programming in discounted MDPs [Chung and Sobel, 1987]. This approach requires modifying the functional to optimize at each step (the step h is optimized with the utility $x \mapsto \exp(\gamma^{-h} \lambda\, x)$), but it also implies a loss of policy stationarity, property usually obtained in dynamic programming for discounted finite-horizon MDPs [Sutton and Barto, 2018].

Utilizing functionals to optimize expected return. DistRL has also been used in Deep Reinforcement learning to optimize non-Bellman-optimizable functionals such as distorted means[Ma et al., 2020, Dabney et al., 2018a]. While, as we proved so, such algorithms cannot lead to optimal policies in terms of these functionals, experiments show that in some contexts they can lead to better expected return and faster convergence in practice. The change of functional can be interpreted as a change in the exploration process, and the resulting risk-sensitive behaviors seem to be relevant in adequate environments.

Dynamic programming for the optimization of other functionals To optimize other statistical functionals such as CVaR and other utilities such as moments with Dynamic Programming, Bäuerle and Ott [2011] and Bäuerle and Rieder [2014] propose to extend the state space of the original MDP to $\mathcal{X}' = \mathcal{X} \times \mathbb{R}$. By theoretically adding a continuous dimension to store the current cumulative rewards. This idea does of course not contradict our results, and the resulting algorithms remain empirically much more expensive.

7 Conclusion

Our work closes an important open problem in the theory of MDP: we exactly characterize the families of statistical functionals that can be evaluated and optimized by dynamic programming. We also put into perspective the DistRL framework: the only functionals of the return that can be optimized with DistRL can actually be handled exactly by dynamic programming. Its benefit lie elsewhere, and notably in the improved stability of behavioral properties it allows. We believe that, by narrowing down the avenues to explain its empirical successes, our work can contribute to clarify the further research to conduct in the theory of DistRL.

[5]Either explicitly (model-based RL) or implicitly (model-free RL, considered here).

Acknowledgements

Alexandre Marthe and Aurélien Garivier acknowledge the support of the Project IDEXLYON of the University of Lyon, in the framework of the Programme Investissements d'Avenir (ANR-16-IDEX-0005), and Chaire SeqALO (ANR-20-CHIA-0020-01).
Claire Vernade is funded by the Deutsche Forschungsgemeinschaft (DFG) under both the project 468806714 of the Emmy Noether Programme and under Germany's Excellence Strategy – EXC number 2064/1 – Project number 390727645. Claire Vernade also thanks the international Max Planck Research School for Intelligent Systems (IMPRS-IS).

References

M. Achab and G. Neu. Robustness and risk management via distributional dynamic programming, Dec. 2021. URL http://arxiv.org/abs/2112.15430. arXiv:2112.15430 [cs, math].

P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, and T. Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017.

M. G. Bellemare, W. Dabney, and M. Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023. http://www.distributional-rl.org.

J. Bernhard, S. Pollok, and A. Knoll. Addressing inherent uncertainty: Risk-sensitive behavior generation for automated driving using distributional reinforcement learning. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2148–2155, 2019. doi: 10.1109/IVS.2019.8813791.

V. S. Borkar. Q-learning for risk-sensitive control. *Mathematics of operations research*, 27(2): 294–311, 2002.

V. S. Borkar. Learning algorithms for risk-sensitive control. In *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems–MTNS*, 2010.

N. Bäuerle and J. Ott. Markov Decision Processes with Average-Value-at-Risk criteria. *Mathematical Methods of Operations Research*, 74(3):361–379, Dec. 2011. ISSN 1432-2994, 1432-5217. doi: 10.1007/s00186-011-0367-0. URL http://link.springer.com/10.1007/s00186-011-0367-0.

N. Bäuerle and U. Rieder. More Risk-Sensitive Markov Decision Processes. *Mathematics of Operations Research*, 39(1):105–120, Feb. 2014. ISSN 0364-765X. doi: 10.1287/moor.2013.0601. URL https://pubsonline.informs.org/doi/abs/10.1287/moor.2013.0601. Publisher: INFORMS.

K.-J. Chung and M. J. Sobel. Discounted MDP's: Distribution Functions and Exponential Utility Maximization. *SIAM Journal on Control and Optimization*, 25(1):49–62, Jan. 1987. ISSN 0363-0129, 1095-7138. doi: 10.1137/0325004. URL http://epubs.siam.org/doi/10.1137/0325004.

W. Dabney, G. Ostrovski, D. Silver, and R. Munos. Implicit Quantile Networks for Distributional Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1096–1105. PMLR, July 2018a. URL https://proceedings.mlr.press/v80/dabney18a.html. ISSN: 2640-3498.

W. Dabney, M. Rowland, M. Bellemare, and R. Munos. Distributional Reinforcement Learning With Quantile Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018b. ISSN 2374-3468. URL https://ojs.aaai.org/index.php/AAAI/article/view/11791. Number: 1.

A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R. Ruiz, J. Schrittwieser, G. Swirszcz, D. Silver, D. Hassabis, and P. Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, Oct. 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-05172-4. URL https://www.nature.com/articles/s41586-022%20-05172-4. Number: 7930 Publisher: Nature Publishing Group.

H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. De Gruyter, Berlin, Boston, 2016. ISBN 9783110463453. doi: doi:10.1515/9783110463453. URL https://doi.org/10.1515/9783110463453.

M. Ghavamzadeh, A. Lazaric, and M. Pirotta. Exploration in reinforcement learning. Tutorial at AAAI'20, 2020. URL https://rlgammazero.github.io/.

M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining Improvements in Deep Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11796. URL https://ojs.aaai.org/index.php/AAAI/article/view/11796.

R. A. Howard and J. E. Matheson. Risk-Sensitive Markov Decision Processes. *Management Science*, 18(7):356–369, Mar. 1972.

C. Jin, Z. Allen-Zhu, S. Bubeck, and M. I. Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

N. Lazic, C. Boutilier, T. Lu, E. Wong, B. Roy, M. Ryu, and G. Imwalle. Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems*, 31, 2018.

H. Liang and Z.-Q. Luo. Bridging Distributional and Risk-sensitive Reinforcement Learning with Provable Regret Bounds, Oct. 2022. URL http://arxiv.org/abs/2210.14051. arXiv:2210.14051 [cs, stat] version: 1.

C. Lyle, M. G. Bellemare, and P. S. Castro. A Comparative Analysis of Expected and Distributional Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 4504–4511, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33014504. URL https://ojs.aaai.org/index.php/AAAI/article/view/4365. Number: 01.

X. Ma, L. Xia, Z. Zhou, J. Yang, and Q. Zhao. DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning, June 2020. URL http://arxiv.org/abs/2004.14547. arXiv:2004.14547 [cs].

A. Mirhoseini, A. Goldie, M. Yazgan, J. Jiang, E. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, S. Bae, et al. Chip placement with deep reinforcement learning. *arXiv preprint arXiv:2004.10746*, 2020.

M. Mowbray, D. Zhang, and E. A. D. R. Chanona. Distributional reinforcement learning for scheduling of chemical production processes, 2022.

I. Osband, D. Russo, and B. Van Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 26, 2013.

M. Popova, O. Isayev, and A. Tropsha. Deep reinforcement learning for de novo drug design. *Science advances*, 4(7):eaap7885, 2018.

R. T. Rockafellar, S. Uryasev, et al. Optimization of conditional value-at-risk. *Journal of risk*, 2: 21–42, 2000.

M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh. An Analysis of Categorical Distributional Reinforcement Learning. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 29–37. PMLR, Mar. 2018. URL https://proceedings.mlr.press/v84/rowland18a.html. ISSN: 2640-3498.

M. Rowland, R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, and W. Dabney. Statistics and Samples in Distributional Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5528–5536. PMLR, May 2019. URL https://proceedings.mlr.press/v97/rowland19a.html. ISSN: 2640-3498.

M. Rowland, Y. Tang, C. Lyle, R. Munos, M. G. Bellemare, and W. Dabney. The Statistical Benefits of Quantile Temporal-Difference Learning for Value Estimation. 2023.

Y. Shen, M. J. Tobia, T. Sommer, and K. Obermayer. Risk-sensitive reinforcement learning. *Neural Computation*, 26(7):1298–1328, 2014. doi: 10.1162/NECO_a_00600.

M. J. Sobel. The variance of discounted Markov decision processes. *Journal of Applied Probability*, 19(4):794–802, Dec. 1982. ISSN 0021-9002, 1475-6072. doi: 10.2307/3213832. URL https://www.cambridge.org/core/product/identifier/S0021900200023123/type/journal_article.

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

C. Szepesvári. Algorithms for Reinforcement Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 4(1):1–103, Jan. 2010. ISSN 1939-4608, 1939-4616. doi: 10.2200/S00268ED1V01Y201005AIM009. URL http://www.morganclaypool.com/doi/abs/10.2200/S00268ED1V01Y201005AIM009.

A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

C. Villani. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003. ISBN 9780821833124. URL https://books.google.fr/books?id=idyFAwAAQBAJ.

A. A. Volk, R. W. Epps, D. T. Yonemoto, B. S. Masters, F. N. Castellano, K. G. Reyes, and M. Abolhasani. Alphaflow: autonomous discovery and optimization of multi-step chemistry using a self-driven fluidic lab guided by reinforcement learning. *Nature Communications*, 14(1):1403, 2023.

J. von Neumann, O. Morgenstern, and A. Rubinstein. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944. ISBN 978-0-691-13061-3. URL https://www.jstor.org/stable/j.ctt1r2gkx.

C. J. Watkins and P. Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, L. Gilpin, P. Khandelwal, V. Kompella, H. Lin, P. MacAlpine, D. Oller, T. Seno, C. Sherstan, M. D. Thomure, H. Aghabozorgi, L. Barrett, R. Douglas, D. Whitehead, P. Dürr, P. Stone, M. Spranger, and H. Kitano. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228, Feb. 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04357-7. URL https://www.nature.com/articles/s41586-021-04357-7. Number: 7896 Publisher: Nature Publishing Group.

A Additional remarks

The Wasserstein metric is defined as $W_1(\nu_1, \nu_2) = \int_0^1 \left|F_{\nu_1}^{-1}(u) - F_{\nu_2}^{-1}(u)\right| \, \mathrm{d}u$ and the Cramer metric as $\ell_2(\nu_1, \nu_2) = \left(\int_{-\infty}^{+\infty} |F_{\nu_1}(u) - F_{\nu_2}(u)|^2 \, \mathrm{d}u\right)^{\frac{1}{2}}$. For both metrics, we define their supremum $\overline{\ell_2}(\eta_1, \eta_2) = \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} \ell_2(\eta_1^{(x,a)}, \eta_2^{(x,a)})$ and $\overline{W}_1(\eta_1, \eta_2) = \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} W_1(\eta_1^{(x,a)}, \eta_2^{(x,a)})$.

A.1 Remarks on the recursive definition of the Q-value distribution

The notation of the Q-value distribution η is often deceivingly complex compared to the actual object it means to represent. While the 'usual' expected Q-function $Q(x, a)$ is simply understood as the expected return of a policy at state-action pair (x, a), DistRL requires us to keep a notation for the complete distribution of the return. In other words, the Q-value distribution $\eta_{\pi,h}^{(x,a)}$ should be understood as the distribution of the *random variable* $Z = R + Z(S')$, which is the convolution of the individual distributions of these two independent random variable. It can also be written:

$$\forall x, a, h, \quad \eta_{\pi,h}^{(x,a)} = \sum_{x',a'} \int p_h^a(x, x')\pi_{h+1}^{x'}(a')\eta_{\pi,h+1}^{(x',a')}(\cdot - r)R_h^{(x,a)}(dr) \; . \tag{10}$$

A.2 Linear and Exponential Utilities satisfy the properties of Lemma 2

Independence Property Any utility U_f verifies the independence property. Let $\nu_1, \nu_2, \nu_3 \in \mathscr{P}(\mathbb{R}),\ \lambda \in [0, 1]$. Assume $U_f(\nu_1) \geq U_f(\nu_2)$. Then,

$$\begin{aligned}
U_f(\lambda\nu_1 + (1-\lambda)\nu_3) &= \int f \mathrm{d}(\lambda\nu_1 + (1-\lambda)\nu_3) \\
&= \lambda \underbrace{\int f \mathrm{d}\nu_1}_{U_f(\nu_1)} + (1-\lambda) \int f \mathrm{d}\nu_3 \\
&\geq \lambda \underbrace{\int f \mathrm{d}\nu_2}_{U_f(\nu_2)} + (1-\lambda) \int f \mathrm{d}\nu_3 \\
&= \int f \mathrm{d}(\lambda\nu_2 + (1-\lambda)\nu_3) \\
&= U_f(\lambda\nu_2 + (1-\lambda)\nu_3)
\end{aligned}$$

In particular, the mean and the exponential utility do.

Translation Property This property comes from the linearity of the mean and the multiplicative morphism of the exponential. Let $\nu_1, \nu_2 \in \mathscr{P}(\mathbb{R}),\ c \in \mathbb{R}$. Assume that $U_{\exp}(\nu_1) \geq U_{\exp}(\nu_2)$ and $U_{\text{mean}}(\nu_1) \geq U_{\text{mean}}(\nu_2)$. Then,

$$\begin{aligned}
U_{\exp}(\tau_c\nu_1) &= \int \exp(r) \mathrm{d}\tau_c\nu_1(r) \\
&= \int \exp(r + c) \mathrm{d}\nu_1(r) \\
&= \exp(c) \int \exp(r) \mathrm{d}\nu_1(r) \\
&= \exp(c) U_{\exp}(\nu_1) \\
&\geq \exp(c) U_{\exp}(\nu_2) \\
&= \dots \\
&= U_{\exp}(\tau_c\nu_2)
\end{aligned}$$

$$
\begin{aligned}
U_{\text{mean}}(\tau_c \nu_1) &= \int \lambda\tau + b + c \mathrm{d}\tau \\
&= c + U_{\text{mean}}(\nu_1) \\
&\geq c + U_{\text{mean}}(\nu_2) \\
&= \dots \\
&= U_{\text{mean}}(\tau_c \nu_2)
\end{aligned}
$$

A.3 Policy Evaluation for linear combinations of moments

Rowland et al. [2019] prove a necessary condition on Bellman-closed utilities, namely that they should be a family of the form $\{x \mapsto x^\ell \exp(\lambda x) | 0 \leq \ell \leq L\}$ for the undiscounted, finite-horizon setting. In the discounted case, the necessary condition is only valid for $\lambda = 0$, that is, without the exponential. They also prove that moments (without the exponential) also verify the sufficient condition such that in that setting they are the only Bellman-closed families of utilities.

In the undiscounted setting, to the best of our knowledge, a similar result has not yet been proved. We provide here the sufficient condition for families of the form $\{x \mapsto x^\ell \exp(\lambda x) | 0 \leq \ell \leq L\}$. We show that they are Bellman-closed and that this implies that they can be computed by DP.

Let's consider the family $s_k(\nu) = \int r^k \exp(\lambda r) \mathrm{d}\nu(r)$ for $k \in [n]$ and some fixed $\lambda \in \mathbb{R}$.

$$
\begin{aligned}
&s_n(\eta_{\pi,h}^{(x,a)}) \\
&= \mathbb{E}\left[Z_h^\pi(x,a)^n \exp(\lambda Z_h^\pi(x,a))\right] \\
&= \mathbb{E}\left[(R_h(x,a) + Z_{h+1}^\pi(X',A'))^n \exp(\lambda(R_h(x,a) + Z_{h+1}^\pi(X',A'))\right] \\
&= \mathbb{E}_{x',a'}\left[\mathbb{E}_{R_h,Z_{h+1}}\left[(R_h(x,a) + Z_{h+1}^\pi(x',a'))^n \exp(\lambda(R_h(x,a) + Z_{h+1}^\pi(x',a'))|X'=x', A'=a'\right]\right] \\
&= \sum_{x',a'} p_h^a(x,x')\pi_h^{x'}(a')\mathbb{E}_{R_h,Z_{h+1}}\left[(R_h(x,a) + Z_{h+1}^\pi(x',a'))^n \exp(\lambda(R_h(x,a) + Z_{h+1}^\pi(x',a'))\right] \\
&= \sum_{x',a'} p_h^a(x,x')\pi_h^{x'}(a')\mathbb{E}_{R_h,Z_{h+1}}\left[\sum_{k=0}^n \binom{n}{k} R_h(x,a)^{n-k} \exp(\lambda R_h(x,a)) Z_{h+1}^\pi(x',a')^k \exp(\lambda Z_{h+1}^\pi(x',a')\right] \\
&= \sum_{x',a'} p_h^a(x,x')\pi_h^{x'}(a') \sum_{k=0}^n \binom{n}{k} \mathbb{E}_{R_h}\left[R_h(x,a)^{n-k} \exp(\lambda R_h(x,a))\right] \mathbb{E}_{Z_{h+1}}\left[Z_{h+1}^\pi(x',a')^k \exp(\lambda Z_{h+1}^\pi(x',a')\right] \\
&= \sum_{x',a'} p_h^a(x,x')\pi_h^{x'}(a') \sum_{k=0}^n \binom{n}{k} \mathbb{E}_{R_h}\left[R_h(x,a)^{n-k} \exp(\lambda R_h(x,a))\right] s_k(\eta_{\pi,h+1}^{(x',a')})
\end{aligned}
$$

This first proves that this family of statistical functional is Bellman closed: they can be expressed as a linear combination of the others. Moreover, on the right-hand side, the expression only depends on the distributions and functionals at the current step h and at the next step $h+1$. Thus, it provides a natural way to evaluate these functionals by DP.

B Categorical Projection: an alternative parametrization

The categorical projection was proposed and studied in Bellemare et al. [2017], Rowland et al. [2018]. For a bounded return distribution, it spreads a fixed number N of Diracs evenly over the support and used weight parameters to represent the true distribution. The parameter N is often referred to as the *resolution* of the projection. More precisely, on a support $[V_{\min}, V_{\max}]$, we write $\Delta = \frac{V_{\max} - V_{\min}}{N-1}$ the step between atoms $z_i = V_{\min} + i\Delta, \quad i \in [\![0, N-1]\!]$. We define the projection of a given Dirac distribution δ_y on the parametric space $\mathscr{P}_C(\mathbb{R}) = \{\sum_i p_i \delta_{z_i} |\ 0 \leq p_i \leq 1,\ \sum_i p_i = 1\}$ by

$$
\Pi_C(\delta_y) = \begin{cases} \delta_{z_0} & y \leq z_0 \\ \frac{z_{i+1}-y}{z_{i+1}-z_i}\delta_{z_i} + \frac{y-z_i}{z_{i+1}-z_i}\delta_{i+1} & z_i < y < z_{i+1} \\ \delta_{z_{N-1}} & y \geq z_{N-1} \end{cases} \tag{11}
$$

This definition can naturally be extended to any bounded distribution ν, and by extension, $\Pi_C \eta = (\Pi_C \eta^{(s,a)})_{(s,a)\in\mathcal{X}\times\mathcal{A}}$. This linear operator minimizes the Cramér distance of the parametrization to the parametric space [Rowland et al., 2018].

This projection verifies this approximation bound, analog to the quantile projection,

$$\sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} \ell_2(\Pi_C \eta^{(x,a)}, \eta^{(x,a)}) \leq \frac{\Delta_\eta}{N} \ . \tag{12}$$

Using the property that $W_1(\nu_1, \nu_2) \leq \sqrt{\Delta}_\eta \ell_2(\nu_1, \nu_2)$, the results with the quantile projection can be adapted for the categorical projection, adding $\sqrt{\Delta}_\eta$ factors to the bounds.

C Proofs for Policy Evaluation with parameterized distributions

C.1 Proof of Proposition 1

We recall the statement of Proposition 1: Let π be a policy and η_π the associated Q-value distributions. Assume the return is bounded on a interval of length $\Delta_\eta \leq H\Delta_R$, where Δ_R is the support size of the reward distribution. Let $\hat{\eta}_\pi$ be the Q-value distributions obtained by dynamic programming (Algorithm 1) using the quantile projection Π_Q with resolution N. Then,

$$\sup_{(x,a,h)\in(\mathcal{X},\mathcal{A},[H])} W_1(\hat{\eta}_{\pi,h}^{(x,a)}, \eta_{\pi,h}^{(x,a)}) \leq H\frac{\Delta_\eta}{2N} \leq H^2\frac{\Delta_R}{2N} \ .$$

To avoid clutter of notation, we denote $\overline{W}_1(\hat{\eta}, \eta) := \sup_{(x,a)\in\mathcal{X}\times\mathcal{A}} W_1(\hat{\eta}^{(x,a)}, \eta^{(x,a)})$.

Proof. First recall that for any Q-value distribution $(\eta_h)_{h\in[H]}$, with the return bounded on an interval of length $\Delta_\eta \leq H\Delta_R$, and Π one of the projection operator of interest with resolution n, we have the following bound on the projection estimation error due to Rowland et al. [2019] (Eq (6)):

$$\overline{W}_1(\Pi\eta, \eta) \leq \frac{\Delta_\eta}{2N} \ . \tag{13}$$

At a fixed step $h \in [H]$, we have the following inequality:

$$\begin{aligned}
\overline{W}_1(\hat{\eta}_{\pi,h}, \eta_{\pi,h}) &= \overline{W}_1(\Pi\mathcal{T}_h^\pi \hat{\eta}_{\pi,h+1}, \mathcal{T}_h^\pi \eta_{\pi,h+1}) \\
&\leq \overline{W}_1(\Pi\mathcal{T}_h^\pi \hat{\eta}_{\pi,h+1}, \mathcal{T}_h^\pi \hat{\eta}_{\pi,h+1}) + \overline{W}_1(\mathcal{T}_h^\pi \hat{\eta}_{\pi,h+1}, \mathcal{T}_h^\pi \eta_{\pi,h+1}) \qquad (14) \\
&\leq H\frac{\Delta_R}{2N} + \overline{W}_1(\hat{\eta}_{\pi,h+1}, \eta_{\pi,h+1}) \ . \qquad (15)
\end{aligned}$$

Where (14) is due to the triangular inequality with $\mathcal{T}_h^\pi \hat{\eta}_{\pi\ h+1}$ as the middle term. In (15), the first term comes from applying (13) to the first term of the previous line. The second term is a consequence of the non-expansive property of the Bellman operator [Bellemare et al., 2017]:

$$\overline{W}_1(\mathcal{T}\eta_1, \mathcal{T}\eta_2) \leq \overline{W}_1(\eta_1, \eta_2) \ .$$

Using it recursively starting from $h = 1$, and using the fact that $\hat{\eta}_{\pi,H} = \eta_{\pi,H}$ we get:

$$\overline{W}_1(\hat{\eta}_{\pi,1}, \eta_{\pi,1}) \leq H\frac{\Delta_R}{2N} + \overline{W}_1(\hat{\eta}_{\pi,2}, \eta_{\pi,2}) \leq 2H\frac{\Delta_R}{2N} + \overline{W}_1(\hat{\eta}_{\pi,3}, \eta_{\pi,3}) \leq \cdots \leq H^2\frac{\Delta_R}{2N} \ .$$

□

C.2 Proof of Lemma 1

We recall the statement of Lemma 1: Let s be either a utility or a distorted mean and let L be the Lipschitz coefficient of its characteristic function. Let ν_1, ν_2 be return distributions. Then:

$$|s(\nu_1) - s(\nu_2)| \leq L W_1(\nu_1, \nu_2) \ .$$

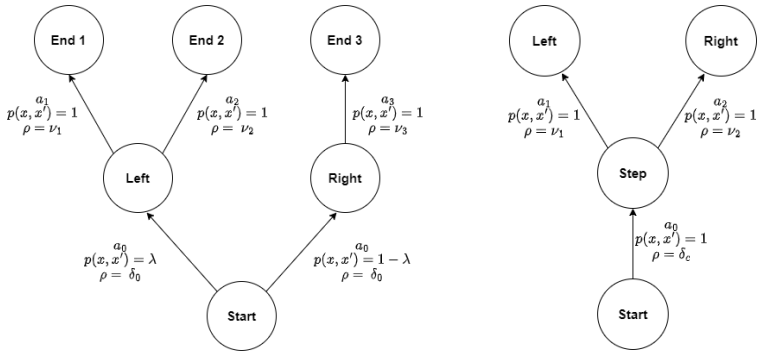

Figure 3: Left: Independence Property Counter Example, Right: Translation Property Counter Example. Each arrow represents a state transition, which is characterized by the action leading to the transition, the probability of such transition, and the reward distribution of the transition.

Proof. We prove the property for each family of utilities separately:

Case 1: s is a utility. There exists f such that $s(\nu) = \int f \mathrm{d}\nu$. Let L_f be its Lipsichtz constant. The Kantorovitch-Rubenstein duality [Villani, 2003] states that:

$$W_1(\nu_1, \nu_2) = \frac{1}{L_f} \sup_{||g||_L \leq L_f} \left(\int g \, \mathrm{d}\nu_1 - \int g \, \mathrm{d}\nu_2 \right) , \tag{16}$$

where $|| \cdot ||_L$ is the Lipschitz norm. We then immediatly get:

$$L_f W_1(\nu_1, \nu_2) \geq \left| \int f \, \mathrm{d}\nu_1 - \int f \, \mathrm{d}\nu_2 \right| = |s(\nu_1) - s(\nu_2)| \, . \tag{17}$$

Case 2: s is a distorted mean. There exists g such that $s(\nu) = \int_0^1 g'(\tau) F_\nu^{-1}(\tau) \mathrm{d}\tau$. Let L_g be its Lipschitz coefficent. Thus:

$$\begin{aligned} |s(\nu_1) - s(\nu_2)| &= \left| \int_0^1 g'(\tau) \left(F_{\nu_1}^{-1} - F_{\nu_2}^{-1}(\tau) \right) \mathrm{d}\tau \right| \\ &\leq ||g'||_\infty \int_0^1 \left| F_{\nu_1}^{-1}(\tau) - F_{\nu_2}^{-1}(\tau) \right| \mathrm{d}\tau \\ &\leq L_g W_1(\nu_1, \nu_2) \, . \end{aligned}$$

□

D Proof of the main results

The proof of the result is divided in parts. First we show that Bellman optimizable functionals verify the two properties of Lemma 2 (Independence and Translation). Then, using those properties, we prove that Bellman optimizable functionals can only be exponential utilities (Theorem 2). Using the known fact that exponential utilities are bellman optimizable, we obtain the full characterization.

D.1 Proof of Lemma 2

Proof. **To prove that each property is necessary**, we use a proof by contradiction, and exhibit MDPs where the algorithm is not optimal when the property is not verified.

Independence Property Let s be a Bellman optimizable statistical functional that does not satisfy the Independence property. That is, there exists $\nu_1, \nu_2, \nu_3 \in \mathscr{P}(\mathbb{R})$ and $\lambda \in [0, 1]$ such that $s(\nu_1) \geq s(\nu_2)$ but $s(\lambda\nu_1 + (1-\lambda)\nu_3) < s(\lambda\nu_2 + (1-\lambda)\nu_3)$.

Then consider the MDP in Fig.3 (left) with horizon $H = 2$ corresponding to the depth of the tree: The agent starts in `Start` and must take 2 actions, a unique but random and non-rewarding one (a_0) and a final deterministic step (a_1 or a_2) to a rewarding state. Thus, by construction, the optimal strategy is (a_0, a_2) that leads to `End 2` with probability λ (and `End 3` with probability $1 - \lambda$). The true optimal distribution at `Start` state is $\eta_0^* = \lambda\nu_2 + (1 - \lambda)\nu_3$. We compute the distributions output by the algorithm:

$$
\begin{aligned}
H = 2: &\qquad \eta_2^{(\text{End } 1, a^*)} = \delta_0, \quad \eta_2^{(\text{End } 2, a^*)} = \delta_0, \quad \eta_2^{(\text{End } 3, a^*)} = \delta_0 \\
H = 1: &\qquad \eta_1^{(\text{Left}, a^* = \arg\max_a s(\nu_a))} = \nu_1, \quad \eta_2^{(\text{Right}, a^* = a_3)} = \nu_3 \\
H = 0: &\qquad \eta_0^{(\text{Start}, a^* = a_0)} = \lambda\nu_1 + (1 - \lambda)\nu_3
\end{aligned}
$$

The output return distribution η_0 is not the true optimal η_0^* for s so the algorithm is incorrect which is a contradiction as s is assumed to be Bellman optimizable. Hence the property is needed.

Translation Property Let s be a Bellman optimizable statistical functional that does not verify the Translation Property, *i.e.* there exists $\nu_1, \nu_2 \in \mathscr{P}(\mathbb{R})$, $c \in \mathbb{R}$ such that $s(\nu_1) \geq s(\nu_2)$ but $s(\tau_c\nu_1) < s(\tau_c\nu_2)$. Then consider MDP in Fig.3 (right). The optimal strategy is again (a_0, a_2) by construction. The algorithm output the following distribution:

$$
\begin{aligned}
H = 2: &\qquad \eta_2^{(\text{Left}, a^*)} = \delta_0, \quad \eta_2^{(\text{Right}, a^*)} = \delta_0 \\
H = 1: &\qquad \eta_1^{(\text{Step}, a^*)} = \nu_1 \\
H = 0: &\qquad \eta_0^{(\text{Start}, a^*)} = \tau_c\nu_1
\end{aligned}
$$

So here again, the algorithm does not output an optimal distribution for s, hence the necessity of the property.

This proof shows that both properties are necessary, but not that they are sufficient. The other implication could be proven, but the proof would be unnecessary as those properties are enough to restrict to only 1 class of function for which we already know is bellman optimizable.

□

D.2 Proof of Theorem 2

Proof. by Lemma 2, all Bellman optimizable statistical functionals satisfy both the Independence and the Translation property. As already mentioned in the main part of the paper, by the Von Neumann-Morgenstein theorem and because of the Independence property, all Bellman optimizable functionals are (or correspond to) utilities.

Let f be the function characterizing the utility $s = s_f$. One may assume, without loss of generality, that f is twice differentiable and that $f'(x) \neq 0$, $\forall x \in \mathbb{R}$. We can then define $\phi(h) = f^{-1}(\frac{1}{2}(f(0) + f(h)))$, so that $\frac{1}{2}(f(0)+f(h)) = f(\phi(h))$. Note that this definition implies that $\phi(0) = f^{-1}(f(0)) = 0$.

Using the inversion theorem, with f twice differentiable and f' always non zero, ϕ is also twice differentiable.

Fix some $h > 0$, and define two simple probability distributions $\nu_1 = \frac{1}{2}(\delta_0 + \delta_h)$ and $\nu_2 = \delta_{\phi(h)}$. Note that $s_f(\nu_1) = \frac{1}{2}(\delta_0 + \delta_h)$ and $s_f(\nu_2) = f(\phi(h))$, so $s_f(\nu_1) = s_f(\nu_2)$ by definition of ϕ. Then, by the Translation property, we have for any $x \in \mathbb{R}$,

$$
\frac{1}{2}\big(f(x) + f(x + h)\big) \geq f(x + \phi(h)) \quad \text{and} \quad f(x + \phi(h)) \geq \frac{1}{2}\big(f(x) + f(x + h)\big), \tag{18}
$$

so $\forall x \in \mathbb{R}$, $\frac{1}{2}\big(f(x) + f(x + h)\big) = f(x + \phi(h))$. We have the inequality both ways above because $s_f(\nu_1) = s_f(\nu_2)$.

This equation can be twice differentiated with respect to h, for any value of x, we obtain:

$$\frac{1}{2}f'(x+h) = \phi'(h)f'(x+\phi(h)) \quad \text{and} \tag{19}$$

$$\frac{1}{2}f''(x+h) = \phi''(h)f'(x+\phi(h)) + \phi'(h)^2 f''(x+\phi(h)) \ . \tag{20}$$

Recall that by definition, $\phi(0) = 0$). Now, for $x = 0$, Eq. (19) yields

$$\frac{1}{2}f'(0) = \phi'(0)f'(\phi(0)) = \phi'(0)f'(0) \quad \Longrightarrow \quad \phi'(0) = \frac{1}{2} \text{ because } f'(0) \neq 0.$$

Now, choosing $h = 0$ in (20) and plugging in the values of $\phi(0)$ and $\phi'(0)$, we obtain for all $x \in \mathbb{R}$:

$$\frac{1}{4}f''(x) = \phi''(0)f'(x) \ .$$

We then consider two cases, depending on whether $\phi''(0)$ is null or not.

Case 1: $\phi''(0) = 0$. The equation simply becomes $f''(x) = 0$, hence f is affine: $\exists\, a, b \in \mathbb{R},\ f(x) = ax + b$.

Case 2: $\phi''(0) \neq 0$. We write $\beta = 4\phi''(0)$. The differential equation becomes $f''(x) = \beta f'(x)$, whose solutions are of the form

$$\exists\, c_1, c_2, \beta, \quad f(x) = c_1 \exp(\beta x) + c_2$$

Hence, f can only be the identity or the exponential, up to an affine transformation.

□