

Reinforcement learning with dynamic convex risk measures

Anthony Coache¹ | Sebastian Jaimungal^{1,2}

¹Department of Statistical Sciences,
University of Toronto, Toronto, Canada

²Oxford-Man Institute, University of
Oxford, Oxford, United Kingdom

Correspondence

Sebastian Jaimungal.

Email: sebastian.jaimungal@utoronto.ca

Funding information

Natural Sciences and Engineering

Research Council of Canada,

Grant/Award Numbers:

RGPIN-2018-05705, CGSD3-2019-534435,

RGPAS-2018-522715

Abstract

We develop an approach for solving time-consistent risk-sensitive stochastic optimization problems using model-free reinforcement learning (RL). Specifically, we assume agents assess the risk of a sequence of random variables using dynamic convex risk measures. We employ a time-consistent dynamic programming principle to determine the value of a particular policy, and develop policy gradient update rules that aid in obtaining optimal policies. We further develop an actor–critic style algorithm using neural networks to optimize over policies. Finally, we demonstrate the performance and flexibility of our approach by applying it to three optimization problems: statistical arbitrage trading strategies, financial hedging, and obstacle avoidance robot control.

KEYWORDS

actor–critic algorithm, dynamic risk measures, financial hedging, policy gradient, reinforcement learning, robot control, time-consistency, trading strategies

1 | INTRODUCTION

Reinforcement learning (RL) provides a (model-free) framework for learning-based control. RL problems aim at learning dynamics in the underlying data and finding optimal behaviors while collecting data via an interactive process. It differs from *supervised learning* that attempts to learn classification functions from labeled data, and *unsupervised learning* that seeks hidden patterns

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](#) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 The Authors. *Mathematical Finance* published by Wiley Periodicals LLC.

in unlabeled data. During the training phase, the agent makes a sequence of decisions while interacting with the data-generating process and observing feedback in the form of costs. The agent aims to discover the best possible actions by interacting with the environment and consistently updating their actions based on their experience, while often, also taking random actions to assist in exploring the state space—the classic exploration–exploitation trade-off (Sutton & Barto, 2018).

In RL, *uncertainty* in the data-generating process can have substantial effects on performance. Indeed, the environmental randomness may result in algorithms optimized for “on-average” performance to have large variance across scenarios. For example, consider a portfolio selection problem: a risk-neutral optimal strategy (where one optimizes the expected terminal return) focuses on assets with the highest returns while ignoring the risks associated with them. As a second example, consider an autonomous car which should account for environmental uncertainties such as the weather and road conditions when learning the optimal strategy. Such an agent may wish to account for variability in the environment and the results of its actions to avoid large potential “losses.” For an overview and outlook on RL in financial mathematics see, for example, Jaimungal (2022).

In the extant literature, there are numerous proposals for accounting for risk sensitivity, where most of them replace the expectation in the optimization problem by risk measures—we provide an overview in Section 2. *Risk-awareness* or *risk-sensitivity* in RL offers a remedy to the data-generating process uncertainty by quantifying low-probability but high-cost outcomes, provides strategies that are more robust to the environment, and allows more flexibility than traditional approaches as it is tuned to the agent’s risk preference. The specific choice of risk measure is a decision the agent makes considering their goals and risk tolerances. We do not address here how the agent makes this specific choice and instead refer the reader to, for example, Dhaene et al. (2006) for an overview.

An interesting approach to risk-aware learning stems from Tamar et al. (2015, 2016), where they provide policy search algorithms to solve risk-aware RL problems in a dynamic framework. Both studies investigate *stationary policies*, restrict themselves to *coherent* risk measures, and apply their methodology to simple financial engineering applications. More specifically, they evaluate their algorithms when learning policies for (perpetual) American options and trading in static portfolio optimization problems. Several real-world applications require a level of flexibility that these limitations preclude.

In this paper, we develop a model-free approach to solve a wide class of (nonstationary) risk-aware RL problems in a time-consistent manner. Our contributions may be summarized as follows: (i) we extend Tamar et al. (2015, 2016); Ahmadi et al. (2021); Kose and Ruszczyński (2021); Huang et al. (2021) by focusing on the broad class of *dynamic convex* risk measures and consider finite-horizon problems with *nonstationary policies*; (ii) we devise an *actor–critic* algorithm to solve this class of RL problems using neural networks to allow continuous state–action spaces; (iii) we derive a recursive formula for efficiently computing the policy gradients; and (iv) we demonstrate the performance and flexibility of our proposed approach on three important applications: optimal trading for statistical arbitrage, hedging financial options, and obstacle avoidance in robot control. We demonstrate that our approach appropriately accounts for uncertainty and leads to strategies that mitigate risk.

The remainder of the paper is structured as follows. In Section 2, we discuss related work and introduce our RL notation in Section 3. Section 4 formalizes the evaluation of risk in both static and dynamic frameworks. We introduce the class of sequential decision making problems with dynamic convex risk measures in Section 5 and devise an actor–critic style algorithm to solve them in Section 6. Finally, Section 7 illustrates the performance of our proposed algorithm, and we discuss our work’s limitations and possible extensions in Section 8.

2 | RELATED WORK

The literature in risk evaluation for sequential decision making can be divided between those that apply a risk measure to a single cost random variable (e.g., discounted sum of costs or terminal profit and loss), and those that apply risk measures recursively to a sequence of cost random variables (e.g., cash-flows). The former approach optimizes the risk of a single random variable, which does not account for the temporal structure in what generates it, while the latter optimizes the risk of sequences of random variables in a *dynamic* framework as additional information becomes available.

Several authors address sequential decision making problems by minimizing the risk of a cost *over a whole episode*. For example, Prashanth and Ghavamzadeh (2013) focus on objective functions for variance-related criteria, while Chow et al. (2017) take a risk-constrained approach with an objective function that includes a penalty on the conditional value-at-risk (CVaR). Some lines of research look at risk-sensitive RL problems using a broader classes of risk measures, such as comonotonic (Petrik & Subramanian, 2012), entropic (Nass et al., 2019), or spectral (Bäuerle & Glauner, 2021b) risk measures. Di Castro et al. (2019) consider a risk-neutral objective function, but includes an additional constraint on the risk of the cumulative discounted cost.

Other works extend optimization in Markov decision processes (MDPs) by evaluating the risk *at each period*. For instance, Ruszczyński (2010) evaluates the risk at each period using dynamic Markov coherent risk measures, while Chu and Zhang (2014) and Bäuerle and Glauner (2021a) propose iterated coherent risk measures, where they both derive risk-aware dynamic programming (DP) equations and provide policy iteration algorithms. While they focus on coherent risk measures, various classes of risk measures have already been extended to a dynamic framework, such as distribution-invariant risk measures (Weber, 2006), coherent risk measures (Riedel, 2004), convex risk measures (Detlefsen & Scandolo, 2005; Frittelli & Gianin, 2004), and dynamic assessment indices (Bielecki et al., 2016), among others—however, these works do not look at how to perform model free optimization, that is, they do not look at RL. For an overview of dynamic risk measures, see, for example, Acciaio and Penner (2011).

Another way to account for uncertainty in the data-generating process is by allowing the parameters of the model, or the entire distribution itself, to be unknown. The class of *robust MDPs* (Delage & Mannor, 2010) focuses on optimizing the worst-case expectation when the parameters vary within a certain set. There exists relationships between risk-aware and robust MDPs, as shown in Osogami (2012) and Bäuerle and Glauner (2022). Indeed, minimizing a Markov coherent risk measure in a risk-aware context is equivalent to minimizing a certain worst-case expectation where the uncertainty set is characterized by a concave function. Several researchers have developed algorithms to solve robust MDPs, for an overview see, for example, Rahimian and Mehrotra (2019).

While we consider a model-free approach, several researchers tackle related, but distinct, problems in a *model-based framework*. For instance, Weinan et al. (2017) and Han et al. (2018) use deep learning methods to solve nonlinear partial differential equations (PDEs). Using the nonlinear Feynman–Kac representation, they reformulate the PDEs as backward stochastic differential equations (BSDEs), they then parametrize the co-adjoint process and initial condition using an ensemble of neural networks, and use the mean squared error in the terminal condition as the loss function. Despite there being an equivalence between BSDEs and dynamic risk measures (Drapeau et al., 2016; Peng, 1997), the dual representation does not directly help when we aim to optimize a dynamic risk measure in a model-free manner.

Other types of algorithms exist in the literature to find a solution to risk-aware RL problems. Among others, Galichet et al. (2013) use a *multi-armed bandit approach*, Shen et al. (2014) devise a risk-sensitive *Q-learning method* when optimizing utility functions, Bellemare et al. (2017) use a *distributional perspective* to learn the whole distribution of the value function, Yu et al. (2018) employ an approximate DP approach to devise a *value iteration* algorithm for Markov risk measures, and Kalogerias et al. (2020) address risk-aware problems from a *Bayesian perspective*. The shortcomings of these approaches are that they apply to a specific class of risk measures and the developed methodologies are tuned to them.

3 | REINFORCEMENT LEARNING

In this section, we introduce the necessary theoretical background for the RL problems we study. We describe each problem as an *agent* who tries to learn an optimal behavior, or *agent's policy*, by interacting with a certain *data-generating process*.

Let S and $\mathcal{A}$ be arbitrary state and action spaces, respectively, and let $C \subset \mathbb{R}$ be a cost space. The data-generating process is often represented as a MDP with the tuple $(S, \mathcal{A}, c, \mathbb{P})$, where $c(s, a, s') \in C$ is a deterministic, state-action-dependent cost function and $\mathbb{P}$ characterizes the transition probabilities $\mathbb{P}(s_{t+1} = s' | s_t = s, a_t = a)$, unknown to the agent. The transition probability is assumed stationary, although time may be a component of the state, for example, we usually assume that time is embedded in the state space without loss of generality. A single *episode* consists of T periods, where $T \in \mathbb{N}$ is known and finite. At each period, the agent begins in state $s_t \in S$, takes an action $a_t \in \mathcal{A}$, moves to the next state $s_{t+1} \in S$, and receives a cost $c_t = c(s_t, a_t, s_{t+1}) \in C$. A directed graph representation of the described MDP is shown in Figure 1. A *trajectory* consists of all states, actions and costs that occur during a single episode and we denote it by the tuple $\tau := (s_0, a_0, c_0, \dots, s_{T-1}, a_{T-1}, c_{T-1}, s_T)$.

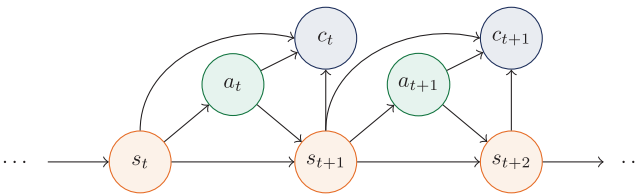


FIGURE 1 Directed graph representation of an MDP. MDP, Markov decision process. [Color figure can be viewed at wileyonlinelibrary.com]

The agent follows a strategy described by a randomized (also known as exploratory control) *policy* $\pi : S \rightarrow \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ is the space of probability measures on $\sigma(\mathcal{A})$. More specifically when in state s at time t , the agent selects the action a with probability $\pi(a | s_t = s)$. This also allows for nonstationary policies when dealing with finite-horizon problems.

Standard RL usually deals with *risk-neutral* objective functions of the (γ -discounted) cost of a trajectory induced by a MDP with a policy π , for instance

$$\min_{\pi} \mathbb{E} \left[\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t, s_{t+1}) \right], \quad (1)$$

where $\gamma \in (0, 1]$ is a discount factor. At period T , there is no action and hence any cost based on the terminal state is encapsulated in c_{T-1} . Contrastingly, risk-sensitive RL considers problems in which the objective takes into account the *variability* of the cost with a risk measure ρ , whether

optimizing

$$\min_{\pi} \rho \left(\sum_{t=0}^{T-1} \gamma^t c(s_t, a_t, s_{t+1}) \right), \quad (2)$$

or Equation (1) with an additional constraint on the risk measure of the trajectory cost. We discuss thoroughly risk measures and their properties in Section 4 next.

In all cases, the goal of RL is to learn the optimal policy π that attains the minimum in the corresponding objective function, and do so in a manner that makes no explicit assumptions on the data-generating process.

4 | RISK MEASURES

In this section, we formalize how we quantify the risk of random variables. We start by providing a review of static risk measures, and then continue by reviewing the framework of Ruszczyński (2010) for building time-consistent dynamic risk measures. Static risk measures evaluate the immediate risk of a financial position, while dynamic risk measures allow its monitoring at different times, and lead to time-consistent optimal strategies. In our work, we make use of static risk measures on conditional distributions of random variables to construct dynamic risk measures.

4.1 | Static setting

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, $\bar{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$ the extended real line, and define $\mathcal{Z} := \mathcal{L}_p(\Omega, \mathcal{F}, \mathbb{P})$ and $\mathcal{Z}^* := \mathcal{L}_q(\Omega, \mathcal{F}, \mathbb{P})$ with $p, q \in [1, \infty]$ such that $1/p + 1/q = 1$. They represent the space of p -integrable, respectively q -integrable, $\mathcal{F}$ -measurable random variables. A *risk measure* is a mapping $\rho : \mathcal{X} \rightarrow \bar{\mathbb{R}}$, where $\mathcal{X}$ is a space of random variables, that satisfies additional properties. In what follows we assume that $\mathcal{X} = \mathcal{Z}$ and $Z \in \mathcal{Z}$ is interpreted as a random cost. We next enumerate some properties of various risk measures.

Definition 4.1. Let $m \in \mathbb{R}$, $\beta > 0$ and $\lambda \in [0, 1]$. A map $\rho : \mathcal{X} \rightarrow \bar{\mathbb{R}}$ is said to be

1. *law-invariant* if $\rho(Z_1) = \rho(Z_2)$ if $Z_1 \stackrel{d}{=} Z_2$;
2. *monotone* if $Z_1 \leq Z_2$ implies $\rho(Z_1) \leq \rho(Z_2)$;
3. *translation invariant* if $\rho(Z + m) = \rho(Z) + m$;
4. *positive homogeneous* if $\rho(\beta Z) = \beta \rho(Z)$;
5. *comonotonic additive* if $\rho(Z_1 + Z_2) = \rho(Z_1) + \rho(Z_2)$ for all comonotonic pairs (Z_1, Z_2) ;
6. *subadditive* if $\rho(Z_1 + Z_2) \leq \rho(Z_1) + \rho(Z_2)$;
7. *convex* if $\rho(\lambda Z_1 + (1 - \lambda)Z_2) \leq \lambda \rho(Z_1) + (1 - \lambda)\rho(Z_2)$.

There is a consensus in the literature that any risk measure should satisfy the monotonicity and translation invariance properties. More specifically, a portfolio with a higher cost for every possible scenario is indeed riskier, and the deterministic part of a portfolio does not contribute to its risk.

Definition 4.2. A map ρ is said to be a *monetary risk measure* (Föllmer et al., 2004) if and only if it is monotone and translation invariant.

In addition to being monetary, additional requirements may be assumed so that the risk measure reflects *observed investor behavior*.

Definition 4.3. A map ρ is said to be a *coherent risk measure* (Artzner et al., 1999) if and only if it is monotone, translation invariant, positive homogeneous, and subadditive.

The additional properties for coherent risk measures guarantee that doubling a position doubles its risk, and diversifying a portfolio reduces its risk. The CVaR (Rockafellar et al., 2000) is a well-known example of a coherent risk measure commonly used in the literature. Criticisms of positive homogeneity and subadditivity led to the study of a broader class of risk measures, where these axioms are weakened and replaced by the notion of convexity. Indeed, the risk might increase in a nonlinear way, which is not permitted under coherent risk measures.

Definition 4.4. A map ρ is said to be a *convex risk measure* (Föllmer & Schied, 2002) if and only if it is monotone, translation invariant, and convex.

Another advantage of convex risk measures is that we can combine several risk measures into a linear combination to create a trade-off between different risk-aware objectives. Indeed, one can easily show that given two convex risk measures ρ_1, ρ_2 and nonnegative coefficients $\beta_1, \beta_2 > 0$, then $\rho := \beta_1 \rho_1 + \beta_2 \rho_2$ and $\rho := \max\{\rho_1, \rho_2\}$ are both also convex. The set of coherent risk measures is a strict subset of the set of convex risk measures.

A dual representation of convex (and, as a special case, coherent) risk measures provides us with a key result for developing our practical algorithm. The dual representation requires us to introduce the expectation under what is effectively a distorted probability measure and denote $\mathbb{E}^\xi[Z] := \sum_{\omega} Z(\omega) \xi(\omega) \mathbb{P}(\omega)$ with $Z \in \mathcal{Z}$ and $\xi \in \mathcal{Z}^*$.

Definition 4.5. The *conjugate* (Shapiro et al., 2014) of the risk measure ρ , denoted $\rho^* : \mathcal{Z}^* \rightarrow \bar{\mathbb{R}}$, is given by

$$\rho^*(\xi) = \sup_{Z \in \mathcal{Z}} \{ \mathbb{E}^\xi[Z] - \rho(Z) \}. \quad (3)$$

Definition 4.6. The *biconjugate* (Shapiro et al., 2014), or conjugate of the conjugate, is a mapping $\rho^{**} : \mathcal{Z} \rightarrow \bar{\mathbb{R}}$ with

$$\rho^{**}(Z) = \sup_{\xi \in \mathcal{Z}^*} \{ \mathbb{E}^\xi[Z] - \rho^*(\xi) \}. \quad (4)$$

The dual representation of the risk measure is given in the following theorem.

Theorem 4.7 Representation theorem (see Theorem 6.4 in Shapiro et al., 2014). *A convex risk measure ρ is proper (i.e., $\rho(Z) > -\infty$ and its domain is nonempty) and lower semicontinuous (i.e., $\rho(W) \leq \liminf_{Z \rightarrow W} \rho(Z)$) iff there exists a set*

$$\mathcal{V}(P) \subset \left\{ \xi \in \mathcal{Z}^* : \sum_{\omega} \xi(\omega) \mathbb{P}(\omega) = 1, \xi \geq 0 \right\},$$

often referred to as the risk envelope, such that

$$\rho(Z) = \sup_{\xi \in \mathcal{U}(\mathbb{P})} \{ \mathbb{E}^\xi[Z] - \rho^*(\xi) \}. \quad (5)$$

Moreover, we have that ρ is coherent (e.g., satisfies also the positive homogeneity) iff

$$\rho(Z) = \sup_{\xi \in \mathcal{U}(\mathbb{P})} \{ \mathbb{E}^\xi[Z] \}.$$

If we assume ρ is a convex, proper, lower semicontinuous risk measure, then by Theorem 4.7, it may be written as an optimization problem where the distortion ξ is chosen adversarially from a subset of the set of all probability densities. Moreover, the notable difference between coherent and convex risk measures is the conjugate term that appears in its dual representation. Coherent risk measures are *uniquely* characterized by their risk envelope.

4.2 | Dynamic setting

Optimizing a controlled performance criteria using static risk measures is known to lead to time-inconsistent solutions—we discuss the precise definition below. Adapting risk measures to properly account for the flow of information requires additional care to ensure that the risk evaluation is done in a time-consistent manner, especially with DP models for MDPs. There are multiple extensions of static risk measure to the dynamic case. Indeed, various classes of risk measures have already been extended to a dynamic framework, such as distribution-invariant (Weber, 2006), coherent (Riedel, 2004), convex risk measures (Detlefsen & Scandolo, 2005; Frittelli & Gianin, 2004), and dynamic assessment indices (Bielecki et al., 2016), among others. Here, we closely follow the work of Ruszczyński (2010).

To this end, let $\mathcal{T} := \{0, \dots, T\}$ denote the sequence of periods in an episode. Consider a filtration $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \dots \subseteq \mathcal{F}_T \subseteq \mathcal{F}$ on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \in \mathcal{T}}, \mathbb{P})$ and $\{\mathcal{Z}_t\}_{t \in \mathcal{T}}$ with $\mathcal{Z}_t := \mathcal{L}_p(\Omega, \mathcal{F}_t, \mathbb{P})$. Define $\mathcal{Z}_{t,T} := \mathcal{Z}_t \times \dots \times \mathcal{Z}_T$.

Definition 4.8. A conditional risk measure is a map $\rho_{t,T} : \mathcal{Z}_{t,T} \rightarrow \mathcal{Z}_t$, which satisfies the monotonicity property, that is, $\rho_{t,T}(Z) \leq \rho_{t,T}(W)$ for all $Z, W \in \mathcal{Z}_{t,T}$ such that $Z \leq W$ a.s.

Definition 4.9. A dynamic risk measure is a sequence of conditional risk measures $\{\rho_{t,T}\}_{t \in \mathcal{T}}$.

We may interpret $\rho_{t,T}(Z)$, for $Z \in \mathcal{Z}_{t,T}$, as a $\mathcal{F}_t$ -measurable charge the agent would be willing to incur at time t instead of the sequence of costs Z . Developing a dynamic programming principle (DPP) for dynamic risk measures crucially depends on the property of *time-consistency*. One wishes to evaluate the risk of future outcomes, but it must not lead to inconsistencies at different points in time.

Definition 4.10. $\{\rho_{t,T}\}_{t \in \mathcal{T}}$ is said to be *time-consistent* (see Definition 3 in Ruszczyński, 2010) iff for any sequence $Z, W \in \mathcal{Z}_{t_1,T}$ and any $t_1, t_2 \in \mathcal{T}$ such that $(0 \leq t_1 < t_2 \leq T)$,

$$Z_k = W_k, \forall k = t_1, \dots, t_2 \quad \text{and} \quad \rho_{t_2,T}(Z_{t_2}, \dots, Z_T) \leq \rho_{t_2,T}(W_{t_2}, \dots, W_T)$$

implies that $\rho_{t_1,T}(Z_{t_1}, \dots, Z_T) \leq \rho_{t_1,T}(W_{t_1}, \dots, W_T)$.

Definition 4.10 may be interpreted as follows: if Z will be at least as good as W at time t_2 and they are identical between t_1 and t_2 , then Z should not be worse than W at time t_1 .

Furthermore, we introduce one additional concept that aids in developing a recursive relationship for dynamic risk measures.

Definition 4.11. A one-step conditional risk measure is a map $\rho_t : \mathcal{Z}_{t+1} \rightarrow \mathcal{Z}_t$, which satisfies $\rho_t(Z) = \rho_{t,t+1}(0, Z)$ for any $Z \in \mathcal{Z}_{t+1}$.

One may assume even stronger properties for conditional risk measures, for example, one may assume that the risk measures are static (across time) and convex. In the next section, we do precisely this. Therefore, the one-step conditional risk measure $\rho_t(\cdot | \mathcal{F}_t)$ outputs an $\mathcal{F}_t$ -measurable random variable obtained when conditioning on $\mathcal{F}_t$. As a consequence of Definitions 10 and 11, we have the following recursive relationship for time-consistent dynamic risk measures.

Theorem 4.12 Recursive relationship (see Theorem 1 and Equation 10 in Ruszczyński, 2010). Let $\{\rho_{t,T}\}_{t \in \mathcal{T}}$ be a time-consistent, dynamic risk measure. Suppose that it satisfies $\rho_{t,T}(Z_t, Z_{t+1}, \dots, Z_T) = Z_t + \rho_{t,T}(0, Z_{t+1}, \dots, Z_T)$ and $\rho_{t,T}(0, \dots, 0) = 0$ for any $Z \in \mathcal{Z}_{t,T}$, $t \in \mathcal{T}$. Then, $\{\rho_{t,T}\}_{t \in \mathcal{T}}$ can be expressed as

$$\rho_{t,T}(Z_t, \dots, Z_T) = Z_t + \rho_t(Z_{t+1} + \rho_{t+1}(Z_{t+2} + \dots + \rho_{T-2}(Z_{T-1} + \rho_{T-1}(Z_T)) \dots)). \quad (6)$$

As we are working with MDPs, the agent observes at each period $t \in \mathcal{T}$ the current state s_t , that is $\mathcal{F}_t$ -measurable. We, thus, define dynamic Markov risk measures, in a similar manner to Ruszczyński (2010), where ρ_t is a Markov one-step conditional risk measure that is $\sigma(s_t)$ -measurable. Those are originally obtained from risk transition mappings with respect to a controlled Markov process, which our framework with MDPs satisfies (for a thorough exploration, see Ruszczyński, 2010, especially Section 4).

5 | PROBLEM SETUP

In this section, we formally introduce the optimization problems that we face. Briefly, we are interested in RL problems where the agent wants to minimize a dynamic convex risk measure in order to obtain a time-consistent solution.

Let $(S, \mathcal{A}, c, \mathbb{P})$ be a MDP, $\mathcal{T} := \{0, \dots, T-1\}$ be the sequence of periods in an episode, and suppose that the agent's policy π is parametrized by some parameters $\theta \in \Theta$. We consider a time-consistent, Markov, dynamic convex risk measure $\{\rho_{t,T}\}_{t \in \mathcal{T}}$, that is, the one-step conditional risk measures ρ_t are static convex risk measures, real-valued, and lower semicontinuous. Using Equation (6) from Theorem 4.12, we aim to solve the following T -period risk-sensitive RL problem:

$$\min_{\theta} \rho_{0,T}(Z^{\theta}) = \min_{\theta} \rho_0 \left(c_0^{\theta} + \rho_1 \left(c_1^{\theta} + \dots + \rho_{T-2} \left(c_{T-2}^{\theta} + \rho_{T-1}(c_{T-1}^{\theta}) \right) \right) \dots \right), \quad (P1)$$

where $c_t^{\theta} = c(s_t, a_t^{\theta}, s_{t+1}^{\theta})$ is a $\mathcal{F}_{t+1}$ -measurable random cost and the trajectory may be modulated by the policy π^{θ} —that is why we include a θ index to actions and states. In the sequel, we may omit the superscript θ when obvious for readability purposes.

Let us denote transition probabilities by $\mathbb{P}^\theta(a, s' | s_t = s) := \mathbb{P}(s' | s, a) \pi^\theta(a | s_t = s)$ and ξ -weighted conditional expectations by $\mathbb{E}_t^\xi[Z] := \sum_{(a, s')} \xi(a, s') \mathbb{P}^\theta(a, s' | s_t) Z(a, s')$ for any $t \in \mathcal{T}$.

All one-step conditional risk measures ρ_t of the dynamic risk measure $\{\rho_{t,T}\}_{t \in \mathcal{T}}$ satisfy the assumptions in Theorem 4.7, and thus using the dual representation from Theorem 4.7, the problem in Equation (P1) may be written equivalently as

$$\begin{aligned} \min_{\theta} \max_{\xi_0 \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s_0 = s_0))} \left\{ \mathbb{E}_0^{\xi_0} \left[c_0^\theta + \max_{\xi_1 \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s_1 = s_1))} \left\{ \mathbb{E}_1^{\xi_1} \left[c_1^\theta + \dots \right. \right. \right. \right. \\ \left. \left. \left. + \max_{\xi_{T-1} \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s_{T-1} = s_{T-1}))} \left\{ \mathbb{E}_{T-1}^{\xi_{T-1}} [c_{T-1}^\theta] - \rho_{T-1}^*(\xi_{T-1}) \right\} \dots \right] - \rho_1^*(\xi_1) \right\} \right] - \rho_0^*(\xi_0) \right\}. \end{aligned} \quad (\text{P2})$$

Assumption 5.1. We restrict to convex risk measures ρ such that the risk envelope $\mathcal{U}$ may be written as

$$\mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s)) = \left\{ \xi : \sum_{(a, s')} \xi(a, s') \mathbb{P}^\theta(a, s' | s) = 1, \xi \geq 0, g_e(\xi, \mathbb{P}^\theta) = 0, \forall e \in \mathcal{E}, f_i(\xi, \mathbb{P}^\theta) \leq 0, \forall i \in \mathcal{I} \right\}, \quad (7)$$

where $g_e(\xi, \mathbb{P})$ are affine functions wrt ξ , $f_i(\xi, \mathbb{P})$ are convex functions wrt ξ , and $\mathcal{E}$ (resp. $\mathcal{I}$) denotes the finite set of equality (resp. inequality) constraints. Furthermore, for any given $\xi \in \{\xi : \sum_{(a, s')} \xi(a, s') = 1, \xi \geq 0\}$, $g_e(\xi, p)$ and $f_i(\xi, p)$ are twice differentiable in p , and there exists $M > 0$ such that for all $(a, s') \in \mathcal{A} \times \mathcal{S}$ we have

$$\max \left\{ \max_{i \in \mathcal{I}} \left| \frac{df_i(\xi, p)}{dp(a, s')} \right|, \max_{e \in \mathcal{E}} \left| \frac{dg_e(\xi, p)}{dp(a, s')} \right| \right\} \leq M.$$

As noted by Tamar et al. (2016), “all coherent risk measures we are aware of in the literature are already captured by [that] risk envelope.” Note, however, here we use convex (which subsumes coherent) risk measures, while still keeping the structure of the risk envelope induced by this observation for coherent risk measures. This risk envelope still covers most cases of convex risk measures typically of interest in the literature such as entropic risk measures, with Shannon entropy as the convex penalty ρ^* , and more generally expected utility risk measures (see, e.g., Rockafellar & Uryasev, 2013). Hence, we view the restrictions of Assumption 5.1 on the explicit form of $\mathcal{U}$ as being not too restrictive.

We now wish to derive DP equations with a view of solving problems of the form (P2). To this end, define the *value function* V as the running risk-to-go

$$V_t(s; \theta) := \rho_t \left(c_t^\theta + \rho_{t+1}(c_{t+1}^\theta + \dots + \rho_{T-1}(c_{T-1}^\theta)) \mid s_t = s \right), \quad (8)$$

for all $s \in \mathcal{S}$ and $t \in \mathcal{T}$. Here, as is standard in the RL literature, the value function is to be understood as a tool to evaluate the quality of a given policy at any state of the MDP. In our case, it represents the risk at a certain time when being in a specific state and following the policy π^θ . The DP equations (DPE) for a specific policy π^θ are

$$V_{T-1}(s; \theta) = \rho_{T-1} \left(c_{T-1}^\theta \mid s_{T-1} = s \right), \quad \text{and} \quad (9a)$$

$$V_t(s; \theta) = \rho_t \left(c_t^\theta + V_{t+1}(s_{t+1}^\theta; \theta) \mid s_t = s \right), \quad (9b)$$

for any $s \in S$ and $t \in \mathcal{T} \setminus \{T-1\}$. The DPE allows us to recursively assess the full extent of the risk associated with a fixed policy π_θ , and in particular, the recursion can be seen to include the risk associated with the current (random) cost and the one-step ahead running risk-to-go, both of which depend explicitly on the next state. Using the dual representation in Theorem 4.7, the DPE in Equations (9a) and (9b) may be written as

$$V_{T-1}(s; \theta) = \max_{\xi \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot \mid s_{T-1}=s))} \left\{ \mathbb{E}_{T-1,s}^\xi [c_{T-1}^\theta] - \rho_{T-1}^*(\xi) \right\}, \quad \text{and} \quad (10a)$$

$$V_t(s; \theta) = \max_{\xi \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot \mid s_t=s))} \left\{ \mathbb{E}_{t,s}^\xi [c_t^\theta + V_{t+1}(s_{t+1}^\theta; \theta)] - \rho_t^*(\xi) \right\}, \quad (10b)$$

where $\mathbb{E}_{t,s}^\xi[\cdot]$ denotes the conditional expectation $\mathbb{E}^\xi[\cdot \mid s_t = s]$.

We aim to optimize the value function V over policies π^θ using a *policy gradient approach* (Sutton et al., 2000). Policy gradient proposes to optimize by updating parameters of the policy using the update rule $\theta \leftarrow \theta + \eta \nabla_\theta V(\cdot; \theta)$, which requires an estimation of the gradient. In order to obtain the gradient of V , we need an additional assumption on the transition probabilities, more specifically on the policy.

Assumption 5.2. We suppose that the logarithm of transition probabilities $\log \mathbb{P}^\theta(a, s' \mid s)$ is a differentiable function in θ when $\mathbb{P}^\theta(a, s' \mid s) \neq 0$, and its gradient wrt θ is bounded for any $(a, s) \in A \times S$.

Assumption 5.2 is written for completeness, as it is a common restriction with policy gradient methods (see, e.g., Sutton et al., 2000). It ensures that the agent chooses a differentiable policy π^θ so that policy gradient can be applied. Indeed, the transition probabilities depend on the policy's parameters θ only through the policy π^θ itself—therefore, the gradient of the log-probability of a transition $\nabla_\theta \log \mathbb{P}^\theta$ may be represented as

$$\begin{aligned} \nabla_\theta \log \mathbb{P}^\theta(a, s' \mid s_{t-1} = s) &= \nabla_\theta (\log \mathbb{P}(s' \mid s, a) + \log \pi^\theta(a \mid s_t = s)) \\ &= \nabla_\theta \log \pi^\theta(a \mid s_t = s). \end{aligned} \quad (11)$$

Theorem 5.3 provides the gradient formulae in our proposed methodology.

Theorem 5.3 (Gradient of V). *Let Assumptions 13 and 14 hold. For any state $s \in S$, the gradient of the value function at period $T-1$ is then*

$$\begin{aligned} \nabla_\theta V_{T-1}(s; \theta) &= \mathbb{E}_{T-1}^{\xi^*} \left[\left(c(s, a_{T-1}^\theta, s_T^\theta) - \lambda^* \right) \nabla_\theta \log \pi^\theta(a_{T-1}^\theta \mid s_{T-1} = s) \right] \\ &\quad - \nabla_\theta \rho_{T-1}^*(\xi^*) - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_\theta g_e(\xi^*, \mathbb{P}^\theta)) - \sum_{i \in I} (\lambda^{*, I}(i) \nabla_\theta f_i(\xi^*, \mathbb{P}^\theta)), \end{aligned} \quad (12a)$$

and the gradient of the value function at periods $t \in \mathcal{T} \setminus \{T - 1\}$ is

$$\begin{aligned} \nabla_{\theta} V_t(s; \theta) = & \mathbb{E}_t^{\xi^*} \left[(c(s, a_t^{\theta}, s_{t+1}^{\theta}) + V_{t+1}(s_{t+1}^{\theta}; \theta) - \lambda^*) \nabla_{\theta} \log \pi^{\theta}(a_t^{\theta} | s_t = s) + \nabla_{\theta} V_{t+1}(s_{t+1}^{\theta}; \theta) \right] \\ & - \nabla_{\theta} \rho_t^*(\xi^*) - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_{\theta} g_e(\xi^*, \mathbb{P}^{\theta})) - \sum_{i \in \mathcal{I}} (\lambda^{*, \mathcal{I}}(i) \nabla_{\theta} f_i(\xi^*, \mathbb{P}^{\theta})), \end{aligned} \quad (12b)$$

where $(\xi^*, \lambda^*, \lambda^{*, \mathcal{E}}, \lambda^{*, \mathcal{I}})$ is any saddle-point of the Lagrangian function of Equations (10a) and (10b), respectively.

Proof of Theorem 5.3. In order to have an expression for the gradient of the value function, we first compute the gradient of the last period and then obtain the recursive relation for subsequent periods. Using Theorem 4.7 and Assumption 5.1, the Lagrangian of the maximization problem in Equation (10a) (with $t = T - 1$) for any state $s \in \mathcal{S}$ can be written as

$$\begin{aligned} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = & \sum_{(a, s')} \xi(a, s') \mathbb{P}^{\theta}(a, s' | s_{T-1} = s) c_{T-1}(s, a, s') - \rho_{T-1}^*(\xi) \\ & - \lambda \left(\sum_{(a, s')} \xi(a, s') \mathbb{P}^{\theta}(a, s' | s_{T-1} = s) - 1 \right) \\ & - \sum_{e \in \mathcal{E}} (\lambda^{\mathcal{E}}(e) g_e(\xi, \mathbb{P}^{\theta})) - \sum_{i \in \mathcal{I}} (\lambda^{\mathcal{I}}(i) f_i(\xi, \mathbb{P}^{\theta})). \end{aligned} \quad (13)$$

By the convexity of Equation (10a) and Assumption 5.1, L^{θ} in Equation (13) has at least one saddle-point. We emphasize here that the saddle-points $(\xi^*, \lambda^*, \lambda^{*, \mathcal{E}}, \lambda^{*, \mathcal{I}})$ depend on the state s . Using Slater's condition and the convexity of the problem, strong duality holds, that is,

$$V_{T-1}(s; \theta) = \max_{\xi \geq 0} \min_{\lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}). \quad (14)$$

We then recall a what is known in the ML literature as the “likelihood-ratio” trick, which states that

$$p(x; \theta) \nabla_{\theta} \log(p(x; \theta)) = \nabla_{\theta} p(x; \theta). \quad (15)$$

Next, we apply the *Envelope theorem for saddle-point problems* (see Theorem 4 and Corollary 5 in Milgrom & Segal, 2002)—which relies on the equidifferentiability of the objective function and the absolute continuity of its gradient. These properties are satisfied as we work under Assumption 5.1. Under these assumptions, we have

$$\max_{\xi \geq 0} \min_{\lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = \max_{\xi \geq 0} \min_{\lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}} L^0(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) + \int_0^{\theta} \nabla_{\theta} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \Big|_{\theta=u} du,$$

which implies that the gradient wrt θ of the objective function $V_{T-1}(s; \theta)$ equals to the gradient wrt θ of the Lagrangian evaluated at one of its saddle-points.

Using Assumption 5.2, the Envelope theorem, the Lagrangian in Equation (13), the strong duality in Equation (14) and the “likelihood-ratio” trick in Equation (15), we obtain that

$$\begin{aligned}
 \nabla_{\theta} V_{T-1}(s; \theta) &= \nabla_{\theta} \max_{\xi \geq 0} \min_{\lambda, \lambda^{\mathcal{E}}, \lambda^I} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^I) \\
 &= \nabla_{\theta} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^I) \Big|_{\xi^*, \lambda^*, \lambda^{*, \mathcal{E}}, \lambda^{*, I}} \\
 &= \sum_{(a, s')} \xi^*(a, s') c_{T-1}(s, a, s') \nabla_{\theta} \mathbb{P}^{\theta}(a, s' | s_{T-1} = s) - \nabla_{\theta} \rho_{T-1}^*(\xi^*) \\
 &\quad - \sum_{(a, s')} \lambda^* \xi^*(a, s') \nabla_{\theta} \mathbb{P}^{\theta}(a, s' | s_{T-1} = s) \\
 &\quad - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_{\theta} g_e(\xi^*, \mathbb{P}^{\theta})) - \sum_{i \in I} (\lambda^{*, I}(i) \nabla_{\theta} f_i(\xi^*, \mathbb{P}^{\theta})) \\
 &= \mathbb{E}_{T-1}^{\xi^*} [(c_{T-1}(s, a_{T-1}^{\theta}, s_T^{\theta}) - \lambda^*) \nabla_{\theta} \log \pi^{\theta}(a_{T-1}^{\theta} | s_{T-1} = s)] \\
 &\quad - \nabla_{\theta} \rho_{T-1}^*(\xi^*) - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_{\theta} g_e(\xi^*, \mathbb{P}^{\theta})) - \sum_{i \in I} (\lambda^{*, I}(i) \nabla_{\theta} f_i(\xi^*, \mathbb{P}^{\theta})). \quad (16)
 \end{aligned}$$

The gradient for others periods is obtained in a similar manner. The Lagrangian of the problem in Equation (10b) (with $t = T - 2, \dots, 0$) is

$$\begin{aligned}
 L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^I) &= \sum_{(a, s')} \xi(a, s') \mathbb{P}^{\theta}(a, s' | s_t = s) (c_t(s, a, s') + V_{t+1}(s'; \theta)) - \rho_t^*(\xi) \\
 &\quad - \lambda \left(\sum_{(a, s')} \xi(a, s') \mathbb{P}^{\theta}(a, s' | s_t = s) - 1 \right) \\
 &\quad - \sum_{e \in \mathcal{E}} (\lambda^{\mathcal{E}}(e) g_e(\xi, \mathbb{P}^{\theta})) - \sum_{i \in I} (\lambda^I(i) f_i(\xi, \mathbb{P}^{\theta})). \quad (17)
 \end{aligned}$$

As the value function depends on the policy π^{θ} , the gradient formula will have an additional term. We obtain

$$\begin{aligned}
 \nabla_{\theta} V_t(s; \theta) &= \nabla_{\theta} L^{\theta}(\xi, \lambda, \lambda^{\mathcal{E}}, \lambda^I) \Big|_{\xi^*, \lambda^*, \lambda^{*, \mathcal{E}}, \lambda^{*, I}} \\
 &= \sum_{(a, s')} \xi^*(a, s') \mathbb{P}^{\theta}(a, s' | s_t = s) \nabla_{\theta} V_{t+1}(s'; \theta) - \sum_{(a, s')} \lambda^* \xi^*(a, s') \nabla_{\theta} \mathbb{P}^{\theta}(a, s' | s_t = s) \\
 &\quad + \sum_{(a, s')} \xi^*(a, s') (c_t(s, a, s') + V_{t+1}(s'; \theta)) \nabla_{\theta} \mathbb{P}^{\theta}(a, s' | s_t = s) \\
 &\quad - \nabla_{\theta} \rho_t^*(\xi^*) - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_{\theta} g_e(\xi^*, \mathbb{P}^{\theta})) - \sum_{i \in I} (\lambda^{*, I}(i) \nabla_{\theta} f_i(\xi^*, \mathbb{P}^{\theta})) \\
 &= \mathbb{E}_t^{\xi^*} [(c_t(s, a_t^{\theta}, s_{t+1}^{\theta}) + V_{t+1}(s_{t+1}^{\theta}; \theta) - \lambda^*) \nabla_{\theta} \log \pi^{\theta}(a_t^{\theta} | s_t = s) + \nabla_{\theta} V_{t+1}(s_{t+1}^{\theta}; \theta)] \\
 &\quad - \nabla_{\theta} \rho_t^*(\xi^*) - \sum_{e \in \mathcal{E}} (\lambda^{*, \mathcal{E}}(e) \nabla_{\theta} g_e(\xi^*, \mathbb{P}^{\theta})) - \sum_{i \in I} (\lambda^{*, I}(i) \nabla_{\theta} f_i(\xi^*, \mathbb{P}^{\theta})). \quad (18)
 \end{aligned}$$

This concludes the proof. $\square$

While the term $\rho_t^*(\xi^*)$ appears at first not to depend on the policy, and therefore the term $\nabla_{\theta} \rho_t^*(\xi^*)$ in Theorem 5.3 appears to vanish, this is not necessarily so. To see why, let us consider convex penalties of the form $\rho_t^*(\xi) = \mathbb{E}_t^{\xi}[f_t(\xi)]$ for convex functions $f_t : \mathcal{Z}^* \rightarrow \mathbb{R}$. In this case, using the Envelope theorem (Milgrom & Segal, 2002) and the “likelihood-ratio” trick—the derivation is similar to the proof of Theorem 5.3—the gradient wrt the policy is given by

$$\nabla_{\theta} \rho_t^*(\xi^*) = \mathbb{E}_t^{\xi^*}[f_t(\xi^*) \nabla_{\theta} \log \pi^{\theta}(a_t^{\theta} | s_t = s)].$$

While we do not restrict to the above specific form for ρ_t^* , this result illustrates why, in general, this contribution to the gradient cannot be ignored.

6 | ACTOR–CRITIC ALGORITHM

In this section, we provide details on our proposed algorithm and the architecture of the implemented objects. Our policy gradient algorithm has an actor–critic style (Konda & Tsitsiklis, 2000), in the sense that we must learn two functions, and we do so in an alternating fashion: (i) a value function V , which plays the role of the *critic*; and (ii) the policy π , which plays the role of the *actor*. Actor–critic algorithms are on-policy policy search methods that maintain an estimate of a value function, which is then used to update the agent’s policy parameters. Such algorithms have been developed in the RL community for their ability to find optimal policies using low variance gradient estimates (Grondman et al., 2012). To obtain an approximation of the optimal policy, we perform the steps described in Algorithm 1.

ALGORITHM 1 Main steps.

Input: ANNs for value function V^{ϕ} and policy π^{θ} ,
number of episodes N , transitions M , epochs K ,
mini-batch size B

- 1 Set initial guesses for V^{ϕ} , π^{θ} ;
- 2 Initialize environment and optimizers;
- 3 **for** each epoch $\kappa = 1, \dots, K$ **do**
- 4 Simulate trajectories under π^{θ} ;
- 5 Freeze $\tilde{\pi} = \pi^{\theta}$;
- 6 *Critic:* Estimate V^{ϕ} using $\tilde{\pi}$
 Algorithm 2(V^{ϕ} , $\tilde{\pi}$, N , M , K , B);
- 7 Freeze $\tilde{V} = V^{\phi}$;
- 8 *Actor:* Update π^{θ} using $\tilde{V}$
 Algorithm 3($\tilde{V}$, π^{θ} , N , M , K , B);
- 9 Store results;

Output: Optimal $\pi^{\theta} \approx \pi^{\theta^*}$ and $V^{\phi} \approx V(s; \theta^*)$

We propose to use function approximations, more specifically *neural network structures*, as they are useful when dealing with continuous state–action spaces and are known to be universal approximators. In recent years, deep neural network modeling has shown remarkable success in

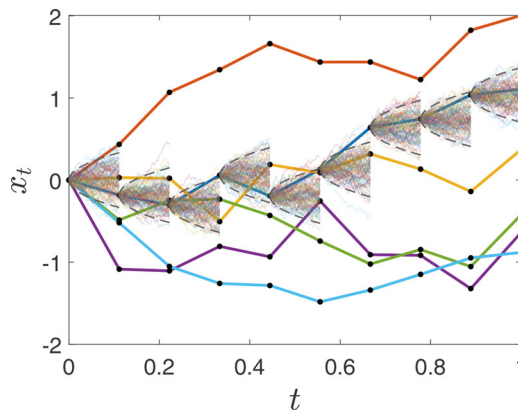


FIGURE 2 Representation of (outer) episodes and (inner) transitions for the simulation upon simulation framework. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1111/mml.12888)]

approximating complex functions (see, e.g., Goodfellow et al., 2016; LeCun et al., 2015; Silver et al., 2016), especially in financial mathematics (see, e.g., Al-Arabi et al., 2018; Casgrain et al., 2022; Cuchiero et al., 2020; Campbell et al., 2021; Carmona & Laurière, 2021; Hu, 2019; Horvath et al., 2021; Hambly et al., 2021; Ning et al., 2021). The use of compositions of simple functions (usually referred to as propagation and activation functions) through several layers approximates complicated functions to arbitrary accuracy (with arbitrarily large structures). Neural networks also avoid the curse of dimensionality issue of representing nonlinear functions in high dimensions.

There are several approaches for modeling the policy and value function with neural network structures. We consider a policy π characterized by a single (fully connected, multilayered feed forward) artificial neural network (ANN) with parameters θ , which takes a state s and time t as inputs and outputs a distribution over the space of actions $\mathcal{A}$. We also suppose that the value function V is characterized by another single (fully connected, multilayered feed forward) ANN with parameters ϕ . $V_t^\phi(s; \theta)$ approximates the value function when the system is in state s during the period t under policy π^θ , previously denoted $V_t(s; \theta)$ in Equations (10a) and (10b). We, therefore, refer to the policy and value function, respectively, by π^θ and V^ϕ .

Our proposed algorithm uses a *nested simulation* or *simulation upon simulation* approach, where we simulate transitions at each visited state. This simulation upon simulation approach results in full (outer) episodes and batches of (inner) transitions for every state, as illustrated in Figure 2. This allows us to easily compute and estimate quantities of interest for each state, for example, saddle-points or one-step conditional risk measures. Such nested simulation approaches are computationally expensive—part of our future work aims to develop a more efficient algorithm when simulations are costly.

We next provide additional details on our actor–critic algorithm in Algorithm 1, that is, how to (i) estimate the value function in Section 6.1 (step 5), and (ii) update the policy in Section 6.2 (step 7). Additional implementation details are provided in Section A.

6.1 | Value function estimation

The value function V^ϕ may be estimated by using the recursion in Equations (10a) and (10b). We use the simulation upon simulation approach mentioned previously. More precisely, when

ALGORITHM 2 Estimation of the value function V .

Input: Value function V^ϕ , policy π^θ , number of episodes N , transitions M , epochs K , batch size B

- 1 **for** each epoch $k = 1, \dots, K$ **do**
- 2 Set the gradients of V^ϕ to zero;
- 3 Sample B states $s_t^{(b)}$, $b = 1, \dots, B$, $t \in \mathcal{T}$;
- 4 Obtain from π^θ the associated transitions $(a_t^{(b,m)}, s_{t+1}^{(b,m)}, c_t^{(b,m)})$, $m = 1, \dots, M$;
- 5 **for** each state $b = 1, \dots, B$, $t \in \mathcal{T}$ **do**
- 6 Compute the predicted values $\hat{v}_t^b = V_t^\phi(s_t^{(b)}; \theta)$;
- 7 **if** $t = T - 1$ **then**
- 8 Set the target value as

$$v_{T-1}^b = \max_{\xi \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s_{T-1} = s_{T-1}^{(b)}))} \left\{ \mathbb{E}_{T-1, s_{T-1}^{(b)}}^\xi \left[c_{T-1}^{(b,m)} \right] + \rho_{T-1}^*(\xi) \right\};$$
- 9 **else**
- 10 Set the target value as

$$v_t^b = \max_{\xi \in \mathcal{U}(\mathbb{P}^\theta(\cdot, \cdot | s_t = s_t^{(b)}))} \left\{ \mathbb{E}_{t, s_t^{(b)}}^\xi \left[c_t^{(b,m)} + V_{t+1}^\phi(s_{t+1}^{(b,m)}; \theta) \right] + \rho_t^*(\xi) \right\};$$
- 11 Compute the expected square loss between v_t^b and $\hat{v}_t^b$;
- 12 Update ϕ by performing an Adam optimizer step;

Output: An estimate of the value function $V_t^\phi(s; \theta) \approx V_t(s; \theta)$

sampling a certain state s_t , we also generate M (inner) transitions from the policy π^θ to obtain the tuples $(s_t, a_t^{(m)}, s_{t+1}^{(m)}, c_t^{(m)})$, $m = 1, \dots, M$. We can then estimate the one-step conditional risk measures for any given state using those additional transitions. To update the value function, we perform this simulation process for a mini-batch of states, compute the predicted (i.e., output of V^ϕ) and target values (i.e., Equations 10a and 10b), and calculate the expected square loss between these values. We update the parameters using the Adam optimizer (Kingma & Ba, 2014) and repeat this process for several epochs in order to provide a good approximation of the value function. We recall that the policy π^θ is fixed while we optimize V^ϕ . The algorithm is provided in Algorithm 2.

A powerful result for neural networks structures is the *universal approximation theorem*—see, for example, Cybenko (1989); Hornik (1991); Leshno et al. (1993); Pinkus (1999).

Theorem 6.1 Universal approximation (Cybenko, 1989). *Let $d_1, d_2 \in \mathbb{N}$ and σ be an activation function. Then σ is not a polynomial iff for any continuous function $f : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$, any compact subset $K \subset \mathbb{R}^{d_1}$ and any $\epsilon > 0$, there exists a neural network $\hat{f}_\epsilon : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^{d_2}$ with representation $\hat{f}_\epsilon = W_2 \circ \sigma \circ W_1$ such that $\sup_{x \in K} \|f(x) - \hat{f}_\epsilon(x)\| < \epsilon$.*

We prove next that for a fixed policy π^θ , we can approximate its corresponding value function $V_t(s; \theta)$ with an ANN using the procedure devised in Algorithm 2. Theorem 6.2 follows from the universal approximation theorem.

Theorem 6.2 (Approximation of V). *Let π^θ denote a fixed policy, with corresponding value function as defined in Equation (8), which we denote $V_t(s; \theta)$. Then for any $\epsilon^* > 0$, there exists an ANN $V_t^\phi : \mathcal{S} \rightarrow \mathbb{R}$ such that $\text{ess sup}_{s \in \mathcal{S}} \|V_t(s; \theta) - V_t^\phi(s; \theta)\| < \epsilon^*$, for any $t \in \mathcal{T}$.*

Proof of Theorem 6.2. First, we prove a lemma that states convex risk measures are absolutely continuous since they are in fact monetary. In what follows, the norm is to be understood as the ∞ -norm.

Lemma 6.3. *Monetary one-step conditional risk measures ρ_t are absolutely continuous a.s.*

Proof of Lemma 6.3. Indeed, starting from the inequality $Z \leq W + \|Z - W\|$, where $Z, W \in \mathcal{Z}_{t+1}$, and using the monotonicity and translation invariance properties, we have

$$\rho_t(Z) \leq \rho_t(W + \|Z - W\|) \Rightarrow \rho_t(Z) - \rho_t(W) \leq \|Z - W\|. \quad (19)$$

Repeating this with $W \leq Z + \|Z - W\|$ yields to

$$\text{ess sup} \|\rho_t(Z) - \rho_t(W)\| \leq \|Z - W\|. \quad (20)$$

Therefore, convex risk measures are Lipschitz continuous a.s. wrt the essential supremum norm, and hence they are absolutely continuous a.s. $\square$

Next, recall that the value function given in Equation (8) is a dynamic convex risk measure, and therefore may be written recursively with the DPE in Equations (9a) and (9b) as

$$V_t(s; \theta) = \rho_t \left(c_t^\theta + V_{t+1}(s_{t+1}^\theta; \theta) \mid s_t = s \right). \quad (21)$$

We prove that the ANN ϕ approximate the value function by induction. Without loss of generality, let us consider the case where the first dimension of the state space S corresponds to the time $t \in \mathcal{T}$. At the period $T - 1$, we have that

$$V_{T-1}(s; \theta) = \rho_{T-1} \left(c_{T-1}^\theta \mid s_{T-1} = s \right). \quad (22)$$

This is a convex risk measure, which is absolutely continuous a.s. by Lemma 6.3. Using the universal approximation theorem given in Theorem 6.1, we obtain that for any $\epsilon_{T-1} > 0$, there exists a neural net ϕ_{T-1} such that

$$\text{ess sup}_{s \in S} \left\| V_{T-1}(s; \theta) - V_{T-1}^{\phi_{T-1}}(s; \theta) \right\| < \epsilon_{T-1}. \quad (23)$$

This proves the base case of our proof by induction.

For the induction step, the ANN ϕ_t approximates the value function at period t as long as the value function at the next period $t + 1$ is adequately approximated. Assume that for any $\epsilon_{t+1} > 0$, there exists an ANN ϕ_{t+1} such that

$$\text{ess sup}_{s \in S} \left\| V_{t+1}(s; \theta) - V_{t+1}^{\phi_{t+1}}(s; \theta) \right\| < \epsilon_{t+1}. \quad (24)$$

Using the translation invariance, the triangle inequality, Lemma 6.3 and Equation (24), we have

$$\begin{aligned} & \text{ess sup}_{s \in S} \left\| V_t(s; \theta) - V_t^{\phi_t}(s; \theta) \right\| \\ &= \text{ess sup}_{s \in S} \left\| \rho_t \left(c_t^\theta + V_{t+1}(s_{t+1}^\theta; \theta) \mid s_t = s \right) - V_t^{\phi_t}(s; \theta) \right\| \end{aligned}$$

$$\begin{aligned}
&\leq \operatorname{ess\,sup}_{s \in S} \left\| \rho_t \left(c_t + V_{t+1}^{\phi_{t+1}}(s_{t+1}^\theta; \theta) \mid s_t = s \right) - V_t^{\phi_t}(s; \theta) \right\| \\
&\quad + \operatorname{ess\,sup}_{s \in S} \left\| \rho_t \left(c_t^\theta + V_{t+1}(s_{t+1}^\theta; \theta) \mid s_t = s \right) - \rho_t \left(c_t + V_{t+1}^{\phi_{t+1}}(s_{t+1}^\theta; \theta) \mid s_t = s \right) \right\| \\
&\leq \operatorname{ess\,sup}_{s \in S} \left\| \rho_t \left(c_t + V_{t+1}^{\phi_{t+1}}(s_{t+1}^\theta; \theta) \mid s_t = s \right) - V_t^{\phi_t}(s; \theta) \right\| \\
&\quad + \operatorname{ess\,sup}_{s \in S} \left\| V_{t+1}(s; \theta) - V_{t+1}^{\phi_{t+1}}(s; \theta) \right\| \\
&< \operatorname{ess\,sup}_{s \in S} \left\| \rho_t \left(c_t + V_{t+1}^{\phi_{t+1}}(s_{t+1}^\theta; \theta) \mid s_t = s \right) - V_t^{\phi_t}(s; \theta) \right\| + \epsilon_{t+1}.
\end{aligned} \tag{25}$$

Using the universal approximation theorem on Equation (25), we get for any $\epsilon_t > 0$, there exists an ANN ϕ_t such that

$$\operatorname{ess\,sup}_{s \in S} \left\| V_t(s; \theta) - V_t^{\phi_t}(s; \theta) \right\| < \epsilon_t + \epsilon_{t+1} \tag{26}$$

We apply this argument recursively for any period $t \in \mathcal{T}$, which completes the proof by induction.

Finally, we need an additional lemma to prove that one can approximate the ensemble of ANNs $\{V_t^{\phi_t}(s; \theta)\}_{t \in \mathcal{T}}$ with a single ANN $V_t^\phi(s; \theta)$.

Lemma 6.4. *Suppose $\{\hat{f}_t(x)\}_{t \in \mathcal{T}}$, $x \in K \subseteq \mathbb{R}^{d_1}$ is an ensemble of a finite number of ANNs. Then for any $\epsilon > 0$, there exists an ANN $\hat{g}_t(x)$ such that $\operatorname{ess\,sup}_{x \in K} \|\hat{f}_t(x) - \hat{g}_t(x)\| < \epsilon$, $\forall t \in \mathcal{T}$.*

Proof of Lemma 6.4. Throughout the proof, we label the ensemble using $\mathcal{T} := \{0, 1, \dots, T\}$ without loss of generality. We create an extension of the ensemble $\{\hat{f}_t(x)\}_{t \in \mathcal{T}}$ to obtain a function $\tilde{f}_t(x)$ that is absolutely continuous a.s. wrt both x and t .

First, note that, by construction, $\hat{f}_t(x)$ s are absolutely continuous a.s. wrt x . Between each pair of indices t and $t + 1$, we extend the function via a polynomial interpolation on the compact set $[t, t + 1]$. Since we work on a closed interval and polynomials are continuously differentiable, the interpolation $\tilde{f}_t(x)$ must be Lipschitz, and thus absolutely continuous. We then use this argument on all pairs of indices, which shows that there exists a function $\tilde{f}_t(x)$ such that (i) $\tilde{f}_t(x) = \hat{f}_t(x)$ for any $x \in K$ and $t \in \mathcal{T}$, and (ii) $\tilde{f}$ is absolutely continuous on $K \times [0, T]$.

Using the universal approximation theorem, for any $\epsilon > 0$, there exists an ANN $\hat{g}_t(x)$ such that

$$\operatorname{ess\,sup}_{(x,t) \in K \times [0,T]} \left\| \tilde{f}_t(x) - \hat{g}_t(x) \right\| < \epsilon. \tag{27}$$

This also holds for $\hat{f}_t(x)$, which yields the desired result. $\square$

Using the triangle inequality and Lemma 6.4, for any $\hat{\epsilon} > 0$, there exists an ANN ϕ such that

$$\operatorname{ess\,sup}_{s \in S} \left\| V_t(s; \theta) - V_t^\phi(s; \theta) \right\| < \hat{\epsilon} + \epsilon_t, \quad \forall t \in \mathcal{T}. \tag{28}$$

As Theorem 6.1 is valid for any $\epsilon > 0$, we can perform the training procedure in order to construct a sequence of ϵ_t , $t \in \mathcal{T}$ that satisfies a global error ϵ^* , such as $\hat{\epsilon} + \sum \epsilon_t < \epsilon^*$. $\square$

6.2 | Update of the policy

The update of the policy is done using the gradients provided in Equations (12a) and (12b) of Theorem 5.3. Some points worth mentioning concern the policy, the saddle-points, and the gradient formula. When implementing the algorithm, we ensure that the policy uses the so-called *reparametrization trick*, which allows the existence of pathwise gradient estimators from random samples. Usually there are three basic approaches to perform a reparametrization:

- (i) Use a location-scale transformation—we can view the standard random variable as an auxiliary variable Z (such as $\mathcal{N}(0, 1)$) and simulate $\mu^\theta + Z\sigma^\theta$ (distributed as $\mathcal{N}(\mu^\theta, \sigma^\theta)$);
- (ii) Use the inverse cumulative distribution function—if it is tractable, we can use the inverse transform sampling method to simulate realizations from uniform random variables;
- (iii) Use a transformation of auxiliary variables—common examples are the log-normal distribution, which can be expressed by exponentiation of a Gaussian distribution, and the gamma distribution, which can be rewritten as a sum of exponentially distributed random variables.

Also since we assume that the form of the risk envelope is known in an explicit form in Assumption 5.1, we can obtain a saddle-point $(\xi^*, \lambda^*, \lambda^{*,\mathcal{E}}, \lambda^{*,\mathcal{I}})$ of the Lagrangian of Equations (10a) and (10b) for any given risk measure, either analytically or using a sample average approximation (see Chapter 5 of Shapiro et al., 2014). The approach to obtain these saddle-points is illustrated for common risk measures in Section 7.

We recall that the value function V^ϕ is fixed while we optimize π^θ . When we compute the gradient of the value function to optimize the policy, we fix the parameters of the value function ϕ . This can be interpreted as taking a copy of the ANN structure, which implies that the value function used in the actor part of the algorithm does not depend explicitly on θ . Therefore, the additional expectation of the gradient of the value function at $t + 1$ in Equation (12b) vanishes. The value function gradient is then estimated averaging over a batch of states and different periods. The algorithm is given in Algorithm 3.

7 | EXPERIMENTS

In this section, we provide three illustrative examples to understand the potential gain of using dynamic risk measures in RL, and more specifically the advantages of our proposed approach on several examples. In our experiments, we consider several risk measures in order to compare their performance and highlight their differences.¹

The first risk measure we consider is the *dynamic expectation*, where the one-step conditional risk measures are $\rho_{\mathbb{E}}(Z) = \mathbb{E}[Z]$, which serves as a benchmark for the risk-neutral approach. It is a convex risk measure, and its saddle-point (ξ^*, λ^*) is given by $\xi^*(\omega) = 1$ and $\lambda^* = 0$.

¹We implemented more dynamic convex risk measures in our code available on Github, and users can easily add their own in the Python files—see Section A for a description of the code architecture.

ALGORITHM 3 Update of the policy π .

Input: Value function V^ϕ , policy π^θ , number of episodes N , transitions M , epochs K , batch size B

- 1 **for** each epoch $k = 1, \dots, K$ **do**
- 2 Set the gradients of π^θ to zero;
- 3 Sample B states $s_t^{(b)}$, $b = 1, \dots, B$, $t \in \mathcal{T}$;
- 4 Obtain from π^θ the associated transitions $(a_t^{(b,m)}, s_{t+1}^{(b,m)}, c_t^{(b,m)})$, $m = 1, \dots, M$;
- 5 **for** each state $b = 1, \dots, B$, $t \in \mathcal{T}$ **do**
- 6 Obtain $\hat{z}_t^{(b,m)} = \nabla_\theta \log \pi^\theta(a_t^{(b,m)} | s_t^{(b)})$ with the reparametrization trick;
- 7 Obtain $\hat{v}_{t+1}^{(b,m)} = V_{t+1}^\phi(s_{t+1}^{(b,m)}; \theta)$;
- 8 Get a saddle-point $(\xi^*, \lambda^*, \lambda^{*,\mathcal{E}}, \lambda^{*,\mathcal{I}})$ and compute $\xi_t^{*,(b,m)} = \xi^*(a_t^{(b,m)}, s_{t+1}^{(b,m)})$;
- 9 Obtain $\hat{\rho}_t^{(b)} = \nabla_\theta \rho_t^*(\xi^*)$, $\hat{g}_{e,t}^{(b)} = \nabla_\theta g_e(\xi^*, \mathbb{P}^\theta)$, and $\hat{f}_{i,t}^{(b)} = \nabla_\theta f_i(\xi^*, \mathbb{P}^\theta)$;
- 10 **if** $t = T - 1$ **then**
- 11 Calculate the gradient $\nabla_\theta V_t(s_t^{(b)}; \theta)$ from Eq. (12a)
- 12
$$\ell_t^{(b)} = \frac{1}{M} \sum_{m=1}^M \left(\xi_t^{*,(b,m)} \left(c_t^{(b,m)} - \lambda^* \right) \hat{z}_t^{(b,m)} - \hat{\rho}_t^{(b)} - \sum_{e \in \mathcal{E}} \lambda^{*,\mathcal{E}}(e) \hat{g}_{e,t}^{(b)} - \sum_{i \in \mathcal{I}} \lambda^{*,\mathcal{I}}(i) \hat{f}_{i,t}^{(b)} \right);$$
- 13 **else**
- 14 Calculate the gradient $\nabla_\theta V_t(s_t^{(b)}; \theta)$ from Eq. (12b)
- 15
$$\ell_t^{(b)} = \frac{1}{M} \sum_{m=1}^M \left(\xi_t^{*,(b,m)} \left(c_t^{(b,m)} + \hat{v}_{t+1}^{(b,m)} - \lambda^* \right) \hat{z}_t^{(b,m)} - \hat{\rho}_t^{(b)} - \sum_{e \in \mathcal{E}} \lambda^{*,\mathcal{E}}(e) \hat{g}_{e,t}^{(b)} - \sum_{i \in \mathcal{I}} \lambda^{*,\mathcal{I}}(i) \hat{f}_{i,t}^{(b)} \right);$$
- 16 Take the average $\ell = \frac{1}{BT} \sum_{b=1}^B \sum_{t=0}^{T-1} \ell_t^{(b)}$;
- 17 Update θ by performing an Adam optimizer step;

Output: An updated policy π^θ

The second risk measure is the *dynamic CVaR* with threshold $\alpha \in (0, 1)$, where the one-step conditional risk maps are

$$\rho_{\text{CVaR}}(Z; \alpha) = \sup_{\xi \in \mathcal{U}(\mathbb{P})} \{\mathbb{E}^\xi[Z]\}, \quad (29)$$

where

$$\mathcal{U}(\mathbb{P}) = \left\{ \xi : \sum_{\omega} \xi(\omega) \mathbb{P}(\omega) = 1, \xi \in \left[0, \frac{1}{\alpha}\right] \right\}. \quad (30)$$

The CVaR is a coherent risk measure widely used in the financial mathematics literature (Rockafellar et al., 2000). As shown in Shapiro et al. (2014), any saddle-point (ξ^*, λ^*) satisfies $\xi^*(\omega) = \frac{1}{\alpha}$ if $Z(\omega) > \lambda^*$ and $\xi^*(\omega) = 0$ otherwise, where λ^* is any $(1 - \alpha)$ -quantile of Z . Here, note that despite the static CVaR being time-inconsistent (see Cheridito and Stadje (2009)), the dynamic CVaR is time-consistent by construction.

The third risk measure is a *dynamic penalized CVaR* where we add a relative entropy term with respect to the uniform distribution. This is a convex but not coherent dynamic risk measure. The one-step conditional risk measures are given by

$$\rho_{\text{CVaR-p}}(Z; \alpha, \beta) = \sup_{\xi \in \mathcal{U}(\mathbb{P})} \{\mathbb{E}^\xi[Z] - \beta \mathbb{E}^\xi[\log \xi]\}, \quad \beta > 0, \quad (31)$$

with the same risk envelope given in Equation (30). Obtaining saddle-points is not as straightforward as with the CVaR, since it requires solving a convex optimization problem. The Lagrangian

with the risk envelope constraints is

$$\mathcal{L}(\xi, \lambda, \eta) = \sum_{\omega} \xi(\omega) \mathbb{P}(\omega) (Z(\omega) - \beta \log \xi(\omega)) - \lambda \left(\sum_{\omega} \xi(\omega) \mathbb{P}(\omega) - 1 \right) - \sum_{\omega} \eta(\omega) \left(\xi(\omega) - \frac{1}{\alpha} \right), \quad (32)$$

with $\lambda \in \mathbb{R}$ and $\eta(\omega) \geq 0$ for all ω . Setting the derivative of Equation (32) wrt $\xi(\omega_i)$ to zero leads to

$$\xi(\omega_i) = \exp \left(\frac{Z(\omega_i) - \lambda - \beta}{\beta} - \frac{\eta(\omega_i)}{\beta \mathbb{P}(\omega_i)} \right). \quad (33)$$

When imposing the constraint on the η 's on Equation (33), we obtain the following expression:

$$\xi(\omega_i) = \begin{cases} \exp \left(\frac{Z(\omega_i) - \lambda - \beta}{\beta} \right) & \text{if } \eta(\omega_i) = 0 \\ \frac{1}{\alpha} & \text{if } \eta(\omega_i) > 0 \end{cases}, \quad (34)$$

and that constraint is active when $Z(\omega_i) > -\beta \log(\alpha) + \beta + \lambda$. We combine Equation (34) with the constraint on λ to get

$$\sum_{i : Z(\omega_i) \leq -\beta \log(\alpha) + \beta + \lambda} \mathbb{P}(\omega_i) \left(\exp \left(\frac{Z(\omega_i) - \lambda - \beta}{\beta} \right) - \frac{1}{\alpha} \right) = 1 - \frac{1}{\alpha}. \quad (35)$$

Any saddle-point (ξ^*, λ^*) then satisfies $\xi^*(\omega) = \max(1/\alpha, e^{(Z(\omega_i) - \lambda^* - \beta)/\beta})$, where λ^* is a root of Equation (35).

We note here that the penalized CVaR contains both risk measures as special cases. Indeed, for $\beta = 0$, it reduces to the CVaR, while we recover the expectation as β tends to ∞ .

7.1 | Statistical arbitrage example

This collection of experiments is performed on an algorithmic trading environment problem. The agent begins each episode with zero inventory, and on each period, the agent wishes to trade quantities of an asset, whose price fluctuates according to some data-generating processes. For each period $t \in \mathcal{T}$, the agent observes the asset's price $S_t \in \mathbb{R}_+$ and their inventory $q_t \in (-q_{\max}, q_{\max})$, performs a trade $a_t^\theta \in (-a_{\max}, a_{\max})$, resulting in wealth $y_t \in \mathbb{R}$ according to

$$\begin{cases} y_0 = 0, \\ y_t = y_{t-1} - a_{t-1}^\theta S_{t-1} - \varphi(a_{t-1}^\theta)^2, & t = 1, \dots, T-1, \\ y_T = y_{T-1} - a_{T-1}^\theta S_{T-1} - \varphi(a_{T-1}^\theta)^2 + q_T S_T - \psi q_T^2, \end{cases} \quad (36)$$

with coefficients $\varphi = 0.005$ and $\psi = 0.5$ for the cost transactions and terminal penalty imposed by the market respectively. We suppose that $T = 5$, $q_{\max} = 5$, $a_{\max} = 2$, and the asset price follows

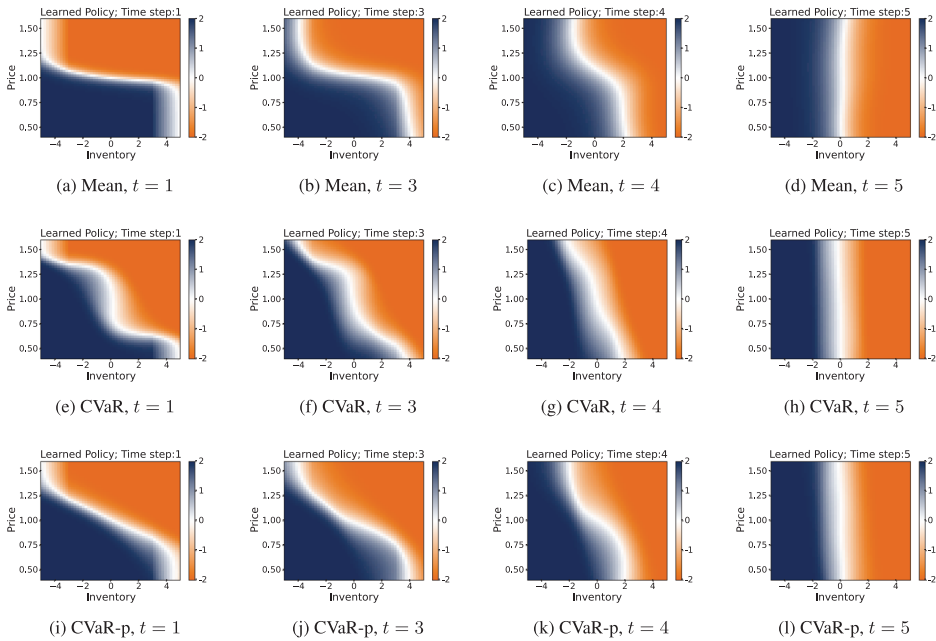


FIGURE 3 Learned policy by the actor-critic algorithm as a function of time (from left to right) when optimizing the dynamic expectation (top), the dynamic CVaR with $\alpha = 0.2$ (middle) and the dynamic penalized CVaR with $\alpha = 0.2$ and $\beta = 0.1$ (bottom) in the statistical arbitrage example. CVaR, conditional value-at-risk. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1111/mafi.12388)]

an Ornstein–Uhlenbeck process, and hence mean-reverts:

$$dS_t = \kappa(\mu - S_t)dt + \sigma dW_t, \quad (37)$$

where $\kappa = 2$, $\mu = 1$, $\sigma = 0.2$ and W_t is a standard $\mathbb{P}$ -Brownian motion.² The risk-aware agent tries to optimize the RL problem stated in Equation (P1), where for all periods $t \in \mathcal{T}$, the actions are determined by the trades a_t , the costs by the differences in wealth $c_t = y_t - y_{t+1}$, and the states by the tuples (t, S_t, q_t) .

Figure 3 shows a comparison of the learned policy between the dynamic expectation, the dynamic CVaR with $\alpha = 0.2$ and the dynamic penalized CVaR with $\alpha = 0.2$ and $\beta = 0.1$. When optimizing a risk-neutral objective function (see Figures 3a–3d), in the beginning of the episode, the agent aims to sell quantities of the asset when its price is higher than the mean-reversion level, and buy it when its price is lower. As periods evolve, the pattern shifts to ensure that the agent concludes the episode with zero inventory to avoid the terminal penalty. With other dynamic risk measures (see Figures 3e–3l), we observe that the agent is less aggressive, and instead waits until there are more significant price deviations from the mean-reversion level before taking actions that would benefit from the price reverting back to its mean. This reflects the *risk-sensitive behavior* of the agent.

² Our approach is model-free, which implies that we can easily replace the asset price dynamics with more complex models, for instance including a stochastic volatility.

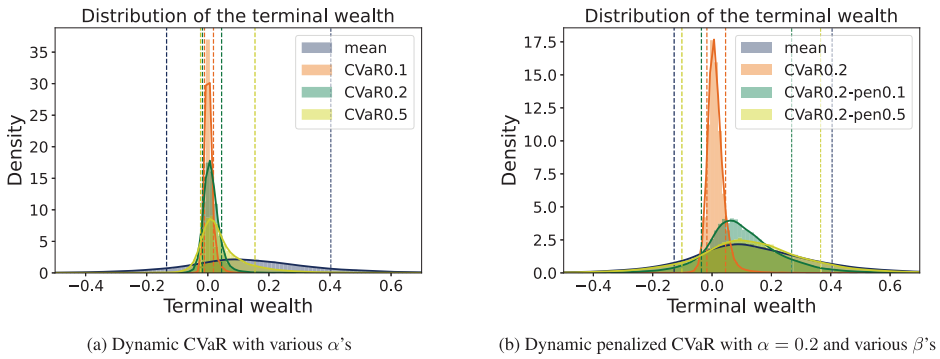


FIGURE 4 Estimated distribution of the terminal wealth when following the learned policy in the statistical arbitrage example for several risk measures over 30,000 episodes. Vertical dashed lines indicate the 10 and 90% quantiles. [Color figure can be viewed at wileyonlinelibrary.com]

The distribution of the terminal reward when the agent follows the learned policy for the dynamic mean, CVaR, and penalized CVaR is illustrated in Figure 4. In general, risk-sensitive approaches lead to a distribution with a *smaller variance* and *fewer large losses*. The agent's tolerance to risk can be adjusted by modifying the threshold α of the dynamic CVaR (see Figure 4a) or the relative entropy penalty constant β of the dynamic penalized CVaR (see Figure 4b). When increasing α or β , the distribution of the terminal wealth converges to the distribution for the risk-neutral objective function, as expected.

7.2 | Hedging with friction example

A corner stone problem of mathematical finance is the question of how to hedge the exposure to financial options. In this section, we illustrate how our approach may be applied to hedging a call option in an environment where the underlying asset dynamics follows the Heston (1993) model in a market with trading frictions. In principle, one can swap out the specific stochastic volatility model for other models and/or use historical sample paths.

We denote the price of an underlying asset by $(S_t)_{t \in \mathcal{T}}$ and the call option's strike price by $K = 10$. We consider the case where an agent sells the call option and aims to dynamically hedge it trading solely in the underlying asset and the bank account. We assume that there are $T = 10$ periods (corresponding to one month). Hence, the agent must pay $(S_T - K)_+$ at the terminal time.

We denote the stochastic variance process by $(\nu_t)_{t \in \mathcal{T}}$, and, as a reminder, in the Heston model, the price evolution of the underlying asset is given by

$$dS_t = \mu S_t dt + \sqrt{\nu_t} S_t dW_t^S, \quad (38a)$$

$$d\nu_t = \kappa(\vartheta - \nu_t) dt + \varsigma \sqrt{\nu_t} dW_t^\nu, \quad (38b)$$

where $\mu = 0.1$ is the drift, $\kappa = 9$ the mean-reversion rate, $\vartheta = (0.25)^2$ the mean-reversion level, $\varsigma = 1$ the volatility of the volatility (often referred to as vol-vol), and $(W_t^S)_{t \in \mathcal{T}}, (W_t^\nu)_{t \in \mathcal{T}}$ are two $\mathbb{P}$ -Brownian motions with correlation $\rho = -0.5$ (i.e., $d[W^S, W^\nu]_t = \rho dt$).

For generating sample paths, we use the Milstein discretization scheme (Mil'shtejn, 1975) to simulate the dynamics of the stock price and volatility, where for each $t \in \mathcal{T}$,

$$S_{t+1} = S_t \exp \left\{ \left(\mu - \frac{1}{2} \nu_t^+ \right) \Delta t + \sqrt{\nu_t^+ \Delta t} W_t^S \right\}, \quad (39a)$$

$$\nu_{t+1} = \nu_t + \kappa(\vartheta - \nu_t^+) \Delta t + \varsigma \sqrt{\nu_t^+ \Delta t} W_t^\nu + \mathbb{1}(\nu_t \geq 0) \left(\frac{1}{4} \varsigma^2 \Delta t ((W_t^\nu)^2 - 1) \right), \quad (39b)$$

with $\nu_t^+ = \max(\nu_t, 0)$, $\Delta t = 1/(12T)$, and the initial price and volatility, respectively, of $S_0 = 10$ and $\nu_0 = (0.2)^2$.

At each period $t \in \mathcal{T}$, the agent's wealth y_t is determined by its hedges a_t and bank account B_t . In what follows, we assume: (i) there are market frictions in the form of transaction costs of $\varepsilon = 0.005$ (per share); (ii) the interest rate of the bank account, denoted r , is zero; and (iii) the agent starts with an initial wealth of $y_0 = B_0$.

We next describe the dynamic hedging procedure we employ and the cash-flow it induces. For each $t \in \mathcal{T}$, the agent takes an action a_t^ϑ , which corresponds to the number of shares to hold over the next time interval, based on its policy, and this action influences its bank account and wealth in the following manner:

$$B_{t+} = B_t - (a_t^\vartheta - a_{t-1}^\vartheta) S_t - \left| a_t^\vartheta - a_{t-1}^\vartheta \right| \varepsilon, \quad (40a)$$

$$y_{t+} = B_{t+} + a_t^\vartheta S_t. \quad (40b)$$

The second term in B_{t+} represents the rebalancing costs, while the third term represents a cost due to trading frictions. The asset price, and bank account, evolves over the next period and induces a change in the agent's wealth process. Thus, for each $t \in \mathcal{T} \setminus \{T-1\}$, we have the following relationships:

$$B_{t+1} = e^{r\Delta t} B_{t+}, \quad (41a)$$

$$y_{t+1} = B_{t+1} + a_t^\vartheta S_{t+1}. \quad (41b)$$

At the end of the investment horizon, the agent must pay the call option and liquidate its inventory. Therefore, we have

$$B_T = e^{r\Delta t} B_{(T-1)+} + a_{T-1}^\vartheta S_T - \left| a_{T-1}^\vartheta \right| \varepsilon - (S_T - K)_+, \quad (42a)$$

$$y_T = B_T. \quad (42b)$$

In our RL notation, the agent minimizes a dynamic convex risk measure of the costs induced by its policy, where for all periods $t \in \mathcal{T}$, the costs $c \in C$ are given by the change in the agent's wealth process $c_t = y_t - y_{t+1}$, and the states represent all the information the agent possesses

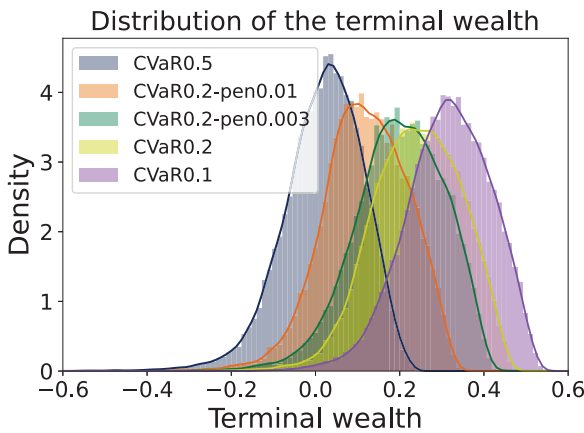


FIGURE 5 Estimated distribution of the terminal wealth when following the learned policy in the hedging example over 30,000 episodes. [Color figure can be viewed at wileyonlinelibrary.com]

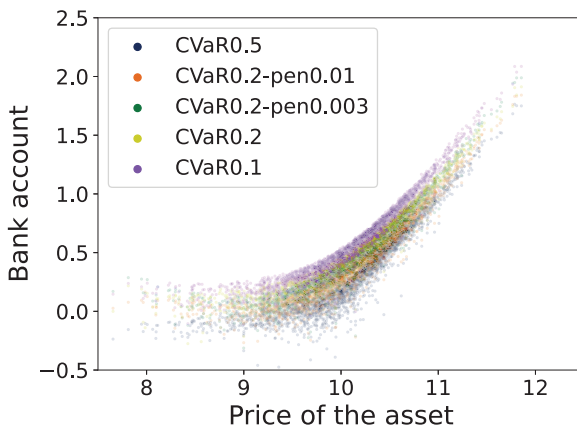


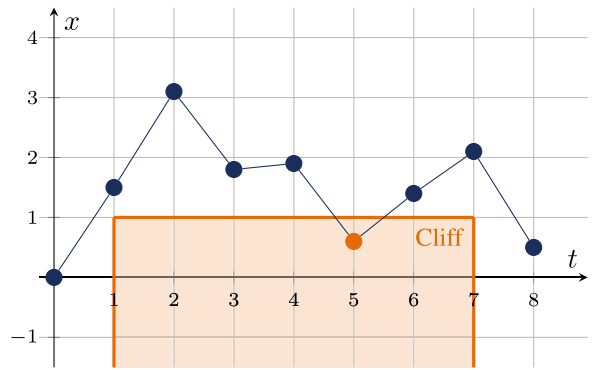
FIGURE 6 Scatter plot of the bank account before paying the call option against the terminal price of the asset when following the learned policy in the hedging example over 3000 episodes. [Color figure can be viewed at wileyonlinelibrary.com]

before making its hedging decision. Depending on the option and the market assumptions, one may easily include several features into the state space, such as more asset price history, market volatility, inventories of other assets the agent holds, and so on. To keep the experiments brief, here, we focus on just a few features.

We consider the case where the states that determine the agent's policy are tuples of the form (t, S_t, a_{t-1}) . Typically, in continuous time financial modeling for the Heston model, one assumes that the volatility process is observed and used as a feature in obtaining the optimal hedge. Here, however, we exclude it from the policy as in real-world trading, volatility itself is not observable. The initial wealth B_0 is obtained by forcing the dynamic risk (whether the dynamic CVaR at level α or dynamic penalized CVaR at level α with relative entropy penalty constant β) to be 0.2. This specific choice is up to the agent to choose and represents the minimal reservation value the agent is willing to take. In implementation, we achieve this by initially setting the price to the Black-Scholes price using the long-run volatility $\sqrt{\vartheta}$, and then adjusting the price until the dynamic risk is 0.2. As coherent and convex dynamic risk measures are translation invariant (see Definition 4.3 and 4.4), the learned policy is not affected by this price adjustment.

In Figure 5, we illustrate the P&L distribution of the optimal strategies for three different confidence levels of dynamic CVaR. Analogous to the other experiments, the figure suggests that risk-sensitive policies lead to reward distributions that become less variable as the confidence level of CVaR increases. In Figure 6, we generate a scatter plot of the underlying asset's price

FIGURE 7 Illustration of the modified cliff walking problem. One sample path (in blue) of the rover is drawn where it falls into the cliff (in orange) at $t = 5$. [Color figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com/doi/10.1111/mafi.12388)]



versus the agent's terminal bank account (corresponding to their terminal wealth). In a situation where the agent can perfectly hedge, we expect to see the “hockey stick” payoff. In the current context, however, while we do see the general “hockey stick” shape, there is additional convexity induced by the agent aiming to hedge in a discrete trading environment with trading frictions and (unobserved) stochastic volatility—which is far from being a complete market.

7.3 | Cliff walking example

This set of experiments is performed on a modified version of the cliff walking problem (Sutton & Barto, 2018) with a continuous action space, illustrated in Figure 7. Consider an autonomous rover exploring the land of a new planet, represented as a Cartesian coordinate system. The rover starts at $(0,0)$ and aims to get to $(T, 0)$, with $T = 9$, while avoiding all coordinates where $x \leq C = 1.0$, illustrated as the cliff. All allowed movements at any period $t \in \mathcal{T}$, that is, moving from (t, x_1) to $(t + 1, x_2)$, incur a cost of $1 + (x_2 - x_1)^2$. Stepping into the cliff region induces an additional cost of 100, while landing further from the goal at (T, x) induces a penalty of size x^2 . Actions taken by the rover are drawn from a Gaussian distribution $a_t^\theta \sim \pi^\theta = \mathcal{N}(\mu^\theta, \sigma)$, with $\mu^\theta \in (-a_{\max}, a_{\max})$, $a_{\max} = 4.0$ and $\sigma = 1.5$. This represents the rover's desire to move in a certain direction μ^θ , but its movements are altered by the unknown terrain (e.g., slipping on sand, crossing shallow water, etc.).³ This introduces randomness in the RL problem that the autonomous rover must account for while making decisions. In our RL notation, for all periods $t \in \mathcal{T}$, the costs $c \in C$ are determined by the rover's movements, and the states by the tuples (t, x_t) .

We next explore the *agent's sensitivity to risk*—whether the rover should reach the goal as quickly as possible by staying close to the cliff, or take a more circuitous route to avoid inadvertently falling into the cliff. Figure 8 shows the 10, 50, and 90% quantiles of the region visited by the rover when following the optimal strategy induced by the four dynamic risk measures. The figure illustrates that, indeed, the agent takes into account the uncertainty of the unknown terrain by staying further and further away from the cliff as they become more risk-averse. Notice, the optimal policy when optimizing the risk-neutral expectation induces the agent to stay close to the cliff to avoid the costs of vertical movements. Ultimately, using a risk-sensitive approach gives a reward distribution with mitigated tail risk for the autonomous rover, as shown in Figure 9. We

³The environment can be modified in order to place different obstacles on the coordinate system, and the policy parametrized with atypical distributions (e.g., one-sided or skewed) to illustrate different terrains.

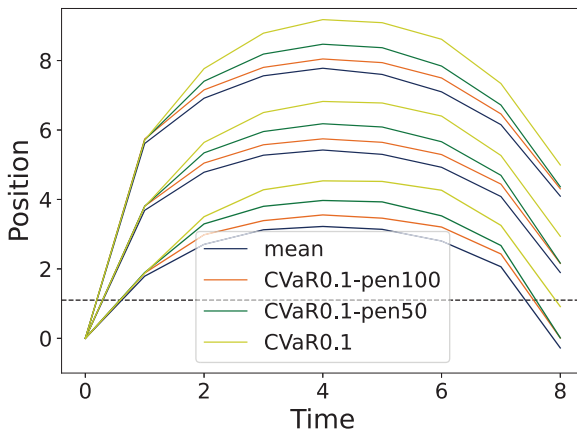


FIGURE 8 Estimated 10, 50, and 90% quantiles of the region visited by the rover when following the learned policy over 30,000 episodes. [Color figure can be viewed at wileyonlinelibrary.com]

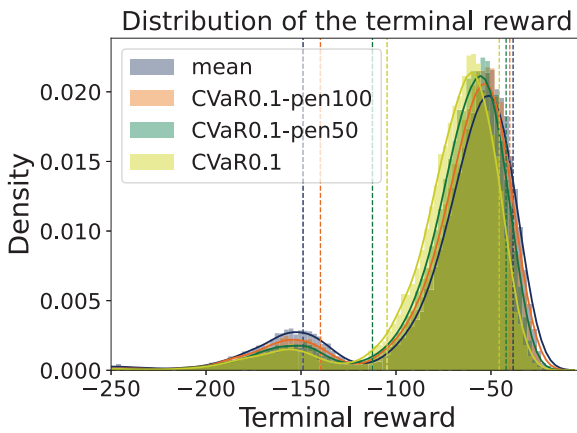


FIGURE 9 Estimated distribution of the terminal reward when following the learned policy in the cliff walking example for several risk measures over 30,000 episodes. [Color figure can be viewed at wileyonlinelibrary.com]

remark that we must consider larger relative entropy penalty constants for the dynamic penalized CVaR than in Sections 7.1 and 7.2 due to the large magnitude of the costs in this example.

Risk-sensitivity in mathematical finance applications is of fundamental importance and requires special attention, which motivates the work provided in this paper. However, the proposed methodology can be applied more generally to other fields of study, as shown in this robot control example. One may see connections between the cliff walking problem and some financial applications: for instance, suppose that an agent aims to hedge a barrier option, which is knocked-out (e.g., falling into the cliff) if the price of the underlying asset (e.g., position of the rover) goes beyond a predetermined price (e.g., position of the cliff). That is another related, yet distinct financial problem our algorithm could be used on.

8 | DISCUSSION

In this work, we extend risk-aware RL procedures found in the literature by developing a methodology for a wide class of sequential decision making problems. Our approach broadens to the whole class of *dynamic convex* risk measures and allows *nonstationary behaviors* from the agent. Our work performs well on three benchmark RL and financial mathematical problems, which opens doors for several applications in credit risk, portfolio optimization, and optimal control among others.

There are still some future directions that we could explore to provide even more flexibility to our proposed approach. For instance, we could devise a *computationally efficient* methodology for large-scale problems when simulations can be costly, propose a *robust framework* for time-consistent dynamic risk measures, and develop a generalization for deterministic policies with dynamic risk measures in a similar manner to *deep deterministic policy gradient*. More recently, efforts were put to verify the theoretical *convergence of policy gradient* methods. Indeed, Agarwal et al. (2021) show that the policy gradient approach with a risk-neutral expectation has global convergence guarantees. On the other hand, it may not converge to a global optimum when the objective function is a dynamic risk measure (Huang et al., 2021). It thus remains an open challenge to prove that actor–critic algorithms with dynamic convex risk measures converge to an optimal policy when both the value function and policy are characterized by ANNs.

ACKNOWLEDGMENTS

The authors acknowledge support from the Natural Sciences and Engineering Research Council of Canada (Grants RGPIN-2018-05705, RGPAS-2018-522715, and CGSD3-2019-534435). The authors also thank the anonymous referees for their thoughtful comments during the review process.

DATA AVAILABILITY STATEMENT

The code, written in Python, that support the findings of this study are openly available in a GitHub repository at <https://github.com/acoache/RL-DynamicConvexRisk>. The data are from a model-based simulation.

ORCID

Sebastian Jaimungal  <https://orcid.org/0000-0002-0193-0993>

REFERENCES

- Acciaio, B., & Penner, I. (2011). Dynamic risk measures. In *Advanced mathematical methods for finance* (pp. 1–34). Springer.
- Agarwal, A., Kakade, S. M., Lee, J. D., & Mahajan, G. (2021). On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98), 1–76.
- Ahmadi, M., Rosolia, U., Ingham, M. D., Murray, R. M., & Ames, A. D. (2021). Constrained risk-averse Markov decision processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13), 11718–11725.
- Al-Arabi, A., Correia, A., Naiff, D., Jardim, G., & Saporito, Y. (2018). Solving nonlinear and high-dimensional partial differential equations via deep learning. *arXiv preprint arXiv:1811.08782*.
- Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent measures of risk. *Mathematical Finance*, 9(3), 203–228.
- Bäuerle, N., & Glauner, A. (2022). Distributionally robust Markov decision processes and their connection to risk measures. *Mathematics of Operations Research*, 47(3), 1757–1780.
- Bäuerle, N., & Glauner, A. (2021b). Minimizing spectral risk measures applied to Markov decision processes. *Mathematical Methods of Operations Research*, 94(1), 35–69.
- Bäuerle, N., & Glauner, A. (2021a). Markov decision processes with recursive risk measures. *European Journal of Operational Research*, 296, 953–966.
- Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In *International conference on machine learning* (pp. 449–458). PMLR.
- Bielecki, T. R., Cialenco, I., Drapeau, S., & Karliczek, M. (2016). Dynamic assessment indices. *Stochastics*, 88(1), 1–44.
- Campbell, S., Chen, Y., Shrivats, A., & Jaimungal, S. (2021). Deep learning for principal-agent mean field games. *arXiv preprint arXiv:2110.01127*.
- Carmona, R., & Laurière, M. (2021). Deep learning for mean field games and mean field control with applications to finance. *arXiv preprint arXiv:2107.04568*.

- Casgrain, P., Ning, B., & Jaimungal, S. (2022). Deep Q-learning for Nash equilibria: Nash-DQN. *Applied Mathematical Finance*, 29(1), 62–78.
- Cheridito, P., & Stadje, M. (2009). Time-inconsistency of VaR and time-consistent alternatives. *Finance Research Letters*, 6(1), 40–46.
- Chow, Y., Ghavamzadeh, M., Janson, L., & Pavone, M. (2017). Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(1), 6070–6120.
- Chu, S., & Zhang, Y. (2014). Markov decision processes with iterated coherent risk measures. *International Journal of Control*, 87(11), 2286–2293.
- Cuchiero, C., Khosrawi, W., & Teichmann, J. (2020). A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4), 101.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.
- Delage, E., & Mannor, S. (2010). Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research*, 58(1), 203–213.
- Detlefsen, K., & Scandolo, G. (2005). Conditional and dynamic convex risk measures. *Finance and Stochastics*, 9(4), 539–561.
- Dhaene, J., Vanduffel, S., Goovaerts, M. J., Kaas, R., Tang, Q., & Vyncke, D. (2006). Risk measures and comonotonicity: A review. *Stochastic Models*, 22(4), 573–606.
- Di Castro, D., Oren, J., & Mannor, S. (2019). Practical risk measures in reinforcement learning. *arXiv preprint arXiv:1908.08379*.
- Drapeau, S., Kupper, M., Gianin, E. R., & Tangpi, L. (2016). Dual representation of minimal supersolutions of convex BSDEs. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 52(2), 868–887.
- Föllmer, H., & Schied, A. (2002). Convex measures of risk and trading constraints. *Finance and Stochastics*, 6(4), 429–447.
- Föllmer, H., Schied, A., & Lyons, T. J. (2004). Stochastic finance. An introduction in discrete time. *The Mathematical Intelligencer*, 26(4), 67–68.
- Frittelli, M., & Gianin, E. R. (2004). Dynamic convex risk measures. In *Risk measures for the 21st century* (pp. 227–248). John Wiley & Sons.
- Galichet, N., Sebag, M., & Teytaud, O. (2013). Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Asian conference on machine learning* (pp. 245–260). PMLR.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
- Grondman, I., Busoniu, L., Lopes, G. A., & Babuska, R. (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6), 1291–1307.
- Hamblly, B., Xu, R., & Yang, H. (2021). Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM Journal on Control and Optimization*, 59(5), 3359–3391.
- Han, J., Jentzen, A., & Weinan, E. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34), 8505–8510.
- Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6(2), 327–343.
- Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2), 251–257.
- Horvath, B., Muguruza, A., & Tomas, M. (2021). Deep learning volatility: A deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1), 11–27.
- Hu, R. (2019). Deep fictitious play for stochastic differential games. *arXiv preprint arXiv:1903.09376*.
- Huang, A., Leqi, L., Lipton, Z. C., & Azizzadenesheli, K. (2021). On the convergence and optimality of policy gradient for Markov coherent risk. *arXiv preprint arXiv:2103.02827*.
- Jaimungal, S. (2022). Reinforcement learning and stochastic optimisation. *Finance and Stochastics*, 26(1), 103–129.
- Kalogerias, D. S., Chamon, L. F., Pappas, G. J., & Ribeiro, A. (2020). Better safe than sorry: Risk-aware nonlinear Bayesian estimation. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5480–5484). IEEE.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Konda, V. R., & Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in Neural Information Processing Systems* (pp. 1008–1014). Citeseer.

- Kose, U., & Ruszczyński, A. (2021). Risk-averse learning by temporal difference methods with Markov risk measures. *Journal of Machine Learning Research*, 22(38), 1–34.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6), 861–867.
- Milgrom, P., & Segal, I. (2002). Envelope theorems for arbitrary choice sets. *Econometrica*, 70(2), 583–601.
- Mil'shtein, G. (1975). Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications*, 19(3), 557–562.
- Nass, D., Belousov, B., & Peters, J. (2019). Entropic risk measure in policy search. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Ning, B., Jaimungal, S., Zhang, X., & Bergeron, M. (2021). Arbitrage-free implied volatility surface generation with variational autoencoders. *arXiv preprint arXiv:2108.04941*.
- Osogami, T. (2012). Robustness and risk-sensitivity in Markov decision processes. *Advances in Neural Information Processing Systems*, 25, 233–241.
- Peng, S. (1997). Backward SDE and related g-expectation. In *Pitman research notes in mathematics series* (pp. 141–160). Longman.
- Petrik, M., & Subramanian, D. (2012). An approximate solution method for large risk-averse Markov decision processes. *arXiv preprint arXiv:1210.4901*.
- Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8, 143–195.
- Prashanth, L., & Ghavamzadeh, M. (2013). Actor-critic algorithms for risk-sensitive MDPs. In *Advances in neural information processing systems* (Vol. 26).
- Rahimian, H., & Mehrotra, S. (2019). Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*.
- Riedel, F. (2004). Dynamic coherent risk measures. *Stochastic Processes and their Applications*, 112(2), 185–200.
- Rockafellar, R. T., & Uryasev, S. (2013). The fundamental risk quadrangle in risk management, optimization and statistical estimation. *Surveys in Operations Research and Management Science*, 18(1-2), 33–53.
- Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2, 21–42.
- Ruszczyński, A. (2010). Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125(2), 235–261.
- Shapiro, A., Dentcheva, D., & Ruszczyński, A. (2014). *Lectures on stochastic programming: Modeling and theory*. SIAM.
- Shen, Y., Tobia, M. J., Sommer, T., & Obermayer, K. (2014). Risk-sensitive reinforcement learning. *Neural Computation*, 26(7), 1298–1328.
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.
- Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).
- Tamar, A., Chow, Y., Ghavamzadeh, M., & Mannor, S. (2015). Policy gradient for coherent risk measures. In *Advances in neural information processing systems* (Vol. 28, pp. 1468–1476).
- Tamar, A., Chow, Y., Ghavamzadeh, M., & Mannor, S. (2016). Sequential decision making with coherent risk. *IEEE Transactions on Automatic Control*, 62(7), 3323–3338.
- Weber, S. (2006). Distribution-invariant risk measures, information, and dynamic consistency. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, 16(2), 419–441.
- Weinan, E., Han, J., & Jentzen, A. (2017). Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4), 349–380.
- Yu, P., Haskell, W. B., & Xu, H. (2018). Approximate value iteration for risk-aware Markov decision processes. *IEEE Transactions on Automatic Control*, 63(9), 3135–3142.

How to cite this article: Coache, A., & Jaimungal, S. (2024). Reinforcement learning with dynamic convex risk measures. *Mathematical Finance*, 34, 557–587.
<https://doi.org/10.1111/mafi.12388>

APPENDIX A: IMPLEMENTATION

In this section, we expand on the experimental setup by giving additional details on the implementation of algorithms given in Section 6.

The structure of the Python files is similar between all sets of experiments. The `envs.py` file contains the environment class for RL problem, as well as functions to interact with it. It has both the PyTorch and NumPy versions of the simulation engine. The `risk_measure.py` file has the class that creates an instance of a risk measure, with functions to compute the risk and calculate the gradient. Risk measures currently implemented are the expectation, the CVaR, the penalized CVaR, the mean-semideviation and a linear combination between the mean and CVaR. There is also a `utils.py` file, which contains some useful functions and variables, such as a function to create new directories and colors for the visualizations.

Models are regrouped under the `models.py` file with classes to build ANN structures using the PyTorch library. In our experiments presented in Section 7, the value function V^ϕ has four layers of 16 hidden nodes each with SiLU activation functions, and no activation function for its output layer. The ANN for the policy π^θ is composed of five layers with 16 hidden nodes each with SiLU activation function, but with an output layer specific to the application—for example, linear transformation of a sigmoid activation function that maps to $(-a_{\max}, a_{\max})$ for the set of experiments in Section 7.1. The learning rates for V^ϕ and π^θ are of the order of, respectively, 1×10^{-3} and 5×10^{-4} . Both V^ϕ and π^θ are updated with mini batches of 300 (outer) episodes and 1000 (inner) transitions during the training phase.⁴

The whole algorithm is wrapped into a single class named `ActorCriticPG`, where input arguments specify which problem the agent faces. The user needs to give an environment, a (convex) risk measure, as well as two neural network structures that play the role of the value function and agent's policy. Each instance of that class has functions to select actions from the policy, whether at random or using the best behavior found thus far, and give the set of invalid actions. There is also a function to simulate (outer) episodes and (inner) transitions using the simulation upon simulation approach discussed in Section 6. Algorithm 2 is wrapped in a function which takes as inputs the mini-batch size B , number of epochs K , and characteristics of the value function neural network structure, such as the learning rate and the number of hidden nodes. Similarly, another function implements Algorithm 3 and takes as inputs the mini-batch size B and number of epochs K .

The `main.py` file contains the program to run the training phase. The first part concerns the importation of libraries and initialization of all parameters, either for the environment, neural networks, or risk measure. Some notable parameters that need to be specified by the user in the `hyperparams.py` file are the numbers of epochs, learning rates, size of the neural networks, and number of episodes/transitions among others. The next section is the training phase and its skeleton is given in Algorithm 1. It uses mostly functions from the `actor_critic.py` file. Finally, the models for the policy and value function are saved in a folder, along with diagnostic plots. Since

⁴ Hyperparameters depend on the specific experiment and are chosen to accelerate the learning procedure.

the nested simulation approach is computationally expensive, running this Python program can take up to 24 h depending on the application and the hyperparameters, especially the number of periods, the complexity of the convex risk measures, and the number of (inner) transitions.

This `main_plot.py` file contains the program to run the testing phase. The first part concerns the importation of libraries and initialization of all parameters. Note that parameters must be identical to the ones used in `main.py`. The next section evaluates the policy found by the algorithm. It runs several simulations using the best behavior found by the actor-critic algorithm. Finally, it outputs graphics to assess the performance of the procedure, such as the preferred action in any possible state and the estimated distribution of the total cost when following the best policy.