

Robust Policy Optimization and Verification with Moment-Based Distributional Constraints

XIAOTONG JI*, Department of Computing, Imperial College London, UK

HANCHUN WANG*, Department of Mathematics, Imperial College London, UK

ANTONIO FILIERI, Department of Computing, Imperial College London, UK

Verification and adaptation methods based on probabilistic model checking for self-adaptive systems traditionally focus on expected values of system properties, which can mask rare but critical outcomes and lead to control policies that optimize average performance but behave poorly in the tail. Existing distributional probabilistic model checking methods typically rely on arbitrary discretization of reward distributions, with limited guarantees for continuous-valued rewards and poor scalability when propagating full distributions for policy optimization in MDPs. Building upon our prior moment-based verification framework for DTMCs [26], we introduce a robust policy optimization method for MDPs that uses moment-based bounds on distributional tail probabilities to enforce chance constraints while optimizing expected cumulative reward. Our approach computes analytical moments of cumulative rewards via first-step analysis and MGF-based recurrences. We then use the moment-based Cantelli bound to obtain a conservative and differentiable surrogate for the non-smooth chance constraint using only a finite number of moments, and use this surrogate to guide policy optimization with conservative probabilistic guarantees. Experimental evaluation on PRISM MDP benchmarks demonstrates that our Cantelli-based value iteration achieves risk-sensitive performance comparable to distributional baselines while reducing runtime by up to $100\times$ and supporting continuous-valued rewards without binning.

ACM Reference Format:

Xiaotong Ji, Hanchun Wang, and Antonio Filieri. 2018. Robust Policy Optimization and Verification with Moment-Based Distributional Constraints. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 24 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

Probabilistic model checking (PMC) [29] is widely used in self-adaptive systems to detect quantitative requirement violations and to support planning of adaptation actions [6, 16, 32, 42]. However, many PMC methods focus only on expected values of system properties, thereby ignoring rare but important outcomes that cause requirement violations. For adaptation and control, expected-value objectives (as in PRISM reward queries such as $R_{\min}=?$ or $R_{\max}=?$) can lead to policies that optimize the average but behave poorly in the tail, which limits robustness in verification and decision-making.

Recent work in PMC [2, 13], inspired by distributional reinforcement learning [5, 11], supports distribution-level reasoning for DTMCs and MDPs by representing probability or reward distributions with histogram approximations. In

*Both authors contributed equally to this work.

Authors' Contact Information: Xiaotong Ji, Department of Computing, Imperial College London, UK, xiaotong.ji16@imperial.ac.uk; Hanchun Wang, Department of Mathematics, Imperial College London, UK, hanchun.wang21@imperial.ac.uk; Antonio Filieri, Department of Computing, Imperial College London, UK, a.filieri@imperial.ac.uk.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

practice, histogram accuracy depends heavily on the binning choice: fewer bins give coarse approximations, while many bins increase cost while still introducing discretization bias. The piecewise-constant representation is non-differentiable, making distribution propagation and transformations harder and less stable. Histogram approximations also struggle to represent multimodality and heavy tails accurately, and convergence guarantees often require integer-valued rewards. These issues are exacerbated in large or continuous-valued reward spaces and in multi-objective settings, where the number of bins grows combinatorially.

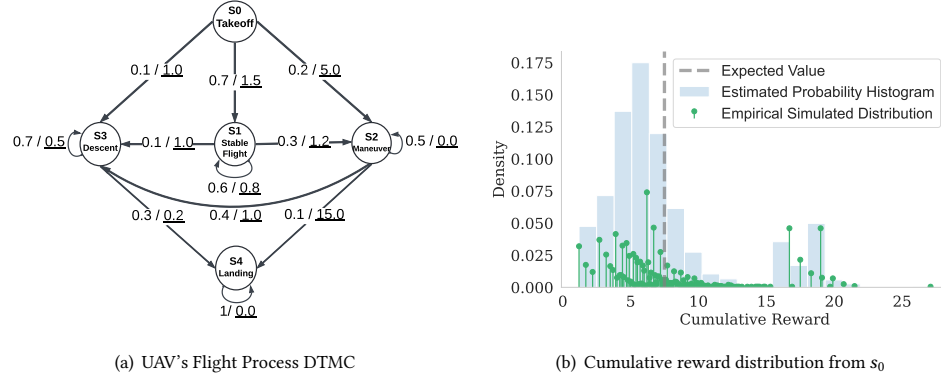


Fig. 1. UAV flight process with states $\{s_0, s_1, s_2, s_3, s_4\}$ representing Takeoff, Stable Flight, Maneuvering, Descent, and Landing, respectively. Transition probabilities and rewards are shown on the edges in the form of p/r . The cumulative reward distribution at s_0 is illustrated using the expected value (grey), histogram approximation (light blue), and empirical samples (green), highlighting the different levels of information provided by various methods on the same distribution.

Example. Consider the UAV flight process in Fig. 1(a) modeled as a DTMC whose states correspond to the specific flight phases Takeoff, Stable Flight, Maneuvering, Descent, and Landing. Transitions are annotated with probabilities and rewards, where the reward encodes the energy consumed by the corresponding maneuver. The cumulative reward at s_0 represents the total energy usage over a flight. Standard PMC focuses on the expected cumulative energy (grey line in Fig. 1(b)), which can be misleading for decision-making because it does not show how probability mass is split across typical and rare outcomes. Histogram-based estimates (light blue bars) provide more detail but may smooth closely spaced peaks and miss isolated peaks and tail behavior, as shown by Monte Carlo samples (green points). This example demonstrates why expectation-based analysis and coarse histogram approximations both miss significant information for model verification and robust action selection.

We previously introduced a moment-based distributional PMC framework for DTMCs in [26], which approximates the cumulative reward distribution up to absorption using analytical moment matching. This method provides efficient, conservative verification via Cantelli inequalities and precise verification via Erlang mixture approximation, enabling distributional analysis beyond expected values. Erlang mixtures can approximate any positive random variable to arbitrary accuracy [40] and align well with Markov reward models. This representation can also support continuous-valued rewards without discretization, captures multimodality and tail behavior more precisely, and preserves higher-order statistics such as skewness and kurtosis through moment matching, therefore addressing potential problems of histogram-based discretized methods. However, the framework in [26] addresses only verification (checking properties of a fixed DTMC), not control (synthesizing policies for MDPs that satisfy distributional requirements).

In this work, we build on the above robust PMC framework and extend it from verification on DTMCs to risk-aware policy synthesis on MDPs. We formulate a chance-constrained control problem that optimizes the expected cumulative reward¹ while enforcing a chance constraint of the form $P(R^\pi \leq \tau) \geq \alpha$, where R^π is the cumulative reward in the induced DTMC under the synthesized policy π . This chance constraint is global and difficult to optimize directly: it is non-differentiable, and the full cumulative reward distribution is not available in closed form. We address this by using analytical moments to derive Cantelli bounds, giving a conservative and differentiable surrogate for the chance constraint. We integrate this surrogate into a value-iteration update, leading to a robust risk-sensitive VI method that synthesizes policies with guaranteed upper/lower bounds on violation probabilities, without Monte Carlo sampling, histogram discretization, or explicit distributional state augmentation, while remaining applicable and efficient for discrete-valued and continuous-valued reward verification.

Our previous work [26] introduced a robust PMC verification framework based on analytical moments, along with error bounds and empirical evaluations. In this work, we make the following additional contributions:

- (1) **Robust risk-sensitive policy optimization.** We extend moment-based robust PMC from DTMC verification to chance-constrained policy synthesis for MDPs. Our method integrates Cantelli bounds into the synthesis loop using only analytical moments of the policy-induced DTMC, resulting in a conservative and differentiable surrogate for the chance constraint.
- (2) **Soundness considerations.** We establish soundness properties induced by the surrogate objective function, including its ability to separate Cantelli-certified policies from uncertified ones, and to prevent the policy search process from escaping regions that satisfy the chance constraint.
- (3) **Empirical evaluation on PRISM case studies.** We evaluate our approach on PRISM MDP benchmarks with both discrete and continuous rewards, and compare against risk-neutral and distributional baselines [13] in terms of expected cumulative reward, tail risk, and computational efficiency.

The remainder of the paper is organized as follows. Section 2 provides background on PMC and moment-based analysis. Section 3 summarizes our prior verification framework, Section 4 presents the policy optimization method. Section 5 evaluates the approach empirically, Section 6 discusses related work, and Section 7 concludes.

2 Background

2.1 Markov Models and Markov Reward Process

Discrete-Time Markov Chain (DTMC). A discrete-time Markov chain (DTMC) is a tuple $D = (S, s_0, P, R, AP, L)$, where S is a finite set of states, $s_0 \in S$ is the initial state, $P : S \times S \rightarrow [0, 1]$ is a stochastic transition matrix with $\sum_{s' \in S} P(s, s') = 1$ for all $s \in S$, and $R : S \rightarrow \mathbb{R}$ is a state reward function,² AP is a set of atomic propositions, $L : S \rightarrow 2^{AP}$ is a labeling function mapping each state to a set of propositions that hold true in that state. The system evolves as a stochastic process $\{X_t\}_{t \geq 0}$ with $X_0 = s_0$ and $\Pr(X_{t+1} = s' \mid X_t = s) = P(s, s')$.

Markov Decision Process (MDP). A finite MDP is a tuple $\mathcal{M} = (S, s_0, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where S is a finite state space, $s_0 \in S$ is the initial state, $\mathcal{A}$ is a finite action set, $\mathcal{A}(s) \subseteq \mathcal{A}$ is the set of actions at state s , $\mathcal{P}(\cdot \mid s, a)$ is the transition kernel on S , $\mathcal{R} : S \times \mathcal{A} \rightarrow \mathbb{R}$ is a one-step reward, and $S_T \subseteq S$ is a set of absorbing states such that for any $s \in S_T$ and $a \in \mathcal{A}(s)$, $\mathcal{P}(s \mid s, a) = 1$. A stationary deterministic policy is a function that maps $\pi : S \rightarrow \mathcal{A}$ with $\pi(s) \in \mathcal{A}(s)$.

¹In this paper we use the term *reward* for PRISM reward structures, and we say *optimize reward* to cover both $R_{\min}$ and $R_{\max}$ objectives.

²Transition rewards are sometimes used for compactness (e.g., Fig. 1). They can be reduced to state rewards without loss of generality by systematically introducing intermediate states that carry the transition reward.

Notably, a specific *induced DTMC* $\mathcal{D}^\pi = (\mathcal{S}^\pi, s_0, \mathcal{P}^\pi, \mathcal{R}^\pi, AP, L)$ can be constructed by a finite MDP $\mathcal{M}$ and a given policy π , where $\mathcal{S}^\pi$ is a sub-space of the finite state space $\mathcal{S}$ of the original MDP, $s_0 \in \mathcal{S}^\pi$ is the initial state, $\mathcal{P}^\pi: \mathcal{S}^\pi \times \mathcal{S}^\pi \rightarrow [0, 1]$ is the stochastic transition matrix.

Reward Process. We consider a Markov model with non-negative reward function $r: \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$, representing the instantaneous reward received in each state, together with a sequence of random variables $\{X_n\}_{n \geq 0}$. The total cumulative reward up to time N is $R(N) = \sum_{n=0}^N r(X_n)$. As the expectation of probabilities and rewards in Markov Models can be computed as the solutions of linear equations [9], the expected total cumulative reward $\mu = \mathbb{E}[\sum_{n=0}^{T-1} r(X_n)]$ can be easily computed by solving the linear system $\mu = r + P\mu$ if the system is transient.

Value Iteration. To analyze quantitative properties in Markov models, standard probabilistic model checkers, such as PRISM, compute optimal expected reward queries ($R_{\min} = ?$ or $R_{\max} = ?$). Value iteration applies the Bellman optimality update over states until convergence (with $V(s) = 0$ for $s \in \mathcal{S}_T$):

$$V(s) = \min / \max_{a \in \mathcal{A}(s)} \left(r(s, a) + \sum_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) V(s') \right)$$

In this paper, we use this standard update as the starting point and modify it to obtain a robust risk-sensitive value-iteration scheme.

2.2 Probabilistic Model Checking

Having established the foundational Markov models, we now describe how probabilistic model checking uses these models to verify quantitative properties.

Probabilistic model checking (PMC) is a formal method for reasoning about quantitative properties of systems modeled as stochastic processes, typically DTMCs and MDPs [29]. In this work, we focus on the verification and policy synthesis in discrete Markov models of regular properties [3], which can be reduced to reachability and reward-to-target objectives: either (i) the probability of reaching a target set of states, or (ii) the cumulative reward accumulated until reaching that target. These properties are commonly specified using formalisms like co-safe LTL [30] or PCTL fragments without nested modalities [3]. Without loss of generality, we consider properties of the form $P_{\bowtie p}[\Diamond \phi]$ and $R_{\bowtie b}[\Diamond \phi]$, where $\bowtie \in \{\leq, \geq\}$, $p \in [0, 1]$ is a probability bound, $b \in [0, \infty)$ is a reward bound, and ϕ is a Boolean state predicate.

The formula $P_{\bowtie p}[\Diamond \phi]$ holds in a state s if the *expected probability* of eventually reaching a ϕ -state from s is $\bowtie p$. Similarly, $R_{\bowtie b}[\Diamond \phi]$ is evaluated by comparing the *expected cumulative reward* until a ϕ -state is reached with the bound b . For DTMCs, both reachability probabilities and expected cumulative rewards reduce to solving linear systems over transient states. For MDPs, the same quantities depend on the chosen policy; PMC tools typically compute extremal values (e.g., minimal or maximal probability/reward over policies) using dynamic programming with the Bellman operator and value iteration [3, 38]. These quantities are expectations, which do not fully characterize the probability distributions of reachability or accumulated rewards that could be relevant when specifying probabilistic requirements.

Example. For the UAV system in Fig. 1, the expected cumulative reward is 7.51, which is rarely achieved across runs and does not capture the bi-modal behavior of the system.

Chance constraints. A common way to formulate distributional requirements for reachability and reward properties is through chance constraints [8, 15]. These constraints support requirements of the form “the cumulative reward stays below (or exceeds) a threshold with at least a given probability.” Reasoning about reachability can be reduced to reasoning about cumulative rewards via standard model transformations [17]; likewise, reaching a target set can be

reduced to absorption into designated absorbing states via standard graph-based reductions [3]. Therefore, without loss of generality, we focus on the distribution of the cumulative reward to absorption.

Formally, let X denote the random cumulative reward to absorption over runs of the Markov process. A chance-constrained requirement can be defined as $\Pr(X \leq r^*) \geq \alpha$, where X represents the cumulative reward to absorption, r^* is a threshold value, and α is the required probability level.

This requirement is equivalent to a Value-at-Risk (VaR) bound, where the α -quantile is $\text{VaR}_\alpha(X) \triangleq \inf\{x \in \mathbb{R} : \Pr(X \leq x) \geq \alpha\}$. Another commonly used tail-risk measure is *Conditional Value-at-Risk (CVaR)*, $\text{CVaR}_\alpha(X) \triangleq \mathbb{E}[X \mid X \geq \text{VaR}_\alpha(X)]$, which summarizes the expected magnitude of the upper tail beyond VaR and often requires additional tail estimation. While VaR, CVaR, and chance constraints are all used in distributional learning and risk-aware verification, we focus on chance constraints in this work to obtain a direct probabilistic guarantee that fits our moment-based analysis.

2.3 Moment Generating Functions and Markov Reward Distributions

To evaluate chance constraints and approximate cumulative reward distributions, we require analytical techniques for computing distributional properties beyond expected values. This subsection introduces the moment generating function (MGF) approach, Cantelli bounds for tail probabilities, and Erlang mixture approximations.

Moment-Generating Function (MGF). We use first-step analysis to compute *analytical* moments of the cumulative reward to absorption for enabling distributional analysis using moment-based information. We consider a set of transient states $C \subseteq S$ and absorbing states $C^c \subseteq S$ (where $C \cap C^c = \emptyset$, $C \cup C^c = S$), with $T = \inf\{n \geq 0 : X_n \in C^c\}$ denoting the first hitting time of a state in C^c . The total cumulative reward up to absorption is represented by $R_T = \sum_{n=0}^{T-1} r(X_n)$. The MGF of R_T starting from $X_0 = x$, denoted by $M_{R_T}(\theta, x)$, is given by:

$$M_{R_T}(\theta, x) = \mathbb{E}_x[e^{\theta R_T}] = \mathbb{E}_x \left[\exp \left(\theta \sum_{n=0}^{T-1} r(X_n) \right) \right] \quad (1)$$

For a Markov model, the MGF of R_T starting from $X_0 = x$ satisfies the linear system:

$$M_{R_T}(\theta) = f(\theta) + G(\theta)M(\theta),$$

where $f(\theta, x) = \sum_{y \in C^c} e^{\theta r(x)} P(x, y)$ for $x \in C$, and $G(\theta, x, y) = e^{\theta r(x)} P(x, y)$ for $x, y \in C$.

Analytical moments from the MGF. The k -th moment $u_k(x) = \mathbb{E}_x[R_T^k]$ starting at $X_0 = x$ can be obtained by differentiating the MGF $u_k = \frac{d^k}{d\theta^k} M_{R_T}(\theta) \Big|_{\theta=0}$ with the following recurrence:

$$u_k(x) = \sum_{y \in S} P(x, y) \sum_{i=0}^k \binom{k}{i} r(x)^{k-i} u_i(y), \quad k \geq 1 \quad (2)$$

Cantelli Inequality. Chance constraints on the finite-horizon cumulative reward R_T can be bounded via Cantelli inequality [20] (also known as one-sided Chebyshev's inequality [39]) using a finite number of analytical moments. We consider two directions that appear in PMC reward objectives: (i) an upper-threshold constraint (typical for $R_{\min}$) $\Pr(R_T \leq \tau \mid X_0 = s_0) \geq \alpha$, and (ii) a lower-threshold constraint (typical for $R_{\max}$) $\Pr(R_T \geq \tau \mid X_0 = s_0) \geq \alpha$.

For any integer $n \geq 2$, we define $u_k = \mathbb{E}[R_T^k \mid X_0 = s_0]$ as the k -th raw moment and $m_k = \mathbb{E}[(R_T - u_1)^k \mid X_0 = s_0]$ as the k -th central moment, with any threshold $\tau > u_1$, and any shift of the central moments $b > -(\tau - u_1)$, the higher-order

Cantelli bound gives

$$\begin{aligned}\Pr(R_T \geq \tau \mid X_0 = s_0) &\leq \frac{\mathbb{E}[(R_T - u_1 + b)^n \mid X_0 = s_0]}{(\tau - u_1 + b)^n} = \frac{\sum_{k=0}^n \binom{n}{k} b^{n-k} m_k}{(\tau - u_1 + b)^n}, \\ \Pr(R_T \leq \tau \mid X_0 = s_0) &\geq 1 - \frac{\sum_{k=0}^n \binom{n}{k} b^{n-k} m_k}{(\tau - u_1 + b)^n}.\end{aligned}\tag{3}$$

For $n = 2$, choosing $b = \frac{u_2 - u_1^2}{\tau - u_1}$ minimizes the bound and yields the classical closed forms of the Cantelli inequality:

$$\Pr(R_T \leq \tau \mid X_0 = s_0) \geq \frac{(\tau - u_1)^2}{(\tau - u_1)^2 + (u_2 - u_1^2)}, \quad \Pr(R_T \geq \tau \mid X_0 = s_0) \leq \frac{u_2 - u_1^2}{(\tau - u_1)^2 + (u_2 - u_1^2)}.\tag{4}$$

These bounds depend only on a finite number of moments of R_T , which we obtain analytically from the MGF equation (2), and hold for any cumulative reward distribution with those moments.

Example. The first moment for our UAV computed with Equation (2) corresponds to the expected reward computed, e.g., by the PRISM model checker [31]: $\mu = 7.51$. However, the standard deviation is $\sigma = 4.91$. Using, for example, Cantelli's inequality, this implies only 50% of the trajectories are guaranteed to fall within the interval $[0, \mu + \sigma]$, leaving a significant portion of the distribution not represented.

This result indicates that the expected value alone provides limited information, motivating the need for accurate distribution approximations and robust model checking in risk-sensitive systems. Next, we introduce the base distribution used for moment-matching approximations.

Erlang Distribution. The Erlang distribution is a special case of both phase-type and Gamma distributions with integer shape parameters [14]. An Erlang distribution with shape parameter $a \in \mathbb{Z}^+$ and rate parameter $\lambda \in \mathbb{R}^+$ is defined by the probability density function and the cumulative distribution function

$$f_X(x; a, \lambda) = \frac{\lambda^a x^{a-1} e^{-\lambda x}}{(a-1)!} \quad F_X(x; a, \lambda) = 1 - \sum_{j=0}^{a-1} \frac{(\lambda x)^j e^{-\lambda x}}{j!}.\tag{5}$$

To approximate the distribution of the cumulative reward to absorption, we use the mixture of Erlang distributions, motivated by its inherent connection to Markov models and its strong ability to approximate positive distributions arbitrarily well [1, 24, 40]. In this work, we use an Erlang mixture with a common rate parameter to reduce the cost of moment matching without sacrificing approximation quality [7]. Matching the first K analytical moments produces a smooth parametric approximation of the cumulative reward distribution, which we use for robust distributional reasoning beyond expected values.

Notation. The above definitions will be used in robust probabilistic model checking in Sec. 3 and in chance-constrained risk-aware policy synthesis in Sec. 4. For reference, Table 1 summarizes the main symbols used in the following sections for both DTMCs and MDPs.

3 Robust Probabilistic Model Checking for DTMCs

In this section, we summarize the relevant part of the robust probabilistic model checking method proposed in [26], which is used as a verification backbone for the policy optimization method. Section 3.1 reviews the Markov reward distribution approximation based on Erlang mixtures, which reconstructs a smooth density on $[0, +\infty)$ from a finite set of analytical moments and avoids the discretization issues of histogram-based approaches. Section 3.2 then describes the verification framework, including a fast sufficient verification based on Cantelli bounds, and a more accurate

Table 1. Summary of key notation.

Symbol	Meaning
T	Absorption time, i.e., the first time a target or absorbing state is reached
R_T	Cumulative reward accumulated until absorption
R_T^π	Cumulative reward until absorption under policy π
X	Random variable denoting the cumulative reward
τ	Reward threshold used in a chance constraint
α	Required satisfaction probability in a chance constraint
u_k	k -th raw moment of R_T
m_k	k -th central moment of R_T
u_k^π	k -th raw moment of R_T^π under policy π
m_k^π	k -th central moment of R_T^π under policy π
$\rho_n(\pi)$	n -th order Cantelli lower bound on the satisfaction probability of policy π
$\mathcal{L}_c(\pi)$	Barrier-augmented surrogate objective used for policy synthesis
c	Penalty weight in the barrier objective
ϵ	Small slack parameter used to stabilize the barrier objective
f_{approx}	Erlang-mixture approximation of the cumulative reward density
K	Number of analytical moments used for moment matching

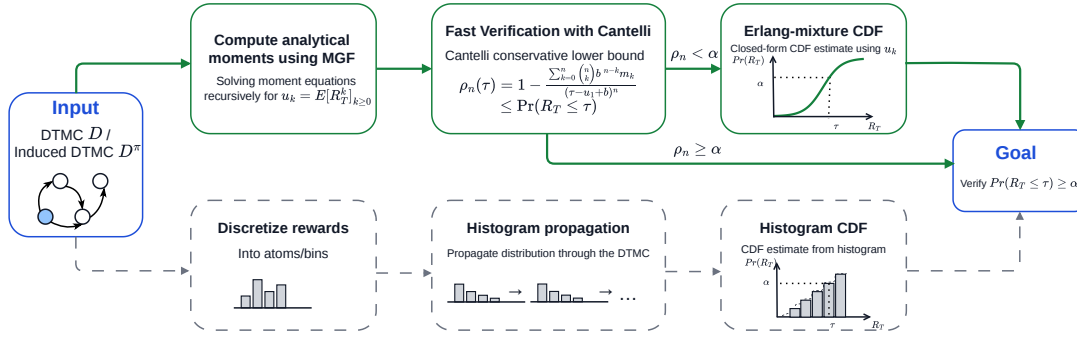


Fig. 2. Workflow for robust probabilistic model checking on DTMCs. The bottom row shows a histogram-based approximation, which discretizes rewards and estimates the satisfaction probability from the propagated histogram distribution. The top row shows the proposed moment-based method, which computes analytical moments and then uses Cantelli’s inequality to provide a conservative lower bound $\rho_n(\tau)$ on the satisfaction probability. If $\rho_n(\tau) \geq \alpha$, the chance constraint is certified directly; otherwise, an Erlang-mixture CDF provides a more accurate estimate.

evaluation based on the Erlang-mixture approximation. Fig. 2 summarizes the overall verification workflow. Our robust probabilistic model checking method avoids reward discretization by computing analytical moments and using them for Cantelli verification and Erlang-mixture approximation.

3.1 Markov Reward Distribution Approximation

To verify chance-constrained properties in Markov models, we start by approximating the distribution of the cumulative reward to absorption R_T in a DTMC D (or induced DTMC D^π) with initial state s_0 and transition matrix P . We model

Algorithm 1 Markov Reward Distribution Approximation

Input: DTMC D (or induced DTMC $\mathcal{D}^\pi = (S, s_0, P, r, AP, L)$), tolerance $\epsilon > 0$, the absorption time T , the number K of moments, and number n of Erlang components

Output: Approximated mixture of n Erlang distributions $f_{\text{approx}}(x)$

- 1: $f = \sum_{y \in T} e^{kr(x)} P(x, y) \quad x \in S$ and $G = e^{kr(x)} P(x, y) \quad x, y \in S$
- 2: $\mu_k = r^k f + \sum_{i=0}^k \binom{k}{i} G^{(k-i)} \quad \forall k \in \{1, \dots, K\}$ ▷ Compute analytical moments with Eq. (2)
- 3: Initialize $\mathcal{L} = \infty$ and $\omega_i, a_i, \lambda \quad \forall i \in \{1, \dots, n\}$
- 4: **repeat**
- 5: $\mathcal{L}_{\text{prev}} = \mathcal{L}$
- 6: $\hat{\mu}_k = \sum_{i=1}^n \omega_i \frac{(a_i+k-1)!}{(a_i-1)!} \frac{1}{\lambda^k} \quad \forall k \in \{1, \dots, K\}$ ▷ Compute approximate moments
- 7: $H(f_{\text{approx}}) = - \int_0^\infty f_{\text{approx}}(x) \log f_{\text{approx}}(x) dx$ ▷ Calculate the differential entropy
- 8: $\min_{\omega_i, a_i, \lambda} \mathcal{L} = \sum_{k=1}^K (\mu_k - \hat{\mu}_k)^2 - \gamma H(f_{\text{approx}})$
- 9: subject to $\sum_{i=1}^n \omega_i = 1, \quad \omega_i \geq 0, \quad a_i \in \mathbb{N}^+, \quad \lambda > 0$
- 10: $\omega_i, a_i, \lambda \leftarrow \arg \min_{\omega_i, a_i, \lambda} \mathcal{L}$ ▷ Update parameters
- 11: **until** $|\mathcal{L}_{\text{prev}} - \mathcal{L}| < \epsilon$
- 12: **return** $f_{\text{approx}}(x) = \sum_{i=1}^n \omega_i f_{\text{Erlang}}(x; a_i, \lambda)$

the approximating density f_{approx} as a mixture of n Erlang components:

$$f_{\text{approx}}(x) = \sum_{i=1}^n \omega_i f_{\text{Erlang}}(x; a_i, \lambda), \quad (6)$$

where $\omega_i \geq 0$ are non-negative mixture weights with $\sum_{i=1}^n \omega_i = 1$, $a_i \in \mathbb{Z}^+$ are shape parameters, and $\lambda \in \mathbb{R}^+$ is a common rate parameter. This mixture defines a smooth and differentiable density on $[0, +\infty)$, which avoids the piecewise-constant artifacts of histogram approximations.

We formulate this approximation as a truncated Stieltjes moment problem on $[0, +\infty)$ [19]. Given the first K analytical moments $\mu_1, \dots, \mu_K$ of R_T (computed as in Section 2.3), we choose the mixture parameters (ω_i, a_i, λ) so that the first K moments $\hat{\mu}_1, \dots, \hat{\mu}_K$ of f_{approx} match the target moments. We also include an entropy regularizer [18, 35] to encourage a less biased approximation by maximizing entropy.

The parameters in the mixture are obtained by solving the following moment-matching problem:

$$\min_{\omega_i, a_i, \lambda} \mathcal{L} = \sum_{k=1}^K (\mu_k - \hat{\mu}_k)^2 - \gamma H(f_{\text{approx}}), \quad (7)$$

where $H(f_{\text{approx}}) = - \int_0^\infty f_{\text{approx}}(x) \log f_{\text{approx}}(x) dx$ is the differential entropy of the mixture, and $\gamma \geq 0$ controls the trade-off between moment fit and entropy. For an Erlang mixture with common rate λ , the k -th raw moment is $\hat{\mu}_k = \sum_{i=1}^n \omega_i \frac{(a_i+k-1)!}{(a_i-1)!} \frac{1}{\lambda^k}$.

Main Algorithm. Algorithm 1 takes a DTMC D (or an induced DTMC $\mathcal{D}^\pi$), a tolerance ϵ , the number of moments K , and mixture size n , and returns an Erlang-mixture approximation of the reward distribution to absorption. For a fixed K , $\lfloor K/2 \rfloor + 1$ Erlang components are often sufficient to match K moments while keeping the mixture compact [28]. The method first computes the target moments μ_k analytically (lines 1-2) and the differential entropy $H(f_{\text{approx}})$. Then it iteratively updates (ω_i, a_i, λ) to minimize the objective $\mathcal{L}$ under the simplex and positivity constraints until the convergence criteria are satisfied. The algorithm then returns the approximated distribution with the optimized parameters for further analysis via robust model checking. To improve numerical stability when matching higher

moments, we optimize over standardized moments $\bar{\mu}_k = \mu_k / c^k$ for $k \geq 3$, using $c = \sqrt{u_2 - u_1^2}$ (the standard deviation) as a scale factor. We further provide the initial values and a hyperparameter ablation study in Section 5.

Algorithm 1 can approximate the cumulative reward distribution arbitrarily closely by increasing the number of moments K and the number of Erlang components n [19, 40]. The entropy term encourages selecting a less biased result and avoids pathological fits when only a small number of moments is available, which is useful in robust model checking where conservative decisions are preferred over overfitting to limited statistics [35].

Complexity Observations. We discuss the worst-case complexity of this algorithm in two parts, and note that in practice, the complexity can be further reduced.

i) Computing the first K moments requires solving K linear systems whose dimension matches the state-space size ξ . A worst-case complexity for solving this linear system is $O(\xi^3)$, standard numerical solvers (e.g., Numpy [23]) usually improve over such a bound and can exploit the sparsity and possible symmetries in the transition matrix for more efficient computations.

ii) Fitting an Erlang-mixture to match the moments leads to a mixed-integer non-linear optimization problem, which is NP-hard in general. In practice, we reduce the computational burden by fixing the Erlang shape parameters to a prescribed well-spread set and optimizing only the remaining continuous parameters (rates and mixture weights). This converts the problem into a more tractable continuous optimization that can be solved efficiently with state-of-the-art optimizers (such as Gurobi [22] or SciPy [41]). We refer to the detailed discussion of these practical choices and the accompanying empirical evaluation in [26], where we also report on the accuracy and computation time required for different spreading strategies and the number of components.

The theoretical justification for approximating the cumulative reward distribution of a DTMC using mixtures of Erlang distributions is provided in [26]. The argument proceeds in three steps: (i) discretize rewards to obtain a discrete phase-type (DPH) approximation with a uniform bound on the CDF error; (ii) convert this to a continuous phase-type (PH) approximation via Erlangization; and (iii) approximate the resulting nonnegative distribution by a mixture of Erlang distributions. Overall, this establishes a constructive approximation pipeline from the Markov reward distribution to an Erlang-mixture representation with a provable error bound.

3.2 Robust Property Evaluation

We now detail the robust property evaluation formulation used in probabilistic model checking. This formulation verifies that the probability that a performance measure violates a threshold remains below or above a predefined level α , rather than only validating whether the expected reward stays below or above this threshold. We present two complementary evaluation methods for the robust property evaluation: (i) a fast sufficient check based on Cantelli's inequality, and (ii) a more accurate verification based on Erlang mixtures.

We formalize robust property evaluation for chance-constrained requirements [15] on the distribution of the cumulative reward until absorption, denoted by X , as

$$\Pr(X \leq r^*) \geq \alpha, \quad (8)$$

where r^* is a verification threshold controlling tail behavior and α is the required probability level. Constraints of the form $X \geq r^*$, or more generally $r_* \leq X \leq r^*$, can be handled in the same way using standard properties of the cumulative distribution function.

The almost-sure properties, where the probability of satisfying a temporal formula is required to be 1 (or 0), can also be phrased as chance-constrained requirements; whereas chance constraints additionally enable controlled relaxation of

requirements. This makes our distributional approach well-suited to verifying probabilistic safety requirements [29], and to robustness analysis of policies synthesized for MDPs via reinforcement learning [21], by reasoning over the induced DTMC model [4, Ch. 10].

Efficient Sufficient Conditions. This robust property can be validated efficiently once the moments of the reward distribution have been computed by using probabilistic inequalities derived from Markov’s theorem [20]. These kinds of inequalities bound the probability that a random variable exceeds a threshold using only finitely many moments, independent of the distribution’s exact shape. As a result, these bounds can be conservative, as we illustrate later. Under our assumption of non-negative rewards, the sharpest Markov-type bound is Cantelli’s inequality, as we detailed in Section 2. We can reuse the general and classical Cantelli bound in Equation 3 and 4

$$\Pr(R_T \leq \tau \mid X_0 = s_0) \geq 1 - \frac{\sum_{k=0}^n \binom{n}{k} b^{n-k} m_k}{(\tau - u_1 + b)^n} \quad \Pr(R_T \geq \tau \mid X_0 = s_0) \leq \frac{u_2 - u_1^2}{(\tau - u_1)^2 + (u_2 - u_1^2)}. \quad (9)$$

For any integer $n \geq 2$, we define $u_k = \mathbb{E}[R_T^k \mid X_0 = s_0]$ as the k -th raw moment and $m_k = \mathbb{E}[(R_T - u_1)^k \mid X_0 = s_0]$ as the k -th central moment, for any threshold $\tau > u_1$ and any shift $b > -(\tau - u_1)$. For $n = 2$, choosing $b = \frac{u_2 - u_1^2}{\tau - u_1}$ minimizes the bound and yields the classical closed form of Cantelli’s inequality above.

After computing the first K moments of the reward distribution in the initialization phase, Equation (3) can provide a fast and robust decision on the probabilistic model checking problem by comparing its right-hand side with the target value α . If the chance constraint holds, the decision procedure terminates. Otherwise, it is still possible that Cantelli’s inequality is not satisfied due to its conservativeness – the inequality must hold for *any* distribution. In such a case, reconstructing f_{approx} is required to obtain a more accurate answer.

When Cantelli’s inequality does not yield a decisive evaluation of the robust property (see the example below), we instead use the Erlang-mixture approximation f_{approx} constructed in Algorithm 1. Given f_{approx} , evaluating the chance-constrained requirement in Equation (8) reduces to computing the cumulative distribution function of a finite Erlang mixture:

$$\Pr(X \leq r^*) = \sum_i \omega_i F_{X_i}(r^*; a_i, \lambda_i) =: F_{\text{approx}}(r^*), \quad (10)$$

where the cumulative distribution functions $F_{X_i}(x; a_i, \lambda_i)$ are defined for each Erlang component of the mixture as in Equation (5).

Example. Fig. 3 shows the cumulative distribution for our UAV running example. The solid green line shows the CDF F_{approx} (integral of f_{approx}) estimated with our method by matching the first three moments to construct a mixture of three Erlang distributions. The shaded bars show the empirical distribution drawn from one million simulations of the system, which we use to approximate the ground truth. The dashed vertical lines show the probability bounds computed using Cantelli’s inequality for the chance requirement of Equation (8), with values of r^* being on the x-axis and each dashed line showing a different probability threshold α . Consider the requirement $\Pr(X \leq 15) \geq 0.65$. Cantelli’s inequality ($n = 3$) ensures that $\Pr(X \leq 15) \geq 0.68$, therefore the requirement holds and there is no need to compute f_{approx} (optimization loop in Algorithm 1) to draw a conclusion. If we were instead to verify $\Pr(X \leq 15) \geq 0.8$, due to its conservative nature, Cantelli’s bound would not be able to prove that the system indeed satisfies the requirement, as correctly concluded by evaluating $F_{\text{approx}}(15)$.

The moment matching process ensures that the approximation closely represents the true stochastic behavior of the system using only the first K moments [19]. The use of multiple moments captures both central tendencies and tail

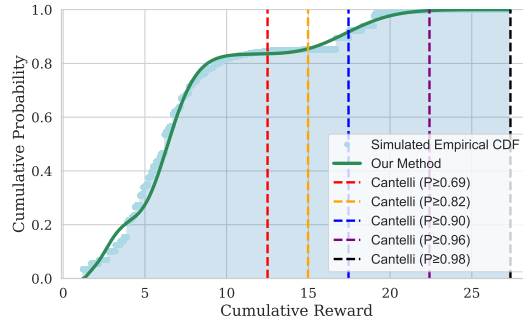


Fig. 3. CDF-based evaluation of the cumulative reward distribution in the UAV example. The light-blue samples show the simulated empirical CDF, while the green curve shows the Erlang-mixture CDF estimated by our method. The dashed vertical lines mark thresholds certified by Cantelli’s inequality at different probability levels, illustrating that Cantelli provides fast conservative certificates, while the Erlang-mixture CDF gives a more accurate distributional estimate.

behaviors, which are essential for assessing system robustness. We assess experimentally the quality of the approximants with different numbers of moments, as well as the computation time required to construct them (see [26] for details).

4 Robust Chance-constrained Policy Optimization

In this section, we move from *verification* to *control*. In standard PMC, policy synthesis for expected-reward objectives is solved by dynamic programming, typically via value iteration [29]. A direct distributional extension would maintain an approximation of the cumulative reward distribution during policy updates (e.g., histograms, kernel density estimation, or Erlang mixtures) and improve the policy using tail probabilities to evaluate the chance constraint. This is expensive because each update propagates distributional objects, and approximate CDFs are not automatically conservative, so they may underestimate tail risk for some policies.

To avoid explicit distribution tracking inside the optimization loop, we propose the use of analytical moment equations (Section 2.3) and a Cantelli bound to obtain a conservative bound on the satisfaction probability. Given a finite MDP $\mathcal{M}$, we synthesize a stationary policy that satisfies a chance constraint on the cumulative reward up to absorption while optimizing expected performance. We then embed this bound into a piecewise differentiable penalty and derive a value-iteration update that trades off the mean reward against higher-order moment penalties. We refer to this construction as a *conservative surrogate*: it replaces the original chance constraint with a sufficient condition based on a lower bound of the satisfaction probability. The resulting unconstrained objective is a *differentiable relaxation* because the hard, non-smooth chance constraint is replaced by a differentiable penalty term that can guide policy improvement. We present the $R_{\min}$ case (i.e., minimizing expected cumulative reward) to simplify notation; the same construction applies to $R_{\max}$ by switching the tail direction and the greedy improvement step. Fig. 4 summarizes the proposed policy-synthesis workflow. At each iteration, the current policy induces a DTMC whose moments define a Cantelli-based conservative bound and a barrier-augmented objective, which then guides the greedy policy update.

Problem Formulation. We consider a finite MDP $\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, \mathcal{P}, \mathcal{R})$ with absorption time T . For a stationary policy π , the cumulative reward up to absorption is $R_T^\pi = \sum_{t=0}^{T-1} r(X_t, \pi(X_t))$. We assume that the expected cumulative reward to absorption is bounded under any policy by a value $R_{\max} < \infty$. Given a threshold $\tau \in \mathbb{R}_{\geq 0}$ and a confidence level $\alpha \in (0, 1)$, we aim to solve the chance-constrained $R_{\min}$ problem

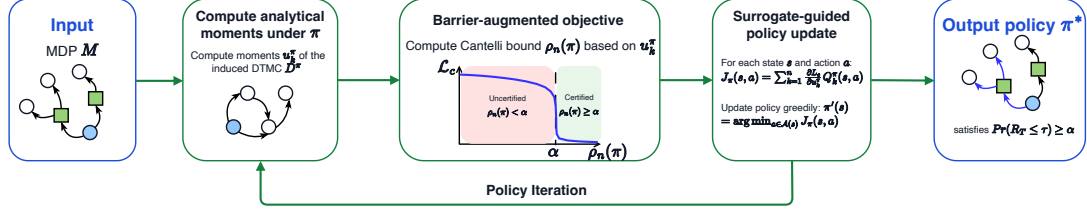


Fig. 4. Workflow for robust chance-constrained policy optimization. For the current policy π , the method computes moments on the induced DTMC $\mathcal{D}^\pi$, forms the Cantelli bound $\rho_n(\pi)$ and barrier-augmented objective $\mathcal{L}_c$, and uses the resulting state-action score $J_\pi(s, a)$ for a greedy policy update. The iteration returns a policy π^* that is certified when $\rho_n(\pi^*) \geq \alpha$.

$$\min_{\pi} \mathbb{E}[R_T^\pi | X_0 = s_0] \quad \text{s.t.} \quad \Pr(R_T^\pi \leq \tau | X_0 = s_0) \geq \alpha. \quad (11)$$

Because any policy π of the MDP $\mathcal{M}$ induces a DTMC $\mathcal{D}^\pi$, one can in principle compute the moments of the cumulative reward distribution analytically and approximate the CDF using the Erlang approximation recalled in Section 2.3. Since the policy search space is finite, such a process terminates, yielding a solution to the problem in Equation (11). However, this approach would be intractable in practice, due to both the combinatorial explosion of the policy search space and the cost of fitting an Erlang distribution for each explored policy. Additionally, it may be difficult to decide a priori the size of an Erlang mixture that would yield satisfactory accuracy for the tail behavior of each possible policy, which would further exacerbate the cost of the fitting problem if large mixtures are conservatively used.

Instead, in the remaining of this section, we propose a different policy search method that relies on the n -th order Cantelli bound in (3) to assess whether a policy satisfies the chance constraint in Equation (11) and define a differentiable relaxation of the problem that guides policy optimization toward policies satisfying the conservative certificate, while optimizing in best-effort the expected reward of the system.

Barrier-Augmented Surrogate Objective. A common approach for constrained MDPs is to replace the hard constraint and convert (11) into an unconstrained problem using a log-barrier [33, 43]. Following this idea, we introduce a barrier-augmented surrogate objective $\mathcal{L}_c$ that reformulates the chance-constrained problem into an unconstrained optimization, while retaining the constraint behavior through a conservative satisfaction bound based on the Cantelli inequality. The objective is called *barrier-augmented* because it augments the expected-reward objective with an additive barrier penalty. The log-barrier term grows sharply as the conservative satisfaction bound approaches α from the feasible side, while the linear extension penalizes policies that violate the conservative bound. This barrier we formulate serves the dual purpose of efficiently guiding the exploration of the policy space towards regions that satisfy the chance constraint, while optimizing the expected reward within such satisfactory regions.

For any fixed policy π , we first compute a conservative lower bound on the satisfaction probability $\Pr(R_T^\pi \leq \tau | X_0 = s_0)$ using the n -th order Cantelli bound following (3)

$$\rho_n(\pi) := 1 - \frac{\sum_{k=0}^n \binom{n}{k} b^{n-k} m_k^\pi}{(\tau - u_1^\pi + b)^n},$$

where u_k^π denotes the k -th raw moment of the cumulative reward distribution, m_k^π denotes the corresponding k -th central moment, $\tau > u_1^\pi$ is the threshold and b is the shift applied to the central moments as defined in Section 2.3. Since $\rho_n(\pi)$ is a lower bound on the true satisfaction probability, the condition $\rho_n(\pi) \geq \alpha$ is a conservative certificate that

$\Pr(R_T^\pi \leq \tau \mid X_0 = s_0) \geq \alpha$. We assume per-step rewards are non-negative and bounded, which implies $\max_\pi u_1^\pi(s_0) < \infty$; $R_{\max}$ is an upper bound on the expected reward under any policy, i.e., $\max_\pi u_1^\pi(s_0) < R_{\max} < \infty$.

We define the barrier-augmented surrogate objective $\mathcal{L}_c$, following the smoothed log-barrier idea in [43] to enable stable optimization both inside and outside the policy search region that satisfies the chance constraint. In particular, we define a piecewise log-barrier with a linear extension based on whether the chance constraint is satisfied according to the conservative Cantelli bound (i.e., $\rho_n(\pi) - \alpha \geq 0$) or not:

$$\mathcal{L}_c(\pi) = u_1^\pi - \begin{cases} c \log(\rho_n(\pi) - \alpha + \varepsilon), & \rho_n(\pi) - \alpha \geq 0, \\ \frac{c(\rho_n(\pi) - \alpha)}{\varepsilon} + c \log(\varepsilon) - R_{\max}, & \rho_n(\pi) - \alpha < 0, \end{cases} \quad (12)$$

where $c > 0$ is a penalty weight and $\varepsilon > 0$ is a small slack to ensure the logarithm is well defined. The sign of $\rho_n(\pi) - \alpha$ separates the policy space into two regions: policies with $\rho_n(\pi) \geq \alpha$ are certified by the Cantelli bound to satisfy the chance constraint, while policies with $\rho_n(\pi) < \alpha$ are not certified and are treated as infeasible by the surrogate objective.

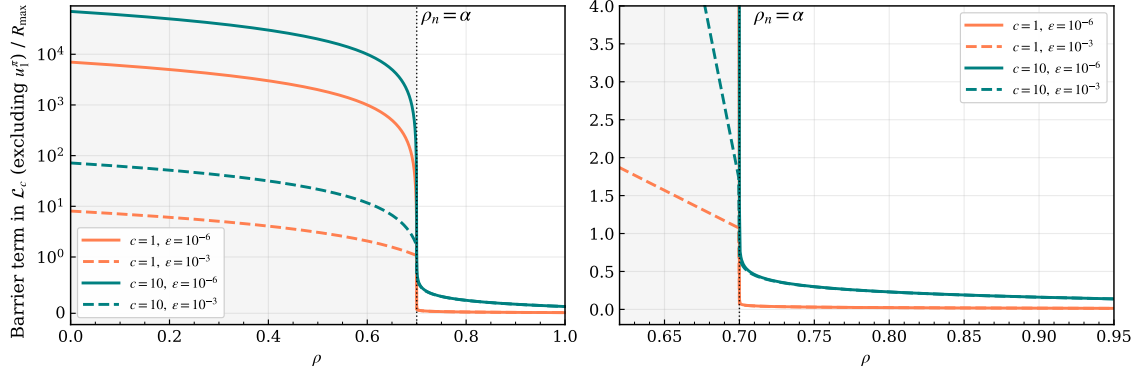


Fig. 5. Log-barrier objective with linear extension for the chance constraint. The barrier term in $\mathcal{L}_c$ (excluding u_1^π) is plotted as a function of $\rho_n(\pi) \in [0, 1]$ with $\alpha = 0.7$, comparing two penalty weights $c \in \{1, 10\}$, two values $\varepsilon \in \{10^{-6}, 10^{-3}\}$ and $R_{\max} = 100$. The dotted line marks the chance constraint boundary $\rho_n(\pi) = \alpha$, and the shaded region indicates the constraint violation set $\rho_n(\pi) < \alpha$; the right panel zooms into the neighborhood of α .

Within the satisfaction region $\rho_n(\pi) - \alpha \geq 0$, the log-barrier term $c \log(\rho_n(\pi) - \alpha + \varepsilon)$ discourages the policy from approaching the constraint boundary. While in the violation region, we use a linear extension $\frac{c(\rho_n(\pi) - \alpha)}{\varepsilon}$ to provide a non-vanishing gradient that pushes the policy towards reducing the constraint violation, together with a fixed infeasibility penalty $c \log(\varepsilon) - R_{\max}$ that ensures that any policy violating the chance constraint receives a penalty in excess of the maximum expected reward achievable along any policy. Figure 5 illustrates visually how policies with $\rho_n(\pi)$ close to the target α incur a large barrier penalty, while policies well inside the satisfactory region (right-hand side of the figure) are mainly driven by u_1^π (the expected reward) for reward optimization. Therefore, $\mathcal{L}_c$ can be viewed as an interior-point approximation of the original chance-constrained optimization. The value c controls the strength of the penalty and we will discuss its adaptive tuning later.

Dynamic Programming Updates. In the finite MDP setting with known transition dynamics, we would like to leverage dynamic programming (e.g., value iteration) to obtain a stable, model-based policy improvement scheme. However, while

the barrier objective $\mathcal{L}_c(\pi)$ removes the explicit chance constraint, it remains a *policy-level* functional: the satisfaction bound on $\rho_n(\pi)$ depends on moments of the entire reward distribution induced by π . This dependence is not additively separable across states in a trajectory, and therefore $\mathcal{L}_c(\pi)$ does not admit a standard Bellman decomposition.

To obtain a Bellman-style greedy improvement step, we linearize $\mathcal{L}_c$ with respect to the moments $(u_k^\pi)_{k=1}^n$ and take a first-order expansion around the current policy π , yielding the following practical update rule:

$$d\mathcal{L}_c \approx \sum_{k=1}^n \frac{\partial \mathcal{L}_c}{\partial u_k^\pi} du_k^\pi \quad (13)$$

$$\kappa_1 := \frac{\partial \mathcal{L}_c}{\partial u_1^\pi} = 1 - \frac{c}{\max(\rho_n(\pi) - \alpha, 0) + \varepsilon} \frac{\partial \rho_n(\pi)}{\partial u_1^\pi}, \quad \kappa_k := \frac{\partial \mathcal{L}_c}{\partial u_k^\pi} = -\frac{c}{\max(\rho_n(\pi) - \alpha, 0) + \varepsilon} \frac{\partial \rho_n(\pi)}{\partial u_k^\pi}, \quad k = 2, \dots, n. \quad (14)$$

The coefficients κ_k measure the sensitivity of the barrier objective to changes in the k -th moment. When $\rho_n(\pi)$ is less than or close to α , these weights drive towards changes that increase $\rho_n(\pi)$.

We then score each state-action choice by its effect on the propagated moments under the current policy π . For each state-action pair (s, a) , we define the local one-step propagated k -th raw moment under π as the k -th moment of the cumulative reward obtained by starting in state s , taking action a at $t = 0$, and thereafter following π until absorption:

$$Q_k^\pi(s, a) := \mathbb{E} \left[(r(s, a) + R^\pi(X_1))^k \mid X_0 = s, A_0 = a \right] = \sum_{s'} P(s' \mid s, a) \sum_{j=0}^k \binom{k}{j} r(s, a)^{k-j} u_j^\pi(s'). \quad (15)$$

This leads to the state-wise surrogate (following the first-order approximation $d\mathcal{L}_c \approx \sum_{k=1}^n \kappa_k du_k^\pi(s_0)$), which scores each action by its effect on the propagated moments:

$$J(s, a) := \sum_{k=1}^n \kappa_k Q_k^\pi(s, a). \quad (16)$$

We construct a candidate policy π' by choosing, at each state s , an action a that minimizes $J(s, a)$, which guides a greedy improvement step. This update is consistent with decreasing $\mathcal{L}_c$ under the local linearization. We then evaluate the candidate policy using the barrier-augmented objective $\mathcal{L}_c$ for a best feasible policy that optimizes the objective value.

Main Algorithm. Algorithm 2 summarizes our robust risk-sensitive value iteration method. Given a policy π , we first evaluate the moments $(u_1^\pi, \dots, u_n^\pi)$ by solving the linear systems on the induced DTMC $\mathcal{D}^\pi$, as introduced in Section 2.3. We then compute the Cantelli bound $\rho_n(\pi)$ at the initial state to derive the coefficients $(\kappa_1, \dots, \kappa_n)$ in the surrogate objective (16). We compute the propagated moments $Q_k^\pi(s, a)$ and the state-action surrogate $J(s, a) = \sum_{k=1}^n \kappa_k Q_k^\pi(s, a)$, and perform a greedy policy-improvement step $\pi'(s) \leftarrow \arg \min_{a \in A(s)} J(s, a)$ (when the goal is to maximize the expected reward/cost, we can reverse the chance-constraint direction and replace the greedy step by $\arg \max$).

After constructing π' , we re-evaluate its moments $(u_1^{\pi'}, \dots, u_n^{\pi'})$, compute the corresponding Cantelli bound $\rho_n(\pi')$, and update the current best policy π^* when the current barrier-augmented objective $\mathcal{L}_c(\pi)$ improves. We also update the penalty weight based on the constraint satisfaction signal according to

$$c \leftarrow \begin{cases} \min\{\beta c, c_{\max}\}, & \text{if } \rho_n(\pi') < \alpha, \\ \max\{c/\beta, c_{\min}\}, & \text{if } \rho_n(\pi') \geq \alpha. \end{cases} \quad (17)$$

Algorithm 2 Robust Risk-sensitive Value Iteration

Input: MDP $\mathcal{M} = (S, s_0, \mathcal{A}, \mathcal{P}, r)$, initial policy π_0 , target (α, τ) , penalty weight $c_0 > 0$, tolerance $\varepsilon > 0$, moment order $n \geq 2$

Output: Policy π^*

```

1: Initialize  $\pi \leftarrow \pi_0, c \leftarrow c_0$ 
2:  $(u_1^\pi, \dots, u_n^\pi) \leftarrow \text{evaluate\_moments}(\pi)$  ▷ via Eq. (2) on  $\mathcal{D}^\pi$ 
3:  $\pi^* \leftarrow \pi_0$ 
4: Compute  $\rho_n(\pi^*)$  from  $(u_k^\pi(s_0))_{k=1}^n$  using the  $n$ -th order Cantelli bound
5: repeat
6:   Compute  $(\kappa_1, \dots, \kappa_n)$  from  $(u_k^\pi(s_0))_{k=1}^n, \tau, \alpha$ , and  $c$ 
7:   for  $s \in S$  do
8:     for  $a \in A(s)$  do
9:       for  $k = 1, \dots, n$  do
10:         $Q_k(s, a) \leftarrow \sum_{s'} P(s' | s, a) \sum_{j=0}^k \binom{k}{j} r(s, a)^{k-j} u_j^\pi(s')$ 
11:      end for
12:       $J(s, a) \leftarrow \sum_{k=1}^n \kappa_k Q_k(s, a)$ 
13:    end for
14:     $\pi'(s) \leftarrow \arg \min_{a \in A(s)} J(s, a)$ 
15:  end for
16:   $(u_1^{\pi'}, \dots, u_n^{\pi'}) \leftarrow \text{evaluate\_moments}(\pi')$ 
17:  Compute  $\rho_n(\pi')$  from  $(u_k^{\pi'}(s_0))_{k=1}^n$  using the  $n$ -th order Cantelli bound
18:  if  $\mathcal{L}_c(\pi') < \mathcal{L}_c(\pi^*)$  then
19:     $\pi^* \leftarrow \pi'$ 
20:  end if
21:  if  $\rho_n(\pi') < \alpha$  then
22:     $c \leftarrow \min\{\beta c, c_{\max}\}$ 
23:  else
24:     $c \leftarrow \max\{c/\beta, c_{\min}\}$ 
25:  end if
26:   $\pi \leftarrow \pi', \quad u_k^\pi \leftarrow u_k^{\pi'} \text{ for } k = 1, \dots, n$ 
27: until convergence in  $\rho_n(\pi)$  or  $\pi$ 
28: return  $\pi^*$ 

```

This schedule increases the barrier weight whenever the Cantelli bound falls below the target level α , and relaxes it otherwise, while the bounds prevent c from diverging. In particular, when the constraint is violated, increasing c prioritizes driving the policy toward the satisfactory region; once the policy enters and remains in the feasible region, the update reduces c , which allows the optimization to focus more on optimizing the expected reward $u_1^{\pi'}$ rather than aggressively enforcing feasibility. Intuitively, when c approaches zero in Equation (14), the coefficient κ_1 converges to 1, while all the others converge to 0, driving Algorithm 2 towards optimizing the expected reward.

Each outer iteration requires solving n linear systems of size $|S|$ to evaluate the moments under the current policy, and performing standard dynamic-programming backups to compute Q_k and J . All quantities are obtained analytically from the model, without maintaining a full cumulative reward distribution over state-action pairs inside the optimization loop, which allows for tractable practical complexity, as evaluated in Section 5.

Notice that Algorithm 2 does not, in general, guarantee global optimality with respect to the original chance constraint on the CDF of the reward distribution. Since the Cantelli bound provides only a sufficient condition, the method is generally conservative (as shown in Section 3.2). Additionally, the greedy nature of the algorithm may, in general,

converge to a local optimum. Therefore, the objective value returned by the algorithm should be interpreted as the performance of a feasible policy obtained under a conservative surrogate, rather than as the optimal value of the original chance-constrained problem. In Section 5, we empirically evaluate this trade-off and show that the resulting policies achieve strong tail behavior with tractable computational costs.

Soundness considerations. The soundness argument is based on the one-sided nature of the Cantelli bound. For a fixed policy π , $\rho_n(\pi)$ is a lower bound on the true satisfaction probability $\Pr(R_T^\pi \leq \tau \mid X_0 = s_0)$. Therefore, if $\rho_n(\pi) \geq \alpha$, then the original chance constraint is guaranteed to hold. Conversely, if $\rho_n(\pi) < \alpha$, the policy is not necessarily unsafe with respect to the true CDF; it is only not certified by the conservative bound. The barrier objective therefore partitions the policy space into Cantelli-certified feasible policies and uncertified policies. The infeasibility penalty in $\mathcal{L}_c$ creates an objective-value separation between these two regions: uncertified policies are assigned a larger surrogate value than any certified feasible policy under the stated bound on $u_1^\pi(s_0)$.

We characterize two soundness properties induced by the barrier-augmented objective in detail. Our analysis relies on an objective-value separation between feasible policies and infeasible policies under the barrier-augmented objective $\mathcal{L}_c$, which is enforced by the infeasibility step penalty. As a consequence, provided that a policy satisfying the chance constraint based on the Cantelli conservative bound exists, (i) any global minimizer of $\mathcal{L}_c$ over deterministic stationary policies must be feasible, and (ii) in Algorithm 2, once the policy π^* becomes feasible, the acceptance rule based on $\mathcal{L}_c$ prevents it from being replaced by an infeasible policy in subsequent iterations.

Proposition 4.1. *Assume $0 \leq u_1^\pi(s_0) < R_{\max}$ for any policy π , and there exists a feasible policy, then the global minimizer of $\mathcal{L}_c$ satisfies the chance constraint.*

PROOF. Let $F := \{\pi : \rho_n(\pi) \geq \alpha\}$ be the set of feasible policies. By definition, we have

$$\inf_{\pi \in F^c} \mathcal{L}_c(\pi) > R_{\max} - c \log \varepsilon \geq \sup_{\pi \in F} \mathcal{L}_c(\pi)$$

therefore $\arg \min_{\pi} \mathcal{L}_c(\pi) \in F$. □

Proposition 4.2. *Let $F := \{\pi : \rho_n(\pi) \geq \alpha\}$ be the set of feasible policies. Assume $0 \leq u_1^\pi(s_0) < R_{\max}$ for every deterministic stationary policy π . Define $\mathcal{L}_c$ as in (12). For Algorithm 2, if there exists an iteration t_0 such that $\pi_{t_0}^* \in F$, then $\pi_t^* \in F$ for all $t \geq t_0$; hence the final synthesized policy is feasible and satisfies chance constraint: $\Pr(R_T^{\pi^*} \leq \tau \mid s_0) \geq \alpha$.*

PROOF. For the set of infeasible policies $F^c := \{\pi : \rho_n(\pi) < \alpha\}$, assume $\pi^* \in F$ at some iteration and any infeasible candidate $\pi' \in F^c$. Then

$$\mathcal{L}_c(\pi') \geq \inf_{\pi \in F^c} \mathcal{L}_c(\pi) > \sup_{\pi \in F} \mathcal{L}_c(\pi) > \mathcal{L}_c(\pi^*),$$

Therefore, a feasible policy π^* cannot be replaced by an infeasible policy (line 18 of Algorithm 2), proving the claimed invariance. □

Even if intermediate policies oscillate across the boundary due to approximate greedy updates and adaptive change of c , the returned π^* remains protected by the objective-value separation.

5 Evaluation

We evaluate our chance-constrained policy synthesis method on MDP case studies adapted from PRISM [31], and compare against standard risk-neutral value iteration and a distributional value iteration baseline [13] in risk-aware settings. The goal is to assess how our method trades off expected cumulative reward and tail risk while enforcing a

chance constraint on the cumulative reward to absorption, and to quantify the resulting computational cost relative to classical value iteration and distributional value-iteration methods. In particular, we report the expected cumulative reward, tail-risk metrics, chance-constraint satisfaction, and run time across benchmarks, highlighting how our approach can achieve a favorable robustness–efficiency trade-off without maintaining full cumulative reward distributions during policy updates. A proof-of-concept implementation is available at <https://github.com/xtji/robust-pmc>.

Benchmarks. All benchmarks are adapted from PRISM case studies [13, 31, 34] and model stochastic control tasks as MDPs. We consider (i) *Betting Game*, where an agent starts with \$5 and repeatedly chooses a bet size $0 \leq \lambda \leq 5$ for 10 stages; each bet wins with probability 0.7, loses with probability 0.25, or hits a jackpot with probability 0.05. The terminal reward is defined as $V_{\max} = 100$ minus the final wealth. (ii) *Deep Sea Treasure*, where a submarine navigates to collect one treasure within 15 steps: each move succeeds with probability 0.6 and deviates to an adjacent cell with probability 0.2 (per deviation direction); the reward combines travel cost and an opportunity-cost term $V_{\max} = 800$ minus the collected treasure values. (iii) *GridWorld*, an $N \times N$ navigation task with scattered obstacles, where stochastic actions may fail and lead to unintended neighboring cells. (iv) *Future Investor*, a continuous-valued reward benchmark modeling a two-player market–investor interaction, in which the investor chooses a reservation month to maximize the eventual sale value, while the market evolves stochastically; this continuous-valued reward benchmark is not directly compatible with histogram-based distributional methods that rely on discretized augmented state spaces [13].

Baselines. We compare our proposed *robust risk-sensitive value iteration* (robust risk-sens. VI), which optimizes expected cumulative reward subject to a chance constraint enforced via the Cantelli bound, against two dynamic-programming baselines: (1) *Risk-neutral value iteration* (risk-neut. VI), which minimizes the expected cumulative reward using the standard Bellman operator; (2) *Risk-sensitive distributional value iteration* (risk-sens. DVI) from [13], which maintains an explicit approximation of the cumulative reward distribution and optimizes a CVaR_α objective (as defined in Sec. 2.2) via an augmented MDP and a histogram-based representation. Because this augmented construction relies on a discretized slack variable, risk-sens. DVI is only applicable to the integer-reward benchmarks and cannot be run on the continuous-valued Future Investor benchmark. In contrast, robust risk-sens. VI operates on the original MDP state space and applies to both discrete- and continuous-valued reward settings, optimizing the chance-constrained objective using finite analytical moments.

Experimental Settings. In our implementation, the linear systems required to compute the analytical moments of the reward process are solved with NumPy [23]. We evaluate the quality of policies synthesized by risk-neut. VI, risk-sens. DVI (for the integer-reward benchmarks where it is applicable), and our robust risk-sens. VI on the MDP benchmarks, comparing optimality, tail-risk metrics, and computational cost. We implement risk-neut. VI following the standard Bellman optimality formulation. Our Cantelli-based chance-constraint surrogate uses the first $n=2$ moments. For risk-sens. DVI [13], we rerun the open-source implementation on GridWorld ($N=10$). This environment is not included in the main results table of [13], whereas the remaining environments are reported there. For the ablation study involving an explicit distributional approximation, we fit Erlang-mixture models via moment matching with a maximum-entropy regularizer, using SciPy SLSQP [41] and Gurobi [22] for optimization.

For each benchmark, we report the expected cumulative reward $\mathbb{E}[R_T^r]$ to capture average performance, the chance-constraint satisfaction, the conditional value-at-risk $\text{CVaR}_\alpha(R_T^r)$ at the same risk level α , and the runtime measured on the same device across methods. The chance constraint corresponds to a VaR/quantile requirement at level α that controls violation frequency, while CVaR_α captures the expected severity of outcomes in the corresponding tail beyond that quantile. The *States* column in Table 2 reports the number of states on which each method operates, highlighting the

Table 2. Performance comparison of runtime, risk metrics, and chance-constraint satisfaction across PRISM benchmarks, evaluated with respect to the chance constraint $\Pr(R_T^\pi \leq \tau) \geq \alpha$.

Model	Method	States	Time (s)	$\mathbb{E}[R_T^\pi]$	$\text{CVaR}_\alpha[R_T^\pi]$	Satisfaction?
Betting Game	risk-neut. VI	891	0.28	61.92	96.43	no
	risk-sens. DVI	90,000	36	85.3	92.2	yes
	robust risk-sens. VI	891	5.51	70.31	91.94	yes
DS Treasure	risk-neut. VI	1,756	5.45	359.3	458.44	no
	risk-sens. DVI	120,000	72	370.1	458.6	yes
	robust risk-sens. VI	1,756	31.10	379.69	447.11	yes
Grid World ($N = 10$)	risk-neut. VI	396	0.01	25.31	41.32	no
	risk-sens. DVI	39,600	2.02	28.0	38.32	yes
	robust risk-sens. VI	396	0.77	27.08	39.2	yes
Grid World ($N = 150$)	risk-neut. VI	22,500	36.44	402.8	479.4	yes
	risk-sens. DVI	2.3×10^6	15,051	402.9	478.4	yes
	robust risk-sens. VI	22,500	138.17	402.9	478.4	yes
Future Investor ($R_{\max}$)	risk-neut. VI	12,096	2.11	5.70	3.87	no
	risk-sens. DVI	-	-	-	-	-
	robust risk-sens. VI	12,096	8.99	5.56	4.10	yes

overhead introduced by the augmented state space in distributional methods [13]. For comparability, we fix $\alpha = 0.7$ for all benchmarks and instantiate the chance constraint $\Pr(R_T^\pi \leq \tau) \geq \alpha$ using the threshold $\tau = R_{\min} + \alpha (R_{\max} - R_{\min})$.

Main Results. Table 2 summarizes the results of this comparison. Following the PRISM properties in each benchmark, we minimize the expectation $\mathbb{E}$ and CVaR_α in Betting Game, DS Treasure, and GridWorld, and maximize them in Future Investor. In all these environments, risk-neut. VI attains the best expected reward, while exhibiting the worst tail behavior (CVaR_α and constraint violation). The risk-sens. DVI method moves along the risk–return trade-off: it reduces CVaR_α compared to risk-neut. VI at the price of more conservative expected reward, moreover, it requires an augmented MDP that is one to two orders of magnitude larger (e.g. 90,000 vs. 891 states for the Betting Game, and 120,000 vs. 1,756 for DS Treasure), resulting in substantially higher computational times.

Our robust risk-sensitive VI method consistently improves tail-risk over risk-neut. VI while operating on the original MDP state space. On all benchmarks where risk-sens. DVI is applicable, robust risk-sensitive VI achieves the best or comparable CVaR_α (e.g. 91.94 vs. 92.20 on Betting Game and 447.11 vs. 458.60 on DS Treasure), often with a smaller or comparable expected cost than risk-sens. DVI, and with markedly lower runtime. In the Gridworld ($N = 10$) benchmark, robust risk-sensitive VI reduces CVaR_α from 41.32 (risk-neut. VI) to 39.2 while keeping the expected reward close to that of risk-neut. VI and remaining faster than risk-sens. DVI. On the Future Investor model, the continuous-valued reward setting is infeasible in risk-sens. DVI, whereas robust risk-sensitive VI still runs on the original MDP and improves lower-tail CVaR_α over the risk-neutral baseline from 3.87 to 4.10. These results support our claim that Cantelli-based policy synthesis offers a practical and efficient alternative for risk-aware control, providing strong tail-risk mitigation without the state-space blowup of distributional CVaR optimization.

Computational Complexity. We compare the per-iteration costs of the baselines and our method. Let N be the number of states and A the maximum number of actions per state. Risk-neut. VI performs Bellman updates on the original MDP, with worst-case cost $O(N^2A)$ when each action can transition to all states. Risk-sens. DVI [13] augments each state with a discretized slack variable b for the risk objective. If b has B grid points, the state space grows from

N to NB . Since DVI also stores and updates a reward distribution with m bins or atoms at each augmented state, and assuming that the distributional projection is linear in m , a simplified worst-case cost is $O(mBN^2A)$. In practice, B and m are often in the tens to hundreds [13]. Our method does not introduce state augmentation or propagate full reward distributions: for a fixed policy, it computes n analytical moments on the induced DTMC and performs moment-based updates on the original MDP state space, with cost $O(n \text{Solve}(N) + n^2N^2A)$, where $\text{Solve}(N)$ denotes the cost of solving one linear system on the N -state induced DTMC. In our main experiments, we use $n = 2$, and moment-based approximation methods generally use only a small finite number of moments because reconstruction from high-order moments becomes increasingly ill-conditioned [27]. Therefore, the observed speedup in Table 2 mainly comes from avoiding the augmented state space and full discretized reward distributions, with n much smaller than B and m , rather than from a different asymptotic dependence on N and A .

Ablation Study. The central motivation for our Cantelli-based surrogate objective is computational: enforcing chance constraints or optimizing risk measures using full reward distributions requires maintaining and updating a distribution at each state, which is expensive in both memory and runtime (e.g., histogram-based approximations, kernel density estimators, or Erlang mixtures). In contrast, the Cantelli bound leverages analytically computed moments to provide a fast and conservative estimate of tail risk, enabling efficient robust policy optimization. At the same time, Erlang mixtures can yield a more accurate distributional representation than histograms, especially for capturing tail behavior and multi-modality in both discrete and continuous reward settings; see the detailed discussion and experiments in [26].

We conduct two ablations to disentangle (i) conservatism and (ii) distributional approximation accuracy from computational cost in our robust policy optimization method based on the Cantelli bound. In the first ablation, we evaluate whether the Cantelli bound is overly conservative and how this conservatism affects synthesized policies. In the second ablation, we evaluate the accuracy–efficiency trade-off of full-distribution methods by replacing the histogram representation in distributional value iteration with an Erlang-mixture approximation, to test whether a better distributional approximation improves performance and at what computational cost. Overall, these ablations highlight the intended trade-off: Cantelli-based synthesis is efficient and robust, while full-distribution approaches (histogram or Erlang) are substantially slower but may sometimes result in better policies when a more accurate approximation matters.

For the first ablation, we compare chance-constraint *predictions* from: (i) robust risk-sensitive value iteration using the Cantelli bound (first two moments), (ii) the same method but replacing the Cantelli-based chance-constraint check with an Erlang-mixture approximation (three mixture components, matching the first two moments; see [26] for the relationship between the number of mixture components, the number of matched moments, and approximation accuracy), and (iii) empirical Monte Carlo estimation. Fig. 6 (left) reports results on GridWorld with $N=10$ over a grid of (α, τ) values. Each cell is categorized by whether the synthesized policy *satisfies* the chance constraint under the three estimation methods. The red region indicates that all three estimators agree that the chance constraint is satisfied. The orange region indicates cases where Monte Carlo and the Erlang approximation both indicate satisfaction, but the Cantelli bound predicts a violation, suggesting that Cantelli is overly conservative. The light-peach region indicates cases where only Monte Carlo suggests satisfaction, i.e., the Erlang approximation is also conservative there. The blue region corresponds to settings where all three estimators agree that the chance constraint is violated. Overall, Cantelli matches the Erlang and Monte Carlo outcomes in the red and blue regions, and differs primarily in the orange region where the Erlang approximation supports more aggressive policies while still satisfying the constraint under Monte Carlo. Fig. 6 (right) reports average runtime across methods (notice the y-axis in log scale): Cantelli-based

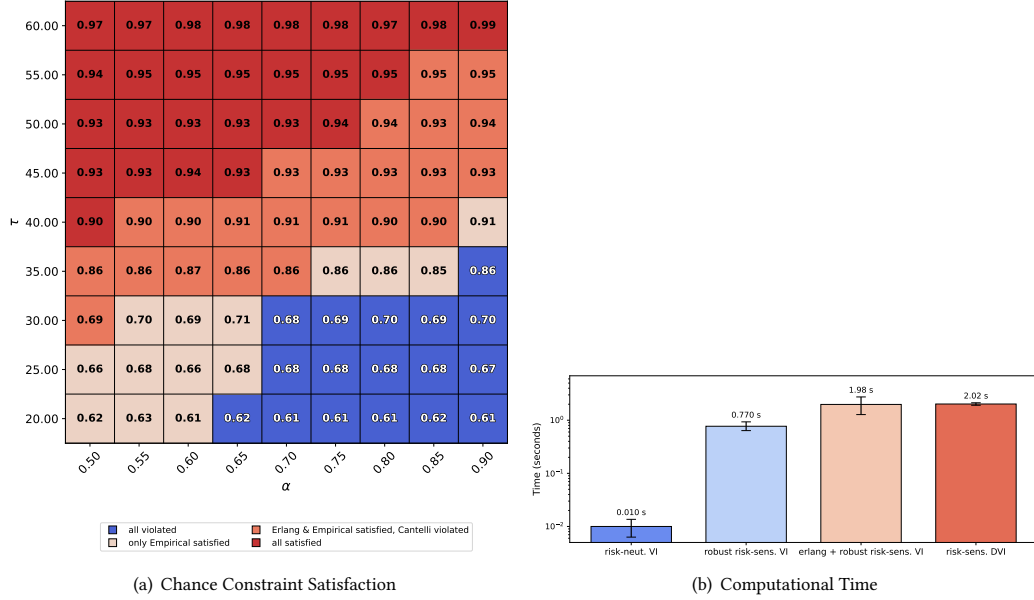


Fig. 6. Comparison of chance-constraint satisfaction and runtime for GridWorld ($N = 10$) over different (α, τ) settings. The left panel shows whether the constraint is satisfied according to Cantelli, Erlang-mixture, and empirical evaluation, while the right panel reports the computational time. Cantelli-based synthesis provides conservative certificates and is substantially faster than full-distribution variants.

evaluation is faster than Erlang-mixture evaluation, while the risk-sensitive DVI baseline is substantially slower due to the augmented state space and the histogram representation. These results support our design choice: Cantelli-based chance-constrained synthesis provides a favorable accuracy–efficiency trade-off for risk-aware planning.

For the second ablation, we compare our method against full-distribution optimization approaches that maintain a cumulative return distribution at each state (histogram-based DVI and Erlang-mixture DVI) and optimize a risk measure such as CVaR. Specifically, we compare robust risk-sensitive value iteration based on the Cantelli bound with a risk-sensitive DVI variant that replaces the histogram approximation with an Erlang-mixture approximation, and the original risk-sensitive DVI method based on a histogram approximation, on small benchmarks (Betting Game and GridWorld ($N=10$)). Table 3 shows that the Erlang-based DVI variant can provide a more accurate distributional representation than histograms and can improve CVaR or expected return in some settings. However, the per-state Erlang-mixture optimization remains costly. In contrast, the Cantelli bound provides a fast, distribution-free method with conservative guarantees. Although it can be conservative, this conservatism aligns with the goals of robust probabilistic model checking and safe policy synthesis. We leave further acceleration of distributional Erlang-mixture methods (e.g., via fast polynomial approximations) to future work.

Summary. Across the evaluated benchmarks, we observe several consistent patterns: Risk-neut. VI achieves the best expected reward in most benchmarks, but fails the chance constraint in most cases. Risk-sens. DVI satisfies the constraint on the discrete reward benchmarks, but it requires a substantially larger augmented state space and cannot be applied to the benchmark with continuous reward domain. In contrast, our method satisfies the chance constraint in

Table 3. Ablation comparison of robust risk-sens. VI (Cantelli), risk-sens. DVI (Histogram), and risk-sens. DVI (Erlang) in terms of policy quality, risk metrics, and runtime.

Model	Method	States	Time (s)	$\mathbb{E}[R_T^\pi]$	$\text{CVaR}_\alpha[R_T^\pi]$	Satisfaction?
Betting Game	robust risk-sens. VI	891	5.51	70.31	91.94	yes
	risk-sens. DVI	90,000	36	85.3	92.2	yes
	Erlang risk-sens. DVI	891	96.02	70.27	92.2	yes
GridWorld ($N = 10$)	robust risk-sens. VI	396	0.77	27.08	39.2	yes
	risk-sens. DVI	39,600	2.02	28.0	38.32	yes
	Erlang risk-sens. DVI	396	18.65	27.59	36.44	yes

all benchmarks while using the original MDP state space. The ablation further confirms that the Cantelli certificate is conservative: for some (α, τ) settings, Monte Carlo simulation or the Erlang-mixture CDF estimates the policy as satisfying the constraint, whereas the Cantelli bound does not certify it. The second ablation shows that Erlang mixtures can sharpen some distributional estimates compared with histograms, but fitting a mixture at each state substantially increases runtime. These results clarify the role of the moment-based Cantelli surrogate: it provides a conservative certificate using only a small number of analytical moments, avoids propagating full reward distributions during policy updates, and remains applicable without extra discretization in reward domains.

6 Related Work

Moment Matching Approximation. Moment matching is a classical approach for approximating an unknown distribution from a finite number of moments [27, 36]. A key design choice is the family of approximants used to represent the density, as the achievable accuracy depends on how well that family captures the true shape. Prior work has studied flexible nonparametric estimators (e.g., kernel-based reconstructions) as well as structured parametric families such as Erlang mixtures [19, 24, 28]. In the context of Markov decision processes, Dolgov and Durfee [12] propose a moment-based construction for approximate probabilistic constraints and risk-sensitive criteria by augmenting an LP formulation to compute higher-order moments of total rewards. In our work, we employ Erlang mixtures not only for their approximation power on $[0, +\infty)$, but also because of their close connection to Markovian reward accumulation, enabling smooth distributional reconstructions from analytically computed moments.

Robust Model Checking. Robustness is one of the most important concerns in model checking methods. Some simulation-based approaches provide probabilistic bounds with confidence guarantees of model correctness [25, 37]. There are also several methods that provide evaluation beyond expected value with distributional information, including quantile-based methods [2] and histogram approximation [13], but these may suffer from discretization errors and information loss due to coarse binning. In contrast, our approach provides smoother approximation while avoiding discretization errors and offering guaranteed accuracy in both discrete-valued and continuous-valued domains.

Distributional Reinforcement Learning. Recent work in distributional reinforcement learning has focused on estimating the full cumulative reward distribution instead of just its expectation. Categorical histogram approximation has been proposed in [5] to compute cumulative reward distributions using discrete bins, optimizing the Wasserstein distance between estimated and target distributions. Extensions of this work, such as quantile regression (QR-DQN) [11] and implicit quantile networks (IQN) [10], improve on this by learning the quantile-based approximation, offering greater flexibility and risk-sensitive policies. Our method can be further extended to the RL domain, providing more

precise estimation and greater flexibility in continuous-valued reward environments than histogram methods, while offering robust verification to guide distributional policy iteration.

7 Conclusion

Building upon the framework in [26], we present a distributional framework for probabilistic model checking and control under chance constraints in discrete Markov models with both discrete-valued and continuous-valued rewards. We showed how to compute analytical moments of the cumulative reward to absorption via first-step analysis and MGF-based recurrences, and how to use these moments in two complementary verification modes that enable users to trade off speed and conservatism: (i) a fast, conservative sufficient check based on Cantelli-type inequalities that requires no distributional fitting, and (ii) a more accurate evaluation obtained by reconstructing a smooth approximation of the cumulative reward distribution via Erlang-mixture moment matching with entropy regularization. This dual approach avoids the discretization artifacts of histogram-based methods while supporting continuous-valued rewards without binning and retaining a principled, moment-driven mechanism for controlling approximation error.

We then moved from verification to control and introduced a robust chance-constrained policy synthesis method for MDPs. Instead of propagating full cumulative reward distributions during dynamic-programming updates, we embedded a Cantelli-certified satisfaction bound into a smooth penalty and derived a value-iteration procedure that trades off expected performance against higher-order moment penalties. The resulting algorithm is fully analytical, uses only a finite number of moments computed from linear systems on induced DTMCs, and provides conservative guarantees for the original chance constraint whenever the Cantelli bound certifies feasibility. We established soundness properties for Cantelli-based feasibility certification and showed that the Erlang mixture can approximate any positive distribution arbitrarily well with sufficient moments.

Our empirical evaluation on PRISM benchmarks demonstrates that the Erlang mixture approximation achieves higher accuracy than histogram methods, while the Cantelli-based value iteration achieves comparable risk-sensitive performance to distributional baselines without the state-space augmentation overhead (up to 100× reduction in state space).

While Cantelli bounds are conservative by design, which can be beneficial in risk-sensitive settings, they may also be loose and thus overly restrictive. An important direction for future work is to develop less conservative policy-improvement methods that propagate compact distributional representations (e.g., Erlang mixtures or other parametric families) while retaining verification-oriented guarantees. We also plan to extend this framework to model-free reinforcement learning settings where the transition dynamics are unknown, developing sample-based methods for estimating analytical moments and chance-constrained policy gradient algorithms with distributional guarantees.

References

- [1] Daniel Ayala, Leonardo Jofré, Luis Gutiérrez, and Ramsés H. Mena. 2022. On a Dirichlet Process Mixture Representation of Phase-Type Distributions. *Bayesian Analysis* 17, 3 (2022), 765 – 790. doi:10.1214/21-BA1272
- [2] Christel Baier, Clemens Dubslaff, Sascha Klüppelholz, Marcus Daum, Joachim Klein, Steffen Märcker, and Sascha Wunderlich. 2014. Probabilistic model checking and non-standard multi-objective reasoning. In *International Conference on Fundamental Approaches to Software Engineering*. Springer, 1–16.
- [3] Christel Baier, Marcus Größer, and Frank Ciesinski. 2009. *Model Checking Linear-Time Properties of Probabilistic Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 519–570. doi:10.1007/978-3-642-01492-5_13
- [4] C. Baier, J.P. Katoen, and K.G. Larsen. 2008. *Principles of Model Checking*. MIT Press.
- [5] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International conference on machine learning*. PMLR, 449–458.

- [6] Radu Calinescu, Carlo Ghezzi, Marta Kwiatkowska, and Raffaella Mirandola. 2012. Self-adaptive software needs quantitative verification at runtime. *Commun. ACM* 55, 9 (Sept. 2012), 69–77. doi:10.1145/2330667.2330686
- [7] S Chakravarthy and Attahiru S Alfa. 1996. *Matrix-analytic methods in stochastic models*. CRC Press.
- [8] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research* 18, 1 (2017), 6070–6120.
- [9] Erhan Cinlar. 2013. *Introduction to stochastic processes*. Courier Corporation.
- [10] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*. PMLR, 1096–1105.
- [11] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. 2018. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.
- [12] Dmitri A. Dolgov and Edmund H. Durfee. 2004. Approximate Probabilistic Constraints and Risk-Sensitive Optimization Criteria in Markov Decision Processes. In *International Symposium on Artificial Intelligence and Mathematics, AI&M 2004, Fort Lauderdale, Florida, USA, January 4-6, 2004*.
- [13] Ingy Elsayed-Aly, David Parker, and Lu Feng. 2024. Distributional probabilistic model checking. In *NASA Formal Methods Symposium*. Springer, 57–75.
- [14] Agner Krarup Erlang. 1909. The theory of probabilities and telephone conversations. *Nyt. Tidsskr. Mat. Ser. B* 20 (1909), 33–39.
- [15] Samira S. Farahani, Rupak Majumdar, Vinayak S. Prabhu, and Sadegh Esmaeil Zadeh Soudjani. 2017. Shrinking Horizon Model Predictive Control with chance-constrained signal temporal logic specifications. In *2017 American Control Conference (ACC)*. 1740–1746. doi:10.23919/ACC.2017.7963204
- [16] Antonio Filieri, Giordano Tamburrelli, and Carlo Ghezzi. 2016. Supporting Self-Adaptation via Quantitative Verification and Sensitivity Analysis at Run Time. *IEEE Transactions on Software Engineering* 42, 1 (2016), 75–99. doi:10.1109/TSE.2015.2421318
- [17] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu. 2011. Quantitative Multi-objective Verification for Probabilistic Systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, Parosh Aziz Abdulla and K. Rustan M. Leino (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 112–127.
- [18] Varun Ganapathi, David Vickrey, John Duchi, and Daphne Koller. 2012. Constrained approximate maximum entropy learning of markov random fields. *arXiv preprint arXiv:1206.3257* (2012).
- [19] PN Gavriliadis and GA Athanassoulis. 2012. The truncated Stieltjes moment problem solved by using kernel density functions. *J. Comput. Appl. Math.* 236, 17 (2012), 4193–4213.
- [20] B. K. Ghosh. 2002. Probability Inequalities Related to Markov’s Theorem. *The American Statistician* 56, 3 (2002), 186–190. <http://www.jstor.org/stable/3087296>
- [21] Dennis Gross, Nils Jansen, Sebastian Junges, and Guillermo A Pérez. 2022. COOL-MC: a comprehensive tool for reinforcement learning and model checking. In *International Symposium on Dependable Software Engineering: Theories, Tools, and Applications*. Springer, 41–49.
- [22] Gurobi Optimization, LLC. 2024. Gurobi Optimizer Reference Manual. <https://www.gurobi.com>
- [23] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.
- [24] Qi-Ming He, Baoliang Liu, and Haoran Wu. 2022. Continuous approximations of discrete phase-type distributions and their applications to reliability models. *Performance Evaluation* 154 (2022), 102284.
- [25] Thomas Héruault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. 2004. Approximate probabilistic model checking. In *International Workshop on Verification, Model Checking, and Abstract Interpretation*. Springer, 73–84.
- [26] Xiaotong Ji, Hanchun Wang, Antonio Filieri, and Ilenia Epifani. 2025. Robust Probabilistic Model Checking With Continuous Reward Domains. In *2025 IEEE/ACM 20th Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. 13–24. doi:10.1109/SEAMS66627.2025.00010
- [27] V John, I Angelov, AA Öncül, and D Thévenin. 2007. Techniques for the reconstruction of a distribution from a finite number of its moments. *Chemical Engineering Science* 62, 11 (2007), 2890–2904.
- [28] Mary A Johnson and Michael R Taaffe. 1989. Matching moments to phase distributions: Mixtures of Erlang distributions of common order. *Stochastic Models* 5, 4 (1989), 711–743.
- [29] Joost-Pieter Katoen. 2016. The Probabilistic Model Checking Landscape. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (New York, NY, USA) (LICS ’16)*. Association for Computing Machinery, New York, NY, USA, 31–45. doi:10.1145/2933575.2934574
- [30] Hadas Kress-Gazit, Georgios E. Fainekos, and George J. Pappas. 2009. Temporal-Logic-Based Reactive Mission and Motion Planning. *IEEE Transactions on Robotics* 25, 6 (2009), 1370–1381. doi:10.1109/TRO.2009.2030225
- [31] Marta Kwiatkowska, Gethin Norman, and David Parker. 2002. PRISM: Probabilistic symbolic model checker. In *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*. Springer, 200–204.
- [32] Marta Kwiatkowska, Gethin Norman, and David Parker. 2022. Probabilistic Model Checking and Autonomy. *Annual Review of Control, Robotics, and Autonomous Systems* 5 (2022), 385–410. doi:10.1146/annurev-control-042820-010947
- [33] Yongshuai Liu, Jiaxin Ding, and Xin Liu. 2020. Ipo: Interior-point policy optimization under constraints. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4940–4947.
- [34] Annabelle McIver and Carroll Morgan. 2007. Results on the quantitative μ -calculus $q\mu$. *ACM Transactions on Computational Logic (TOCL)* 8, 1 (2007), 3–es.
- [35] Lawrence R Mead and Nikos Papanicolaou. 1984. Maximum entropy in the problem of moments. *J. Math. Phys.* 25, 8 (1984), 2404–2417.

- [36] Robert M Mnatsakanov and Artak S Hakobyan. 2009. Recovery of distributions via moments. *Lecture Notes-Monograph Series* (2009), 252–265.
- [37] Angela Pappagallo, Annalisa Massini, and Enrico Tronci. 2020. Monte carlo based statistical model checking of cyber-physical systems: A review. *Information* 11, 12 (2020), 588.
- [38] Martin L Puterman. 1990. Markov decision processes. *Handbooks in operations research and management science* 2 (1990), 331–434.
- [39] John G Saw, Mark CK Yang, and Tse Chin Mo. 1984. Chebyshev inequality with estimated mean and variance. *The American Statistician* 38, 2 (1984), 130–132.
- [40] H.C. Tijms. 1994. *Stochastic Models: An Algorithmic Approach*. Wiley.
- [41] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods* 17, 3 (2020), 261–272.
- [42] Danny Weyns, M. Usman Iftikhar, Didac Gil de la Iglesia, and Tanvir Ahmad. 2012. A survey of formal methods in self-adaptive systems. In *Proceedings of the Fifth International C* Conference on Computer Science and Software Engineering* (Montreal, Quebec, Canada) (C3S2E '12). Association for Computing Machinery, New York, NY, USA, 67–79. doi:10.1145/2347583.2347592
- [43] Baohe Zhang, Yuan Zhang, Lilli Frison, Thomas Brox, and Joschka Bödecker. 2024. Constrained reinforcement learning with smoothed log barrier function. *arXiv preprint arXiv:2403.14508* (2024).