

# Online Robust Planning Under Model Uncertainty: A Sample-Based Approach

Tamir Shazman<sup>1</sup>, Idan Lev-Yehudi<sup>2</sup>, Ron Benchetrit<sup>3</sup>, Vadim Indelman<sup>4,1</sup>

<sup>1</sup>Faculty of Data and Decision Sciences, Technion - Israel Institute of Technology, Haifa 32000, Israel

<sup>2</sup>Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa 32000, Israel

<sup>3</sup>Faculty of Computer Science, Technion - Israel Institute of Technology, Haifa 32000, Israel

<sup>4</sup>Stephen B. Klein Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel  
{tmyr, idanlev, ronbenc}@campus.technion.ac.il, vadim.indelman@technion.ac.il

## Abstract

Online planning in Markov Decision Processes (MDPs) enables agents to make sequential decisions by simulating future trajectories from the current state, making it well-suited for large-scale or dynamic environments. Sample-based methods such as Sparse Sampling and Monte Carlo Tree Search (MCTS) are widely adopted for their ability to approximate optimal actions using a generative model. However, in practical settings, the generative model is often learned from limited data, introducing approximation errors that can degrade performance or lead to unsafe behaviors. To address these challenges, Robust MDPs (RMDPs) offer a principled framework for planning under model uncertainty, yet existing approaches are typically computationally intensive and not suited for real-time use. In this work, we introduce *Robust Sparse Sampling* (RSS), the first online planning algorithm for RMDPs with finite-sample theoretical performance guarantees. Unlike Sparse Sampling, which estimates the nominal value function, RSS computes a robust value function by leveraging the efficiency and theoretical properties of Sample Average Approximation (SAA), enabling tractable robust policy computation in online settings. RSS is applicable to infinite or continuous state spaces, and its sample and computational complexities are independent of the state space size. We provide theoretical performance guarantees and empirically show that RSS outperforms standard Sparse Sampling in environments with uncertain dynamics.

## 1 Introduction

Markov Decision Processes (MDPs) provide a mathematical framework for modeling sequential decision-making under uncertainty, where an agent interacts with a stochastic environment to maximize cumulative expected rewards. Exact solutions to MDPs are often computationally infeasible, as they've been shown to be P-complete (Papadimitriou and Tsitsiklis 1987), and practical methods often resort to approximate solutions (Littman, Dean, and Kaelbling 1995).

*Online methods* try to circumvent the complexity of computing a policy by planning online only for the current state (Koenig 2001; Ross et al. 2008), making it particularly suitable for large or dynamic environments where computing optimal actions for all states in advance is infeasible. Among the most popular online planning methods are sample-based

algorithms like Sparse Sampling (Kearns, Mansour, and Ng 2002) and Monte Carlo Tree Search (MCTS) (Coulom 2006), which approximate near-optimal decisions using limited computation at runtime.

Sparse Sampling has historical, theoretical and algorithmic significance: being the first algorithm to provide finite-time guarantees for online planning, and computational complexity scaling only by the planning horizon and approximation budget, rather than state-space size. It has inspired many practical tree-based online planning algorithms like MCTS (Kocsis and Szepesvári 2006; Browne et al. 2012), popular algorithms for partially observable settings like POMCP (Silver and Veness 2010) and DESPOT (Soman et al. 2013), and has recently been extended to theoretical guarantees of particle-belief approximations in POMDP planning (Lim, Tomlin, and Sunberg 2019; Lim et al. 2023).

A major limitation of Sparse Sampling, MCTS and existing online planning methods is that they typically assume access to a *generative model*, i.e. a simulator that provides samples of next states and rewards. In practice, however, such models are often estimated from data and may introduce approximation errors. If these discrepancies are ignored, they can lead to poor or even unsafe decision-making (Mannor et al. 2007).

*Robust Markov Decision Processes* (RMDPs) offer a theoretical framework to address this issue by explicitly modeling uncertainty in the transition dynamics (Iyengar 2005; Nilim and Ghaoui 2005). RMDPs define sets of plausible models and optimize for the worst-case within these sets, thereby guaranteeing performance robustness. However, solving RMDPs typically involves a computationally demanding min-max optimization over both policies and model perturbations, making them difficult to apply in online or large-scale settings.

To enhance scalability, various approaches have been proposed within the robust reinforcement learning (RL) community. Some methods utilize robust variants of function approximation (Tamar, Mannor, and Xu 2014), while others introduce sample-based algorithms that learn robust policies by interacting with an environment affected by model uncertainty (Wang and Zou 2021; Panaganti et al. 2022; Panaganti and Kalathil 2022; Dong et al. 2022). Although these techniques show promise, they are generally not tailored for online planning, as they aim to learn global policies across

the entire state space rather than allocating computational effort to the specific decision at hand. In contrast to robust reinforcement learning, few works address the challenges of robust online planning, being limited to parametric uncertainty structures or deterministic MDPs (Sharma et al. 2019; Kohankhaki et al. 2024). This highlights the need for robust online planning methods that are both general-purpose and theoretically grounded.

To address the challenge of online planning under model uncertainty, we adopt the RMDP framework to formalize robustness and introduce a new sample-based planning algorithm: *Robust Sparse Sampling (RSS)*. RSS extends Sparse Sampling to explicitly handle model uncertainty. To enable efficient robust decision-making in online settings, it leverages the theoretical properties and computational efficiency of the Sample Average Approximation framework (Shapiro, Dentcheva, and Ruszczyński 2021). We also establish theoretical performance guarantees.

## 1.1 Contributions

This work addresses *online planning under model uncertainty* by formulating a sample-based robust planner and establishing its theoretical and empirical merits. Our main contributions are:

- **Algorithmic novelty.** We propose *Robust Sparse Sampling (RSS)*, which, to the best of our knowledge, is the first sample-based online planning algorithm that directly addresses robust MDPs while providing *finite-sample performance guarantees*. Notably, the complexity of RSS is independent of the size of the state space, making it suitable for environments with infinite or continuous state spaces.
- **Theoretical guarantees.** By leveraging the convergence theory of Sample Average Approximation technique, we derive an error bound between the value of the policy induced by RSS and the true optimal robust value function. This bound can be made arbitrarily small by appropriately setting the planning parameters.
- **Empirical validation.** Experiments on two benchmark domains demonstrate that RSS substantially reduces catastrophic failures and achieves higher empirical returns than classical Sparse Sampling when the transition dynamics are misspecified.

## 1.2 Related Work

Robustness in *online MDP planning* has been explored in only a couple of recent studies. Sharma et al. (2019) introduced Robust Adaptive Monte Carlo Planning (RAMCP), which embeds Monte Carlo Tree Search in the Bayes-adaptive framework. That framework requires a prior over the transition model, and misspecifying this prior can harm performance; RAMCP seeks to hedge against such misspecification by computing a policy that is robust to prior errors. However, RAMCP still assumes that transition uncertainty follows a specific parametric form, which limits its applicability in settings where the dynamics are non-parametric or deviate from that model.

Kohankhaki et al. (2024) introduced Uncertainty Adapted MCTS (UA-MCTS), an MCTS variant for deterministic MDPs that adjusts node selection based on estimated transition uncertainty. Although UA-MCTS demonstrates strong empirical performance in deterministic settings, it lacks formal robustness guarantees and does not extend naturally to stochastic environments, limiting its general applicability.

## 2 Preliminaries

### 2.1 Robust Markov Decision Process (RMDP)

We consider a Markov Decision Process (MDP) defined as  $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \gamma)$ . The (possibly infinite) state space is  $\mathcal{S}$ . We assume the action space  $\mathcal{A}$  is finite. We assume a bounded reward function  $r : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ , yet our analysis can be trivially extended to any time-dependent bounded reward. For the transition kernel  $P$ ,  $P_{s,a}(s')$  denotes the probability of transitioning to state  $s'$  given state  $s$  and action  $a$ .  $\gamma \in [0, 1)$  is the discount factor.

During planning, the agent has access only to an approximate generative model of the transition kernel  $P^o$ , which is an estimate of the true transition model  $P$ . We assume that there exists a state-action dependent bound between the true and approximate transition kernels of the form:

$$\forall (s, a) \in \mathcal{S} \times \mathcal{A}, \quad D(P_{s,a}, P_{s,a}^o) \leq \rho, \quad (1)$$

where  $\rho \in [0, 1]$ , and  $D(\cdot, \cdot)$  is a distance metric between two probability distributions.  $\rho$  quantifies the maximum allowable deviation between the true transition model  $P_{s,a}$  and the estimated model  $P_{s,a}^o$ . Higher values of  $\rho$  indicate greater uncertainty, with  $\rho = 0$  meaning perfect model accuracy. This uncertainty bound can be estimated from statistical confidence intervals (Berend and Kontorovich 2012) or explicitly defined based on domain-specific knowledge. In this work, we focus on  $D(\cdot, \cdot)$  being the Total Variation (TV) distance, i.e.,  $D(P_{s,a}, P_{s,a}^o) = \frac{1}{2} \|P_{s,a} - P_{s,a}^o\|_1$ .

Planning directly with the empirical model  $P^o$  can lead to suboptimal or unsafe policies (Mannor et al. 2007). To guard against model error, we adopt the Robust MDP framework (RMDP). Instead of a single transition kernel, RMDP consider an uncertainty set of transition kernels. Adopting the common rectangularity assumption (Iyengar 2005; Nilim and Ghaoui 2005), according to which the uncertainty in the transition kernels is independent for each state-action pair, we define the uncertainty set as:

$$\mathcal{P} = \bigotimes_{(s,a) \in \mathcal{S} \times \mathcal{A}} \mathcal{P}_{s,a}, \quad (2)$$

$$\mathcal{P}_{s,a} = \{P_{s,a} \in \Delta(\mathcal{S}) : D(P_{s,a}, P_{s,a}^o) \leq \rho\}, \quad (3)$$

where  $\Delta(\mathcal{S})$  is the set of probability distributions over  $\mathcal{S}$ . For a fixed model  $P'$  and policy  $\pi$ , the (non-robust) value function is

$$V^{\pi, P'}(s) = \mathbb{E}_{P', \pi} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_t = \pi(s_t) \right]. \quad (4)$$

The *robust* value function takes the worst case model in  $\mathcal{P}$ :

$$V^{\pi}(s) = \min_{P' \in \mathcal{P}} V^{\pi, P'}(s), \quad (5)$$

and our planning objective is to find a policy that maximizes this worst-case return, i.e.  $V^*(s) = \max_{\pi} V^{\pi}(s)$  where  $\pi^* \in \arg \max_{\pi} V^{\pi}(s)$ . We denote the corresponding robust action-value function by  $Q^*$ . A deterministic robust optimal policy is known to exist (Iyengar 2005), and its value function satisfies the robust Bellman equation:

$$V^*(s) = \max_{a \in \mathcal{A}} \left[ r(s, a) + \gamma \min_{P_{s,a} \in \mathcal{P}_{s,a}} \mathbb{E}_{s' \sim P_{s,a}} [V^*(s')] \right]. \quad (6)$$

This formulation guarantees that the robust value serves as a lower-bound for the true value, providing explicit protection against transition-model misspecification.

## 2.2 Robust Action-Value Function Dual Form

Computing a robust policy under an imperfect transition model using online, sample-based methods is challenging, due to the infinite number of possible transition distributions within the uncertainty set (2). This makes a direct optimization of the robust Bellman's equation (6) intractable. In their recent work, Panaganti et al. (2022) show that the dual form of the optimal robust action-value function admits the following closed-form expression:

$$\begin{aligned} Q^*(s, a) = & r(s, a) - \\ & \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \left( \mathbb{E}_{s' \sim P_{s,a}^o} [(\eta - V^*(s'))_+] - \eta + \right. \\ & \left. \rho \left( \eta - \inf_{s''} V^*(s'') \right) \right), \end{aligned} \quad (7)$$

where  $[x]_+ \triangleq \max\{0, x\}$ . The dual variable  $\eta$  serves as a Lagrange multiplier, balancing the trade-off between the expected value and the worst-case value.

However, estimating the infimum of the robust value function  $V^*(s'')$  over all states  $s''$  is generally intractable, and particularly problematic in large or continuous state spaces, where computing the infimum term is computationally prohibitive. To simplify the dual formulation, we assume the existence of a fail-state, stated in the following assumption.

**Assumption (Fail-State).** There exists a state  $s_f \in \mathcal{S}$  such that  $r(s_f, a) = 0$  and  $P'_{s_f, a}(s_f) = 1$  for all actions  $a \in \mathcal{A}$  and all transition probabilities  $P' \in \mathcal{P}$ . This implies  $V^*(s_f) = 0$ , and hence  $\inf_{s''} V^*(s'') = 0$ . Under this assumption, equation (7) simplifies to:

$$\begin{aligned} Q^*(s, a) = & r(s, a) - \\ & \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \left( \mathbb{E}_{s' \sim P_{s,a}^o} [(\eta - V^*(s'))_+] - \eta(1 - \rho) \right). \end{aligned} \quad (8)$$

To simplify the notation in the remainder of the paper, we define for each state-action pair  $(s, a)$  the function:

$$F_{s,a}^{\rho}(\eta) \triangleq \mathbb{E}_{s' \sim P_{s,a}^o} [(\eta - V^*(s'))_+] - \eta(1 - \rho). \quad (9)$$

Hence, we can rewrite the dual action-value in equation (8):

$$Q^*(s, a) = r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} F_{s,a}^{\rho}(\eta). \quad (10)$$

## 2.3 Sparse Sampling (SS)

Sparse Sampling (SS) (Kearns, Mansour, and Ng 2002) is a model-based online planning algorithm assuming a known transition kernel  $P$ , that approximates the optimal action-value function  $Q^{*,P}$  with high probability by constructing a stochastic lookahead tree of finite depth  $H$ . It operates by building a recursive search tree. At each node corresponding to a state  $s$ , the algorithm explores each action  $a \in \mathcal{A}$  by drawing  $C$  independent next-state samples from  $P^{s,a}(\cdot)$ . For each sampled successor state  $s'$ , the process recursively continues until the maximum depth  $H$  is reached. The recursive computation of the action-value function at depth  $d$  proceeds as follows:

$$\begin{aligned} \hat{Q}_d^P(s, a) &= r(s, a) + \gamma \cdot \frac{1}{C} \sum_{i=1}^C \hat{V}_{d-1}^P(s'_i), \quad s'_i \sim P_{s,a}, \\ \hat{V}_{d-1}^P(s) &= \max_{a \in \mathcal{A}} \hat{Q}_{d-1}^P(s, a), \\ \hat{V}_0^P(s) &= \tilde{V}_{\theta}^P(s), \quad \forall s \in \mathcal{S} \text{ (leaf terminal value)}. \end{aligned} \quad (11)$$

Here,  $\tilde{V}_{\theta}^P(s)$  denotes a terminal value estimator for  $V^*(s')$ , which may be a learned function or set to zero. In this work, unless stated otherwise, we assume  $\tilde{V}_{\theta}^P(s) = 0$ .

Sparse Sampling provides theoretical guarantees on the gap between the nominal value of the policy it computes and the optimal value function. This difference can be made arbitrarily small by choosing a sufficiently large number of samples  $C$  and planning depth  $H$ .

## 2.4 Sample Average Approximation (SAA)

Stochastic programming (Haneveld and Van der Vlerk 2020) addresses optimization problems under uncertainty, where the objective function involves an expectation over a random variable. The general formulation is given by:

$$\min_{x \in \mathbb{X}} F(x), \quad (12)$$

$$\text{where } F(x) \triangleq \mathbb{E}_{y \sim P_y} [f(y, x)],$$

Here,  $P_y$  denotes a probability distribution over random variables  $y$ ,  $\mathbb{X} \subseteq \mathbb{R}$  is the feasible domain, and  $f(y, x)$  is a real-valued function depending on both the uncertain variable  $y$  and the decision variable  $x$ .

Computing the expectation  $\mathbb{E}_{y \sim P_y} [f(y, x)]$  exactly can be challenging, especially when the distribution  $P_y$  is high-dimensional or analytically intractable. Sample Average Approximation (SAA) (Shapiro, Dentcheva, and Ruszczyński 2021) replaces the expectation with an empirical average based on a finite number of samples drawn from  $P_y$ . Given  $C$  i.i.d. samples  $\{y_i\}_{i=1}^C$  from  $P_y$ , the empirical approximation of the objective becomes:

$$\hat{F}(x) = \frac{1}{C} \sum_{i=1}^C f(y_i, x). \quad (13)$$

SAA is widely used across domains such as operations research, finance, and machine learning to address optimization problems under uncertainty (Burroni, Domke, and Sheldon 2023; Shapiro and Li 2025). Its convergence properties are well-established (Sinha and Chakrabarty 2024); under suitable regularity conditions on the function  $f(y, x)$

and the feasible set  $\mathbb{X}$ , the solution of the empirical problem converges to the true optimum of the original stochastic program as the number of samples increases (Shapiro, Dentcheva, and Ruszczyński 2021).

### 3 Robust Sparse Sampling (RSS)

#### 3.1 Robust Action-Value Estimation via SAA

Although the simplified dual formulation in Equation (10) offers valuable theoretical insight, it remains intractable to solve directly when the robust value function  $V^*(\cdot)$  is unknown. Even if  $V^*(\cdot)$  were available, evaluating  $Q^*(s, a)$  would still require solving a stochastic programming problem over the function  $F_{s,a}^\rho(\eta)$ , which involves an intractable expectation, particularly in large or continuous state spaces.

To address this challenge, we employ the SAA method, replacing  $F_{s,a}^\rho(\eta)$  with an empirical estimate  $\hat{F}_{s,a}^\rho(\eta)$  based on  $C$  samples of the next states  $\{s'_i\}_{i=1}^C \sim P_{s,a}^o(\cdot)$ . This leads to the following approximate formulation:

$$\begin{aligned} \hat{Q}^*(s, a) &= r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \hat{F}_{s,a}^\rho(\eta), \quad \text{where} \\ \hat{F}_{s,a}^\rho(\eta) &= \frac{1}{C} \sum_{i=1}^C (\eta - V^*(s'_i))_+ - \eta(1 - \rho). \end{aligned} \quad (14)$$

The function  $\hat{F}_{s,a}^\rho(\eta)$  is piecewise linear and convex in  $\eta$ , with non-differentiable breakpoints occurring at the sampled values  $\{V^*(s'_i)\}_{i=1}^C$ . This structure makes the optimization problem in Equation (14) efficiently solvable.

#### 3.2 RSS Algorithm

The Robust Sparse Sampling (RSS) algorithm, inspired by the Sparse Sampling algorithm, incorporates robustness against model uncertainty while using a finite number of samples online in a recursive manner. Instead of estimating the nominal action-value function  $Q^{*,P}(s, a)$ , RSS estimates the robust action-value function  $Q^*(s, a)$ . The complete procedure is described in Algorithm 1.

Specifically, RSS recursively estimates the robust action-value function at depth  $d$  by sampling  $C$  successor states from the estimated generative model  $P_{s,a}^o$ . Then, in contrast to the standard Sparse Sampling, RSS computes the robust action-value function by solving the SAA problem at each depth  $d$ :

$$\begin{aligned} \hat{Q}_d(s, a) &= r(s, a) - \gamma \min_{\eta \in [0, \frac{2}{\rho(1-\gamma)}]} \hat{F}_{s,a}^{\rho,d}(\eta), \quad \text{where} \\ \hat{F}_{s,a}^{\rho,d}(\eta) &= \frac{1}{C} \sum_{i=1}^C (\eta - \hat{V}_{d-1}(s'_i))_+ - \eta(1 - \rho), \\ \hat{V}_{d-1}(s) &= \max_{a \in \mathcal{A}} \hat{Q}_{d-1}(s, a), \\ \hat{V}_0(s) &= 0, \quad \forall s \in \mathcal{S} \text{ (leaf terminal value)}. \end{aligned} \quad (15)$$

where each successor  $s'_i$  is drawn i.i.d. from the approximate model  $P_{s,a}^o(\cdot)$ . The routine is invoked recursively from

---

#### Algorithm 1: Robust Sparse Sampling (RSS)

---

**Input:** Current state  $s$ , current depth  $d$

**Parameter:** Sample width  $C$ , planning horizon  $H$  computed based on Theorem 1

**Output:** Estimated optimal action and its value

---

```

1: if  $d = 0$  then
2:   return 0
3: end if
4: for all  $a \in \mathcal{A}$  do
5:    $V_{\text{list}} \leftarrow []$ 
6:   for  $i = 1$  to  $C$  do
7:     Sample  $s'_i \sim P^o(\cdot | s, a)$ 
8:      $(-, \hat{V}_{d-1}(s'_i)) \leftarrow \text{RSS}(s'_i, d-1)$ 
9:     Append  $\hat{V}_{d-1}(s'_i)$  to  $V_{\text{list}}$ 
10:  end for
11:  Update  $\hat{Q}_d(s, a)$  using Equation (15) with  $V_{\text{list}}$ 
12: end for
13: return  $\arg \max_{a \in \mathcal{A}} \hat{Q}_d(s, a), \max_{a \in \mathcal{A}} \hat{Q}_d(s, a)$ 

```

---

the current state  $s$  and remaining depth  $d$ . The recursion terminates at  $d = 0$ , where the leaf value is fixed at 0.

It is important to emphasize that the function  $\hat{F}_{s,a}^\rho(\eta)$  in Equation (14) is a theoretical construct, as it depends on the true robust value function  $V^*(\cdot)$ , which is not accessible to the algorithm in practice. In contrast, the RSS algorithm avoids this dependency by estimating robust values recursively.

At depth  $d$ , RSS replaces  $V^*(s')$  with  $\hat{V}_{d-1}(s')$ , the estimated robust value of the sampled successor state  $s'$  from the previous depth. This substitution yields an empirical estimate  $\hat{F}_{s,a}^{\rho,d}(\eta)$  that approximates  $F_{s,a}^\rho(\eta)$  without requiring knowledge of the exact robust value function.

## 4 Theoretical Analysis of RSS

### 4.1 Performance Guarantees

Our main theoretical result establishes that the value of the policy returned by the RSS algorithm can be made arbitrarily close to the optimal robust value function. Specifically, RSS guarantees the following bound:

**Theorem 1.** *For any  $s \in \mathcal{S}$  and any  $\epsilon > 0$ , the Robust Sparse Sampling algorithm returns a policy  $\pi$  such that:*

$$|V^\pi(s) - V^*(s)| \leq \epsilon,$$

with the following hyperparameters:

$$\begin{aligned} \lambda &= \frac{\epsilon}{3}, \quad \delta = \lambda(1 - \gamma), \quad H = \lceil \log_\gamma(\lambda) \rceil \\ C &= \frac{2}{\lambda^2 \rho^2 (1 - \gamma)^2} \\ &\quad \left( 2H \ln \left( \frac{2|A| \cdot H}{\lambda^2 \rho^2 (1 - \gamma)^2} \right) + \ln \left( \frac{2(8 - 4\rho)}{\delta \lambda (1 - \gamma) \rho} \right) \right). \end{aligned}$$

The proof is provided in the supplementary material. Similar result was originally shown for the nominal (non-robust) setting by Kearns, Mansour, and Ng (2002), where

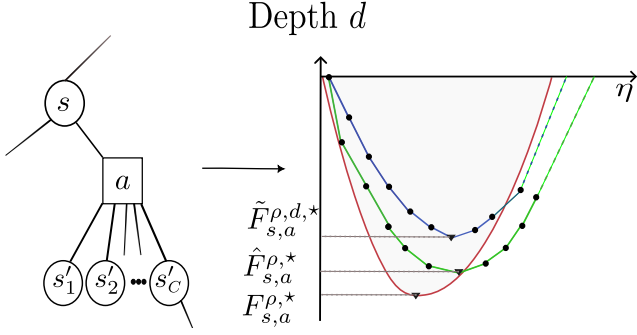


Figure 1: Illustration of the RSS algorithm. At each depth  $d$ , RSS samples  $C$  successor states and recursively estimates their robust values  $\hat{V}_{d-1}(s'_i)$ . The robust action-value estimate  $\hat{Q}_d(s, a)$  is obtained by minimizing the piecewise-linear convex function  $\tilde{F}_{s,a}^{\rho,d}(\eta)$ . The plot displays  $F_{s,a}^{\rho}(\eta)$ ,  $\hat{F}_{s,a}^{\rho}(\eta)$ , and  $\tilde{F}_{s,a}^{\rho,d}(\eta)$  in red, green, and blue, respectively, along with their corresponding minima  $F_{s,a}^{\rho,*}$ ,  $\hat{F}_{s,a}^{\rho,*}$ , and  $\tilde{F}_{s,a}^{\rho,d,*}$ . Triangles indicate the minima, and black dots represent the breakpoints. As the number of samples  $C$  increases, both  $\hat{F}_{s,a}^{\rho}(\eta)$  and  $\tilde{F}_{s,a}^{\rho,d}(\eta)$  converge to  $F_{s,a}^{\rho}(\eta)$ . Since both  $\tilde{F}_{s,a}^{\rho,d}(\eta)$  and  $\hat{F}_{s,a}^{\rho}(\eta)$  are piecewise-linear and convex, their minima can be computed efficiently.

the Sparse Sampling algorithm approximates the optimal value function  $V^{*,P}(s)$ . Here, we extend that result to the robust setting.

This extension is non-trivial, as the robust formulation must account for worst-case transitions within an uncertainty set, which do not arise in the nominal case.

**Proof Sketch.** The proof of Theorem 1 follows a structure similar to the original Sparse Sampling analysis (Kearns, Mansour, and Ng 2002), but extends it using tools from SAA theory to handle robustness.

First, we show that both  $F_{s,a}^{\rho}(\eta)$ , defined in (10), and its empirical counterpart  $\hat{F}_{s,a}^{\rho}(\eta)$ , defined in (14), are Lipschitz continuous with respect to  $\eta$ . This property allows us to apply concentration inequalities from SAA theory (Shapiro, Dentcheva, and Ruszczyński 2021), yielding probabilistic bounds between  $F_{s,a}^{\rho}(\eta)$  and  $\hat{F}_{s,a}^{\rho}(\eta)$ . Consequently, we obtain bounds on the difference between the true robust action-value function  $Q^*(s, a)$  and the SAA-based estimate  $\hat{Q}^*(s, a)$ . Next, we establish a concentration bound between  $F_{s,a}^{\rho}(\eta)$  and  $\tilde{F}_{s,a}^{\rho,d}(\eta)$ , the estimator used by RSS at depth  $d$ , as defined in (15). This step uses the bound from the previous stage, combined with an inductive argument over the estimated robust value function  $\hat{V}_{d-1}$  at depth  $d - 1$ . The relationship and differences between  $F_{s,a}^{\rho}(\eta)$ ,  $\hat{F}_{s,a}^{\rho}(\eta)$ , and  $\tilde{F}_{s,a}^{\rho,d}(\eta)$  are illustrated in Figure 1.

We then apply a union bound over every state-action pair in the search tree, guaranteeing that the concentration inequalities hold simultaneously at all nodes.

Finally, to relate the robust value of the policy  $V^{\pi}(s)$  re-

turned by RSS to the optimal robust value  $V^*(s)$ , we generalize a key lemma from (Kearns, Mansour, and Ng 2002) to the robust setting. This lemma bounds the value gap in terms of the maximum approximation error in the robust action-value function. Combining all steps yields the final bound stated in Theorem 1.

## 4.2 Computational Complexity

The RSS algorithm builds a lookahead tree of depth  $H$ , where each node branches into  $|A| \cdot C$  children—corresponding to  $|A|$  actions and  $C$  sampled next states per action. Therefore, the total number of nodes in the tree is  $(|A| \cdot C)^H$ .

At each node, the algorithm performs two main operations: (1) sampling  $C$  next states using the generative model, and (2) solving the SAA optimization problem defined in Equation 15. The sampling step incurs a cost of  $O(C)$ . For the optimization step, the algorithm minimizes a piecewise-linear convex function  $\tilde{F}_{s,a}^{\rho,d}(\eta)$ , which has  $C$  breakpoints at the values  $\{\hat{V}_{d-1}(s'_i)\}_{i=1}^C$ . The minimum is guaranteed to lie at one of the breakpoints or at the boundary points  $\eta = 0$  and  $\eta = \frac{2}{\rho(1-\gamma)}$ . The optimal solution can thus be found by first sorting the  $C$  breakpoints in  $O(C \log C)$  time, followed by a linear scan to identify the minimizer, resulting in a total per-node complexity of:

$$O(C \log C).$$

Multiplying this by the total number of nodes yields the overall computational complexity of RSS:

$$O(|A| \cdot C \log C)^H.$$

For comparison, the standard Sparse Sampling algorithm has complexity  $O(|A| \cdot C^H)$ , implying that RSS introduces only an additional logarithmic factor due to the robust optimization step. Importantly, in most practical applications, sampling successor states from the generative model dominates the computational cost. As a result, the added  $\log C$  factor in RSS is typically negligible in practice and does not significantly affect overall runtime.

## 5 Experiments

We evaluate the performance of the proposed RSS algorithm in two benchmark environments: `FrozenLake` and `CartPole`, aiming to empirically assess its robustness under model misspecification and compare it to standard Sparse Sampling (SS).

All experiments are conducted in the setting of *online planning with model uncertainty*. The agent computes actions by simulating future trajectories using an inaccurate generative model, differing from the true environment dynamics.

In these environments, uncertainty is present only in certain regions, while others are accurately modeled. This reflects a common real-world scenario in which hazardous or rarely visited states lack sufficient data, resulting in higher model uncertainty, whereas frequently visited safe regions benefit from more reliable transition estimates. To capture

this structure, we apply the robust backup update (15) exclusively in states with uncertainty. In all other states, we use the standard expected backup (11). Full algorithmic details are provided in the supplementary material.

This selective use of robust backups preserves the theoretical guarantees established in Theorem 1, while avoiding overly conservative behavior in well-modeled regions. Further discussion of this design choice appears in the supplementary material.

## 5.1 FrozenLake

**Environment.** The `FrozenLake` task is played on an  $8 \times 8$  grid. The agent begins in the upper-left cell and must navigate to the goal in the lower-right cell without falling into any of the "hole" cells scattered throughout the grid. At each time step, the agent chooses one of four actions: up, down, left or right, but movement is stochastic: with probability  $p$  the agent moves in the intended direction, and with probability  $(1 - p)/2$  it instead slips to one of the two orthogonal neighbors.

The immediate reward at each state is defined as:  $r(s) = \frac{1}{(d(s)+1)^3}$ , where  $d(s)$  is the Manhattan distance from state  $s$  to the goal. A terminal reward of 1 is granted upon reaching the goal, while falling into a hole yields a reward of 0. Each episode ends when a terminal state is reached or after 150 time steps.

**Model Uncertainty.** In our setup, the true transition dynamics are defined using  $p = 0.4$ . However, the agent plans using an approximate model that differs only in states adjacent to holes. In these uncertain regions, the probability of moving in the intended direction is increased to  $p^o = p + \rho$ , while the probabilities of deviating to either perpendicular direction are adjusted to  $(1 - p^o)/2$ . This modification satisfies the uncertainty condition defined in (1). Elsewhere, the approximate model matches the true dynamics exactly. A visualization of the environment is shown in Figure 2.

**Experimental Setup and Results.** Experimental results are summarized in Table 1. We evaluate RSS and standard SS under varying levels of model uncertainty, each over 1000 different seeds. As a benchmark, we also evaluate SS with full access to the true dynamics, achieving an average discounted return of  $0.249 \pm 0.012$ . All methods use a planning horizon of  $H = 3$ , sample width  $C = 50$ , and discount factor  $\gamma = 0.99$ . The uncertainty budget  $\rho$  is varied across the set  $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ .

As expected, both RSS and SS underperform compared to the SS variant with full access to the true environment dynamics. However, RSS consistently demonstrates better performance than SS across all values of  $\rho$ , with the performance gap widening as uncertainty increases. This highlights RSS’s robustness to model misspecification and its ability to maintain stronger performance under growing uncertainty.

## 5.2 CartPole

**Environment.** We use the `CartPole` environment, where the agent must balance a pole on a moving cart by

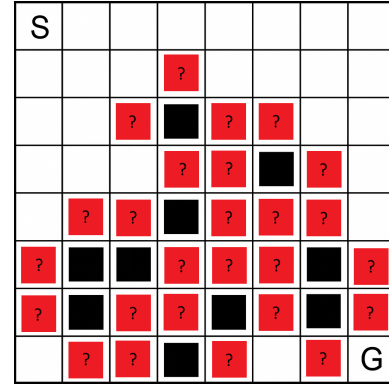


Figure 2: Visualization of the  $8 \times 8$  `FrozenLake` environment, a stochastic grid-world where the agent starts in the top-left cell (S) and aims to reach the goal in the bottom-right cell (G), while avoiding hazardous holes represented by black squares. Due to the stochastic nature of the environment, the agent’s actions may not always result in the intended direction. Cells adjacent to holes are marked with red squares containing question marks, highlighting regions of model uncertainty where the agent’s planning model deviates from the true environment dynamics.

$\rho$	RSS	SS
0.1	<b>0.177</b> $\pm 0.011$	0.172 $\pm 0.011$
0.2	<b>0.171</b> $\pm 0.011$	0.123 $\pm 0.009$
0.3	<b>0.145</b> $\pm 0.010$	0.109 $\pm 0.009$
0.4	<b>0.126</b> $\pm 0.009$	0.098 $\pm 0.008$
0.5	<b>0.127</b> $\pm 0.009$	0.080 $\pm 0.007$
0.6	<b>0.118</b> $\pm 0.009$	0.080 $\pm 0.008$

Table 1: Performance of RSS and SS in the `FrozenLake` environment under varying uncertainty levels  $\rho$ . The reported values are the average discounted return with the standard error over 1000 different seed. The best-performing algorithm for each  $\rho$  is highlighted in bold. The average discounted return of SS with access to the true dynamics is  $0.249 \pm 0.012$ .

applying discrete left or right forces. The continuous state is defined by the cart’s position  $x$ , velocity  $\dot{x}$ , pole angle  $\theta$ , and angular velocity  $\dot{\theta}$ . An episode terminates if  $|\theta| > 0.2$  radians,  $|x| > 2.4$ , or after 200 time steps. The reward is defined as  $r(\theta) = 1 - 0.2|\theta|$  in non-terminal states, and 0 otherwise, encouraging the pole to remain upright.

At the start of each episode, the cart is centered and the pole is vertical. At each step, the agent selects a force, transitioning to the next state according to deterministic dynamics, with added Gaussian noise  $\mathcal{N}(0, \sigma_\theta^2(x))$  on the pole angle. The noise variance depends on the current cart position  $x$ , defined as:

$$\sigma_\theta^2(x) = \begin{cases} \sigma_{high}^2, & \text{if } x_a < |x| < x_b \\ \sigma_{low}^2, & \text{otherwise} \end{cases} \quad (16)$$

This models a narrow “hazard zone”  $x \in \pm[x_a, x_b]$ , where the system is more unstable due to higher noise.



**Model Uncertainty.** The hazard zone is assumed to be narrow and difficult to model accurately. As such, the planning model assumes a constant low noise  $\sigma_{low}^2$  across all states, underestimating the true noise in the hazard zone. This mismatch induces localized model uncertainty. Full noise specifications and uncertain total variance calculations are detailed in the supplementary material.

**Experimental Setup and Results.** We set  $x_a = 0.02$ ,  $x_b = 0.03$ ,  $\sigma_{low} = 10^{-3}$ , and vary  $\sigma_{high}$  from 0.07 to 0.15. We compare RSS against standard SS, using a planning horizon  $H = 5$ , width  $C = 10$ , and discount factor  $\gamma = 0.999$ . As a reference, we also evaluate SS with access to the true noise model. Each configuration is averaged over 500 random seeds.

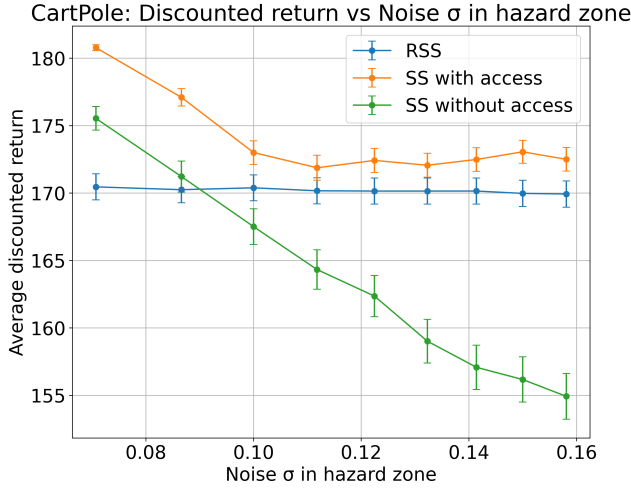


Figure 3: Average discounted return comparison of RSS and SS (with/without access to true dynamics) under increasing noise variance in the hazard zone. Error bars denote standard error across 500 different seeds.

Figures 3 show the average discounted performance as a function of the noise standard deviation  $\sigma_{high}$  within the hazard zone. An additional figure in the supplementary material presents the success rate—defined as completing 200 steps without termination—under varying noise levels. As expected, the SS variant with full access to the true dynamics achieves the highest performance across all noise levels, as it can plan optimally using accurate environment information. The performance of both SS variants (with and without model access) degrades as noise increases, indicating their increased sensitivity to unmodeled uncertainty.

In contrast, RSS maintains stable performance across all levels of noise, demonstrating its robustness to model misspecification. Notably, in low-noise regimes, RSS underperforms compared to SS without access. This is a known phenomenon in robust planning: robust policies are inherently conservative, as they optimize for the worst-case plausible dynamics within an uncertainty set, leading to overly cautious behavior that sacrifices performance for safety (Manor, Mebel, and Xu 2012).

However, as the noise variance increases, RSS maintains a near-constant performance level in this scenario. SS without access to the true model continues to rely on an underestimated noise model, resulting in unsafe and suboptimal actions, while RSS anticipates and mitigates adverse dynamics. As a result, RSS eventually outperforms SS without access in both return and success rate. This crossover point highlights the fundamental trade-off in robust planning: while robust methods may underperform in low-risk settings, they provide significant benefits in high-uncertainty environments by reducing risk and failure rates.

## 6 Conclusions

In this work, we introduced the Robust Sparse Sampling (RSS) algorithm, the first online planning algorithm for RMDPs with finite-sample theoretical performance guarantees. RSS extends the Sparse Sampling algorithm by incorporating robustness against model errors, leveraging the dual formulation of robust value functions and Sample Average Approximation (SAA) techniques. Our theoretical analysis establishes finite-time error bounds for RSS, and we demonstrate its effectiveness in simulative experiments in environments with uncertain transition dynamics.

We hope that the RSS algorithm and methods will serve as a foundation for future research in robust online planning, both for methods that can scale better for large state and action spaces, and for online methods that can handle model uncertainty. Moreover, we wish to extend our methods to anytime-fashion Monte Carlo Tree Search (MCTS), and to Partially Observable Markov Decision Process (POMDP) settings.

**Limitations** While the RSS algorithm is the first to address robust online planning under model uncertainty with formal performance guarantees, it exhibits several important limitations.

Similar to Sparse Sampling, RSS suffers from significant sample and computational inefficiency, as the complexity grows exponentially with the planning horizon  $H$ . This severely restricts its practical applicability in environments requiring long-horizon planning. However, in settings where short planning horizons are sufficient—e.g., due to low discount factors or inherently short episodes—RSS may still offer a viable and effective alternative.

Second, the algorithm assumes prior knowledge of the uncertainty budget parameter  $\rho$ . In real-world applications, accurately estimating  $\rho$  is often nontrivial, especially in non-stationary or partially observed environments where the transition dynamics may evolve over time. Estimating such parameters reliably remains an open problem in robust decision-making (Kumar et al. 2024; Suilen et al. 2022).

A current limitation that leads to over-conservatism is the rectangularity assumption of the uncertainty set. Recent works have shown promising directions to address this issue (Goyal and Grand-Clement 2023), and we hope to incorporate those in online robust planning as well.

## References

- Berend, D.; and Kontorovich, A. 2012. On the convergence of the empirical distribution. *arXiv preprint arXiv:1205.6711*.
- Browne, C. B.; Powley, E.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1): 1–43.
- Burrone, J.; Domke, J.; and Sheldon, D. 2023. Sample average approximation for Black-Box VI. *arXiv preprint arXiv:2304.06803*.
- Coulom, R. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*, 72–83. Springer.
- Dong, J.; Li, J.; Wang, B.; and Zhang, J. 2022. On-line policy optimization for robust mdp. *arXiv preprint arXiv:2209.13841*.
- Goyal, V.; and Grand-Clement, J. 2023. Robust markov decision processes: Beyond rectangularity. *Mathematics of Operations Research*, 48(1): 203–226.
- Haneveld, K.; and Van der Vlerk, M. H. 2020. *Stochastic programming*. Springer.
- Iyengar, G. N. 2005. Robust Dynamic Programming. *Mathematics of Operations Research*, 30(2): 257–280.
- Kearns, M.; Mansour, Y.; and Ng, A. Y. 2002. A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes. *Machine Learning*, 49(2-3): 193–208.
- Kocsis, L.; and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, 282–293. Springer.
- Koenig, S. 2001. Agent-centered search. *AI Magazine*, 22(4): 109–109.
- Kohankhaki, F.; Aghakasiri, K.; Zhang, H.; Wei, T.-H.; Gao, C.; and Müller, M. 2024. Monte Carlo tree search in the presence of transition uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 20151–20158.
- Kumar, N.; Wang, K.; Gadot, U.; Levy, K. Y.; and Mannor, S. 2024. Learning the Uncertainty Set in Robust Markov Decision Process. In *The Second Tiny Papers Track at ICLR 2024*.
- Lim, M. H.; Becker, T. J.; Kochenderfer, M. J.; Tomlin, C. J.; and Sunberg, Z. N. 2023. Optimality guarantees for particle belief approximation of pomdps. *Journal of Artificial Intelligence Research*, 77: 1591–1636.
- Lim, M. H.; Tomlin, C. J.; and Sunberg, Z. N. 2019. Sparse tree search optimality guarantees in pomdps with continuous observation spaces. *arXiv preprint arXiv:1910.04332*.
- Littman, M. L.; Dean, T. L.; and Kaelbling, L. P. 1995. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI’95, 394–402.
- Mannor, S.; Mebel, O.; and Xu, H. 2012. Lightning does not strike twice: Robust MDPs with coupled uncertainty. *arXiv preprint arXiv:1206.4643*.
- Mannor, S.; Simester, D.; Sun, P.; and Tsitsiklis, J. N. 2007. Bias and variance approximation in value function estimates. *Management Science*, 53(2): 308–322.
- Nilim, A.; and Ghaoui, L. E. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research*, 53(5): 780–798.
- Panaganti, K.; and Kalathil, D. 2022. Sample complexity of robust reinforcement learning with a generative model. In *International Conference on Artificial Intelligence and Statistics*, 9582–9602. PMLR.
- Panaganti, K.; Xu, Z.; Kalathil, D.; and Ghavamzadeh, M. 2022. Robust Reinforcement Learning using Offline Data. *arXiv:2208.05129*.
- Papadimitriou, C. H.; and Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of operations research*, 12(3): 441–450.
- Ross, S.; Pineau, J.; Paquet, S.; and Chaib-Draa, B. 2008. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research*, 32: 663–704.
- Shapiro, A.; Dentcheva, D.; and Ruszczyński, A. 2021. *Lectures on stochastic programming: modeling and theory*. SIAM.
- Shapiro, A.; and Li, Y. 2025. Risk-averse formulations of Stochastic Optimal Control and Markov Decision Processes. *arXiv preprint arXiv:2505.16651*.
- Sharma, A.; Harrison, J.; Tsao, M.; and Pavone, M. 2019. Robust and adaptive planning under model uncertainty. In *Proceedings of the international conference on automated planning and scheduling*, volume 29, 410–418.
- Silver, D.; and Veness, J. 2010. Monte-Carlo planning in large POMDPs. *Advances in neural information processing systems*, 23.
- Sinha, D.; and Chakrabarty, S. P. 2024. Multilevel Monte Carlo in Sample Average Approximation: Convergence, Complexity and Application. *arXiv preprint arXiv:2407.18504*.
- Somani, A.; Ye, N.; Hsu, D.; and Lee, W. S. 2013. DESPOT: Online POMDP planning with regularization. *Advances in neural information processing systems*, 26.
- Suilen, M.; Simão, T. D.; Parker, D.; and Jansen, N. 2022. Robust anytime learning of Markov decision processes. *Advances in Neural Information Processing Systems*, 35: 28790–28802.
- Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling Up Robust MDPs using Function Approximation. In Xing, E. P.; and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 181–189. Beijing, China: PMLR.
- Wang, Y.; and Zou, S. 2021. Online robust reinforcement learning with model uncertainty. *Advances in Neural Information Processing Systems*, 34: 7193–7206.