

Failure Probability Estimation for Black-Box Autonomous Systems using State-Dependent Importance Sampling Proposals

Harrison Delecki¹, Sydney M. Katz¹, Mykel J. Kochenderfer¹

Abstract—Estimating the probability of failure is a critical step in developing safety-critical autonomous systems. Direct estimation methods such as Monte Carlo sampling are often impractical due to the rarity of failures in these systems. Existing importance sampling approaches do not scale to sequential decision-making systems with large state spaces and long horizons. We propose an adaptive importance sampling algorithm to address these limitations. Our method minimizes the forward Kullback–Leibler divergence between a state-dependent proposal distribution and a relaxed form of the optimal importance sampling distribution. Our method uses Markov score ascent methods to estimate this objective. We evaluate our approach on four sequential systems and show that it provides more accurate failure probability estimates than baseline Monte Carlo and adaptive importance sampling techniques. Our implementation is available at <https://github.com/sisl/SPAIS.jl>.

I. INTRODUCTION

Autonomous systems are increasingly being considered for safety-critical domains such as aviation [1], [2], autonomous driving [3], and home robotics [4]. These autonomous systems operate sequentially by taking actions after receiving observations of the environment. For example, an autonomous vehicle must continuously process sensor data, predict the movements of other vehicles, and make decisions about steering and braking. The safety of these systems must be rigorously validated in simulation before deployment. One way to quantify system safety is to estimate the probability of a system failure such as a collision of an autonomous vehicle with other agents. Estimating the probability of failure for autonomous systems may highlight weaknesses and uncover potentially dangerous scenarios.

There are three key challenges associated with estimating the probability of failure in sequential autonomous systems. First, failures tend to be rare when the system is designed to be safe. Estimating the probability of failure using simple methods like Monte Carlo sampling may require an enormous number of expensive simulations. Second, the search space over failure events is very high-dimensional because autonomous systems operate over large state spaces and long time horizons. Third, autonomous systems can exhibit multimodal failures, requiring validation techniques that can adequately capture diverse behaviors.

This work was supported in part by the National Science Foundation (NSF) Graduate Research Fellowship under grant #DGE-2146755. Any opinions, findings, or conclusions expressed here do not reflect the views of the NSF. The NASA University Leadership initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.

¹Department of Aeronautics and Astronautics, Stanford University (e-mail: {hdelecki, smkatz, mykel}@stanford.edu).

Many previous approaches have attempted to address these challenges through importance sampling techniques. Importance sampling techniques aim to sample from alternative distributions that assign higher likelihood to failure events [5]. These methods often struggle to scale to high-dimensional, long-horizon problems because they rely on simple parametric proposals that are difficult to design. Extensions of importance sampling to sequential problems using dynamic programming and reinforcement learning show promise but remain limited to small discrete state spaces and short time horizons [6], [7].

In this work, we aim to enable efficient failure probability estimation for black-box sequential systems through importance sampling. Inspired by previous work, we decompose the high-dimensional sampling problem over system trajectories using a sequential state-dependent proposal distribution. We optimize the proposal by minimizing the forward Kullback–Leibler (KL) divergence between the sequential proposal and an approximate distribution over failure trajectories. We estimate gradients of the objective using a Markov score ascent method [8], [9], which approximates samples from the target distribution using MCMC. In addition to optimizing the proposal, the algorithm uses these samples to compute an importance sampling estimate of the probability of failure. The impact of our approach is illustrated in Fig. 1. In summary, our contributions are as follows:

- We propose a sequential state-based importance sampling proposal for failure probability estimation.
- We optimize the proposal by minimizing the forward KL divergence to a tractable approximation of the optimal importance sampling distribution.
- We demonstrate that the proposed approach achieves more accurate estimates of the probability of failure compared to baselines on four sample problems.

II. RELATED WORK

White-box vs. black-box: Formal verification methods like model-checking take a white-box approach. These methods use internal information about the system under test to prove safety properties [10], [11]. White-box methods do not scale to large systems because they consider every possible simulation of the system under test. In this work, we take a black-box approach. Black-box methods generally scale to large, complex systems because they do not require access to an internal model, but only pass inputs to the system and observe outputs. See the survey by Corso *et al.* [12] for more on black-box methods.

Falsification: A common approach to black-box safety validation of sequential systems is falsification, which aims

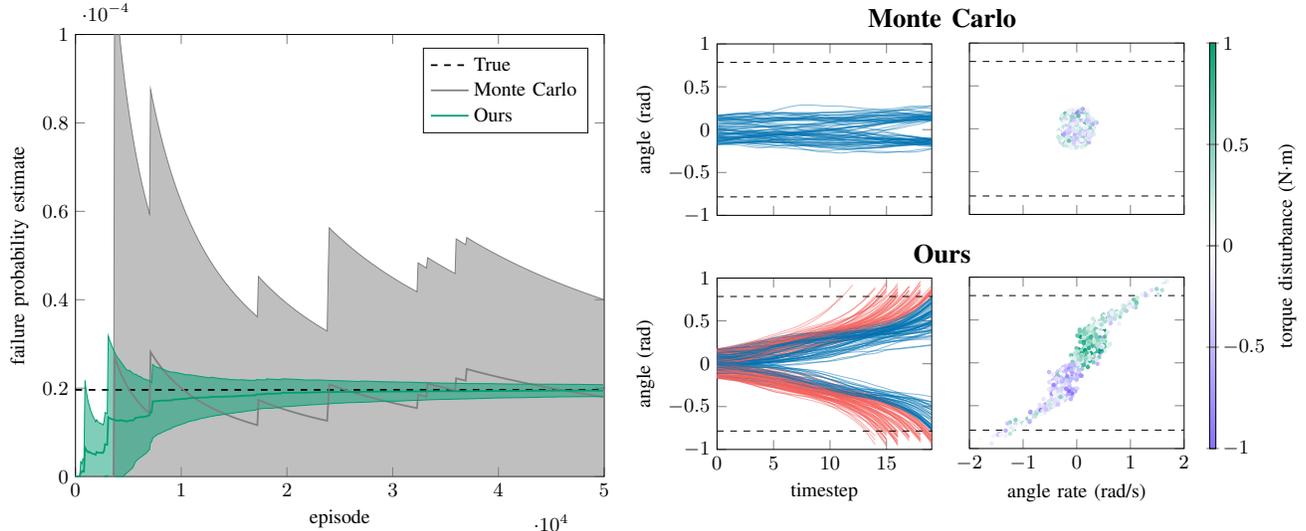


Fig. 1: Comparison of the proposed approach and Monte Carlo (MC) sampling for estimating failure probabilities in an inverted pendulum with a rule-based controller under torque disturbances. Failures occur when the magnitude of the pendulum’s angle from the vertical exceeds a threshold. The left figure shows the estimated failure probability using our method during training compared to MC. Our method is more accurate with fewer samples than MC. The middle two figures compare 200 pendulum angle trajectories over time using samples from MC and from ours after training. Our method discovers failure trajectories (red) across modes, while all MC samples are safe (blue). The two right figures show torque disturbances applied at each state in the sampled trajectories. Our method learns to add positive torque disturbances for positive angles (left from vertical) and vice versa, enabling efficient sampling for failure probability estimation.

to find individual failure trajectories. Falsification approaches use optimization [13], trajectory planning [14], and reinforcement learning [15] to find specific inputs that lead to failure. Adaptive Stress Testing further incorporates trajectory likelihood to search for the most-likely failure event [16]. The drawback of falsification methods is that they tend to converge on a single failure, such as the most-likely or most severe, and do not explore the space of potential failures.

Failure probability estimation: Failure probability estimation provides a more comprehensive assessment of system safety by characterizing the distribution over failures. Three common techniques for failure probability estimation are Sequential Monte Carlo (SMC) [17], multilevel splitting [18], and importance sampling [19].

MCMC-based methods: SMC and splitting methods typically rely on Markov chain Monte Carlo (MCMC) methods to directly sample many failure trajectories which are then used to estimate the failure probability. These approaches may require domain knowledge to perform well [20], [21], or the system under test to be differentiable to scale to larger problems [22]. In contrast, our method uses MCMC only to help optimize our proposal distribution through Markov score ascent, rather than for generating samples for estimation. This key distinction allows our approach to avoid the scalability limitations of direct MCMC sampling.

Importance sampling methods: Adaptive Importance Sampling (AIS) methods iteratively improve a proposal distribution, often by minimizing the KL divergence between the proposal and the true failure distribution [23], [24]. Existing AIS methods face challenges with long time horizons, because they typically model only a small set of input parameters using simple exponential family propos-

als [25], making them intractable for sequential problems. Our approach addresses these challenges by decomposing the high-dimensional sampling problem using a sequential state-dependent proposal distribution. Importance sampling algorithms have been extended to sequential problems by incorporating dynamic programming, but this approach is limited to small discrete state spaces [6]. Recent work also builds on policy gradient methods for AIS, but face challenges in scaling due to high variance gradient estimates. [7]. In this work, we propose an AIS algorithm for sequential problems with continuous state spaces and long horizons that overcomes these limitations by optimizing a flexible state-dependent proposal distribution using Markov score ascent to estimate lower-variance objective gradients.

III. BACKGROUND

In this section, we provide necessary background on safety validation for sequential autonomous systems, failure probability estimation, and importance sampling.

A. Safety Validation for Sequential Systems

Consider a system under test that takes actions a in an environment after receiving observations o of its state s . We assume states may transition stochastically according to a transition model $p(s' | s, a)$ and state observations may depend on imperfect sensors modeled as $p(o | s)$. The system under test may also act stochastically depending on the observation with policy $\pi(a | o)$.

In safety validation, we search for failure events by controlling sources of randomness in the state transitions, observations, and system under test. We use disturbances, denoted by x , to control sources of randomness. We assume

disturbances are continuous values in the space $\mathcal{X} \subseteq \mathbb{R}^{D_x}$ where $D_x \in \mathbb{N}$ is the dimension of the disturbance space. Given a state s_t and disturbance x_t at time t , we can simulate the next timestep $t + 1$ according to $o_t \sim p(\cdot | s_t, x_t)$, $a_t \sim \pi(\cdot | o_t, x_t)$, and $s_{t+1} \sim p(\cdot | s_t, a_t, x_t)$. Different components of the disturbance x_t may be applied to the observation, system under test, and transition. Given a set of disturbances up to horizon T , we can simulate the system under test with disturbances to compute a system trajectory $\tau = (s_1, a_1, o_1, x_1, \dots, s_T, a_T, o_T, x_T)$.

We assume that disturbances are sampled according to a distribution that may depend on the current system state $d(x | s)$. The disturbance distribution reflects how likely different disturbances are to occur in the real operating environment and is typically designed using real-world data or expert knowledge. This formulation could also consider disturbances that depend on observations and actions, but we focus only on state-dependent disturbances for simplicity. Using the disturbance distribution, we can define a distribution over system trajectories

$$p(\tau) = p(s_1) \prod_{t=1}^T p(s_{t+1}, o_t, a_t | s_t, x_t) d(x_t | s_t) \quad (1)$$

where $p(s_1)$ is the initial state distribution and

$$p(s_{t+1}, o_t, a_t | s_t, x_t) = p(s_{t+1} | s_t, a_t, x_t) p(o_t | s_t, x_t) \pi(a_t | o_t, x_t)$$

We denote this the nominal trajectory distribution since it represents the anticipated trajectory distribution in operation.

We evaluate the safety of system trajectories using an evaluation function $f(\tau) \in \mathbb{R}$. When $f(\tau)$ exceeds a failure threshold γ , we say that the trajectory is a failure event, with higher values indicating greater proximity to failure. We can express the distribution over all failure trajectories as the conditional distribution

$$p(\tau | f(\tau) \geq \gamma) = \frac{p(\tau) \mathbb{1}\{f(\tau) \geq \gamma\}}{\int p(\tau') \mathbb{1}\{f(\tau') \geq \gamma\} d\tau'} \quad (2)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function and the denominator integrates over all trajectories. The denominator is the total probability of failure, which we denote as μ . Due to the high dimensionality of trajectories and the complexity of autonomous systems, exact computation of this quantity is intractable. Instead, we rely on black-box evaluations of the system to estimate this probability.

B. Failure Probability Estimation

Failure probability estimation is the problem of estimating the denominator of Eq. (2), or the expected value:

$$\mu = \mathbb{E}_{p(\tau)} [\mathbb{1}\{f(\tau) \geq \gamma\}] \quad (3)$$

A simple approach to estimate μ is using Monte Carlo (MC) sampling, which uses N independent samples from $p(\tau)$ to compute the empirical estimate

$$\hat{\mu}_{\text{MC}} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}\{f(\tau_i) \geq \gamma\} \quad (4)$$

For very rare failures (e.g. $\mu \approx 10^{-9}$), standard Monte Carlo may require billions of samples, making direct estimation infeasible for complex systems.

C. Importance Sampling

Importance sampling can reduce the variance of failure probability estimates by focusing samples on regions of interest. IS draws samples from a proposal distribution $q(\tau)$, and computes an estimate

$$\hat{\mu}_{\text{IS}} = \frac{1}{N} \sum_{i=1}^N w(\tau_i) \mathbb{1}\{f(\tau_i) \geq \gamma\} \quad (5)$$

where the samples are weighted according to their importance weight $w(\tau) = p(\tau)/q(\tau)$. The minimum variance importance sampling distribution is the failure distribution itself, meaning that samples are only drawn from the failure region with likelihood proportional to $p(\tau)$. The estimator is unbiased if $q(\tau) > 0$ whenever $\mathbb{1}\{f(\tau_i) \geq \gamma\} p(\tau) > 0$.

The key challenge of importance sampling is designing a good proposal distribution. The variance of the estimator increases roughly exponentially with the problem dimension, but it can be reduced by a well-designed proposal distribution. However, modeling $D_x T$ -dimensional trajectory distributions using previous IS approaches can be extremely difficult.

IV. METHODS

Next, we introduce our method for failure probability estimation in sequential systems. We propose a sequential state-based IS proposal, discuss proposal optimization, and detail our algorithm.

A. Sequential Proposal Distribution

To address the challenges of fitting a complex, high-dimensional proposal distribution, we decompose the problem using a proposal applied at each timestep. Since the system behavior and disturbance distribution may depend on the system state, we condition the distribution on the current state. These proposals take the form $q_\theta(x | s)$ with parameters θ . We can draw samples from $q_\theta(\tau)$ by sampling an initial state $s_1 \sim p(s_1)$ and simulating each timestep $x_t \sim q_\theta(\cdot | s_t)$, $o_t \sim p(\cdot | s_t, x_t)$, $a_t \sim \pi(\cdot | o_t, x_t)$, and $s_{t+1} \sim p(\cdot | s_t, a_t, x_t)$. System trajectories sampled using the proposal $q_\theta(x | s)$ have the density:

$$q_\theta(\tau) = p(s_1) \prod_{t=1}^T p(s_{t+1}, o_t, a_t | s_t, x_t) q_\theta(x_t | s_t) \quad (6)$$

In computing the importance weight as the ratio of Eq. (1) to Eq. (6), all terms cancel except the disturbance distribution and the proposal. The importance weight reduces to:

$$w(\tau) = \prod_{t=1}^T \frac{d(x_t | s_t)}{q_\theta(x_t | s_t)} \quad (7)$$

This formulation reduces the problem of learning a proposal over full disturbance trajectories to learning a proposal over a single timestep given the current state. However, obtaining a reliable IS estimate of the failure probability still requires careful design of this proposal.

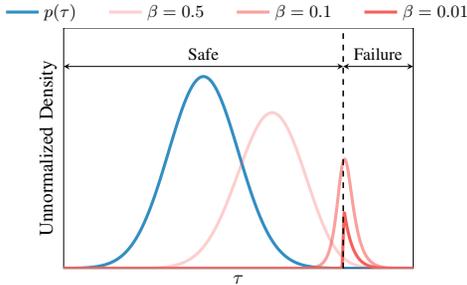


Fig. 2: Illustration of $\tilde{p}(\tau)$. The approximate failure distribution (red) approaches the true failure distribution as β decreases.

B. Proposal Optimization

The optimal importance sampling distribution is known to be proportional to the true distribution over failure trajectories. Therefore, we propose to learn the proposal parameters by minimizing the forward KL divergence between the proposal and the failure distribution:

$$\theta^* = \operatorname{argmin}_{\theta} D_{KL}(p(\tau | f(\tau) \geq \gamma) || q_{\theta}(\tau)) \quad (8)$$

Minimizing the forward KL divergence is desirable for importance sampling because its mass covering properties help ensure the IS estimator remains unbiased [19]. This objective is equivalent up to a constant of proportionality to minimizing the cross-entropy:

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \mathbb{E}_{p(\tau | f(\tau) \geq \gamma)} [-\log q_{\theta}(\tau)] \quad (9)$$

However, this formulation creates a chicken-and-egg problem, since our goal is to approximate $p(\tau | f(\tau) \geq \gamma)$ with $q_{\theta}(\tau)$, but the cross-entropy involves an expectation over $p(\tau | f(\tau) \geq \gamma)$. A common approach optimizes an IS estimate of the expectation. Unfortunately, the variance of the IS estimate increases rapidly with problem dimension, making optimization very difficult especially when the proposal is far from the target.

Building on previous work in variational Bayesian inference, we optimize the objective in Eq. (9) using Markov score ascent methods [8], [9]. Markov score ascent methods can achieve much lower variance estimates of the objective. The key idea of Markov score ascent methods is to estimate the cross-entropy using samples from an MCMC kernel that is ergodic with respect to the target distribution. The kernel is chosen to directly use the approximation of the target distribution $q_{\theta}(\tau)$. With repeated application of the kernel, the samples will be approximately distributed according to the target. Given N trajectories approximately distributed according to the target, we may compute a Monte Carlo estimate of the objective:

$$\mathcal{L}(\theta) \approx \frac{1}{N} \sum_{n=1}^N -\log q_{\theta}(\tau_n) \quad (10)$$

Expanding the objective and removing terms that do not

depend on θ , the loss can be computed as

$$\mathcal{L}(\theta) \approx \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^T -\log q_{\theta}(x_{t,n} | s_{t,n}) \quad (11)$$

where subscript t, n denotes timestep t in trajectory n . This estimated cross-entropy objective is optimized using stochastic gradient descent.

In this work, we construct an MCMC kernel by performing a single step of Independent Metropolis-Hastings (IMH) [26]. Given an initial trajectory τ , we can generate a new sample from IMH by first sampling a proposal trajectory $\tau' \sim q_{\theta}(\cdot)$. IMH either accepts the new trajectory τ' according to the Metropolis-Hastings acceptance probability a :

$$a(\tau, \tau') = \min \left(1, \frac{p(\tau' | f(\tau') \geq \gamma) q_{\theta}(\tau)}{p(\tau | f(\tau) \geq \gamma) q_{\theta}(\tau')} \right) \quad (12)$$

Otherwise, IMH rejects the proposal and τ remains the same.

One challenge in applying IMH to sample from the true failure distribution is that the failure distribution applies zero density to safe trajectories. Since failures are rare, many sampled trajectories may be safe. In turn, the acceptance probabilities will be zero, making it difficult for the MCMC step to move trajectories closer to the failure distribution.

C. Approximating the Failure Distribution

To address the issue of low acceptance rates, we use a smooth approximation of the discontinuous failure distribution. Specifically, we approximate the indicator in the target by $P_{\beta}(f(\tau) - \gamma)$ where P_{β} is the cumulative distribution function of a logistic distribution with zero mean and scale β . This approximation replaces the discontinuous indicator with a smooth sigmoid curve from zero to one. The approximate failure distribution $\tilde{p}(\tau)$ has unnormalized density

$$\tilde{p}(\tau) \propto p(\tau) P_{\beta}(f(\tau) - \gamma) \quad (13)$$

which relaxes the failure distribution, enabling stable sampling during optimization. This approximation is visualized in Fig. 2 for a few values of β .

Using the approximate posterior, our objective becomes:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tilde{p}(\tau)} [-\log q_{\theta}(\tau)] \quad (14)$$

We can now apply IMH to generate samples for Markov score ascent by computing the acceptance probability as $\min(1, \tilde{w}(\tau')/\tilde{w}(\tau))$ where $\tilde{w}(\tau) = \tilde{p}(\tau)/q_{\theta}(\tau)$. Note that we can still compute the acceptance probability using $\tilde{w}(\tau)$ without knowing the normalizing constant of the approximate failure distribution, because the constants in the numerator and denominator in the ratio of Eq. (12) cancel.

D. Algorithm Details

The complete algorithm, which we call State-dependent Proposal Adaptive Importance Sampling (SPAIS), is shown in Algorithm 1. The algorithm maintains a set of N trajectories that will be updated using the IMH kernel and a buffer to store evaluations $f(\tau)$ and the corresponding IS weight of samples from $q_{\theta}(\tau)$. At each iteration, the algorithm samples N new proposal trajectories τ' . These samples are

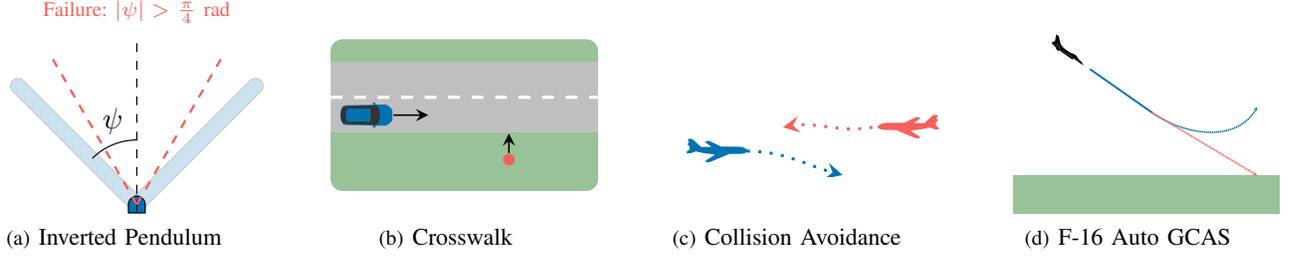


Fig. 3: The environments used in failure probability estimation experiments.

Algorithm 1 State-dependent Proposal Adaptive Importance Sampling (SPAIS)

Require: nominal distribution $p(\tau)$, evaluation function $f(\tau)$, proposal $q_{\theta_1}(x | s)$, failure threshold γ , no. particles N , no. iterations N_{iter}

- 1: **function** SPAIS($p(\tau), f(\tau), q_{\theta_1}(x | s), \gamma, N, N_{\text{iter}}$)
- 2: $\tau_n \sim q_{\theta_1}(\tau)$ for $n \in 1 : N$ \triangleright Sample initial trajectories
- 3: $\mathcal{D} \leftarrow \{f(\tau_n), p(\tau_n)/q_{\theta_1}(\tau_n)\}_{n=1}^N$ \triangleright Initialize IS buffer
- 4: $\tilde{w}_n \leftarrow p(\tau_n)P_{\beta}(f(\tau_n) - \gamma)/q_{\theta_1}(\tau_n)$ for $n \in 1 : N$
- 5: **for** $k \in 1 : N_{\text{iter}}$ **do**
- 6: $\tau'_n \sim q_{\theta_k}(\tau)$ for $n \in 1 : N$ \triangleright Sample trajectories
- 7: $\mathcal{D} \leftarrow \mathcal{D} \cup \{f(\tau'_n), p(\tau'_n)/q_{\theta_k}(\tau'_n)\}_{n=1}^N$ \triangleright Update IS buffer
- 8: $\tilde{w}'_n \leftarrow p(\tau'_n)P_{\beta}(f(\tau'_n) - \gamma)/q_{\theta_k}(\tau'_n)$ for $n \in 1 : N$
- 9: $\tilde{w}_n \leftarrow p(\tau_n)P_{\beta}(f(\tau_n) - \gamma)/q_{\theta_k}(\tau_n)$ for $n \in 1 : N$
- 10: $\tau_n \leftarrow \text{MHACCEPT}(\tilde{w}'_n, \tilde{w}_n)$ for $n \in 1 : N$
- 11: $\mathcal{L}(\theta) \leftarrow \sum_{n=1}^N \sum_{t=1}^T -\log q_{\theta}(x_{t,n} | s_{t,n})$ \triangleright Eq. (11)
- 12: $\theta_{k+1} \leftarrow \text{SGD}(\mathcal{L}, \theta_k)$ \triangleright Stochastic gradient descent on \mathcal{L}
- 13: **return** $\hat{\mu} \leftarrow \text{IMPORTANCESAMPLINGESTIMATE}(\mathcal{D}, \gamma)$ \triangleright Eq. (5)

first added to the IS buffer. Next, they are used to update the trajectory set. Each proposed trajectory is accepted or rejected according to the MH acceptance probability. At each iteration, our algorithm performs N simulations of the system under test, and requires $2N$ evaluations of the proposal. However, the cost of evaluating the proposal is typically insignificant compared to that of simulation. At the end of each iteration, we update the proposal by fitting the proposal to the updated set of trajectories using stochastic gradient descent. After a fixed number of iterations, the algorithm returns an IS estimate of the failure probability using Eq. (5).

We parameterize $q_{\theta}(x | s)$ as a multivariate Gaussian

$$q_{\theta}(x | s) = \mathcal{N}(x | \mu_{\theta}(s), \Sigma_{\theta}(s))$$

where $\mu_{\theta}(\cdot)$ and $\Sigma_{\theta}(\cdot)$ are both neural networks with two hidden layers of size 64 and 32 respectively. We find using any small value of β in the range $[10^{-4}, 10^{-2}]$ gives good performance. We use $\beta = 10^{-2}$ for all evaluations. For each problem, we normalize the state space features and pretrain $q_{\theta}(x | s)$ to match $d(x | s)$.

V. EXPERIMENTS

In this section, we describe our experiments to evaluate the proposed approach. We describe baseline estimation methods, evaluation metrics, and introduce four example safety validation problems.

A. Evaluation Metrics and Baselines

We evaluate the bias and variance of the estimated probability of failure using the relative error ϵ_{rel} and absolute

Problem	μ	D_s	D_x	T
Pendulum	1.96×10^{-5}	2	1	20
Crosswalk	5.90×10^{-5}	8	5	100
Collision Avoidance	2.23×10^{-5}	12	1	40
F-16 GCAS	6.59×10^{-5}	13	6	200

TABLE I: Failure probability μ , state size D_s , disturbance size D_x , and horizon T for each problem.

relative error ϵ_{abs} , respectively:

$$\epsilon_{\text{rel}} = (\hat{\mu} - \mu)/\mu, \quad \epsilon_{\text{abs}} = |\hat{\mu} - \mu|/\mu$$

We obtain an estimate of the ground truth failure probability μ using 10^7 Monte Carlo samples. We give each method a budget of 50000 simulations. Metrics are averaged over 10 trials to arrive at estimates of the empirical bias and variance.

We compare the method against three baseline methods. The simplest baseline is Monte Carlo estimation. We also compare against a variant of the cross-entropy method (CEM) that optimizes a state-independent proposal distribution applied at each timestep. Finally, we evaluate against PG-AIS, which learns a reinforcement learning policy for IS [7].

B. Validation Problems

We demonstrate SPAIS on four systems illustrated in Fig. 3. Each problem's properties are reported in Table I.

Inverted pendulum: We evaluate an underactuated inverted pendulum system with a nonlinear rule-based control policy. The system is subjected to additive torque disturbances over 20 timesteps. We define failure to occur when the magnitude of the pendulum's angle from the vertical exceeds $\pi/4$ rad. This problem has two distinct failure modes corresponding to leftward and rightward falls.

Crosswalk: In the second experiment we evaluate an autonomous vehicle (AV) approaching a pedestrian at a crosswalk inspired by Koren *et al.* [27]. The AV uses the Intelligent Driver Model [28] for longitudinal control, maintaining safe distances from obstacles. We adversarially control the pedestrian's 2D acceleration, perceived position and speed as detected by the AV. Failures occur when the AV collides with the pedestrian.

Aircraft collision avoidance: The third experiment evaluates an aircraft collision avoidance system based on the AVOIDDS dataset and simulator [29]. The system detects nearby intruding aircraft and recommends pilot maneuvers (e.g. climb or descend) to avoid midair collisions with

TABLE II: Relative and absolute error for each method across all four validation problems.

Method	Pendulum		Crosswalk		Collision Avoidance		F-16 GCAS	
	ϵ_{rel}	ϵ_{abs}	ϵ_{rel}	ϵ_{abs}	ϵ_{rel}	ϵ_{abs}	ϵ_{rel}	ϵ_{abs}
MC	-0.39 ± 0.99	0.81 ± 0.65	0.15 ± 0.36	0.28 ± 0.27	0.30 ± 0.55	0.54 ± 0.26	0.39 ± 0.88	0.74 ± 0.70
CEM	-0.54 ± 0.05	0.53 ± 0.05	-0.22 ± 0.11	0.22 ± 0.11	-0.18 ± 0.50	0.26 ± 0.43	0.63 ± 1.75	1.13 ± 1.09
PG-AIS	-0.41 ± 0.28	0.47 ± 0.16	-0.26 ± 0.55	0.41 ± 0.44	-0.91 ± 0.10	0.91 ± 0.10	-0.91 ± 0.45	0.94 ± 0.58
SPAIS	-0.04 ± 0.06	0.06 ± 0.03	-0.02 ± 0.13	0.09 ± 0.09	-0.05 ± 0.27	0.18 ± 0.21	0.06 ± 0.10	0.09 ± 0.07

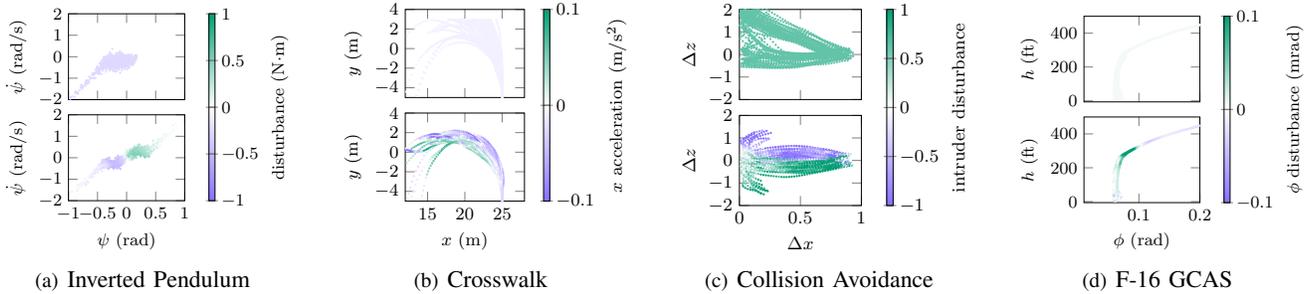


Fig. 4: Visited states and applied disturbances by CEM (above) and SPAIS (below) after training.

the ownship. We evaluate the collision avoidance policy in scenarios with perfect perception of a single intruding aircraft. The intruder’s initial state is randomly sampled according to the AVOIDDS encounter model. We adversarially control the intruder’s vertical rate to cause midair collisions.

F-16 ground collision avoidance: Finally, we consider a model of the ground collision avoidance system (GCAS) for the F-16 aircraft adapted from Heidlauf *et al.* [30]. The GCAS detects impending ground collisions and executes a series of maneuvers to prevent collision. We add disturbances to observations of orientation and angular velocity over 200 timesteps. The aircraft starts at an altitude of 578 ft, pitched down 27 deg, and is rolled 22 deg. The controller first rolls the wings level and then pitches up to avoid collision.

VI. RESULTS

The relative and absolute error for each method across all problems are reported in Table II. The proposed SPAIS algorithm achieves lower relative and absolute error on all four validation problems. The MC estimator generally has low bias, but high variance. The PG-AIS baseline method tends to have large variance, indicating that it struggles to represent the failure distribution. CEM exhibits a bias, which may be due to mode collapse where the proposal does not cover all failure modes or due to the proposal being spread very wide leading to large importance weights.

We visualize sampled system trajectories from the CEM and SPAIS proposals for each problem in Fig. 4. Each plot shows system trajectories in state space with points colored according to a mean disturbance sampled from the proposal.

Inverted pendulum: In Fig. 4(a), we plot the pendulum angle ψ against angular rate $\dot{\psi}$, and color the points according to the torque disturbance. SPAIS generally adds positive torque disturbances for positive angles, and vice versa. In contrast, the CEM proposal only samples negative disturbances and thus only samples from one of the two failure modes.

Crosswalk: CEM and SPAIS proposals for the crosswalk problem are shown in Fig. 4(b). This figure shows the 2D position of the pedestrian with points colored according to the controlled pedestrian x -axis acceleration. Both proposals steer the pedestrian into the street and towards the oncoming AV. The SPAIS proposal is more structured around an important failure mode. The trajectories are more consistent, and use some deceleration to keep the pedestrian in front of the AV for longer. In contrast, the CEM proposal is more spread out, potentially contributing to its larger bias.

Collision avoidance: Figure 4(c) shows the proposals for the aircraft collision avoidance problem, plotting normalized vertical separation Δz against horizontal distance Δx of the ownship and intruder. The CEM proposal exhibits mode collapse, favoring intruder trajectories below the ownship. The SPAIS proposal symmetrically samples trajectories above and below the ownship, contributing to reduced estimation bias.

F-16 GCAS: Figure 4(d) shows F-16 GCAS trajectories of altitude h against roll angle ϕ with points colored by the observed roll angle disturbance. SPAIS increases the error in the observed roll angle just before the controller begins to pull up, causing a delay in the maneuver and more ground collisions. The mean CEM disturbance is small, suggesting that the proposal must be wide to capture the failure trajectories discovered by SPAIS.

VII. CONCLUSION

Accurately estimating failure probabilities is a critical step in developing autonomous systems in safety-critical domains. We propose a method for failure probability estimation that takes advantage of the sequential nature of autonomous systems to construct flexible IS proposals that are optimized using Markov score ascent. Using a relatively small number of samples, our results show that SPAIS achieves under 10% empirical bias with at least two times lower empirical variance than baselines. Future work will investigate using more flexible state-based proposals to further improve estimation.

REFERENCES

- [1] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next-Generation Airborne Collision Avoidance System," *Lincoln Laboratory Journal*, vol. 19, no. 1, 2012.
- [2] M. P. Owen, A. Panken, R. Moss, L. Alvarez, and C. Leeper, "ACAS Xu: Integrated Collision Avoidance and Detect and Avoid Capability for UAS," in *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2019, pp. 1–10.
- [3] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113 816, 2021.
- [4] G. A. Zachiotis, G. Andrikopoulos, R. Gornez, K. Nakamura, and G. Nikolakopoulos, "A survey on the application trends of home service robotics," in *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2018, pp. 1999–2006.
- [5] A. Owen and Y. Zhou, "Safe and effective importance sampling," *Journal of the American Statistical Association*, vol. 95, no. 449, pp. 135–143, 2000.
- [6] J. P. Chryssanthacopoulos, M. J. Kochenderfer, and R. E. Williams, "Improved Monte Carlo sampling for conflict probability estimation," in *AIAA Non-Deterministic Approaches Conference*, 2010.
- [7] A. Corso, K.-Y. Kim, S. Gupta, G. Gao, and M. J. Kochenderfer, "A deep reinforcement learning approach to rare event estimation," *arXiv preprint arXiv:2211.12470*, 2022.
- [8] C. Naesseth, F. Lindsten, and D. Blei, "Markovian score climbing: Variational inference with $KL(p||q)$," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.
- [9] K. Kim, J. Oh, J. Gardner, A. B. Dieng, and H. Kim, "Markov chain score ascent: A unifying framework of variational inference with markovian gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022.
- [10] J. M. Schumann, *Automated Theorem Proving in Software Engineering*. Springer, 2001.
- [11] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, *Handbook of Model Checking*. Springer, 2018.
- [12] A. Corso, R. Moss, M. Koren, R. Lee, and M. Kochenderfer, "A survey of algorithms for black-box safety validation of cyber-physical systems," *Journal of Artificial Intelligence Research*, vol. 72, pp. 377–428, 2021.
- [13] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing cyber-physical systems through Bayesian optimization," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, pp. 1–18, 2017.
- [14] C. E. Tuncali and G. Fainekos, "Rapidly-exploring random trees for testing automated vehicles," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019, pp. 661–666.
- [15] T. Akazaki, S. Liu, Y. Yamagata, Y. Duan, and J. Hao, "Falsification of cyber-physical systems using deep reinforcement learning," in *International Symposium on Formal Methods (FM)*, Springer, 2018, pp. 456–465.
- [16] R. Lee, O. J. Mengshoel, A. Saksena, R. Gardner, D. Genin, J. Silbermann, M. Owen, and M. J. Kochenderfer, "Adaptive stress testing: Finding likely failure events with reinforcement learning," *Journal of Artificial Intelligence Research*, vol. 69, pp. 1165–1201, 2020.
- [17] F. Cérou, P. Del Moral, T. Furon, and A. Guyader, "Sequential Monte Carlo for rare event estimation," *Statistics and Computing*, vol. 22, no. 3, pp. 795–808, 2012.
- [18] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Analysis and Applications*, vol. 25, no. 2, pp. 417–443, 2007.
- [19] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer, 2004.
- [20] M. O’Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [21] J. Norden, M. O’Kelly, and A. Sinha, "Efficient black-box assessment of autonomous vehicle safety," *arXiv preprint arXiv:1912.03618*, 2019.
- [22] A. Sinha, M. O’Kelly, R. Tedrake, and J. C. Duchi, "Neural bridge sampling for evaluating safety-critical autonomous systems," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 6402–6416, 2020.
- [23] M. F. Bugallo, V. Elvira, L. Martino, D. Luengo, J. Miguez, and P. M. Djuric, "Adaptive importance sampling: The past, the present, and the future," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 60–79, 2017.
- [24] O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert, "Population Monte Carlo," *Journal of Computational and Graphical Statistics*, vol. 13, no. 4, pp. 907–929, 2004.
- [25] J. Uesato, A. Kumar, C. Szepesvári, T. Erez, A. Ruderman, K. Anderson, K. Dvijotham, N. Heess, and P. Kohli, "Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures," in *International Conference on Learning Representations*, 2019.
- [26] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.
- [27] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018.
- [28] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, p. 1805, 2000.
- [29] E. Smyers, S. Katz, A. Corso, and M. J. Kochenderfer, "AVOIDDS: Aircraft vision-based intruder detection dataset and simulator," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023.
- [30] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in F-16 ground collision avoidance and other automated maneuvers," in *International Workshop on Applied Verification for Continuous and Hybrid Systems at the Conference on Analysis and Design of Hybrid Systems*, 2018.