FORMAL SAFETY VERIFICATION OF STOCHASTIC SYSTEMS UNDER DIFFUSION-INDUCED UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Formal safety verification of autonomous systems in open environments is challenged by interaction-induced stochasticity. Existing methods typically assume pre-specified stochastic models or handcrafted disturbance distributions, limiting their ability to capture complex disturbance patterns encountered in real-world operation. In this paper, we leverage expressive diffusion models to learn data-driven stochastic behaviors from observed trajectories and adapts it to the evolving system state. By exploiting their intrinsic score-based reverse stochastic dynamics, we analytically propagate the learned uncertainty into an explicit verification-oriented stochastic differential equation (SDE), enabling rigorous probabilistic safety verification via stochastic barrier certificates. Experimental results on representative benchmarks demonstrate the effectiveness of our proposed approach.

1 INTRODUCTION

Autonomous systems have been widely deployed in safety-critical applications, including self-driving vehicles, intelligent robotics, and unmanned aerial vehicles Luckcuck et al. (2019); Mehdipour et al. (2023); Yurtsever et al. (2020). Operating in open and dynamically changing environments, these systems are inevitably subject to various sources of uncertainty, resulting in stochastic behaviors that cannot be adequately captured by deterministic models. Consequently, stochastic dynamical systems have become a fundamental paradigm for modeling uncertainty in autonomy, motivating the need to establish rigorous probabilistic safety guarantees Lavaei et al. (2022); Chatterjee et al. (2023); Laurenti & Lahijanian (2025).

Related Work. Existing probabilistic verification approaches commonly model uncertainty using predefined stochastic processes, transition kernels, or disturbance distributions Žikelić et al. (2023); Yang (2017); Amin et al. (2006); Prajna et al. (2004); Huang et al. (2017); Jagtap et al. (2020), while perception-based methods often parameterize environmental variability using latent variables with prescribed priors Katz et al. (2022); Cai et al. (2025); Ma et al. (2026). Such prior specification, however, can be difficult to justify in realistic settings, where interaction-induced stochasticity is generally unknown and may vary with the system state and operating context. To relax these assumptions, recent studies have developed data-driven verification methods that infer stochastic dynamics or uncertainty from observed trajectories Salamati et al. (2024); Mathiesen et al. (2026); Mazouz et al. (2024). More recently, generative models such as cGANs have been used to capture complex uncertainty distributions Hu et al. (2026), but primarily as black-box samplers for constructing stochastic abstractions from generated transitions, without exploiting their internal generative dynamics.

In this context, diffusion models Ho et al. (2020) provide a particularly suitable framework for modeling complex distributions and have recently been extended from image generation Croitoru et al. (2023) to robotic trajectory planning Janner et al. (2022); Xiao et al. (2025). Crucially, score-based diffusion models Song et al. (2020) admit an explicit stochastic-dynamical representation through forward and reverse stochastic differential equations (SDEs) Øksendal (2003). This intrinsic structure enables the learned distribution to be propagated through the reverse dynamics and transformed into a verification-oriented stochastic model, establishing a principled connection between data-driven uncertainty modeling and formal probabilistic safety verification.

Motivated by these observations, we propose a diffusion-induced framework for formal probabilistic safety verification of stochastic systems. Our approach learns state-dependent transition uncertainty

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

from observed trajectories and introduces reverse-polynomial moment propagation to transform the learned reverse diffusion dynamics into an explicit verification-oriented SDE while preserving the known nominal dynamics. The resulting stochastic model is then integrated with stochastic barrier certificate synthesis to establish certified lower bounds on infinite-horizon safety probabilities.

The main contributions of this paper are summarized as follows.

- We propose a conditional diffusion model to capture unknown uncertainty from observed trajectories while preserving the known nominal dynamics.
- We develop reverse-polynomial moment propagation to bridge learned diffusion processes with neural stochastic barrier certificates based probabilistic safety verification.
- We implement a prototype tool and demonstrate effective and efficient probabilistic safety verification across a range of stochastic benchmarks.

2 Problem Statement

Notations. Let $\mathbb{R}$ and $\mathbb{N}$ denote the sets of real and natural numbers, respectively. Let $\mathbf{x} = [x_1, \ldots, x_n]^\top \in \mathbb{R}^n$ denote the system state, and $\|\cdot\|$ the Euclidean norm. We use $\mathbb{R}[\mathbf{x}]$ and $\mathbb{R}[\mathbf{x}]_d$ to denote the spaces of multivariate real polynomials and polynomials of degree at most d, respectively. The set of sum-of-squares (SOS) polynomials is denoted by $\Sigma[\mathbf{x}]$. For a mapping $f : X \to Y$, $C^k(X, \mathbb{R}^m)$ denotes the space of k-times continuously differentiable functions. And $\mathrm{Tr}(\cdot)$, $P(\cdot)$, and $\mathbb{E}[\cdot]$ denote the trace, probability distribution, and expectation operator, respectively.

We consider stochastic dynamical systems described by the following Itô stochastic differential equation (SDE):

$$d\mathbf{x}_t = f(\mathbf{x}_t)\, dt + g(\mathbf{x}_t)\, d\mathbf{W}_t, \tag{1}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ is the system state, f denotes the drift dynamics, g is the diffusion coefficient, and $\mathbf{W}_t$ is an m-dimensional standard Wiener process. The drift term characterizes the nominal system evolution, while the diffusion term captures stochastic uncertainty affecting the system dynamics. Let $\Psi, \Theta, \Xi \subseteq \mathbb{R}^n$ denote the state domain, initial set, and unsafe set, respectively. We assume that they are compact semi-algebraic sets defined by polynomial inequalities:

$$\Psi := \{\mathbf{x} \mid \psi_j(\mathbf{x}) \geq 0,\ j \in [\kappa]\}, \quad \Theta := \{\mathbf{x} \mid \theta_i(\mathbf{x}) \geq 0,\ i \in [\iota]\}, \quad \Xi := \{\mathbf{x} \mid \xi_k(\mathbf{x}) \geq 0,\ k \in [r]\}.$$

where $\psi_j, \theta_i, \xi_k \in \mathbb{R}[\mathbf{x}]$. The safety property is defined with respect to the avoidance of the unsafe set Ξ. For stochastic case, we can give a formal definition of probabilistic safety as follows.

Definition 1 (Probabilistic Safety) *For the stochastic system (1), define the infinite-horizon safety probability as* $P_{\mathrm{safe}}(\mathbf{x}(0)) := \mathbb{P}\{\forall t \geq 0,\ \mathbf{x}(t) \notin \Xi\}$ *. The system is said to be probabilistically safe with failure upper bound* $\rho \in [0, 1]$ *if, for every* $\mathbf{x}(0) \in \Theta$, $\mathbb{P}\{\exists t \geq 0 : \mathbf{x}(t) \in \Xi\} \leq \rho$. *Equivalently, it admits a safety probability lower bound* $1 - \rho$, *i.e.,*

$$P_{\mathrm{safe}}(\mathbf{x}(0)) \geq 1 - \rho, \qquad \forall\, \mathbf{x}(0) \in \Theta. \tag{2}$$

Barrier certificates establish infinite-time safety by constructing a scalar function that separates the initial and unsafe regions while satisfying appropriate conditions along the system dynamics. For continuous stochastic systems, this principle leads to stochastic barrier certificates (SBCs), which constrain the infinitesimal generator of the barrier function to provide probabilistic safety guarantees Prajna et al. (2007), as formalized in Definition 2.

Definition 2 (Stochastic Barrier Certificate) *Consider the stochastic system with state domain* Ψ, *initial set* Θ, *and unsafe set* Ξ. *A barrier certificate for safety with respect to* Ξ *is a function* $B : \Psi \to \mathbb{R}$ *with* $B \in C^2(\Psi)$ *such that there exists a constant* $\rho \in [0, 1]$ *satisfying*

$$\begin{cases} B(\mathbf{x}) \leq \rho\ (\mathbf{x} \in \Theta), \qquad B(\mathbf{x}) \geq 1\ (\mathbf{x} \in \Xi), \qquad B(\mathbf{x}) \geq 0\ (\mathbf{x} \in \Psi), \\ \mathcal{A}B(\mathbf{x}) := \dfrac{\partial B}{\partial \mathbf{x}} f(\mathbf{x}) + \dfrac{1}{2}\mathrm{Tr}\left(g(\mathbf{x})^T \dfrac{\partial^2 B}{\partial \mathbf{x}^2} g(\mathbf{x}) \right) \leq 0, \qquad \mathbf{x} \in \Psi. \end{cases}$$

Theorem 1 *Given the stochastic dynamical system (1), if there exists an SBC* $B \in C^2(\Psi)$ *satisfying Definition 2, then the probabilistic safety bound (2) holds.*

Proof. By Definition 2, $B(\mathbf{x}) \geq 0$ and its infinitesimal generator Øksendal (2003) satisfies $\mathcal{A}B(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \Psi$. Hence, $B(\mathbf{x}(t))$ is a nonnegative supermartingale. Since $B(\mathbf{x}) \geq 1$ on Ξ, the supermartingale inequality Rogers & Williams (2000) yields, for any $\mathbf{x}(0) \in \Theta$, $\mathbb{P}\{\exists t \geq 0 : \mathbf{x}(t) \in \Xi\} \leq B(\mathbf{x}(0)) \leq \rho$, where the last inequality follows from $B(\mathbf{x}) \leq \rho$ on Θ. Therefore, $P_{\text{safe}}(\mathbf{x}(0)) \geq 1 - \rho$ for all $\mathbf{x}(0) \in \Theta$, which establishes (2). □

While Theorem 1 assumes a specified stochastic dynamics model, accurately prescribing the stochastic effects induced by real-world system interactions is often infeasible. We therefore adopt a data-driven formulation that preserves the known nominal dynamics while characterizing the unknown stochasticity from observed system trajectories.

Assumption 1 *Let $\Psi_{\Delta t} : \mathcal{X} \to \mathcal{X}$ denote the one-step transition map induced by the known nominal dynamics $f(\mathbf{x})$ in (1) over the sampling interval Δt. The actual system evolves according to $\mathbf{x}_{t+1} = \Psi_{\Delta t}(\mathbf{x}_t) + \boldsymbol{\delta}_t$, where $\boldsymbol{\delta}_t \in \mathbb{R}^n$ denotes an trajectory-induced stochastic residual. Let $\mathcal{H}_t = (\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t)$ denote the observable trajectory history. We assume that the conditional law of the residual is determined by the current state and the information contained in $\mathcal{H}_t$, i.e., there exists an unknown conditional probability kernel $P^\star_\Delta(\cdot \mid \mathbf{x}_t, \mathcal{H}_t)$ such that $\boldsymbol{\delta}_t \mid (\mathbf{x}_t, \mathcal{H}_t) \sim P^\star_\Delta(\cdot \mid \mathbf{x}_t, \mathcal{H}_t)$. The kernel $P^\star_\Delta$ is unknown and need not admit an analytical representation. A finite trajectory $\{\mathbf{x}_t\}_{t=0}^N$ generated by $f(\mathbf{x})$ is available, from which the residual observations $\boldsymbol{\delta}_t = \mathbf{x}_{t+1} - \Psi_{\Delta t}(\mathbf{x}_t), t = 0, \ldots, N-1$, can be constructed.*

Based on the above formulation, we state the verification problem considered in this work as follows.

Problem. Given a known nominal dynamics, the trajectory observations satisfying Assumption 1, and sets Ψ, Θ, and Ξ, construct a verification-oriented stochastic model as (1) and synthesize an SBC satisfying Definition 2 to certify the probabilistic safety lower bound in (2).

3 Main Method

In this section, we present a diffusion-induced probabilistic safety verification framework named *DiffSBC*. As illustrated in Fig. 1, the framework consists of three main components: **(i) trajectory-conditioned diffusion learning**, which learns the conditional distribution of interaction-induced transition residuals from observed system trajectories while preserving the known nominal dynamics; **(ii) score-induced stochastic dynamics construction**, which propagates residual moments through the reverse diffusion process and transforms the learned uncertainty into a verification-oriented SDE; and **(iii) stochastic barrier certificate synthesis**, which integrates neural candidate learning with SOS-based formal certification to establish probabilistic safety guarantees.

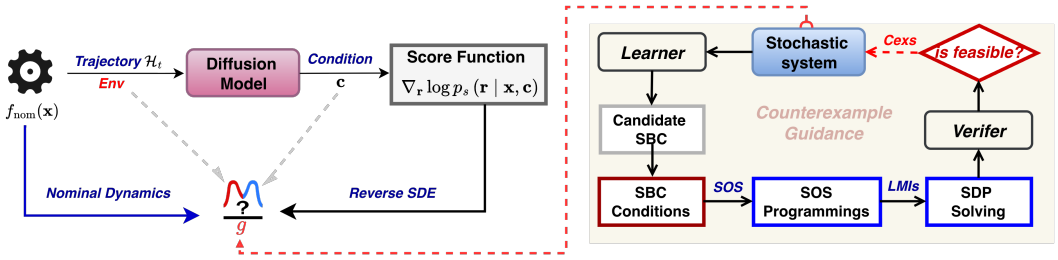


Figure 1: Overview of the proposed *DiffSBC* framework.

3.1 Trajectory-Conditioned Diffusion Learning

Under Assumption 1, each observed transition induces a residual $\boldsymbol{\delta}_t = \mathbf{x}_{t+1} - \Psi_{\Delta t}(\mathbf{x}_t)$, representing the stochastic deviation from the nominal evolution. Here, we characterize its conditional distribution using a score-based diffusion model. To capture its dependence on recent system evolution, we encode the trajectory history $\mathcal{H}_t = (\mathbf{x}_{t-T+1}, \ldots, \mathbf{x}_t)$ as $\mathbf{c}_t = E_\phi(\mathcal{H}_t)$, where E_ϕ denotes a learnable trajectory encoder. We then learn $p_\theta(\boldsymbol{\delta} \mid \mathbf{x}, \mathbf{c})$, conditioned jointly on the current state $\mathbf{x}_t$ and the encoded history $\mathbf{c}_t$, from the dataset $\mathcal{D} = \{(\mathbf{x}_t, \mathbf{c}_t, \boldsymbol{\delta}_t)\}_{t=0}^{N-1}$. The learned distribution p_θ provides a diffusion-based approximation of the unknown conditional residual distribution.

3.1.1 FORWARD DIFFUSION PROCESS

Given a residual sample $\boldsymbol{\delta}_t$ and its condition $(\mathbf{x}_t, \mathbf{c}_t)$, we define the clean diffusion variable as $\mathbf{r}_0 = \boldsymbol{\delta}_t$ and construct a forward noising process over diffusion time $s \in [0, 1]$. Specifically, conditioned on $(\mathbf{x}_t, \mathbf{c}_t)$, the residual variable $\mathbf{r}_s$ evolves according to the variance-preserving (VP) SDE

$$d\mathbf{r}_s = -\frac{1}{2}\beta(s)\mathbf{r}_s\, ds + \sqrt{\beta(s)}\, d\mathbf{W}_s, \tag{3}$$

where $\beta(s) > 0$ denotes the noise schedule and $\mathbf{W}_s$ is a standard Wiener process in diffusion time. The corresponding transition kernel admits the closed-form reparameterization

$$\mathbf{r}_s = \alpha(s)\mathbf{r}_0 + \sigma(s)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

where $\alpha(s) = \exp\big(-\frac{1}{2}\int_0^s \beta(\tau)\, d\tau\big)$ and $\sigma^2(s) = 1 - \alpha^2(s)$. Equivalently,

$$p(\mathbf{r}_s \mid \mathbf{r}_0) = \mathcal{N}\big(\alpha(s)\mathbf{r}_0,\, \sigma^2(s)\mathbf{I}\big)\,.$$

As s approaches the terminal diffusion time, the conditional marginal $p_s(\mathbf{r}_s \mid \mathbf{x}_t, \mathbf{c}_t)$ approaches a standard Gaussian distribution.

3.1.2 CONDITIONAL SCORE ESTIMATION

For the perturbed residual distribution $p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})$, we estimate the conditional score

$$\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}), \tag{5}$$

which governs the reverse-time dynamics for recovering the conditional residual distribution. We adopt the noise-prediction parameterization and train $\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})$ by minimizing

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{\substack{(\mathbf{x},\mathbf{c},\mathbf{r}_0)\sim\mathcal{D},\\ s\sim\mathcal{U}(0,1),\ \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})\|_2^2\right], \tag{6}$$

where $\mathbf{r}_s = \alpha(s)\mathbf{r}_0 + \sigma(s)\boldsymbol{\epsilon}$ as defined in (4). The conditional score is then estimated as

$$\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}) \approx -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})}{\sigma(s)}, \tag{7}$$

which characterizes the residual distribution throughout diffusion time and subsequently determines the reverse-time stochastic dynamics developed next.

3.2 SCORE-INDUCED STOCHASTIC DYNAMICS

A key observation from the score-based diffusion model learned in Sec. 3.1 is that it provides not only a generative model of the residual distribution, but also an explicit stochastic-dynamical representation through its reverse-time SDE. This suggests a natural connection to stochastic formal verification, which likewise operates on stochastic differential dynamics as in (1). However, the two SDEs cannot be directly identified, since reverse diffusion evolves the residual in artificial diffusion time, whereas the system SDE evolves the state in physical time.

We therefore bridge these two stochastic representations through the moments of the learned residual distribution in this section. By propagating these moments through the reverse diffusion dynamics, we integrate the learned stochasticity with the known nominal dynamics to construct a verification-oriented SDE for subsequent probabilistic safety verification.

3.2.1 REVERSE-TIME STOCHASTIC DYNAMICS

Given the conditional score in (5), the forward VP-SDE from (3) induces the reverse-time SDE

$$d\mathbf{r}_s = \left[-\frac{1}{2}\beta(s)\mathbf{r}_s - \beta(s)\nabla_{\mathbf{r}_s} \log p_s\left(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}\right)\right] ds + \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s, \tag{8}$$

where $\overline{\mathbf{W}}_s$ denotes a reverse-time Wiener process and the process evolves from $s = 1$ to $s = 0$ Song et al. (2020). The derivation for the conditional VP-SDE is provided in Appendix A.2.2.

For numerical computation, we substitute the noise-prediction parameterization in (7) into (8), so we can yield $d\mathbf{r}_s = \left[-\frac{1}{2}\beta(s)\mathbf{r}_s + \frac{\beta(s)}{\sigma(s)}\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})\right] ds + \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s$, and then we discretize the resulting reverse process into K steps. Let $\mathbf{r}_k$ denote the latent residual at the k-th diffusion step, with $\mathbf{r}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{r}_0$ representing the reconstructed transition residual. The resulting reverse update can be expressed in the generic form

$$\mathbf{r}_{k-1} = a_k \mathbf{r}_k + b_k \boldsymbol{\epsilon}_\theta\left(\mathbf{r}_k, k, \mathbf{x}, \mathbf{c}\right) + \eta_k \mathbf{z}_k, \qquad \mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{9}$$

where a_k, b_k, and η_k are determined by the noise schedule and reverse discretization. This representation explicitly relates each reverse step to the learned noise predictor and provides the form required for subsequent moment propagation.

3.2.2 Reverse-Polynomial Moment Propagation

The reverse process in (9) is conventionally used to generate samples by numerically evolving individual trajectories from the Gaussian prior to the learned data distribution Song et al. (2020). For stochastic verification, however, our objective is to characterize the conditional moments of the learned residual distribution required for constructing the subsequent verification-oriented stochastic model, rather than to generate individual residual realizations.

Direct moment propagation is nevertheless difficult because the reverse update depends nonlinearly on the neural noise predictor $\boldsymbol{\epsilon}_\theta$. To see this, consider a scalar residual component. Taking the expectation of (9) gives

$$\mu_{k-1} = a_k \mu_k + b_k \mathbb{E}\left[\epsilon_\theta(r_k, k, \mathbf{x}, \mathbf{c})\right],$$

where the Gaussian innovation has zero mean. In general, $\mathbb{E}[\epsilon_\theta(r_k, k, \mathbf{x}, \mathbf{c})]$ cannot be expressed using a finite set of moments of r_k because ϵ_θ is a nonlinear neural network.

To address this difficulty, we develop reverse-polynomial moment propagation (RPM), which replaces the nonlinear neural dependence in the reverse dynamics with a local polynomial representation and propagates the resulting moments through the polynomial stochastic recursion. Specifically, for each conditioning pair $(\mathbf{x}, \mathbf{c})$ and reverse step k, we approximate the dependence of the learned noise predictor on the diffusion latent variable by a quadratic polynomial:

$$\epsilon_\theta\left(r_k, k, \mathbf{x}, \mathbf{c}\right) \approx q_{0,k} + q_{1,k} r_k + q_{2,k} r_k^2,$$

where $q_{0,k}$, $q_{1,k}$, and $q_{2,k}$ are locally fitted for the given $(\mathbf{x}, \mathbf{c}, k)$. Substituting this approximation into (9) yields

$$r_{k-1} = h_{0,k} + h_{1,k} r_k + h_{2,k} r_k^2 + \eta_k z_k, \qquad z_k \sim \mathcal{N}(0, 1), \tag{10}$$

where $h_{0,k} = b_k q_{0,k}$, $h_{1,k} = a_k + b_k q_{1,k}$, and $h_{2,k} = b_k q_{2,k}$. The original neural reverse dynamics are thus transformed into a polynomial stochastic recursion whose moment propagation can be characterized explicitly.

To characterize this dependence more generally, consider a degree-d polynomial reverse update

$$r_{k-1} = \sum_{\ell=0}^{d} h_{\ell,k} r_k^\ell + \eta_k z_k, \qquad z_k \sim \mathcal{N}(0, 1), \tag{11}$$

and define the n-th raw moment as $m_{n,k} := \mathbb{E}[r_k^n], \qquad m_{0,k} := 1.$

Proposition 1 *For the polynomial reverse dynamics in (11), assume that z_k is independent of r_k. Then, for any integer $n \geq 1$, the n-th raw moment at reverse step $k-1$ satisfies*

$$m_{n,k-1} = \sum_{j=0}^{n} \binom{n}{j} \eta_k^j \mathbb{E}[z_k^j]\, \mathbb{E}\left[\left(\sum_{\ell=0}^{d} h_{\ell,k} r_k^\ell\right)^{n-j}\right]. \tag{12}$$

Consequently, $m_{n,k-1}$ depends only on raw moments of r_k up to order dn.

Proof. Taking the n-th power of (11) and applying the binomial theorem gives

$$r_{k-1}^n = \sum_{j=0}^{n} \binom{n}{j} \left(\sum_{\ell=0}^{d} h_{\ell,k} r_k^\ell\right)^{n-j} (\eta_k z_k)^j.$$

Taking expectations and using the independence of r_k and z_k yields (12). Since $\sum_{\ell=0}^{d} h_{\ell,k} r_k^{\ell}$ has degree d, its $(n-j)$-th power has degree at most $d(n-j) \le dn$ in r_k. Therefore, evaluating $m_{n,k-1}$ requires raw moments of r_k of order at most dn. □

For the quadratic approximation in (10), Proposition 1 with $d = 2$ gives

$$m_{n,k-1} = \mathcal{M}_{n,k}\left(m_{0,k}, m_{1,k}, \ldots, m_{2n,k}\right), \tag{13}$$

where $\mathcal{M}_{n,k}$ denotes the polynomial moment map induced by $h_{0,k}$, $h_{1,k}$, $h_{2,k}$, and η_k. Hence, exact propagation of an n-th order moment generally requires moments up to order $2n$ at the preceding reverse step. Consequently, exact recursive propagation requires moments of progressively increasing orders; after ℓ reverse steps, moments of order up to $2^{\ell} n$ may be required. Thus, the quadratic reverse dynamics induce an unclosed moment hierarchy, precluding an exact finite-dimensional recursion over a fixed set of moments.

For stochastic verification, we retain the first two moments of the residual distribution. Setting $n = 1$ in (13) shows that propagation of the mean requires moments up to order two, whereas setting $n = 2$ shows that propagation of the second raw moment requires moments up to order four. Let $\mu_k := \mathbb{E}[r_k] = m_{1,k}$, and $v_k := \mathrm{Var}[r_k]$, the first two raw moments are given exactly by

$$m_{1,k} = \mu_k, \qquad m_{2,k} = \mu_k^2 + v_k. \tag{14}$$

The propagation of the second raw moment additionally requires $m_{3,k}$ and $m_{4,k}$. Propagating these higher-order moments exactly would, according to Proposition 1, require moments of still higher orders and therefore preserve the unclosed hierarchy. To obtain a finite-dimensional recursion in terms of (μ_k, v_k), we adopt Gaussian moment closure and approximate the required third- and fourth-order moments as

$$m_{3,k} \approx \mu_k^3 + 3\mu_k v_k, \qquad m_{4,k} \approx \mu_k^4 + 6\mu_k^2 v_k + 3v_k^2. \tag{15}$$

The approximation error of the aforementioned moment propagation is analyzed in Appendix A.2.5.

With the moment hierarchy closed as above, we now derive the explicit recursions for the first two moments used in RPM. Using (14), the first-order moment recursion obtained from (10) is $\mu_{k-1} = h_{0,k} + h_{1,k} m_{1,k} + h_{2,k} m_{2,k}$. For the second raw moment $m_{2,k-1} := \mathbb{E}[r_{k-1}^2]$, expanding (10) and using $\mathbb{E}[z_k] = 0$, $\mathbb{E}[z_k^2] = 1$, and the independence of z_k and r_k gives

$$m_{2,k-1} = h_{0,k}^2 + 2h_{0,k}h_{1,k}m_{1,k} + \left(h_{1,k}^2 + 2h_{0,k}h_{2,k}\right) m_{2,k} + 2h_{1,k}h_{2,k}m_{3,k} + h_{2,k}^2 m_{4,k} + \eta_k^2. \tag{16}$$

The variance is then recovered from $v_{k-1} = m_{2,k-1} - \mu_{k-1}^2$, yielding, together with (15), a closed recursion over the first two moments. Starting from the standard Gaussian prior $(\mu_K, v_K) = (0, 1)$, the recursion is evaluated from $k = K$ to $k = 1$, yielding the approximate conditional residual moments (μ_0, v_0) used to construct the verification-oriented stochastic dynamics.

3.2.3 Verification-Oriented Diffusion Construction

RPM yields the conditional moments of the learned transition residual at $s = 0$. In the multivariate setting, we denote the resulting conditional residual covariance by $\mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x}, \mathbf{c})$. Since this covariance characterizes the stochastic state deviation over the physical sampling interval Δt, we convert it into the continuous-time covariance rate $Q_\theta(\mathbf{x}, \mathbf{c}) = \frac{\mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x}, \mathbf{c})}{\Delta t}$. This follows from the local covariance relation of an Itô process,

$$\mathrm{Cov}[\Delta\mathbf{x}_t \mid \mathbf{x}_t, \mathbf{c}_t] \approx G(\mathbf{x}_t, \mathbf{c}_t) G(\mathbf{x}_t, \mathbf{c}_t)^{\top} \Delta t,$$

so that $Q_\theta(\mathbf{x}, \mathbf{c}) \approx G(\mathbf{x}, \mathbf{c}) G(\mathbf{x}, \mathbf{c})^{\top}$. The derivation of the resulting diffusion-induced stochastic dynamics is provided in Appendix A.3.

Accordingly, let $G_\theta(\mathbf{x}, \mathbf{c})$ be a diffusion factor satisfying $G_\theta(\mathbf{x}, \mathbf{c}) G_\theta(\mathbf{x}, \mathbf{c})^{\top} = Q_\theta(\mathbf{x}, \mathbf{c})$. The learned residual uncertainty then induces the physical-time stochastic dynamics $d\mathbf{x}_t = f_{\mathrm{nom}}(\mathbf{x}_t)\, dt + G_\theta(\mathbf{x}_t, \mathbf{c}_t)\, d\mathbf{W}_t$. Thus, through RPM and the covariance-rate conversion, the uncertainty represented by the learned reverse diffusion dynamics is mapped into an explicit physical-time stochastic system model. For tractable SOS-based certification, we further construct a constant verification proxy for the state- and context-dependent covariance rate. In our implementation, stochasticity is modeled along a selected state coordinate $\mathbf{e}_j$, yielding the rank-one representation

$\overline{Q}_{\text{diff}} = \lambda_{\text{diff}} \mathbf{e}_j \mathbf{e}_j^\top$, and $\overline{G}_{\text{diff}} = \sqrt{\lambda_{\text{diff}}}\, \mathbf{e}_j$, where λ_{diff} specifies the diffusion rate used for verification. The resulting verification-oriented SDE is

$$d\mathbf{x}_t = f_{\text{nom}}(\mathbf{x}_t)\, dt + \overline{G}_{\text{diff}}\, dW_t, \tag{17}$$

where W_t is a one-dimensional standard Wiener process in physical time. For a barrier function B, the resulting diffusion contributes $\frac{1}{2} \operatorname{Tr}\left(\overline{Q}_{\text{diff}} \nabla^2 B(\mathbf{x})\right) = \frac{1}{2} \lambda_{\text{diff}} \mathbf{e}_j^\top \nabla^2 B(\mathbf{x}) \mathbf{e}_j$ to its infinitesimal generator, which can be directly incorporated into the subsequent SOS-based SBC synthesis.

3.3 Neural Stochastic Barrier Certificate Synthesis

Given the verification-oriented SDE in (17), we synthesize stochastic barrier certificates (SBCs) through a verification-in-the-loop framework. A neural network first constructs a polynomial SBC candidate, which is formally certified through SOS programming while optimizing the failure bound. Failed certification generates counterexamples (Cexs) for candidate refinement, forming an iterative learning–verification loop.

3.3.1 Neural SBC Learning

Let $B_\theta(\mathbf{x})$ denote a neural SBC parameterized by θ. To enable exact SOS verification, we employ quadratic activation functions so that $B_\theta(\mathbf{x})$ is a polynomial. Instead of treating the failure bound ρ as a predefined parameter, we jointly optimize the certificate and the smallest admissible failure bound by solving

$$\min_{\theta,\rho} \ \mathcal{L}(\theta, \rho), \qquad 0 \le \rho \le 1, \tag{18}$$

where $\mathcal{L} = \lambda_\Theta \mathcal{L}_\Theta + \lambda_\Xi \mathcal{L}_\Xi + \lambda_\Psi \mathcal{L}_\Psi + \lambda_\mathcal{A} \mathcal{L}_\mathcal{A} + \lambda_\rho \rho$. The individual loss terms are defined as

$$\begin{cases} \mathcal{L}_\Theta = \dfrac{1}{N_\Theta} \displaystyle\sum_{\mathbf{x}_i \in \Theta} [B_\theta(\mathbf{x}_i) - \rho]_+, & \mathcal{L}_\Xi = \dfrac{1}{N_\Xi} \displaystyle\sum_{\mathbf{x}_i \in \Xi} [1 - B_\theta(\mathbf{x}_i)]_+, \\ \mathcal{L}_\Psi = \dfrac{1}{N_\Psi} \displaystyle\sum_{\mathbf{x}_i \in \Psi} [-B_\theta(\mathbf{x}_i)]_+, & \mathcal{L}_\mathcal{A} = \dfrac{1}{N_\Psi} \displaystyle\sum_{\mathbf{x}_i \in \Psi} [\mathcal{A}B_\theta(\mathbf{x}_i)]_+. \end{cases}$$

Minimizing (18) simultaneously reduces the certificate violations and the failure bound, thereby maximizing the certified lower bound of the safety probability. Since the optimization relies on finitely sampled states, the obtained polynomial candidate must be further certified through SOS relaxation approach Prajna et al. (2007).

3.3.2 SOS-Based Posterior Certification

Since neural learning enforces the SBCs conditions only over finitely sampled states, the resulting candidate does not provide a formal guarantee over the entire verification domain. We therefore fix the learned candidate $B(\mathbf{x}) = B_{\theta^\star}(\mathbf{x})$ and certify its validity through SOS programming.

Theorem 2 (Putinar's Positivstellensatz Putinar (1993)) *Let $S = \{\mathbf{x} \in \mathbb{R}^n \mid p_i(\mathbf{x}) \ge 0,\ i = 1, \dots, m\}$ be a compact basic semi-algebraic set whose quadratic module is Archimedean. If a polynomial $h(\mathbf{x})$ is strictly positive on S, then there exist SOS polynomials σ_i such that $h(\mathbf{x}) = \sigma_0(\mathbf{x}) + \sum_{i=1}^m \sigma_i(\mathbf{x}) p_i(\mathbf{x})$. This result enables polynomial positivity over semi-algebraic sets to be certified through SOS representations.*

Following Theorem 2, we impose SOS certificates for the four SBC conditions over their corresponding semi-algebraic sets and solve

$$\begin{cases} \min\limits_{\rho,\tau,\delta,\sigma,\phi} & \rho \\ \quad \text{s.t.} & \rho - B(\mathbf{x}) - \displaystyle\sum_i \tau_i(\mathbf{x}) \theta_i(\mathbf{x}) \in \Sigma, \quad B(\mathbf{x}) - 1 - \sum_i \delta_i(\mathbf{x}) \xi_i(\mathbf{x}) \in \Sigma, \\ & B(\mathbf{x}) - \displaystyle\sum_i \sigma_i(\mathbf{x}) \psi_i(\mathbf{x}) \in \Sigma, \qquad -\mathcal{A}B(\mathbf{x}) - \sum_i \phi_i(\mathbf{x}) \psi_i(\mathbf{x}) \in \Sigma, \\ & \tau_i, \delta_i, \sigma_i, \phi_i \in \Sigma, \qquad 0 \le \rho \le 1. \end{cases} \tag{19}$$

For the fixed candidate $B_{\theta^\star}$, the SOS program searches for valid multipliers while minimizing the failure bound. If (19) is feasible, the SBC conditions are formally certified over their respective

domains, and the resulting $\rho^\star$ yields the certified infinite-horizon safety lower bound $1-\rho^\star$. Otherwise, counterexamples exhibiting violations of the SBC conditions are generated following the CEGIS framework Solar-Lezama et al. (2006) and added to the training set to refine the candidate. The refined candidate is then recertified, forming the iterative learning–verification loop. More details are provided in Appendix A.4.

4 EXPERIMENTS

We conduct a series of experiments to evaluate the effectiveness of the proposed *DiffSBC*. In particular, we compare it with conventional approaches using manually specified disturbance models Mathiesen et al. (2022) and recent generative-model-based approach that estimates stochastic uncertainty from empirical sample statistics Hu et al. (2026). We first present the experimental protocol and an end-to-end case study, followed by a systematic evaluation on a diverse set of nonlinear dynamical systems with varying dimensions and dynamics. All experiments are conducted on a Windows 11 machine equipped with 32 GB RAM and an AMD Ryzen 9 8945HX CPU running at 2.50 GHz.

4.1 CASE STUDY

Example 1. Klipp et al. (2005) We first consider the following challenging 7-dimensional case whose nominal dynamics are given by

$$\begin{cases} \dot{x}_1 = -0.4x_1 + 5x_3x_4, & \dot{x}_2 = 0.4x_1 - x_2, & \dot{x}_3 = x_2 - 5x_3x_4, & \dot{x}_4 = 5x_5x_6 - 5x_3x_4, \\ \dot{x}_5 = -5x_5x_6 + 5x_3x_4, & \dot{x}_6 = 0.5x_7 - 5x_5x_6, & \dot{x}_7 = -0.5x_7 + 5x_5x_6. \end{cases}$$

with the state space $\Psi = [-2, 2]^7$. The initial region is defined as $\Theta = [-1.01, -0.99]^7$ and unsafe region is $\Xi = [1.8, 2]^7$. We first generate 64 trajectories of horizon 40 with sampling interval $\Delta t = 0.05$, discarding transitions that reach the state-space boundary. A one-dimensional conditional diffusion model is then trained using a cosine schedule with 32 reverse steps, taking 4.85 seconds. Finally, reverse-polynomial moment propagation is evaluated over 580 state probes to construct the fixed covariance proxy $\bar{G}_{\text{diff}} = \text{diag}(0, 0, 0, 0, 0, 0, 0.032581)$. Consequently, the continuous-time proxy verified by the main method is $d\mathbf{x} = f(\mathbf{x})dt + 0.180501\, e_7\, dW$, where e_7 is the 7-th Euclidean basis vector. Next, SOS selects degree $d_B = 2$ with relaxation degree to synthesize a neural SBC as follows and shown in Fig. 2:

$$\begin{aligned} B(\mathbf{x}) = {} & 0.127680 + 0.049029(x_1 + x_2 + x_3) + 0.053284(x_4 + x_5) + 0.000853(x_6 + x_7) \\ & + 0.006999(x_1^2 + x_2^2 + x_3^2) + 0.013998(x_1x_2 + x_1x_3 + x_2x_3) + 0.003358(x_1 + x_2 + x_3)(x_4 + x_5) \\ & + 0.000160(x_1 + x_2 + x_3)(x_6 + x_7) + 0.010712(x_4^2 + x_5^2) + 0.021423x_4x_5 \\ & + 0.000182(x_4 + x_5)(x_6 + x_7) + 0.000002(x_6^2 + x_7^2) + 0.000004x_6x_7. \end{aligned}$$

The SOS optimization returns $\rho_{\text{raw}} = 0.00001236$. To account for numerical solver tolerances, we report the conservative failure bound $\rho = \min\{1, \rho_{\text{raw}} + 10^{-3}\} = 0.001012$, which yields the certified infinite-horizon safety lower bound $1 - \rho = 0.998988$.

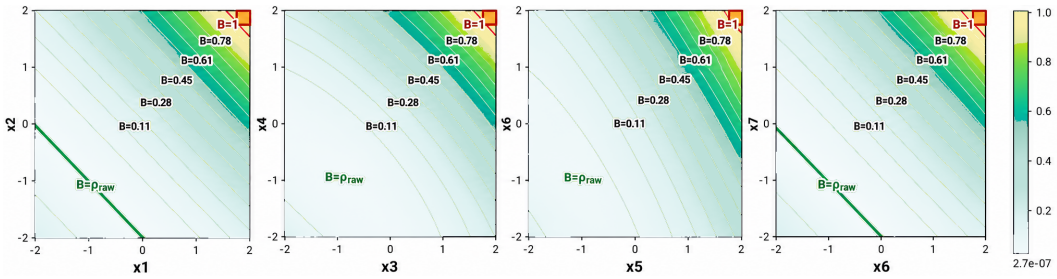


Figure 2: The Phase diagram of SBC on Example 1.

4.2 PERFORMANCE EVALUATION

We evaluate the proposed method against two representative uncertainty-modeling baselines on all benchmark examples, with detailed benchmark specifications and baseline formulations provided in

Appendix A.9. *Additive+Neural* Mathiesen et al. (2022) models the learned uncertainty as a fixed additive disturbance and performs verification on the resulting deterministic system. *Gauss.+c-SBC* Hu et al. (2026) directly fits a Gaussian transition model from sampled successor states and verifies the resulting discrete-time stochastic system using c-SBC conditions. In contrast, *DiffSBC* extracts stochastic transition uncertainty from the learned reverse diffusion dynamics and incorporates the resulting covariance into a continuous-time stochastic abstraction for SBC verification, more details are in Appendix A.8.

Table 1: Algorithm Performance on Benchmark Examples (Time in Seconds)

			Ours			*Additive+Neural*			*Gauss.+c-SBC*		
Ex.	n	d_f	d_B^*	$1-\rho^*$	t	d_B^*	$1-\rho^*$	t	d_B^*	$1-\rho^*$	t
C_1	2	2	6	0.8703	1.24	6	0.8739	11.10	6	0.8855	13.10
C_2	2	3	6	0.9976	1.21	6	0.9593	13.89	6	0.9635	15.85
C_3	2	2	6	0.7186	1.35	6	0.3003	10.39	6	0.3005	12.33
C_4	2	2	6	0.9980	1.29	6	0.9967	11.15	6	0.9981	12.73
C_5	3	3	6	0.9924	8.72	4	0.8130	22.41	6	0.9038	219.98
C_6	7	2	2	0.9990	5.40	2	0.5900	19.58	2	0.9048	21.69
C_7	9	2	2	0.9988	16.22	–	–	–	2	0.8002	298.20
C_8	3	3	4	0.6891	2.22	4	0.5061	19.64	6	0.8174	209.12
C_9	5	3	4	0.6895	23.67	4	0.4134	265.35	4	0.6379	290.34

Note. n and d_f denote the system dimension and the polynomial degree of the nominal dynamics, respectively. For each method, d_B^* denotes the minimum degree of a successfully verified barrier certificate, $1-\rho^*$ the certified infinite-horizon safety lower bound, and t the total certificate synthesis time in seconds. "–" indicates that no valid certificate was obtained.

As shown in Table 1, the proposed framework successfully verifies all examples and consistently achieves the lowest verification time. The efficiency advantage is particularly pronounced on the more computationally challenging examples. For instance, on C_7, our method completes verification in $16.22\,\mathrm{s}$, whereas *Additive+Neural* fails to obtain a valid certificate and *Gauss.+c-SBC* requires $298.20\,\mathrm{s}$. Similar improvements are observed on C_8 and C_9, where the proposed method requires only $2.22\,\mathrm{s}$ and $23.67\,\mathrm{s}$, respectively, compared with $209.12\,\mathrm{s}$ and $290.34\,\mathrm{s}$ for *Gauss.+c-SBC*.

In terms of certified safety, our method generally yields substantially higher safety lower bounds than *Additive+Neural* on the more challenging benchmarks. The improvement is especially evident on C_3, C_5, C_6, C_7, and C_9, indicating that preserving the state-dependent stochastic structure learned by the diffusion model can lead to more informative certificates than reducing the learned uncertainty to an averaged additive disturbance. Compared with *Gauss.+c-SBC*, neither method uniformly dominates the other in terms of the numerical safety bound: *Gauss.+c-SBC* gives slightly higher bounds on C_1, C_4, and C_8, whereas the proposed method yields higher bounds on C_2, C_3, C_5, C_6, C_7, and C_9. Nevertheless, the proposed method consistently requires substantially less verification time, with the difference becoming especially pronounced as the benchmark complexity increases.

The results also show that the proposed method can often obtain valid certificates with lower polynomial degrees. On C_6 and C_7, a quadratic barrier is sufficient; on C_8, our method succeeds with degree 4, compared with degree 6 for *Gauss.+c-SBC*. Together, these results suggest that our diffusion-induced stochastic abstraction provides an effective interface between learned uncertainty and SBC-based formal verification, achieving favorable verification efficiency while maintaining competitive certified safety bounds.

5 Conclusion

This paper presented a diffusion-induced framework for probabilistic safety verification under unknown stochastic uncertainty. By exploiting the reverse stochastic dynamics of score-based diffusion, the proposed approach propagates learned uncertainty into an explicit verification-oriented SDE, which is formally certified using stochastic barrier certificates and Generator-SOS optimization. Experiments on representative benchmarks demonstrate substantially reduced verification cost while maintaining competitive certified safety bounds.

Reproducibility Statement

To facilitate reproducibility, we provide the complete algorithmic procedure in Appendix A.5. Detailed derivations and theoretical analyses are provided in Appendices A.2, A.3, and A.4. Detailed experimental settings, hyperparameters, and numerical verification criteria are reported in Appendix A.6, with complete benchmark specifications provided in Appendix A.9. Implementation details, together with the anonymized code repository containing the source code, scripts, and configuration files required to reproduce the reported results, are available at https://anonymous.4open.science/r/paper_experiment_code_package-1944/README.md.

References

Saurabh Amin, Alessandro Abate, Maria Prandini, John Lygeros, and Shankar Sastry. Reachability analysis for controlled discrete time stochastic hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, pp. 49–63. Springer, 2006.

Feiyang Cai, Chuchu Fan, and Stanley Bak. Scalable surrogate verification of image-based neural network control systems using composition and unrolling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21–30, 2025.

Krishnendu Chatterjee, Thomas A Henzinger, Mathias Lechner, and Đorđe Žikelić. A learner-verifier framework for neural network controllers and certificates of stochastic systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pp. 3–25. Springer, 2023.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10850–10869, 2023.

Godfrey Harold Hardy, John Edensor Littlewood, and George Pólya. *Inequalities*. Cambridge university press, 1952.

Desmond J Higham. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Qianlong Hu, Hanrui Zhao, Liqian Chen, Zhengfeng Yang, Banghu Yin, and Ji Wang. Provably probabilistic safe controller synthesis for vision-based neural network control systems. In *Proceedings of the 63rd ACM/IEEE Design Automation Conference (DAC)*, 2026. URL https://63dac.conference-program.com/presentation/?id=RESEARCH720&sess=sess158. Accepted for publication.

Chao Huang, Xin Chen, Wang Lin, Zhengfeng Yang, and Xuandong Li. Probabilistic safety verification of stochastic hybrid systems using barrier certificates. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(5s):1–19, 2017.

Pushpak Jagtap, Sadegh Soudjani, and Majid Zamani. Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 66(7):3097–3110, 2020.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Sydney M Katz, Anthony L Corso, Christopher A Strong, and Mykel J Kochenderfer. Verification of image-based neural network controllers using generative models. *Journal of Aerospace Information Systems*, 19(9):574–584, 2022.

Edda Klipp, Ralf Herwig, Axel Kowald, Christoph Wierling, and Hans Lehrach. *Systems biology in practice: concepts, implementation and application*. John Wiley & Sons, 2005.

Christian Kuehn. Moment closure—a brief review. *Control of self-organizing nonlinear systems*, pp. 253–271, 2016.

Luca Laurenti and Morteza Lahijanian. A unifying perspective for safety of stochastic systems: From barrier functions to finite abstractions. *IEEE Transactions on Automatic Control*, 2025.

Abolfazl Lavaei, Sadegh Soudjani, Alessandro Abate, and Majid Zamani. Automated verification and synthesis of stochastic hybrid systems: A survey. *Automatica*, 146:110617, 2022.

Matt Luckcuck, Marie Farrell, Louise A Dennis, Clare Dixon, and Michael Fisher. Formal specification and verification of autonomous robotic systems: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–41, 2019.

Xinhang Ma, Junlin Wu, Hussein Sibai, Yiannis Kantaros, and Yevgeniy Vorobeychik. Learning vision-based neural network controllers with semi-probabilistic safety guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35698–35706, 2026.

Xuerong Mao. *Stochastic differential equations and applications*. Elsevier, 2007.

Frederik Baymler Mathiesen, Simeon Calvert, and Luca Laurenti. Safety certification for stochastic systems via neural barrier functions. *arXiv preprint arXiv:2206.01463*, 2022.

Frederik Baymler Mathiesen, Licio Romao, Simeon C Calvert, Luca Laurenti, and Alessandro Abate. A data-driven approach for safety quantification of non-linear stochastic systems with unknown additive noise distribution. *Automatica*, 193:113187, 2026.

Rayan Mazouz, John Skovbekk, Frederik Baymler Mathiesen, Eric Frew, Luca Laurenti, and Morteza Lahijanian. Data-driven permissible safe control with barrier certificates. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 6844–6849. IEEE, 2024.

Noushin Mehdipour, Matthias Althoff, Radboud Duintjer Tebbens, and Calin Belta. Formal methods to comply with rules of the road in autonomous driving: State of the art and grand challenges. *Automatica*, 152:110692, 2023.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, 6 edition, 2003.

Stephen Prajna, Ali Jadbabaie, and George J Pappas. Stochastic safety verification using barrier certificates. In *2004 43rd IEEE conference on decision and control (CDC)(IEEE Cat. No. 04CH37601)*, volume 1, pp. 929–934. IEEE, 2004.

Stephen Prajna, Ali Jadbabaie, and George J Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1428, 2007.

Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

Stefan Ratschan. Simulation based computation of certificates for safety of dynamical systems. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pp. 303–317. Springer, 2017.

Stefan Ratschan and Zhikun She. Providing a basin of attraction to a target region by computation of lyapunov-like functions. In *2006 IEEE International Conference on Computational Cybernetics*, pp. 1–5. IEEE, 2006.

Stefan Ratschan and Zhikun She. Providing a basin of attraction to a target region of polynomial systems by computation of lyapunov-like functions. *SIAM Journal on Control and Optimization*, 48(7):4377–4394, 2010.

Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*. Springer Science & Business Media, 2013.

Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*, volume 2. Cambridge university press, 2000.

Ali Salamati, Abolfazl Lavaei, Sadegh Soudjani, and Majid Zamani. Data-driven verification and synthesis of stochastic systems via barrier certificates. *Automatica*, 159:111323, 2024.

Andrew Sogokon, Khalil Ghorbal, and Taylor T Johnson. Non-linear continuous systems for safety verification (benchmark proposal). In *ARCH@ CPSWeek 2016-3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, volume 43, pp. 42–51. EasyChair, 2016.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Sanjit Seshia, and Vijay Saraswat. Combinatorial sketching for finite programs. In *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pp. 404–415, 2006.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Wei Xiao, Johnson Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, volume 2025, pp. 100391–100412, 2025.

Insoon Yang. A dynamic game approach to distributionally robust safety specifications for stochastic systems. *arXiv preprint arXiv:1701.06260*, 2017.

Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

Hanrui Zhao, Niuniu Qi, Lydia Dehbi, Xia Zeng, and Zhengfeng Yang. Formal synthesis of neural barrier certificates for continuous systems via counterexample guided learning. *ACM Transactions on Embedded Computing Systems*, 22(5s):1–21, 2023.

He Zhu, Zikang Xiong, Stephen Magill, and Suresh Jagannathan. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pp. 686–701, 2019.

Đorđe Žikelić, Mathias Lechner, Thomas A Henzinger, and Krishnendu Chatterjee. Learning control policies for stochastic systems with reach-avoid guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11926–11935, 2023.

A Appendix

A.1 Background and Preliminaries

A.1.1 Itô Stochastic Differential Equations

Stochastic differential equations (SDEs) provide a mathematical framework for modeling dynamical systems affected by random perturbations. Unlike deterministic ordinary differential equations, where the system evolution is uniquely determined by the initial condition, SDEs describe systems whose trajectories are driven by both deterministic dynamics and stochastic processes. In this work, we consider continuous-time stochastic dynamical systems described by the following Itô SDE:

$$d\mathbf{x}_t = f(\mathbf{x}_t)\, dt + g(\mathbf{x}_t)\, d\mathbf{W}_t, \tag{20}$$

where $\mathbf{x}_t \in \mathbb{R}^n$ denotes the system state, $f : \mathbb{R}^n \to \mathbb{R}^n$ is the drift function, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the diffusion coefficient, and $\mathbf{W}_t \in \mathbb{R}^m$ is an m-dimensional standard Wiener process.

The drift term $f(\mathbf{x}_t)$ represents the deterministic evolution of the system, corresponding to the nominal dynamics in the absence of random perturbations. The diffusion term $g(\mathbf{x}_t)d\mathbf{W}_t$ captures the stochastic variation introduced by environmental uncertainty, disturbances, or unmodeled dynamics. The matrix

$$g(\mathbf{x}_t)g(\mathbf{x}_t)^\top$$

characterizes the local covariance rate of the stochastic perturbation and determines how uncertainty propagates along different state dimensions.

The Wiener process $\mathbf{W}_t$ is a continuous-time stochastic process satisfying

$$\mathbf{W}_0 = \mathbf{0},$$

with independent Gaussian increments

$$\mathbf{W}_{t+\Delta t} - \mathbf{W}_t \sim \mathcal{N}(\mathbf{0}, \Delta t\, \mathbf{I}).$$

Therefore, the stochastic increment induced by the diffusion term has zero mean and covariance proportional to the time interval:

$$\mathbb{E}[g(\mathbf{x}_t) d\mathbf{W}_t] = 0,$$

and

$$\mathbb{E}[g(\mathbf{x}_t) d\mathbf{W}_t d\mathbf{W}_t^\top g(\mathbf{x}_t)^\top] = g(\mathbf{x}_t) g(\mathbf{x}_t)^\top dt.$$

This property enables diffusion coefficient to characterize the instantaneous uncertainty intensity of the stochastic system.

For an Itô SDE, the evolution of a sufficiently smooth function $V(\mathbf{x}) \in C^2(\mathbb{R}^n)$ is characterized by Itô's formula:

$$dV(\mathbf{x}_t) = \mathcal{L}V(\mathbf{x}_t)dt + \nabla V(\mathbf{x}_t) g(\mathbf{x}_t) d\mathbf{W}_t, \tag{21}$$

where $\mathcal{L}$ denotes the infinitesimal generator Mao (2007) of the stochastic system:

$$\mathcal{L}V(\mathbf{x}) = \nabla V(\mathbf{x}) f(\mathbf{x}) + \frac{1}{2} \mathrm{Tr}\left(g(\mathbf{x})^\top \nabla^2 V(\mathbf{x}) g(\mathbf{x}) \right). \tag{22}$$

The generator describes the expected infinitesimal change of a function along stochastic trajectories and serves as the fundamental operator for stochastic stability and safety analysis. In particular, stochastic barrier certificates exploit the generator condition to establish supermartingale properties and derive probabilistic safety guarantees.

A.1.2 Stochastic Barrier Certificates

Barrier certificates (BCs) provide a verification-oriented approach for establishing safety guarantees without explicitly computing system trajectories. For deterministic dynamical systems, a valid barrier certificate constructs an invariant separation between the initial and unsafe regions. Stochastic barrier certificates (SBCs) extend this idea to stochastic systems by requiring the barrier function to satisfy a generator condition, which guarantees that the barrier process evolves as a supermartingale.

Consider the stochastic dynamical system

$$d\mathbf{x}_t = f(\mathbf{x}_t)dt + g(\mathbf{x}_t) d\mathbf{W}_t, \tag{23}$$

with state domain Ψ, initial set Θ, and unsafe set Ξ. Let $B : \Psi \to \mathbb{R}$ be a twice continuously differentiable function. The infinitesimal generator of the system applied to B is defined as

$$\mathcal{L}B(\mathbf{x}) = \nabla B(\mathbf{x}) f(\mathbf{x}) + \frac{1}{2} \mathrm{Tr}\left(g(\mathbf{x})^T \nabla^2 B(\mathbf{x}) g(\mathbf{x}) \right). \tag{24}$$

The first term represents the change of the barrier function induced by the deterministic drift, while the second term captures the influence of stochastic perturbations through its curvature.

Definition 3 (Stochastic Barrier Certificate) *Consider the stochastic system (23) with state domain Ψ, initial set Θ, and unsafe set Ξ. A function $B \in C^2(\Psi)$ is a stochastic barrier certificate with respect to Ξ if there exists a constant $\rho \in [0, 1]$ such that*

$$\begin{aligned} &\textit{(i)}\ B(\mathbf{x}) \le \rho, \quad \forall \mathbf{x} \in \Theta, && \textit{(ii)}\ B(\mathbf{x}) \ge 1, \quad \forall \mathbf{x} \in \Xi, \\ &\textit{(iii)}\ B(\mathbf{x}) \ge 0, \quad \forall \mathbf{x} \in \Psi, && \textit{(iv)}\ \mathcal{L}B(\mathbf{x}) \le 0, \quad \forall \mathbf{x} \in \Psi. \end{aligned}$$

The four conditions above jointly establish a probabilistic separation between the initial and unsafe regions. Condition (i) bounds the initial value of the barrier function by ρ. Since the initial state is restricted to Θ, the initial barrier value is guaranteed to be small. Condition (ii) assigns a unit barrier value to the unsafe region. Thus, entering the unsafe set necessarily requires the barrier process to increase from at most ρ to at least one. Condition (iii) ensures that the barrier function remains nonnegative throughout the state domain. This condition allows the barrier function to be

interpreted as a nonnegative stochastic process along system trajectories. Condition (iv) is the key stochastic invariance condition.

According to Itô's formula, the evolution of the barrier process $B(\mathbf{x}_t)$ satisfies

$$dB(\mathbf{x}_t) = \mathcal{L}B(\mathbf{x}_t)dt + \nabla B(\mathbf{x}_t)g(\mathbf{x}_t)d\mathbf{W}_t. \tag{25}$$

When condition (iv) holds, the drift component of $B(\mathbf{x}_t)$ is non-positive. Therefore, $B(\mathbf{x}_t)$ forms a nonnegative supermartingale, i.e.,

$$\mathbb{E}[B(\mathbf{x}_t) \mid \mathcal{F}_s] \leq B(\mathbf{x}_s), \qquad t \geq s, \tag{26}$$

where $\mathcal{F}_s$ denotes the filtration generated by the stochastic process.

A.1.3 Probabilistic Safety Guarantee

The probabilistic safety guarantee follows from the maximal inequality of nonnegative supermartingales. We first state the resulting safety guarantee.

Theorem 3 (Stochastic Barrier Certificate Guarantee) *Given the stochastic system (23), if there exists a stochastic barrier certificate satisfying Definition 3, then the unsafe-reaching probability is bounded by ρ, i.e.,*

$$\mathbb{P}\{\exists t \geq 0, \mathbf{x}_t \in \Xi\} \leq \rho,$$

and the safety probability satisfies

$$P_{\text{safe}}(\mathbf{x}_0) \geq 1 - \rho, \qquad \forall \mathbf{x}_0 \in \Theta.$$

Proof. Define the unsafe hitting time as

$$\tau_\Xi = \inf\{t \geq 0 : \mathbf{x}_t \in \Xi\}. \tag{27}$$

The probability of reaching the unsafe set can be expressed as

$$\mathbb{P}\{\exists t \geq 0, \mathbf{x}_t \in \Xi\} = \mathbb{P}\{\tau_\Xi < \infty\}. \tag{28}$$

According to condition (ii), whenever the trajectory reaches the unsafe region,

$$B(\mathbf{x}_{\tau_\Xi}) \geq 1. \tag{29}$$

Therefore,

$$\{\tau_\Xi < \infty\} \subseteq \{\sup_{t \geq 0} B(\mathbf{x}_t) \geq 1\}. \tag{30}$$

By conditions (iii) and (iv), $B(\mathbf{x}_t)$ is a nonnegative supermartingale. For a nonnegative supermartingale, Doob's maximal inequality Revuz & Yor (2013) gives

$$\mathbb{P}\{\sup_{t \geq 0} B(\mathbf{x}_t) \geq 1\} \leq \mathbb{E}[B(\mathbf{x}_0)]. \tag{31}$$

Since condition (i) guarantees

$$B(\mathbf{x}_0) \leq \rho, \qquad \forall \mathbf{x}_0 \in \Theta, \tag{32}$$

we have

$$\mathbb{P}\{\exists t \geq 0, \mathbf{x}_t \in \Xi\} \leq \rho. \tag{33}$$

Consequently, the safety probability satisfies

$$\begin{aligned} P_{\text{safe}}(\mathbf{x}_0) &= \mathbb{P}\{\forall t \geq 0, \mathbf{x}_t \notin \Xi\} \\ &= 1 - \mathbb{P}\{\exists t \geq 0, \mathbf{x}_t \in \Xi\} \\ &\geq 1 - \rho, \end{aligned} \tag{34}$$

for all initial states $\mathbf{x}_0 \in \Theta$.

Therefore, constructing an SBC satisfying Definition 3 is sufficient to certify an infinite-horizon probabilistic safety guarantee. □

A.2 Theorem Analysis of Reverse SDE and Moment Propagation

Score-based diffusion models provide an explicit stochastic evolution process for transforming a tractable Gaussian prior into a target data distribution. Unlike conventional generative models that only define a sampling mechanism, the underlying SDEs characterize how uncertainty evolves during the generation process. This property enables diffusion models to be transformed into verification-oriented stochastic dynamics.

In this section, we provide the theoretical foundation of the proposed diffusion-induced stochastic abstraction. Specifically, we first introduce the reverse-time SDE associated with score-based diffusion models. Then, we show how the uncertainty encoded by the reverse process can be propagated analytically through moment propagation and converted into an explicit SDE for formal verification.

A.2.1 Reverse-Time Stochastic Differential Equation (SDE)

Consider the conditional residual distribution

$$p_0(\mathbf{r} \mid \mathbf{x}, \mathbf{c}),$$

where $\mathbf{r}$ denotes a state-transition residual and $(\mathbf{x}, \mathbf{c})$ represents the conditioning information. Following the continuous-time formulation of score-based diffusion models Song et al. (2020), the forward noising process can be described by the SDE

$$d\mathbf{r}_s = f_{\text{diff}}(\mathbf{r}_s, s)\, ds + g_{\text{diff}}(s)\, d\mathbf{W}_s, \tag{35}$$

where $s \in [0, 1]$ denotes diffusion time, f_{diff} is the forward drift, and g_{diff} determines the diffusion schedule. Starting from $\mathbf{r}_0 \sim p_0(\mathbf{r} \mid \mathbf{x}, \mathbf{c})$, the forward process progressively perturbs the residual distribution toward a Gaussian prior.

Let $p_s(\mathbf{r} \mid \mathbf{x}, \mathbf{c})$ denote the conditional marginal distribution at diffusion time s. The corresponding reverse-time SDE is Song et al. (2020)

$$\begin{aligned} d\mathbf{r}_s = {} & \left[f_{\text{diff}}(\mathbf{r}_s, s) - g_{\text{diff}}(s)^2 \nabla_{\mathbf{r}} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})\right] ds \\ & + g_{\text{diff}}(s)\, d\overline{\mathbf{W}}_s, \end{aligned} \tag{36}$$

where $\overline{\mathbf{W}}_s$ denotes a reverse-time Wiener process and the SDE is solved from $s = 1$ to $s = 0$. The score

$$\nabla_{\mathbf{r}} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})$$

characterizes the local geometry of the perturbed residual distribution and determines the learned component of the reverse drift.

In this work, we adopt the variance-preserving (VP) SDE, for which

$$f_{\text{diff}}(\mathbf{r}_s, s) = -\frac{1}{2}\beta(s)\mathbf{r}_s, \qquad g_{\text{diff}}(s) = \sqrt{\beta(s)}.$$

Substituting these coefficients into equation 36 gives

$$d\mathbf{r}_s = \left[-\frac{1}{2}\beta(s)\mathbf{r}_s - \beta(s)\nabla_{\mathbf{r}} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})\right] ds + \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s, \tag{37}$$

which is the reverse-time stochastic dynamics used in our method.

A.2.2 Derivation and Discretization of the Conditional Reverse-Time VP-SDE

This section derives the conditional reverse-time VP-SDE used in (8) and its discrete reverse update in (9). Throughout the diffusion process, the conditioning variables $(\mathbf{x}, \mathbf{c})$ are treated as fixed; only the residual variable $\mathbf{r}_s$ evolves with the artificial diffusion time s.

[Forward VP-SDE]. For a residual sample $\mathbf{r}_0 \sim p_0(\mathbf{r} \mid \mathbf{x}, \mathbf{c})$, the variance-preserving (VP) forward diffusion process is defined as

$$d\mathbf{r}_s = -\frac{1}{2}\beta(s)\mathbf{r}_s\, ds + \sqrt{\beta(s)}\, d\mathbf{W}_s, \qquad s \in [0, 1], \tag{38}$$

where $\beta(s) > 0$ denotes the noise schedule and $\mathbf{W}_s$ is a standard Wiener process in diffusion time.

Equation (38) is a linear SDE. Define

$$\alpha(s) = \exp\left(-\frac{1}{2}\int_0^s \beta(u)\,du\right). \tag{39}$$

Its transition distribution conditioned on the clean residual $\mathbf{r}_0$ is Gaussian:

$$p(\mathbf{r}_s \mid \mathbf{r}_0) = \mathcal{N}\left(\alpha(s)\mathbf{r}_0,\, \sigma^2(s)\mathbf{I}\right), \tag{40}$$

where

$$\sigma^2(s) = 1 - \alpha^2(s). \tag{41}$$

Equivalently, the perturbed residual can be represented as

$$\mathbf{r}_s = \alpha(s)\mathbf{r}_0 + \sigma(s)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{42}$$

Since the initial residual follows the conditional distribution $p_0(\mathbf{r}_0 \mid \mathbf{x}, \mathbf{c})$, the corresponding marginal distribution at diffusion time s is

$$p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}) = \int p(\mathbf{r}_s \mid \mathbf{r}_0) p_0(\mathbf{r}_0 \mid \mathbf{x}, \mathbf{c})\, d\mathbf{r}_0. \tag{43}$$

Thus, although the forward transition kernel itself does not depend on $(\mathbf{x}, \mathbf{c})$, the marginal distribution remains conditional on $(\mathbf{x}, \mathbf{c})$ through the initial residual distribution.

[Conditional Reverse-Time VP-SDE]. Consider a general forward diffusion process

$$d\mathbf{r}_s = \mathbf{f}(\mathbf{r}_s, s)\, ds + g(s)\, d\mathbf{W}_s. \tag{44}$$

The corresponding reverse-time SDE takes the form

$$d\mathbf{r}_s = \left[\mathbf{f}(\mathbf{r}_s, s) - g^2(s)\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s)\right] ds + g(s)\, d\overline{\mathbf{W}}_s, \tag{45}$$

where the process is solved backward from $s = 1$ to $s = 0$ and $\overline{\mathbf{W}}_s$ denotes a reverse-time Wiener process Song et al. (2020).

In our conditional setting, $(\mathbf{x}, \mathbf{c})$ remains fixed throughout the diffusion-time evolution. Therefore, for each fixed conditioning pair $(\mathbf{x}, \mathbf{c})$, the same reverse-time result applies to the conditional marginal $p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})$, yielding

$$d\mathbf{r}_s = \left[\mathbf{f}(\mathbf{r}_s, s) - g^2(s)\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c})\right] ds + g(s)\, d\overline{\mathbf{W}}_s. \tag{46}$$

The conditioning variables therefore enter the reverse dynamics through the conditional score

$$\mathbf{s}^\star(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c}) = \nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}), \tag{47}$$

while they do not constitute additional stochastic state variables of the diffusion process.

For the VP-SDE in (38), the forward drift and diffusion coefficient are

$$\mathbf{f}(\mathbf{r}_s, s) = -\frac{1}{2}\beta(s)\mathbf{r}_s, \qquad g(s) = \sqrt{\beta(s)}. \tag{48}$$

Substituting (48) into (46) gives

$$\begin{aligned} d\mathbf{r}_s = &\left[-\frac{1}{2}\beta(s)\mathbf{r}_s - \beta(s)\nabla_{\mathbf{r}_s} \log p_s\left(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}\right)\right] ds \\ &+ \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s, \end{aligned} \tag{49}$$

which is exactly the conditional reverse-time VP-SDE used in (8).

[Noise-Prediction Parameterization]. The conditional score in (49) is approximated using the trained noise-prediction network. Under the VP perturbation kernel in (42), the score is parameterized as

$$\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}) \approx -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})}{\sigma(s)}. \tag{50}$$

Substituting (50) into (49) yields the learned reverse stochastic dynamics

$$d\mathbf{r}_s = \left[-\frac{1}{2}\beta(s)\mathbf{r}_s + \frac{\beta(s)}{\sigma(s)}\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})\right] ds + \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s. \tag{51}$$

Hence, the learned reverse process consists of a linear drift in $\mathbf{r}_s$, a nonlinear drift determined by the conditional noise-prediction network, and a Gaussian diffusion term.

[Reverse-Time Discretization]. Let

$$1 = s_K > s_{K-1} > \cdots > s_0 = 0, \qquad \Delta s_k := s_k - s_{k-1} > 0, \tag{52}$$

denote a discretization of the diffusion-time interval. Since the reverse process evolves from s_k to s_{k-1}, the corresponding time increment is

$$s_{k-1} - s_k = -\Delta s_k. \tag{53}$$

Applying the Euler–Maruyama discretization to (51) gives

$$\begin{aligned} \mathbf{r}_{k-1} = \mathbf{r}_k - \left[-\frac{1}{2}\beta_k \mathbf{r}_k + \frac{\beta_k}{\sigma_k}\boldsymbol{\epsilon}_\theta(\mathbf{r}_k, s_k, \mathbf{x}, \mathbf{c})\right]\Delta s_k \\ + \sqrt{\beta_k \Delta s_k}\,\mathbf{z}_k, \qquad \mathbf{z}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned} \tag{54}$$

where $\beta_k := \beta(s_k)$ and $\sigma_k := \sigma(s_k)$. Rearranging the deterministic terms yields

$$\mathbf{r}_{k-1} = \left(1 + \frac{1}{2}\beta_k \Delta s_k\right)\mathbf{r}_k - \frac{\beta_k \Delta s_k}{\sigma_k}\boldsymbol{\epsilon}_\theta(\mathbf{r}_k, s_k, \mathbf{x}, \mathbf{c}) + \sqrt{\beta_k \Delta s_k}\,\mathbf{z}_k. \tag{55}$$

Therefore, under Euler–Maruyama discretization, the coefficients in (9) are

$$a_k = 1 + \frac{1}{2}\beta_k \Delta s_k, \qquad b_k = -\frac{\beta_k \Delta s_k}{\sigma_k}, \qquad \eta_k = \sqrt{\beta_k \Delta s_k}. \tag{56}$$

More generally, alternative reverse-SDE discretizations or diffusion schedules may result in different coefficient expressions while retaining the same structural form

$$\mathbf{r}_{k-1} = a_k \mathbf{r}_k + b_k \boldsymbol{\epsilon}_\theta(\mathbf{r}_k, k, \mathbf{x}, \mathbf{c}) + \eta_k \mathbf{z}_k. \tag{57}$$

This representation separates the linear reverse term, the learned nonlinear contribution, and the Gaussian innovation. Importantly, RPM does not rely on the particular Euler–Maruyama coefficients in (56); rather, it exploits the generic structure in (57). After locally polynomializing the nonlinear noise predictor, this structure enables the reverse stochastic update to be expressed as a polynomial function of $\mathbf{r}_k$ plus Gaussian innovation, from which the residual moments can be propagated recursively.

A.2.3 Score Approximation

The conditional score in (49) is generally unknown and is approximated by the trained diffusion model. From the forward perturbation representation in (42),

$$\mathbf{r}_s = \alpha(s)\mathbf{r}_0 + \sigma(s)\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

a noise-prediction network $\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})$ is trained to predict the injected Gaussian noise. Under the standard noise-prediction parameterization, the conditional score is represented as

$$\nabla_{\mathbf{r}_s} \log p_s(\mathbf{r}_s \mid \mathbf{x}, \mathbf{c}) \approx -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})}{\sigma(s)}. \tag{58}$$

Substituting (58) into (49) yields the learned reverse dynamics

$$d\mathbf{r}_s \approx \left[-\frac{1}{2}\beta(s)\mathbf{r}_s + \frac{\beta(s)}{\sigma(s)}\boldsymbol{\epsilon}_\theta(\mathbf{r}_s, s, \mathbf{x}, \mathbf{c})\right] ds + \sqrt{\beta(s)}\, d\overline{\mathbf{W}}_s. \tag{59}$$

Hence, the trained conditional noise predictor determines the learned component of the reverse drift, while the VP diffusion schedule determines the remaining drift and diffusion coefficients. This reverse stochastic dynamics provides the basis for the reverse-polynomial moment propagation developed in the main text.

Theorem 4 (Correctness of Reverse Diffusion Process) *Assume that conditional score function*

$$\nabla_{\mathbf{r}} \log p_s(\mathbf{r}|\mathbf{x}, \mathbf{c})$$

is exactly known. Then the reverse-time SDE in (36), initialized from the Gaussian prior $p_1(\mathbf{r}) = \mathcal{N}(0, I)$, recovers the original conditional residual distribution, i.e.,

$$\mathbf{r}_0 \sim p_0(\mathbf{r}|\mathbf{x}, \mathbf{c}).$$

Proof. The result follows from the time-reversal property of Itô stochastic processes. For the forward diffusion process, the marginal density p_s evolves according to the Fokker–Planck equation Øksendal (2003). The reverse-time drift is obtained by subtracting the score correction term

$$g_{\text{diff}}(s)^2 \nabla_{\mathbf{r}} \log p_s(\mathbf{r}_s|\mathbf{x}, \mathbf{c})$$

from the forward drift. Therefore, the reverse process preserves the same marginal probability flow as the forward diffusion but evolves in the opposite temporal direction. Starting from the terminal Gaussian distribution, the reverse SDE reconstructs the original data distribution at $s = 0$. □

A.2.4 Reverse-Polynomial Moment Propagation

Although the reverse SDE theoretically defines the learned residual distribution, directly sampling reverse trajectories to estimate uncertainty introduces Monte Carlo variance and requires a large number of samples Robert et al. (1999). Instead, we directly propagate the moments of the reverse process by the diffusion trajectory using moment-based approximation and closure Kuehn (2016).

In practice, the reverse SDE is discretized into K reverse diffusion steps. Let $\mathbf{r}_k$ denote the diffusion latent variable at step k. A general reverse update is written as

$$\mathbf{r}_{k-1} = a_k \mathbf{r}_k + b_k \epsilon_\theta(\mathbf{r}_k, k, \mathbf{x}, \mathbf{c}) + \eta_k \mathbf{z}_k, \tag{60}$$

where $\mathbf{z}_k \sim \mathcal{N}(0, I)$, and a_k, b_k, and η_k are determined by the diffusion schedule.

The nonlinear noise predictor is locally approximated by a quadratic polynomial:

$$\epsilon_\theta(r_k, k, \mathbf{x}, \mathbf{c}) \approx q_{0,k} + q_{1,k} r_k + q_{2,k} r_k^2. \tag{61}$$

Substituting the approximation into reverse update gives

$$r_{k-1} = h_{0,k} + h_{1,k} r_k + h_{2,k} r_k^2 + \eta_k z_k. \tag{62}$$

Assume that the diffusion latent variable follows Gaussian moment closure $r_k \sim \mathcal{N}(\mu_k, v_k)$. The required moments are

$$\begin{aligned} m_{1,k} &= \mu_k, \\ m_{2,k} &= \mu_k^2 + v_k, \\ m_{3,k} &= \mu_k^3 + 3\mu_k v_k, \\ m_{4,k} &= \mu_k^4 + 6\mu_k^2 v_k + 3v_k^2. \end{aligned} \tag{63}$$

Proposition 2 (Moment Recursion under Gaussian Closure) *For the quadratic reverse update in (62), assuming $r_k \sim \mathcal{N}(\mu_k, v_k)$, the propagated mean and variance are given by*

$$\mu_{k-1} = h_{0,k} + h_{1,k} m_{1,k} + h_{2,k} m_{2,k}, \tag{64}$$

and

$$v_{k-1} = M_{2,k-1} - \mu_{k-1}^2, \tag{65}$$

where

$$\begin{aligned} M_{2,k-1} = h_{0,k}^2 + 2h_{0,k}h_{1,k}m_{1,k} + (h_{1,k}^2 + 2h_{0,k}h_{2,k})m_{2,k} \\ + 2h_{1,k}h_{2,k}m_{3,k} + h_{2,k}^2 m_{4,k} + \eta_k^2. \end{aligned} \tag{66}$$

Proof. Taking expectation of (62) gives

$$\mathbb{E}[r_{k-1}] = h_{0,k} + h_{1,k}\mathbb{E}[r_k] + h_{2,k}\mathbb{E}[r_k^2],$$

because $\mathbb{E}[z_k] = 0$. Applying Gaussian moment identities yields the mean recursion.

Similarly, expanding r_{k-1}^2 and taking expectation gives the second-order moment expression. The variance is obtained from the definition

$$\mathrm{Var}(r_{k-1}) = \mathbb{E}[r_{k-1}^2] - \mathbb{E}[r_{k-1}]^2.$$

□

Starting from

$$\mu_K = 0, \qquad v_K = 1,$$

the above recursion is applied from $k = K$ to $k = 1$. The terminal moments

$$\mu_0, \quad v_0$$

provide the conditional uncertainty of the reconstructed residual:

$$v_0 \approx \mathrm{Var}(\Delta x|\mathbf{x}, \mathbf{c}). \tag{67}$$

A.2.5 Error Analysis of Reverse-Polynomial Moment Propagation

This section establishes error bounds for the reverse-polynomial moment propagation (RPM) procedure introduced in Sec. 3.2. The approximation error arises from two sources: (i) replacing the nonlinear neural noise predictor by a local polynomial, and (ii) closing the resulting moment hierarchy using Gaussian moment closure. We analyze these two sources separately and then derive an overall bound on the terminal residual moments.

Preliminaries. For clarity, we consider a scalar residual component; the same analysis applies componentwise to the modeled residual dimensions. Recall the original learned reverse dynamics

$$r_{k-1} = F_k(r_k) + \eta_k z_k, \qquad z_k \sim \mathcal{N}(0, 1), \tag{68}$$

where

$$F_k(r) := a_k r + b_k \epsilon_\theta(r, k, \mathbf{x}, \mathbf{c}). \tag{69}$$

For a fixed conditioning pair $(\mathbf{x}, \mathbf{c})$, the corresponding polynomialized reverse dynamics are

$$\widetilde{r}_{k-1} = \widetilde{F}_k(\widetilde{r}_k) + \eta_k z_k, \tag{70}$$

where

$$\widetilde{F}_k(r) := h_{0,k} + h_{1,k} r + h_{2,k} r^2. \tag{71}$$

Equivalently,

$$\widetilde{F}_k(r) = a_k r + b_k p_k(r), \qquad p_k(r) := q_{0,k} + q_{1,k} r + q_{2,k} r^2.$$

We distinguish three sets of moments throughout the analysis. Let

$$\mu_k := \mathbb{E}[r_k], \qquad v_k := \mathrm{Var}[r_k]$$

denote the exact moments of the learned neural reverse process (68). Let

$$\widetilde{\mu}_k := \mathbb{E}[\widetilde{r}_k], \qquad \widetilde{v}_k := \mathrm{Var}[\widetilde{r}_k]$$

denote the exact moments of the polynomialized process (70). Finally, let

$$\widehat{\mu}_k, \qquad \widehat{v}_k$$

denote the moments produced by the RPM recursion with Gaussian moment closure.

All three constructions are initialized from the same Gaussian prior:

$$r_K = \widetilde{r}_K \sim \mathcal{N}(0, 1), \qquad \widehat{\mu}_K = 0, \qquad \widehat{v}_K = 1. \tag{72}$$

The total terminal errors admit the decomposition

$$\begin{aligned} |\mu_0 - \widehat{\mu}_0| &\le |\mu_0 - \widetilde{\mu}_0| + |\widetilde{\mu}_0 - \widehat{\mu}_0|, \\ |v_0 - \widehat{v}_0| &\le |v_0 - \widetilde{v}_0| + |\widetilde{v}_0 - \widehat{v}_0|. \end{aligned} \tag{73}$$

The first terms characterize the polynomial approximation error, whereas the second terms characterize the Gaussian-closure error.

[Error Induced by Polynomial Approximation]. We first quantify the discrepancy between the learned neural reverse process and its polynomial approximation.

Assumption 2 (Local Polynomial Approximation) *For each reverse step k, the polynomial approximation p_k satisfies*

$$\mathbb{E}\left[|\epsilon_\theta(\widetilde{r}_k, k, \mathbf{x}, \mathbf{c}) - p_k(\widetilde{r}_k)|^2\right] \le \left(\varepsilon_k^{\text{poly}}\right)^2. \tag{74}$$

Assumption 3 (Reverse-Map Stability) *For each reverse step k, the neural reverse map F_k is Lipschitz on the region reached by the coupled reverse processes, i.e.,*

$$|F_k(r) - F_k(r')| \le L_k|r - r'| \tag{75}$$

for some finite constant $L_k > 0$.

We couple the original and polynomialized reverse processes using the same Gaussian innovations z_k and define their mean-square discrepancy as

$$E_k^{\text{poly}} := \mathbb{E}\left[|r_k - \widetilde{r}_k|^2\right]. \tag{76}$$

Lemma 1 (One-Step Polynomialization Error) *Under Assumptions 2 and 3, for any $\gamma_k > 0$,*

$$E_{k-1}^{\text{poly}} \le \alpha_k E_k^{\text{poly}} + \beta_k, \tag{77}$$

where

$$\alpha_k := (1+\gamma_k)L_k^2, \qquad \beta_k := \left(1 + \frac{1}{\gamma_k}\right) b_k^2 \left(\varepsilon_k^{\text{poly}}\right)^2. \tag{78}$$

Proof. Because the two processes are coupled using the same Gaussian innovation, the stochastic terms cancel:

$$r_{k-1} - \widetilde{r}_{k-1} = F_k(r_k) - \widetilde{F}_k(\widetilde{r}_k). \tag{79}$$

Adding and subtracting $F_k(\widetilde{r}_k)$ gives

$$r_{k-1} - \widetilde{r}_{k-1} = F_k(r_k) - F_k(\widetilde{r}_k) + b_k\left[\epsilon_\theta(\widetilde{r}_k, k, \mathbf{x}, \mathbf{c}) - p_k(\widetilde{r}_k)\right]. \tag{80}$$

For arbitrary $\gamma_k > 0$, Young's inequality Hardy et al. (1952)

$$|u+v|^2 \le (1+\gamma_k)|u|^2 + \left(1 + \frac{1}{\gamma_k}\right)|v|^2$$

together with Assumptions 2 and 3 yields

$$E_{k-1}^{\text{poly}} \le (1+\gamma_k)L_k^2 E_k^{\text{poly}} + \left(1 + \frac{1}{\gamma_k}\right) b_k^2 (\varepsilon_k^{\text{poly}})^2,$$

which proves (77). □

Since both processes start from the same Gaussian prior, $E_K^{\text{poly}} = 0$. Recursively applying Lemma 1 gives the following terminal bound.

Corollary 1 (Accumulated Polynomialization Error) *The terminal mean-square discrepancy satisfies*

$$E_0^{\text{poly}} \le \sum_{j=1}^{K} \beta_j \prod_{i=1}^{j-1} \alpha_i, \tag{81}$$

where an empty product is defined as one.

The mean error follows immediately from Jensen's inequality:

$$|\mu_k - \widetilde{\mu}_k| \le \sqrt{E_k^{\text{poly}}}. \tag{82}$$

To bound the variance error, we additionally assume bounded second moments.

Assumption 4 (Bounded Second Moments) *For each reverse step k, there exists $M_{2,k} < \infty$ such that*

$$\mathbb{E}[r_k^2] \le M_{2,k}, \qquad \mathbb{E}[\widetilde{r}_k^2] \le M_{2,k}. \tag{83}$$

Using

$$\mathrm{Var}[X] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

and the Cauchy–Schwarz inequality Hardy et al. (1952), we obtain

$$|v_k - \widetilde{v}_k| \le 4\sqrt{M_{2,k} E_k^{\mathrm{poly}}}. \tag{84}$$

Therefore, at the terminal reverse step,

$$|\mu_0 - \widetilde{\mu}_0| \le \sqrt{E_0^{\mathrm{poly}}}, \qquad |v_0 - \widetilde{v}_0| \le 4\sqrt{M_{2,0} E_0^{\mathrm{poly}}}. \tag{85}$$

[Error Induced by Gaussian Moment Closure]. We next quantify the approximation introduced by closing the moment hierarchy at the first two moments.

For the exact polynomialized process, define the raw moments

$$\widetilde{m}_{j,k} := \mathbb{E}[\widetilde{r}_k^j].$$

The deviation of its third and fourth moments from their Gaussian counterparts is characterized by the closure residuals

$$\kappa_{3,k} := \widetilde{m}_{3,k} - \left(\widetilde{\mu}_k^3 + 3\widetilde{\mu}_k \widetilde{v}_k\right), \tag{86}$$

and

$$\kappa_{4,k} := \widetilde{m}_{4,k} - \left(\widetilde{\mu}_k^4 + 6\widetilde{\mu}_k^2 \widetilde{v}_k + 3\widetilde{v}_k^2\right). \tag{87}$$

These quantities vanish when $\widetilde{r}_k$ is Gaussian and measure the local error introduced by Gaussian moment closure otherwise.

For fixed first two moments, the mean recursion is exact because the quadratic reverse map requires only the first two raw moments:

$$\widetilde{\mu}_{k-1} = h_{0,k} + h_{1,k}\widetilde{\mu}_k + h_{2,k}\left(\widetilde{\mu}_k^2 + \widetilde{v}_k\right). \tag{88}$$

In contrast, the second raw moment depends on $\widetilde{m}_{3,k}$ and $\widetilde{m}_{4,k}$. Substituting (86)–(87) into the exact second-moment recursion shows that the one-step discrepancy induced by Gaussian closure is

$$\Delta_{2,k-1}^{\mathrm{cl}} = 2h_{1,k}h_{2,k}\kappa_{3,k} + h_{2,k}^2 \kappa_{4,k}. \tag{89}$$

Consequently,

$$\left|\Delta_{2,k-1}^{\mathrm{cl}}\right| \le 2|h_{1,k}h_{2,k}||\kappa_{3,k}| + h_{2,k}^2|\kappa_{4,k}|. \tag{90}$$

Define the closed RPM moment map

$$\widehat{\mathbf{m}}_{k-1} = \mathcal{G}_k(\widehat{\mathbf{m}}_k), \qquad \widehat{\mathbf{m}}_k := \begin{bmatrix} \widehat{\mu}_k \\ \widehat{v}_k \end{bmatrix}, \tag{91}$$

where $\mathcal{G}_k$ denotes the closed mean–variance recursion induced by the polynomial reverse update in (10). Specifically, the mean is propagated using the first-order moment recursion, while the second raw moment is propagated according to (16); the required higher-order moments are approximated using the Gaussian closure in (15), and the variance is recovered as $v_{k-1} = m_{2,k-1} - \mu_{k-1}^2$.

Assumption 5 (Moment-Map Stability) *For each reverse step k, the closed moment map $\mathcal{G}_k$ is Lipschitz on a domain containing both $(\widetilde{\mu}_k, \widetilde{v}_k)$ and $(\widehat{\mu}_k, \widehat{v}_k)$:*

$$\|\mathcal{G}_k(\mathbf{m}) - \mathcal{G}_k(\mathbf{m}')\|_2 \le C_k \|\mathbf{m} - \mathbf{m}'\|_2 \tag{92}$$

for some finite constant $C_k > 0$.

Define the closure-induced moment error

$$E_k^{\mathrm{cl}} := \left\| \begin{bmatrix} \widetilde{\mu}_k \\ \widetilde{v}_k \end{bmatrix} - \begin{bmatrix} \widehat{\mu}_k \\ \widehat{v}_k \end{bmatrix} \right\|_2. \tag{93}$$

Lemma 2 (One-Step Closure Error) *Under Assumption 5, the closure error satisfies*

$$E_{k-1}^{\mathrm{cl}} \le C_k E_k^{\mathrm{cl}} + \delta_k^{\mathrm{cl}}, \tag{94}$$

where

$$\delta_k^{\mathrm{cl}} := 2|h_{1,k}h_{2,k}||\kappa_{3,k}| + h_{2,k}^2|\kappa_{4,k}|. \tag{95}$$

Proof. For fixed exact first two moments, Gaussian closure leaves the mean update unchanged and perturbs only the second-moment, and hence variance, update by $\Delta^{\mathrm{cl}}_{2,k-1}$. Therefore, the local closure perturbation has Euclidean norm bounded by δ^{cl}_k. The remaining discrepancy results from evaluating the closed moment map at different first two moments and is bounded by Assumption 5. Applying the triangle inequality yields (94). □

Because the Gaussian prior is represented exactly at $k = K$,

$$E^{\mathrm{cl}}_K = 0.$$

Unrolling (94) therefore yields

$$E^{\mathrm{cl}}_0 \le \sum_{j=1}^{K} \delta^{\mathrm{cl}}_j \prod_{i=1}^{j-1} C_i. \tag{96}$$

[Overall RPM Error Bound]. We now combine the polynomialization and moment-closure errors.

Theorem 5 (Terminal Error Bound for RPM) *Suppose Assumptions 2–5 hold. Then the terminal mean and variance produced by RPM satisfy*

$$|\mu_0 - \widehat{\mu}_0| \le \varepsilon_\mu, \tag{97}$$

and

$$|v_0 - \widehat{v}_0| \le \varepsilon_v, \tag{98}$$

where

$$\varepsilon_\mu := \sqrt{E^{\mathrm{poly}}_0} + E^{\mathrm{cl}}_0, \tag{99}$$

and

$$\varepsilon_v := 4\sqrt{M_{2,0} E^{\mathrm{poly}}_0} + E^{\mathrm{cl}}_0. \tag{100}$$

Moreover,

$$E^{\mathrm{poly}}_0 \le \sum_{j=1}^{K} \beta_j \prod_{i=1}^{j-1} \alpha_i, \tag{101}$$

and

$$E^{\mathrm{cl}}_0 \le \sum_{j=1}^{K} \delta^{\mathrm{cl}}_j \prod_{i=1}^{j-1} C_i. \tag{102}$$

Proof. By the decomposition in (73),

$$|\mu_0 - \widehat{\mu}_0| \le |\mu_0 - \widetilde{\mu}_0| + |\widetilde{\mu}_0 - \widehat{\mu}_0|.$$

Using (82) and the definition of E^{cl}_0 gives

$$|\mu_0 - \widehat{\mu}_0| \le \sqrt{E^{\mathrm{poly}}_0} + E^{\mathrm{cl}}_0.$$

Similarly,

$$|v_0 - \widehat{v}_0| \le |v_0 - \widetilde{v}_0| + |\widetilde{v}_0 - \widehat{v}_0|.$$

Applying (84) and (93) yields

$$|v_0 - \widehat{v}_0| \le 4\sqrt{M_{2,0} E^{\mathrm{poly}}_0} + E^{\mathrm{cl}}_0.$$

Finally, the accumulated bounds (101) and (102) follow by recursively applying Lemmas 1 and 2, respectively. □

Theorem 5 shows that the RPM approximation error is controlled by three quantities: the local polynomial fitting errors $\varepsilon^{\mathrm{poly}}_k$, the Gaussian-closure residuals $\kappa_{3,k}$ and $\kappa_{4,k}$, and the stability factors of the reverse and moment dynamics. In particular, approximation errors remain controlled when the local errors are small and their amplification through the reverse dynamics is sufficiently limited.

Finally, the variance bound can be transferred directly to the continuous-time covariance rate used in the verification-oriented stochastic model. For a physical sampling interval Δt, define

$$Q_\theta := \frac{v_0}{\Delta t}, \qquad \widehat{Q}_\theta := \frac{\widehat{v}_0}{\Delta t}. \tag{103}$$

Then Theorem 5 immediately gives

$$\left| Q_\theta - \widehat{Q}_\theta \right| \le \frac{\varepsilon_v}{\Delta t}. \tag{104}$$

Thus, the approximation error introduced by RPM induces an explicitly bounded error in the covariance rate of the stochastic model constructed from the learned reverse diffusion dynamics.

A.3 From Residual Covariance to Verification-Oriented Stochastic Dynamics

The propagated terminal covariance represents the uncertainty encoded by the complete reverse diffusion process. For a physical sampling interval Δt, the covariance rate is defined as

$$G_{\text{diff}} = \frac{1}{\Delta t} \operatorname{Cov}(\Delta \mathbf{x} | \mathbf{x}, \mathbf{c}). \tag{105}$$

Proposition 3 (Covariance-Matched Stochastic Abstraction) *Let G_{diff} be the covariance rate obtained from the reverse moment propagation. Then there exists a diffusion coefficient g_{diff} satisfying*

$$g_{\text{diff}} g_{\text{diff}}^T = G_{\text{diff}},$$

such that the stochastic system

$$d\mathbf{x}_t = f_{\text{nom}}(\mathbf{x}_t)dt + g_{\text{diff}} d\mathbf{W}_t \tag{106}$$

preserves the local second-order uncertainty characteristics of the learned diffusion-induced transition model.

proof. The covariance rate of an Itô diffusion process is determined by

$$g_{\text{diff}} g_{\text{diff}}^T.$$

The reverse diffusion process provides the covariance of the residual transition over one physical interval,

$$\operatorname{Cov}(\Delta \mathbf{x}).$$

Dividing this covariance by Δt gives the corresponding continuous-time covariance rate. Since every positive semi-definite matrix admits a matrix factorization, a diffusion coefficient satisfying

$$g_{\text{diff}} g_{\text{diff}}^T = G_{\text{diff}}$$

always exists. Therefore, the constructed SDE preserves the local second-order stochastic behavior of the learned diffusion abstraction. □

A.4 Counterexample-Guided Certificate Refinement

Although neural SBC learning provides an effective mechanism for constructing polynomial certificate candidates, the learned candidate is optimized only over finitely sampled states and may therefore violate the SBC conditions at unseen states. We adopt a counterexample-guided inductive synthesis (CEGIS) strategy Solar-Lezama et al. (2006) to iteratively identify such violations and refine the candidate until posterior SOS certification succeeds.

At refinement iteration k, let B_{θ_k} denote the current polynomial SBC candidate. We first solve the SOS program in (19) with B_{θ_k} fixed. When certification is successful, the optimization returns the certified failure bound $\rho_k^\star$ and the procedure terminates. Otherwise, we search for states that maximally violate the individual SBC conditions under the current candidate and failure bound.

A.4.1 Maximum Counterexample Generation

Rather than searching for an arbitrary violating state, we identify an extremal counterexample for each SBC condition. For the initial-set condition, we solve

$$\mathbf{x}_{\Theta}^{\star} = \arg\max_{\mathbf{x}\in\Theta} \left[B_{\theta_k}(\mathbf{x}) - \rho_k^{\star}\right], \tag{107}$$

with the corresponding violation magnitude

$$\nu_{\Theta}^{\star} = B_{\theta_k}(\mathbf{x}_{\Theta}^{\star}) - \rho_k^{\star}. \tag{108}$$

A positive value $\nu_{\Theta}^{\star} > 0$ indicates a violation of condition (i).

For the unsafe-set condition, we compute

$$\mathbf{x}_{\Xi}^{\star} = \arg\min_{\mathbf{x}\in\Xi} \left[B_{\theta_k}(\mathbf{x}) - 1\right], \tag{109}$$

and define

$$\nu_{\Xi}^{\star} = 1 - B_{\theta_k}(\mathbf{x}_{\Xi}^{\star}). \tag{110}$$

Thus, $\nu_{\Xi}^{\star} > 0$ identifies a violation of condition (ii).

Similarly, violation of the nonnegativity condition is detected through

$$\mathbf{x}_{\Psi,B}^{\star} = \arg\min_{\mathbf{x}\in\Psi} B_{\theta_k}(\mathbf{x}), \tag{111}$$

with

$$\nu_{\Psi,B}^{\star} = -B_{\theta_k}(\mathbf{x}_{\Psi,B}^{\star}), \tag{112}$$

where $\nu_{\Psi,B}^{\star} > 0$ indicates violation of condition (iii).

Finally, for the stochastic generator condition, we solve

$$\mathbf{x}_{\Psi,\mathcal{A}}^{\star} = \arg\max_{\mathbf{x}\in\Psi} \mathcal{A}B_{\theta_k}(\mathbf{x}), \tag{113}$$

and define

$$\nu_{\Psi,\mathcal{A}}^{\star} = \mathcal{A}B_{\theta_k}(\mathbf{x}_{\Psi,\mathcal{A}}^{\star}). \tag{114}$$

A positive value $\nu_{\Psi,\mathcal{A}}^{\star} > 0$ indicates violation of condition (iv).

The resulting set of maximum counterexamples is therefore

$$\mathcal{C}_k^{\star} = \left\{\mathbf{x}_{\Theta}^{\star}, \mathbf{x}_{\Xi}^{\star}, \mathbf{x}_{\Psi,B}^{\star}, \mathbf{x}_{\Psi,\mathcal{A}}^{\star}\right\}, \tag{115}$$

where only counterexamples associated with positive violation magnitudes are retained. These extremal states identify regions in which the current candidate most strongly violates the corresponding SBC conditions.

A.4.2 Counterexample-Set Construction

Using only a single extremal state for retraining may lead to highly localized corrections and require many CEGIS iterations. We therefore expand each maximum counterexample into a local counterexample set. For a maximum counterexample $\mathbf{x}^{\star}$ associated with verification region $\Omega \in \{\Theta, \Xi, \Psi\}$, define its γ-neighborhood as

$$\mathcal{N}_{\Omega}(\mathbf{x}^{\star}, \gamma) = \{\mathbf{x} \in \Omega : \|\mathbf{x} - \mathbf{x}^{\star}\|_{\infty} \le \gamma\}. \tag{116}$$

Candidate states are generated around $\mathbf{x}^{\star}$ according to

$$\widetilde{\mathbf{x}}^{(j)} = \Pi_{\Omega}\left(\mathbf{x}^{\star} + \boldsymbol{\delta}^{(j)}\right), \qquad \boldsymbol{\delta}^{(j)} \sim \mathrm{Unif}([-\gamma, \gamma]^n), \tag{117}$$

where Π_{Ω} projects the perturbed state onto the corresponding verification region. The sampled states are then filtered using the associated SBC condition so that only genuine violating states are retained.

Specifically, the local counterexample sets are defined as

$$\mathcal{C}_{\Theta,k} = \{\mathbf{x} \in \mathcal{N}_{\Theta}(\mathbf{x}_{\Theta}^{\star}, \gamma) : B_{\theta_k}(\mathbf{x}) > \rho_k^{\star}\}, \tag{118}$$

$$\mathcal{C}_{\Xi,k} = \{\mathbf{x} \in \mathcal{N}_{\Xi}(\mathbf{x}_{\Xi}^{\star}, \gamma) : B_{\theta_k}(\mathbf{x}) < 1\}, \tag{119}$$

$$\mathcal{C}_{\Psi,B,k} = \left\{\mathbf{x} \in \mathcal{N}_{\Psi}(\mathbf{x}_{\Psi,B}^{\star}, \gamma) : B_{\theta_k}(\mathbf{x}) < 0\right\}, \tag{120}$$

$$\mathcal{C}_{\Psi,\mathcal{A},k} = \left\{\mathbf{x} \in \mathcal{N}_{\Psi}(\mathbf{x}_{\Psi,\mathcal{A}}^{\star}, \gamma) : \mathcal{A}B_{\theta_k}(\mathbf{x}) > 0\right\}. \tag{121}$$

The complete counterexample set at iteration k is then

$$\mathcal{C}_k = \mathcal{C}_{\Theta,k} \cup \mathcal{C}_{\Xi,k} \cup \mathcal{C}_{\Psi,B,k} \cup \mathcal{C}_{\Psi,\mathcal{A},k}. \tag{122}$$

This construction retains the maximum counterexample itself and augments it with nearby violating states, thereby providing the learner with a local characterization of the violated region rather than a single isolated state.

A.4.3 Certificate Refinement

Let $\mathcal{D}_k$ denote the training dataset at CEGIS iteration k. After generating $\mathcal{C}_k$, the training dataset is augmented as

$$\mathcal{D}_{k+1} = \mathcal{D}_k \cup \mathcal{C}_k. \tag{123}$$

The neural SBC is then retrained over $\mathcal{D}_{k+1}$ to obtain an updated polynomial candidate $B_{\theta_{k+1}}$. The updated candidate is subsequently fixed and passed again to the SOS posterior certification program (19), which re-optimizes the failure bound and returns an updated value $\rho^\star_{k+1}$ whenever the corresponding SOS constraints are feasible.

At each CEGIS iteration, the current certificate candidate is first examined for violations of the SBC conditions. The maximum counterexamples are identified through the condition-specific optimization problems above and expanded into local counterexample sets, which are then incorporated into the training dataset to refine the certificate candidate. The refined candidate is subsequently subjected to SOS posterior certification, and this process is repeated until the SOS program becomes feasible. At termination, the SBC conditions are formally certified over their respective continuous domains, and the optimized failure bound $\rho^\star$ yields the certified infinite-horizon safety lower bound $1-\rho^\star$. The procedure terminates once the SOS program is feasible for the current candidate. At termination, the SBC conditions are formally certified over their respective continuous domains, and the optimized failure bound $\rho^\star$ yields the certified infinite-horizon safety lower bound $1-\rho^\star$.

A.5 Algorithmic Procedure

The complete diffusion-induced probabilistic safety verification procedure is summarized in Algorithm 1. The algorithm integrates trajectory-conditioned uncertainty learning, reverse-moment propagation, covariance-matched stochastic abstraction, and counterexample-guided stochastic barrier certificate synthesis into a unified verification pipeline.

Algorithm 1 proceeds in four main stages. Lines 1–3 construct the transition residuals and trajectory contexts from the observed trajectories, train the conditional diffusion model, and determine the state–context conditions required for stochastic abstraction. This stage separates the environment-induced transition uncertainty from the known nominal dynamics and learns its conditional distribution. Lines 4–16 perform reverse-polynomial moment propagation and construct the verification-oriented stochastic dynamics. For each required state–context condition, the reverse process is initialized from the standard Gaussian prior. At each reverse step, the nonlinear noise predictor is locally approximated by a quadratic polynomial, and the resulting polynomial reverse dynamics are used to recursively propagate the first two residual moments under Gaussian moment closure. The terminal moments yield the conditional residual covariance $\mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x},\mathbf{c})$, which is converted into the continuous-time covariance rate $Q_\theta(\mathbf{x},\mathbf{c}) = \mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x},\mathbf{c})/\Delta t$. A diffusion coefficient G_θ satisfying $G_\theta G_\theta^\top = Q_\theta$ is then used to construct the verification-oriented stochastic dynamics, from which the infinitesimal generator $\mathcal{A}$ is derived for subsequent SBC synthesis and verification.

Lines 17–41 perform counterexample-guided SBC synthesis and posterior certification for each candidate barrier degree. A neural polynomial SBC candidate B_ϑ is first learned jointly with a failure-bound candidate ρ over the sampled states, after which its exact polynomial form and infinitesimal generator $\mathcal{A}B$ are constructed. The four SBC conditions are then encoded as SOS constraints and solved through SDP over the complete semi-algebraic domains. If the SOS certification succeeds, the certified optimum $\rho^\star$ and the corresponding barrier certificate are recorded. Otherwise, four condition-specific optimization problems search for extremal violations of the initial-set, unsafe-set, nonnegativity, and generator conditions, respectively. Only extremal states with positive violation magnitudes are retained and expanded into local counterexample sets, which are added to the corresponding training sets for certificate refinement. If SOS certification fails but no explicit violating

Algorithm 1: Diffusion-Induced Probabilistic Safety Verification

Require: Nominal dynamics f_{nom}, trajectory data $\mathcal{T}$, sampling interval Δt, sets Θ, Ψ, Ξ, reverse steps K, and candidate barrier degrees $\mathcal{D}_B$
Ensure: Verified SBC $B^\star$ and certified safety lower bound $1-\rho^\star$, if found

Compute transition residuals $\Delta\mathbf{x}_t = \mathbf{x}_{t+1} - \Psi_{\Delta t}(\mathbf{x}_t)$ and trajectory contexts $\mathbf{c}_t = E_\phi(\mathcal{H}_t)$
Train the conditional diffusion model $p_\theta(\Delta\mathbf{x} \mid \mathbf{x}, \mathbf{c})$
Determine the state–context conditions $(\mathbf{x}, \mathbf{c})$ required for stochastic abstraction
for each required $(\mathbf{x}, \mathbf{c})$ **do**
  Initialize $\mu_K \leftarrow 0$ and $v_K \leftarrow 1$
  for $k = K, K-1, \ldots, 1$ **do**
    Approximate $\epsilon_\theta(r_k, k, \mathbf{x}, \mathbf{c})$ by $q_{0,k} + q_{1,k} r_k + q_{2,k} r_k^2$
    Construct $r_{k-1} = h_{0,k} + h_{1,k} r_k + h_{2,k} r_k^2 + \eta_k z_k$
    Propagate $(\mu_k, v_k) \to (\mu_{k-1}, v_{k-1})$ using the moment recursions and Gaussian closure
  end for
  Recover the conditional residual covariance $\mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x}, \mathbf{c})$
  Set $Q_\theta(\mathbf{x}, \mathbf{c}) \leftarrow \mathbf{\Sigma}_{\Delta\mathbf{x}}(\mathbf{x}, \mathbf{c}) / \Delta t$
  Choose $G_\theta(\mathbf{x}, \mathbf{c})$ such that $G_\theta G_\theta^\top = Q_\theta(\mathbf{x}, \mathbf{c})$
end for
Construct $d\mathbf{x}_t = f_{\mathrm{nom}}(\mathbf{x}_t)dt + G_\theta(\mathbf{x}_t, \mathbf{c}_t)d\mathbf{W}_t$
Construct the infinitesimal generator $\mathcal{A}$
for each $d_B \in \mathcal{D}_B$ **do**
  Initialize the neural SBC learner and sampled states over Θ, Ψ, and Ξ
  repeat
    Train $B_\vartheta(\mathbf{x})$ jointly with ρ using the sampled SBC losses
    Convert B_ϑ into its exact polynomial form B and construct $\mathcal{A}B$
    Formulate the four SBC conditions as SOS constraints and solve the resulting SDP
    if SOS certification succeeds **then**
      Obtain the certified optimum $\rho^\star$
      Record $(B, \rho^\star, d_B)$
      break
    else
      Compute $\mathbf{x}^\star_\Theta \leftarrow \arg\max_{\mathbf{x}\in\Theta}[B(\mathbf{x}) - \rho]$
      Compute $\mathbf{x}^\star_\Xi \leftarrow \arg\min_{\mathbf{x}\in\Xi}[B(\mathbf{x}) - 1]$
      Compute $\mathbf{x}^\star_{\Psi,B} \leftarrow \arg\min_{\mathbf{x}\in\Psi} B(\mathbf{x})$
      Compute $\mathbf{x}^\star_{\Psi,\mathcal{A}} \leftarrow \arg\max_{\mathbf{x}\in\Psi} \mathcal{A}B(\mathbf{x})$
      Retain extremal states with positive violation magnitudes and construct their local counterexample sets
      Set $\mathcal{C} \leftarrow \mathcal{C}_\Theta \cup \mathcal{C}_\Xi \cup \mathcal{C}_{\Psi,B} \cup \mathcal{C}_{\Psi,\mathcal{A}}$
      if $\mathcal{C} = \emptyset$ **then**
        break
      else
        Add $\mathcal{C}$ to the corresponding training sets
      end if
    end if
  until the refinement budget is exhausted
end for
if at least one candidate is successfully verified **then**
  Select the verified candidate with minimum $\rho^\star$
  return $B^\star$, $\rho^\star$, and $1-\rho^\star$
else
  return UNKNOWN
end if

state is identified, the refinement of the current candidate is terminated rather than interpreting SOS infeasibility itself as evidence of an SBC violation.

Finally, Lines 42–47 examine all successfully certified candidates. If at least one candidate has been verified, the certificate with the smallest certified failure bound $\rho^\star$ is selected and the corresponding infinite-horizon safety lower bound $1-\rho^\star$ is returned. Otherwise, the algorithm returns UNKNOWN, indicating that no certificate has been established within the prescribed search and refinement budgets; this outcome does not imply that the system is unsafe.

A.6 EXPERIMENTAL DETAILS

This supplement expands the experimental study in the main paper. We first summarize the benchmark construction and stochastic models, then report additional degree scans and numerical diagnostics, and finally present the dynamics and barrier geometry of each benchmark. The reported probability bounds certify the fixed stochastic proxy used by each verifier; the statistical relationship between a learned proxy and an unknown physical disturbance is discussed separately at the end of this section.

Table 2: Hardware and software environment used for the reported runs.

Environment item	Value	Software	Version
Operating system	Windows 11	Python	3.14.5
Processor	AMD Ryzen 9 8945HX, 2.50 GHz	NumPy	2.5.1
Memory	32 GB RAM	SciPy	1.18.0
Logical processors	32	scikit-learn	1.9.0
Hardware acceleration	CPU only	SymPy	1.14.0
Solver interface	CVXPY	CVXPY	1.9.2
Conic solver	MOSEK	MOSEK	11.2.2
Figure generation	Pillow	Pillow	12.2.0

A.6.1 BENCHMARK SELECTION

The manuscript reports systems C_1 and $C_3 \sim C_9$ are from the *SynNBC* suite Zhao et al. (2023) and the controlled Duffing system from Example 4.3 of Zhu et al. (2019) as C_2. These 9 cases make the method-wise degree scans and barrier visualizations readable. They span state dimensions from 2 to 9, nominal drift degrees from 2 to 3, and cases in which quadratic, quartic, and sextic barriers are selected. We also require at least one posterior-accepted nontrivial certificate so that a reported case supports a meaningful geometric analysis. The selection is a coverage-oriented case study; no population-level statistical claim over the benchmark suite is made from these nine systems.

Each benchmark implementation supplies a polynomial vector field $f : \mathbb{R}^n \to \mathbb{R}^n$, a compact verification domain Ψ, an initial set Θ, and an unsafe set Ξ. The eight SynNBC cases retain their source specifications. For C_2, the drift uses the published P1 feedback, while the rectangular sets are specified for this experiment below. The nominal vector field is never learned or refitted in the reported experiments. All probability statements use the worst-case initial semantics

$$B(x) \le \rho \quad \text{for every } x \in \Theta, \tag{124}$$

rather than an expectation under an assumed initial distribution. Thus the same certificate covers every initial point in the stated initial box.

A.6.2 EXPERIMENTAL SETUP

[Trajectory generation]. For every benchmark, we simulate the Euler–Maruyama Higham (2001) update

$$x_{k+1} = x_k + \Delta t f(x_k) + 0.03\sqrt{\Delta t}\,\xi_k, \qquad \xi_k \sim \mathcal{N}(0, I_n), \tag{125}$$

with a fresh independent innovation at every physical time step. The common sampling period is $\Delta t = 0.05$. We request 64 trajectories of horizon 40 and use random seed 7. Initial states are sampled from the stated initial box. If an integration step is clipped to remain in the verification domain, that

Table 3: Correspondence between manuscript identifiers and source systems, with state dimension and nominal drift degree.

Paper ID	Source	Registry ID	State dim. n	Drift degree d_f
C_1 Ratschan & She (2006)	C1	4	2	2
C_2 Zhu et al. (2019)	Ex. 4.3	–	2	3
C_3 Sogokon et al. (2016)	C3	7	2	2
C_4 Sogokon et al. (2016)	C5	9	2	2
C_5 Ratschan & She (2010)	C8	5	3	3
C_6 Klipp et al. (2005)	C12	12	7	2
C_7 Klipp et al. (2005)	C14	13	9	2
C_8 Ratschan (2017)	R1	23	3	3
C_9 Ratschan (2017)	R2	24	5	3

transition is removed before model fitting: the clipping offset is an artifact of the numerical simulator and must not be interpreted as physical noise.

For an accepted transition, the increment remaining after subtracting the known nominal drift is

$$r_k = x_{k+1} - x_k - \Delta t f(x_k). \tag{126}$$

Equation (126) is used only to expose the injected stochastic increment. It does not create an additional learned drift term. The same accepted transitions and trajectory-level train/test split are used by all three pipelines in a given case.

[Rank-one stochastic setting]. To isolate the effect of the uncertainty model and control SOS growth, only the final state coordinate is retained as stochastic during learning and verification. Let $j = n$ and let e_j denote the last Euclidean basis vector. Although (125) perturbs all coordinates during data generation, the learned verification proxy deliberately models only $r_{k,j}$. Its covariance therefore has rank one,

$$\bar{G} = e_j \lambda e_j^\top = \mathrm{diag}(0, \ldots, 0, \lambda). \tag{127}$$

This is a modeling ablation, not an assertion that every source system is physically driven by one-dimensional noise.

[Diffusion-induced stochastic proxy]. The proposed pipeline trains a conditional one-dimensional denoising diffusion model on the selected increments. The context is derived from the state history by a statistics encoder of dimension $\min(4, n)$. The diffusion noise predictor has two tanh hidden layers of width 64 and is trained by Adam with mini-batches of 256, learning rate 5×10^{-4}, a maximum of 300 optimizer iterations, and early stopping after 15 non-improving iterations. Each physical increment is paired with eight independently sampled diffusion training times. The schedule is a 32-step cosine schedule with terminal cumulative signal fraction below 0.003. The selected increments are symmetrized as $\{r, -r\}$ so that the learned physical role is diffusion-only and the exact nominal drift remains unchanged.

Instead of drawing a terminal residual bank, the method propagates quadratic Gaussian moments through all 32 learned reverse updates. A degree-2 polynomial surrogate is fitted at normalized nodes of radius 2 and checked at nodes of radius 3 with diagnostic multiplier 1.25. The reverse propagation returns a conditional terminal covariance. It is evaluated on a deterministic state probe set, and the largest sampled terminal covariance rate over the verification box is retained as λ_{diff}. The main proxy is

$$dX_t = f(X_t)dt + e_j \sqrt{\lambda_{\mathrm{diff}}}\, dW_t. \tag{128}$$

The learned λ_{diff} is treated as an exact parameter of (128). The deterministic-node reverse-fit statistics are retained as diagnostics, not presented as global approximation bounds.

[Experimental Parameters]. Tables 4 and 5 summarize the common settings used in all 9 benchmarks. A separate uncertainty model is trained for each source system, while the three certificate pipelines for that system use the same accepted transitions.

For every accepted candidate, the raw optimizer value is converted to the reported value by

$$\rho_{\mathrm{report}} = \min\{1, \rho_{\mathrm{raw}} + 10^{-3}\}, \qquad P_{\mathrm{safe}}^{\mathrm{LB}} = 1 - \rho_{\mathrm{report}}. \tag{129}$$

Table 4: Complete active settings for trajectory generation and stochastic model learning. PCG64 denotes the bit generator used by NumPy's seeded `default_rng`; all model-side random states are derived from the same master seed.

Data/model parameter	Value	Data/model parameter	Value
Random seed	7	RNG / estimator state	PCG64 / 7
Independent runs per configuration	1	Physical step Δt	0.05
Requested trajectories	64	Trajectory horizon	40
Train/test split	80/20 by trajectory	Generator noise amplitude	0.03
Stochastic coordinate	final coordinate	Diffusion latent dimension	1
History length	4	Context encoder	statistics
Context dimension	$\min(4, n)$	Context reference points	1
Diffusion reverse steps	32	Schedule	cosine
β start/end	$10^{-4}/2 \times 10^{-2}$	Maximum β	0.5
Terminal $\bar{\alpha}$ upper target	0.003	Noise-predictor widths	$(64, 64)$
Activation / optimizer	tanh / Adam	MLP ℓ_2 coefficient	10^{-4}
Training copies per increment	8	Batch size	256
Learning rate	5×10^{-4}	Maximum training iterations	300
Early-stopping patience	15	Increment symmetrization	$\{r, -r\}$
Reverse-polynomial degree	2	Fit/validation radii	2 / 3
Validation multiplier	1.25	Reverse-moment evaluations	32
Random abstraction probes	80	Partitions / validation grid	1 / 2
Terminal PAC calibration	off	Global model-error envelope	off

Table 5: Complete active settings for certificate synthesis and posterior verification. Bernstein candidate/verification subdivisions are $(8, 16)$ for $n \leq 2$, $(3, 6)$ for $n = 3$, $(2, 3)$ for $n = 4$, and $(1, 1)$ for $n \geq 5$. Resource budgets define the "R" entries and do not assert infeasibility.

Certificate parameter	**Value**	**Certificate parameter**	**Value**
Initial semantics	worst case	Objective	minimum ρ
Requested d_B	$2, 4, 6$	Optimizer upper bound on ρ	1
Main dynamics semantics	continuous generator	Gaussian semantics	one-step expectation
Transition degree d_T	$\min(3, \max(2, d_f))$	Transition conditions	300
Successors per condition	32	Expectation correction	empirical standard error
Transition validation multiplier	1.25	Baseline CEGIS rounds	1
Bernstein subdivisions	dimension dependent	Posterior sample states	300
Main SOS relaxation degree	even ceiling of $\max(d_B, d_B - 1 + d_f)$	Discrete relaxation degree	even ceiling of $\max(2, d_B, d_T)$
Nominal SOS margin	10^{-6}	SOS regularization	10^{-7}
SOS sample tolerance	10^{-3}	Coefficient tolerance	10^{-7}
Gram eigenvalue tolerance	10^{-7}	MOSEK interior-point tolerance	solver default
Initial-objective weight	0	Finite-horizon slack (c, T)	$(0, 0)$
Reporting allowance on ρ	10^{-3}	Surrogate-error terms	off (paper-fixed)
Main SOS PSD-scalar budget	12,000	c-SBC PSD-scalar budget	2,000
Bernstein-matrix entry budget	2×10^6	C5 degree-6 refinement	2×10^{-6}
Selection rule	min. ρ, then time	Independent executions	1 per case–method–degree

The 10^{-3} term is a common numerical reporting allowance, not a learned uncertainty parameter. A verified candidate with $\rho_{\mathrm{report}} = 1$ remains a valid but trivial certificate and is reported as a zero lower bound. "R" denotes a pre-solve resource-limit decision and "F" denotes a solver or posterior-check failure.

Remark 1 (Parameter-selection rationale) *The settings in Tables 4 and 5 are shared across cases rather than tuned to individual test results. The common trajectory count and horizon impose the same data budget on all three methods, while the grouped 80/20 split avoids placing adjacent transitions from one rollout on both sides of the split. The rank-one stochastic support isolates uncertainty-model effects and limits the combinatorial growth of the SOS programs. Thirty-two reverse steps are chosen to drive the terminal cumulative signal fraction below 0.003 while retaining a fixed reverse-moment cost; the* $(64, 64)$ *predictor and early-stopping rule give a uniform capacity and optimization budget across state dimensions. The degree-2 reverse surrogate, radii 2 and 3, and multiplier 1.25 provide a common central-fit and wider validation region in normalized diffusion coordinates.*

Barrier degrees $2, 4, 6$ *cover the low polynomial orders commonly used in the benchmark studies while exposing whether additional geometric flexibility improves* ρ*. Higher degrees were not included because the monomial and Gram bases already grow combinatorially over this range. The declared conic and Bernstein budgets make that computational limit method independent and reproducible. No case-specific architecture, learning rate, trajectory count, or degree range is selected. The sole numerical exception is the explicitly marked C5 degree-6 tolerance refinement, which changes the posterior numerical acceptance threshold but not the stochastic model or optimization objective.*

[Randomness, repetitions, and statistical scope]. The master seed is 7. Trajectory generation, trajectory-group splitting, diffusion-time sampling, and the scikit-learn estimator use NumPy's seeded generator or estimator `random_state`; Gaussian-transition probes use deterministic offsets computed from the same seed, the source-case identifier, and d_B. Consequently, rerunning the same configuration reproduces the pseudorandom inputs, although floating-point SDP coefficients and wall-clock times can still vary with solver ordering and machine load.

Each case–method–degree configuration is executed once independently. The eight diffusion times paired with each increment enlarge one training set and are not eight repeated experiments. Likewise, the three barrier degrees are a model-order scan rather than statistical replications. The declared C_4 numerical refinement is a numerical follow-up; these configurations are not pooled as repeated trials. Therefore, Tables 8– 12 report single-run wall-clock times and point estimates.

A.6.3 Learned Stochastic Models

In this section, Table 6 gives the retained transition counts and diffusion optimization statistics. Counts differ by case because domain clipping is filtered. The split is performed by trajectory, so transitions from one trajectory do not appear in both training and test sets.

Table 6: Per-case trajectory filtering and diffusion training statistics. "Diff. examples" includes eight noisy training copies per retained training increment. Training time is excluded from certificate time in the main result table.

Paper ID (source)	Retained	Train	Test	Diff. examples	Iter.	Noise MSE	Incr. MAE	Train (s)
C_1 (C1)	68	57	11	912	63	0.4552	0.00461	0.57
C_2 (Ex. 4.3)	2153	1732	421	27712	48	0.4813	0.00593	4.15
C_3 (C3)	592	493	99	7888	57	0.4767	0.00583	1.44
C_4 (C5)	2368	1887	481	30192	69	0.4758	0.00577	5.73
C_5 (C8)	973	805	168	12880	28	0.4767	0.00537	1.23
C_6 (C12)	766	548	218	8768	43	0.4785	0.00546	4.83
C_7 (C14)	159	129	30	2064	124	0.4566	0.00543	14.01
C_8 (R1)	97	74	23	1184	128	0.4703	0.00573	0.71
C_9 (R2)	124	106	18	1696	105	0.4748	0.00427	1.65

And, Table 7 reports the scalar nonzero entry of the rank-one covariance matrix. The diffusion and Gaussian values are estimated by different procedures and are not expected to coincide. In particular, a larger learned diffusion covariance makes the Hessian term more restrictive, but it does not alone determine the optimized probability bound.

Table 7: Learned one-dimensional covariance rates. The data generator uses amplitude 0.03 in every coordinate; the listed verification covariances are estimated from the retained final-coordinate data and are not hard-coded.

Paper ID (source)	j	Probe states	λ_{diff}	$\sqrt{\lambda_{\mathrm{diff}}}$	λ_{G}
C_1 (C1)	2	89	0.006423	0.08014	0.000806
C_2 (Ex. 4.3)	2	89	0.004568	0.06759	0.000862
C_3 (C3)	2	89	0.019796	0.14070	0.000898
C_4 (C5)	2	89	0.003863	0.06215	0.000871
C_5 (C8)	3	107	0.003476	0.05896	0.000957
C_6 (C12)	7	580	0.032581	0.18050	0.000913
C_7 (C14)	9	1223	0.019197	0.13855	0.000853
C_8 (R1)	3	107	0.006018	0.07758	0.000995
C_9 (R2)	5	323	0.009617	0.09806	0.001080

A.7 Supplementary Quantitative Results

A.7.1 Best result for each manuscript case

Table 8 reproduces the large experimental table from the main manuscript and appends the *SynNBC* and Zhu et al. (2019) source identifier. The time is certificate construction and verification time for the selected degree; it does not include trajectory generation or diffusion training.

Table 8: Best verified result among $d_B \in \{2, 4, 6\}$ for the nine manuscript cases. Time is in seconds. The dagger marks the explicitly declared 2×10^{-6} numerical refinement for source C5; all other main SOS entries use the default tolerance.

				Diffusion + GenSOS			Additive + Neural			Gaussian + c-SBC		
Paper	Source	n	d_f	d_B^*	$1-\rho^*$	t	d_B^*	$1-\rho^*$	t	d_B^*	$1-\rho^*$	t
C_1	C1	2	2	6	0.8703	1.24	6	0.8739	11.10	6	0.8855	13.10
C_2	Ex. 4.3	2	3	6	0.9976	1.21	6	0.9593	13.89	6	0.9635	15.85
C_3	C3	2	2	6	0.7186	1.35	6	0.3003	10.39	6	0.3005	12.33
C_4	C5	2	2	$6^\dagger$	0.9980	1.29	6	0.9967	11.15	6	0.9981	12.73
C_5	C8	3	3	6	0.9924	8.72	4	0.8130	22.41	6	0.9038	219.98
C_6	C12	7	2	2	0.9990	5.40	2	0.5900	19.58	2	0.9048	21.69
C_7	C14	9	2	2	0.9988	16.22	–	–	–	2	0.8002	298.20
C_8	R1	3	3	4	0.6891	2.22	4	0.5061	19.64	6	0.8174	209.12
C_9	R2	5	3	4	0.6895	23.67	4	0.4134	265.35	4	0.6379	290.34

A.7.2 Degree scan for the proposed method

Table 9 reports the lower bound and time at every requested degree. Higher degree is not guaranteed to be selected because the larger relaxation may exceed the computational limit or fail the posterior numerical check. For C_2, the degree-two solve fails, while degrees four and six pass the posterior numerical check.

The 7-dimensional C_6 case makes the resource effect especially visible: all three quadratic programs return accepted certificates, while degrees 4 and 6 are rejected before solving because their estimated conic representations exceed the common resource budget. Thus the "R" marks should not be interpreted as a degradation of the quadratic certificate.

A.8 Comparison with Baselines

We compare *DiffSBC* with two representative uncertainty-modeling and probabilistic-verification strategies. The comparisons use the same benchmark definitions and barrier-degree settings whenever applicable. Following Mathiesen et al. (2022), the first baseline, denoted by *Gauss.+Neural*, represents the learned uncertainty as a fixed additive disturbance. The disturbance is obtained by

Table 9: Degree scan for Diffusion + Generator-SOS. "F" denotes a failed solver/posterior check and "R" a pre-solve resource rejection.

Paper	Source	$d_B = 2$		$d_B = 4$		$d_B = 6$	
		$1-\rho$	t	$1-\rho$	t	$1-\rho$	t
C_1	C1	0	0.18	0.7805	0.56	0.8703	1.24
C_2	Ex. 4.3	F	0.23	0.9547	0.56	0.9976	1.21
C_3	C3	0	0.16	0.5579	0.56	0.7186	1.35
C_4	C5	F	0.19	F	0.48	$0.9980^{\dagger}$	1.29
C_5	C8	0	0.38	0.9085	2.00	0.9924	8.72
C_6	C12	0.9990	5.40	R	–	R	–
C_7	C14	0.9988	16.22	R	–	R	–
C_8	R1	0.2398	0.38	0.6891	2.22	F	8.85
C_9	R2	0.2404	1.73	0.6895	23.67	R	–

averaging diffusion-inferred disturbances over M unit-time intervals,

$$\bar{\delta}_{\text{diff}} = \frac{1}{M}\sum_{m=1}^{M}\delta_{\text{diff}}^{(m)}, \qquad \dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{e}_j\bar{\delta}_{\text{diff}}.$$

This baseline therefore incorporates the learned uncertainty through a deterministic additive term rather than through a stochastic diffusion process.

Following Hu et al. (2026), the second baseline, denoted by *Gauss.+c-SBC*, constructs a data-driven discrete-time stochastic model. Multiple successor states are sampled for each state $\mathbf{x}_t$, and the resulting transition residuals are fitted by

$$\delta_t \sim \mathcal{N}\left(\mu_{\text{G}}(\mathbf{x}_t), \Sigma_{\text{G}}(\mathbf{x}_t)\right), \qquad \mathbf{x}_{t+1} = \Phi_{\Delta t}(\mathbf{x}_t) + \delta_t.$$

The resulting discrete-time stochastic system is verified using the corresponding c-SBC conditions Prajna et al. (2004). In contrast to *DiffSBC*, this baseline directly fits a Gaussian transition model from sampled successor states rather than extracting the uncertainty moments through the learned reverse diffusion dynamics.

A.8.1 Main Comparison of Example 1

We first compare the three methods using the same quadratic barrier degree so that differences in the certified safety bounds are not caused by increased certificate complexity. The results for Example 1 are reported in Table 10. *DiffSBC* achieves a failure bound of $\rho^\star = 0.001012$, corresponding to the certified infinite-horizon safety lower bound $1-\rho^\star = 0.998988$. Under the same barrier-degree setting, *Gauss.+Neural* and *Gauss.+c-SBC* obtain safety lower bounds of 0.589986 and 0.904764, respectively. For this example, the comparison indicates that the diffusion-based uncertainty representation and its covariance-matched stochastic abstraction yield a higher certified safety lower bound without increasing the barrier degree. Since the three methods employ barriers of the same degree in this comparison, the difference arises from their uncertainty representations and associated verification formulations rather than from using a more expressive certificate class.

Table 10: Comparison of probabilistic safety verification results for Example 1.

Metric	*DiffSBC*	*Gauss.+Neural*	*Gauss.+c-SBC*
d_B^*	2	2	2
ρ^*	0.001012	0.410014	0.095236
$1-\rho^*$	0.998988	0.589986	0.904764
Time (s)	5.40	19.58	21.69

Note. Gauss.+Neural Mathiesen et al. (2022) represents uncertainty using the averaged diffusion-inferred disturbance $\bar{\delta}_{\text{diff}} = M^{-1}\sum_{m=1}^{M}\delta_{\text{diff}}^{(m)}$ in $\dot{\mathbf{x}} = f(\mathbf{x}) + \mathbf{e}_j\bar{\delta}_{\text{diff}}$. *Gauss.+c-SBC* Hu et al. (2026) fits sampled transition residuals as $\delta_t \sim \mathcal{N}(\mu_{\text{G}}(\mathbf{x}_t), \Sigma_{\text{G}}(\mathbf{x}_t))$ and verifies $\mathbf{x}_{t+1} = \Phi_{\Delta t}(\mathbf{x}_t) + \delta_t$ using discrete-time c-SBC conditions.

A.8.2 DEGREE-WISE COMPARISON

To further examine the effect of certificate complexity, we scan the barrier degrees $d_B \in \{2, 4, 6\}$ for the two baseline methods. Tables 11 and 12 report the resulting certified safety lower bounds together with the corresponding wall-clock times. Each entry is reported as the safety lower bound followed by the verification time in seconds in parentheses. A value of zero is retained when the verifier accepts only a trivial certificate, whereas "R" denotes a case for which no reported result is obtained under the prescribed computational setting.

Table 11: Degree-wise safety lower bounds and computation times for *Gauss.+Neural*. Values in parentheses denote wall-clock time in seconds.

Paper	Source	$d_B = 2$	$d_B = 4$	$d_B = 6$
C_1	C1	0.4164 (0.34)	0.4973 (2.15)	0.8739 (11.10)
C_2	Ex. 4.3	0 (0.42)	0.8944 (2.41)	0.9593 (13.89)
C_3	C3	0 (0.29)	0 (1.95)	0.3003 (10.39)
C_4	C5	0.6771 (0.58)	0.9664 (2.22)	0.9967 (11.15)
C_5	C8	0 (1.40)	0.8130 (22.41)	R
C_6	C12	0.5900 (19.58)	R	R
C_7	C14	R	R	R
C_8	R1	0.2389 (1.35)	0.5061 (19.64)	R
C_9	R2	0.1134 (5.06)	0.4134 (265.35)	R

Table 12: Degree-wise safety lower bounds and computation times for *Gauss.+c-SBC*. Values in parentheses denote wall-clock time in seconds.

Paper	Source	$d_B = 2$	$d_B = 4$	$d_B = 6$
C_1	C1	0.4316 (0.50)	0.5079 (2.71)	0.8855 (13.10)
C_2	Ex. 4.3	0 (0.98)	0.9063 (3.15)	0.9635 (15.85)
C_3	C3	0 (0.45)	0 (2.64)	0.3005 (12.33)
C_4	C5	0.7015 (0.73)	0.9716 (2.87)	0.9981 (12.73)
C_5	C8	0 (2.30)	0.8583 (22.34)	0.9038 (219.98)
C_6	C12	0.9048 (21.69)	R	R
C_7	C14	0.8002 (298.20)	R	R
C_8	R1	0.2500 (2.23)	0.5389 (20.68)	0.8174 (209.12)
C_9	R2	0.2396 (11.75)	0.6379 (290.34)	R

A.8.3 POSTERIOR NUMERICAL DIAGNOSTICS

Table 13 reports numerical diagnostics for the selected Generator-SOS certificates. "PSD" counts scalar entries represented by all positive-semidefinite Gram variables, while "Eq." counts coefficient-matching equalities. The residual is the largest conic/coefficient violation and $\lambda_{\min}$ is the smallest recovered Gram eigenvalue.

Table 13: Posterior numerical diagnostics for the selected proposed-method certificates. The C5 source case is the sole explicitly refined entry.

Paper	Source	d_B	Relax.	PSD	Eq.	Max residual	$\lambda_{\min}$
C_1	C1	6	8	920	180	4.62×10^{-8}	1.64×10^{-8}
C_2	Ex. 4.3	6	8	920	180	1.56×10^{-8}	-1.52×10^{-8}
C_3	C3	6	8	920	180	9.55×10^{-9}	-7.14×10^{-9}
C_4	C5	6	8	920	180	1.07×10^{-6}	-1.03×10^{-6}
C_5	C8	6	8	5040	660	5.74×10^{-8}	-5.60×10^{-8}
C_6	C12	2	4	3672	1320	4.22×10^{-9}	-4.04×10^{-9}
C_7	C14	2	4	8140	2860	2.25×10^{-8}	-2.20×10^{-8}
C_8	R1	4	6	1500	336	7.52×10^{-9}	9.43×10^{-8}
C_9	R2	4	6	11004	1848	7.35×10^{-9}	-6.12×10^{-9}

A.9 BENCHMARK-WISE RESULTS

This section records the implemented dynamics and boxes used in Table 1. Intervals are Cartesian products in the displayed coordinate order. For every case, the stochastic coordinate is the last coordinate and the rank-one proxy covariance is given in Table 7.

A.9.1 BENCHMARK C_1

The first case is a two-dimensional quadratic system,

$$f(x) = \begin{bmatrix} -5.5x_2 + x_2^2 \\ 6x_1 - x_1^2 \end{bmatrix}. \tag{130}$$

Its sets are

$$\Psi = [1, 5]^2, \tag{131}$$
$$\Theta = [4, 4.5] \times [0.9, 1.1], \tag{132}$$
$$\Xi = [1, 2] \times [2, 3]. \tag{133}$$

Only 68 transitions remain after clipping rejection, making this the smallest training set among the low-dimensional C-series cases. The diffusion pipeline learns $\lambda_{\text{diff}} = 0.006423$, whereas the empirical Gaussian fit gives $\lambda_{\text{G}} = 0.000806$. The proposed bound rises from zero at degree two to 0.7805 at degree four and 0.8703 at degree six. The two Gaussian methods select degree six and report 0.8739 and 0.8855. The result illustrates why all requested degrees are evaluated even when the quadratic program is feasible: a feasible quadratic certificate can still be probabilistically trivial.

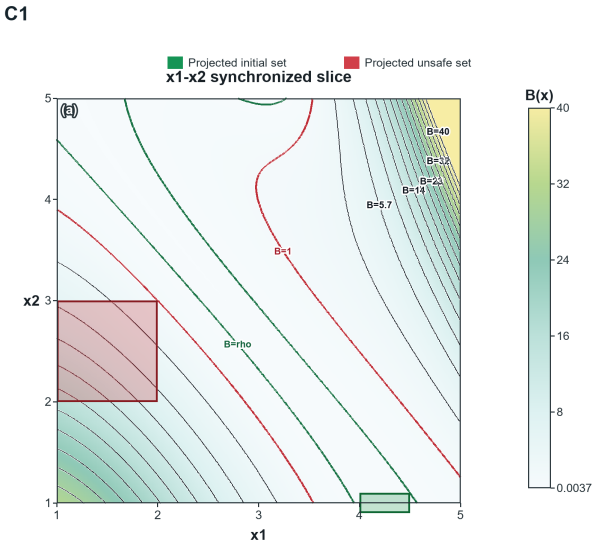


Figure 3: Main-method barrier for paper C_1. The heatmap shows $B(x)$ on the full two-dimensional domain; dense contours expose its geometry, and the green/red boxes are the initial/unsafe sets. The emphasized green and red contours mark $B = \rho_{\text{raw}}$ and $B = 1$, respectively.

A.9.2 BENCHMARK C_2: EXAMPLE 4.3

The source is Example 4.3 of Zhu et al. (2019). Substituting the published P1 feedback $a = 0.39x_1 - 1.41x_2$ gives

$$f(x) = \begin{bmatrix} x_2 \\ -0.6x_2 - x_1 - x_1^3 + 0.39x_1 - 1.41x_2 \end{bmatrix}, \tag{134}$$

with

$$\Psi = [-6, 6]^2, \tag{135}$$

$$\Theta = [-2.5, 2.5] \times [-2, 2], \tag{136}$$

$$\Xi = [5, 6]^2. \tag{137}$$

These rectangular sets specify this experiment; the reported certificate applies to the displayed corner unsafe box. The clipping filter retains 2,153 transitions, of which 1,732 are assigned to training. The learned diffusion and Gaussian covariance rates are 0.004568 and 0.000862. At degree six, the proposed method gives a 0.9976 lower bound, versus 0.9593 and 0.9635 for the Gaussian Neural and c-SBC methods.

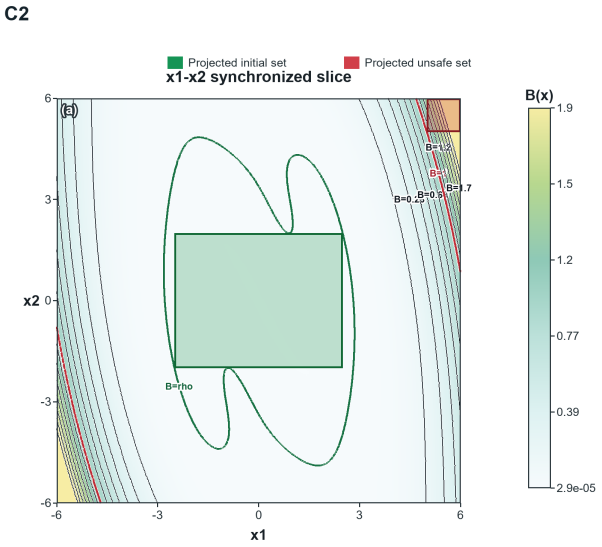


Figure 4: Main-method degree-6 barrier for paper C_2 (Example 4.3 of Zhu et al. (2019)). The heatmap shows the full two-dimensional domain; the green and red boxes are the initial and restricted unsafe sets. The emphasized contours mark $B = \rho_{\text{raw}}$ and $B = 1$, respectively.

A.9.3 BENCHMARK C_3

The third case is

$$f(x) = \begin{bmatrix} -1 + x_1^2 + x_2^2 \\ 5(-1 + x_1 x_2) \end{bmatrix}, \tag{138}$$

on $\Psi = [-2, 2]^2$, with $\Theta = [-1, 0]^2$ and $\Xi = [1, 2]^2$. The training/test split contains 493/99 transitions. This case has $\lambda_{\text{diff}} = 0.019796$, more than twenty times the fitted Gaussian rate 0.000898. Nevertheless, the degree-6 Generator-SOS certificate yields a 0.7186 lower bound, while both Gaussian discrete-time methods yield about 0.300. The comparison should not be read as a monotone covariance experiment: the methods use different stochastic proxies and different continuous- versus discrete-time certificate constraints.

A.9.4 BENCHMARK C_4

The fourth manuscript case is

$$f(x) = \begin{bmatrix} -2x_1 + x_1^2 + x_2 \\ x_1 - 2x_2 + x_2^2 \end{bmatrix}, \tag{139}$$

with $\Psi = [-1, 1]^2$, $\Theta = [-0.1, 0.1]^2$, and $\Xi = [0.5, 1]^2$. It supplies the largest retained data set among the selected cases: 1,887 training and 481 test transitions. The covariance rates are 0.003863 (diffusion) and 0.000871 (Gaussian). Only the sextic proposed candidate becomes nontrivial. Its set and generator sample violations are zero, while the conic residual and minimum Gram eigenvalue

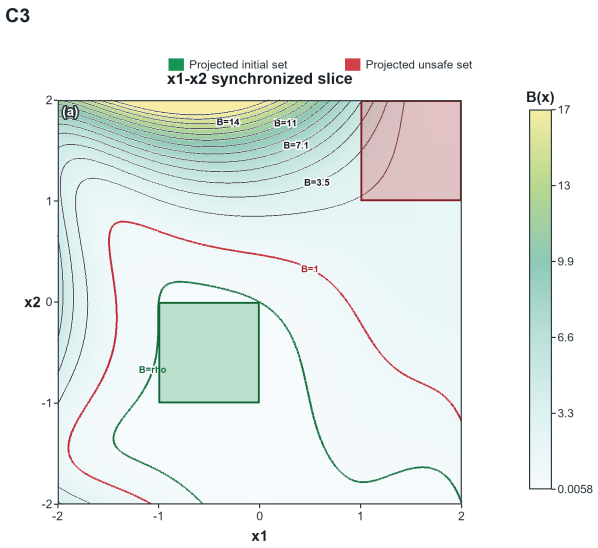


Figure 5: Main-method degree-6 barrier for paper C_3. This is a full-domain two-dimensional evaluation of the stored, posterior-accepted polynomial; the set overlays and emphasized separation levels use the same convention as Fig. 3.

are approximately 1.07×10^{-6} and -1.03×10^{-6}. We therefore report it with the explicit 2×10^{-6} refinement marker rather than silently applying a looser tolerance to the entire benchmark.

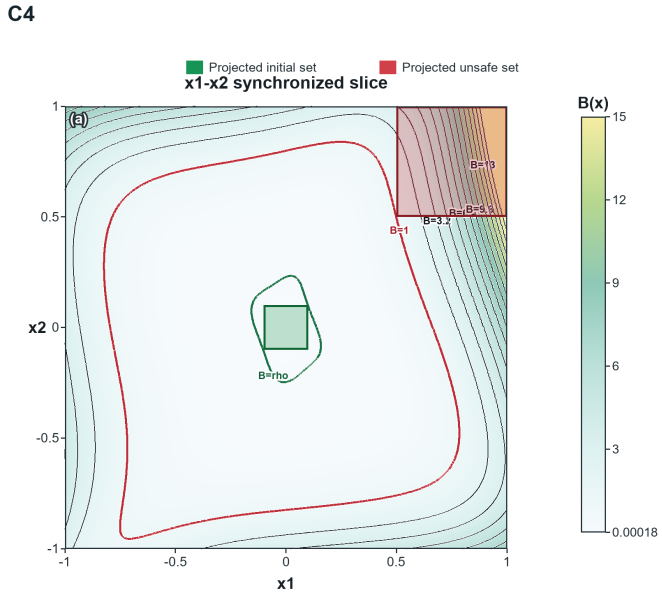


Figure 6: Refined degree-6 main-method barrier for paper C_4. The plot is generated from the retained unrounded coefficients accepted under the explicitly reported numerical refinement.

A.9.5 BENCHMARK C_5

The fifth case is the three-dimensional cubic system

$$f(x) = \begin{bmatrix} -x_2 \\ -x_3 \\ -x_1 - 2x_2 - x_3 + x_1^3 \end{bmatrix}. \tag{140}$$

The domain is $\Psi = [-2, 2]^3$ and

$$\Theta = [-0.25, 0.75] \times [-0.25, 0.75] \times [-0.75, 0.25], \tag{141}$$
$$\Xi = [1, 2] \times [-2, -1] \times [-2, -1]. \tag{142}$$

The model uses 805 training and 168 test transitions. Its learned covariance rates are 0.003476 and 0.000957. The proposed degree scan improves from zero to 0.9085 and then 0.9924. The Neural comparison becomes resource-limited at degree six and therefore selects its quartic 0.8130 result. The multi-candidate c-SBC verifier completes degree six but requires 219.98 seconds, compared with 8.72 seconds for the selected proposed certificate.

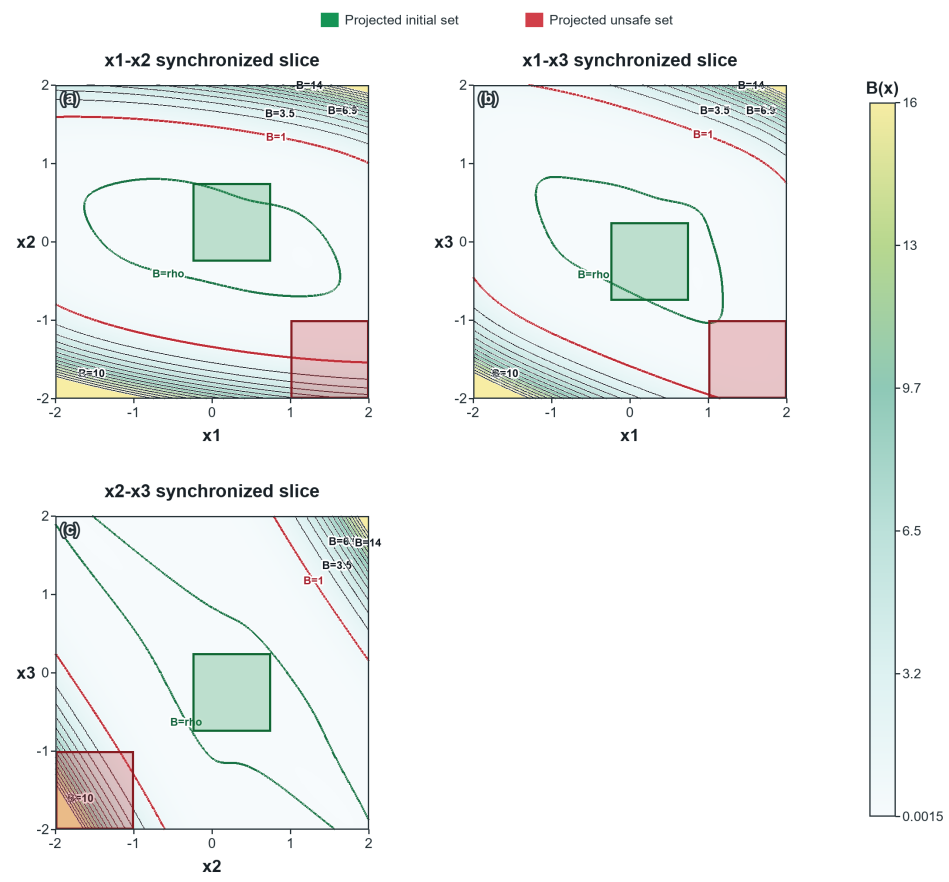


Figure 7: Main-method degree-6 barrier for paper C_5. Because the system is three-dimensional, the panels show synchronized projections onto (x_1, x_2), (x_1, x_3), and (x_2, x_3). They are visual diagnostics of the full three-dimensional SOS certificate.

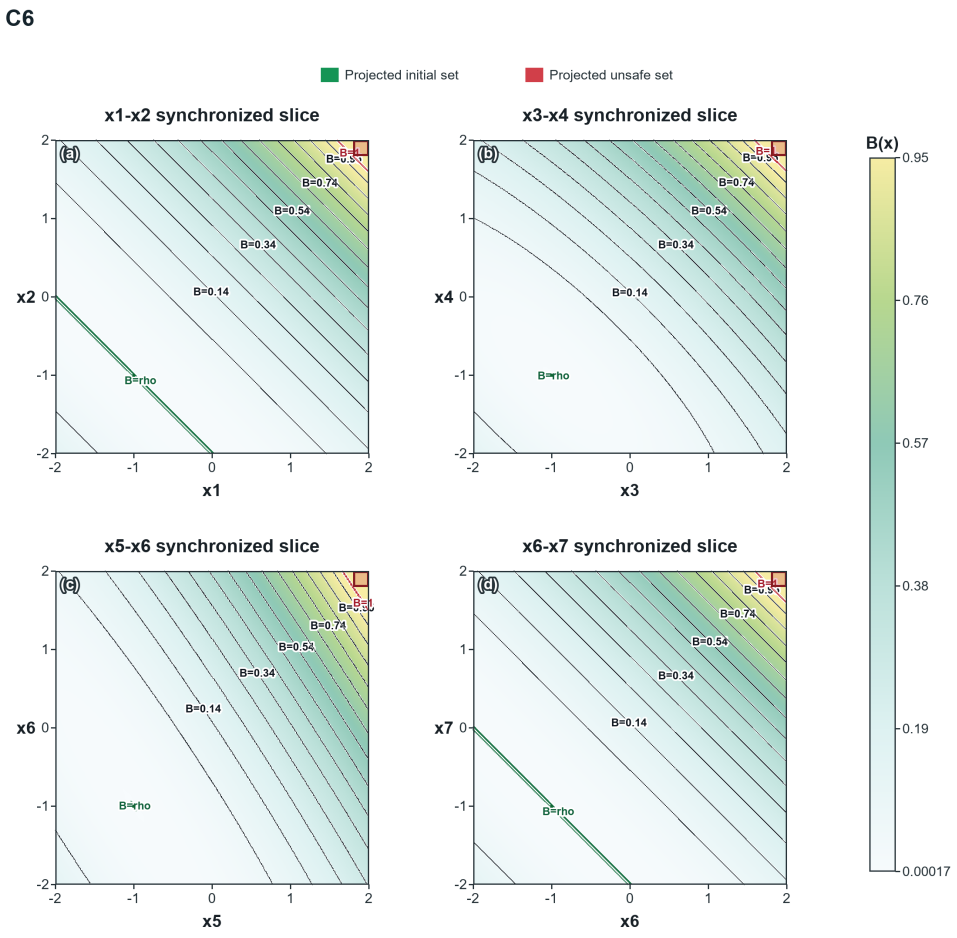


Figure 8: Accepted barrier certificate from C_6. The four panels show synchronized slices through the upstream, nonlinear-coupling, and downstream stochastic coordinates. Dense contours show B and the colored boxes show projected state sets. The SOS and posterior checks remain seven-dimensional; these slices are explanatory visualizations only.

A.9.6 BENCHMARK C_6

This is the seven-dimensional case study used in the main paper. Its exact quadratic drift is

$$\begin{aligned} f(x) = (&-0.4x_1 + 5x_3x_4,\ 0.4x_1 - x_2,\ x_2 - 5x_3x_4,\\ &5x_5x_6 - 5x_3x_4,\ -5x_5x_6 + 5x_3x_4,\\ &0.5x_7 - 5x_5x_6,\ -0.5x_7 + 5x_5x_6)^\top. \end{aligned} \tag{143}$$

The domain, initial set, and unsafe set are $[-2,2]^7$, $[-1.01,-0.99]^7$, and $[1.8,2]^7$. The clipping filter leaves 766 transitions, split into 548/218. The learned diffusion covariance rate is 0.032581, corresponding to amplitude 0.180501 on x_7, while the Gaussian rate is 0.000913.

Define $s_1 = x_1 + x_2 + x_3$, $s_2 = x_4 + x_5$, and $s_3 = x_6 + x_7$. The accepted quadratic proposed barrier, rounded to six decimal places for display, is

$$\begin{aligned} B(x) = {} & 0.127680 + 0.049029s_1 + 0.053284s_2\\ & + 0.000853s_3\\ & + 0.006999(x_1^2 + x_2^2 + x_3^2)\\ & + 0.013998(x_1x_2 + x_1x_3 + x_2x_3)\\ & + 0.003358s_1s_2 + 0.000160s_1s_3\\ & + 0.010712(x_4^2 + x_5^2) + 0.021423x_4x_5\\ & + 0.000182s_2s_3 + 0.000002(x_6^2 + x_7^2)\\ & + 0.000004x_6x_7. \end{aligned} \tag{144}$$

The unrounded 36 coefficients are used for every posterior check. The optimizer returns $\rho_{\mathrm{raw}} = 1.2356 \times 10^{-5}$; after the common reporting allowance, $\rho = 0.001012$ and the lower bound is 0.998988. The SDP has 3,672 scalar Gram variables and 1,320 coefficient equalities. The maximum coefficient residual is 4.22×10^{-9}, the minimum recovered Gram eigenvalue is -4.04×10^{-9}, and the largest sampled generator residual is 1.02×10^{-7}. Quartic and sextic programs exceed the stated resource budget rather than disproving the existence of higher-degree certificates.

The figure makes two aspects of the certificate visible. First, the initial projection lies near the low barrier level ρ_{raw}, whereas the unsafe projection approaches the unit separation level. Second, the contour orientation changes across coupled coordinate pairs even though the displayed quadratic coefficients have repeated symmetric groups. A two-dimensional slice cannot establish the global inequalities in Definition 3; it is therefore used only after the full SOS and posterior checks have succeeded.

A.9.7 BENCHMARK C_7

The seventh case is the largest selected system, with nine states and quadratic drift,

$$\begin{aligned} f_1 &= 3x_3 - x_1x_6, & f_2 &= x_4 - x_2x_6,\\ f_3 &= x_1x_6 - 3x_3, & f_4 &= x_2x_6 - x_4,\\ f_5 &= 3x_3 + 5x_1 - x_5,\\ f_6 &= 5x_5 + 3x_3 + x_4\\ &\quad - x_6(x_1 + x_2 + 2x_8 + 1),\\ f_7 &= 5x_4 + x_2 - 0.5x_7,\\ f_8 &= 5x_7 - 2x_6x_8 + x_9 - 0.2x_8,\\ f_9 &= 2x_6x_8 - x_9. \end{aligned} \tag{145}$$

Here $\Psi = [-2,2]^9$, $\Theta = [0.99,1.01]^9$, and $\Xi = [1.8,2]^9$. Only 159 unclipped transitions remain, so the diffusion model has 129 training transitions and runs for 124 iterations. The proposed quadratic program contains 8,140 scalar PSD entries and 2,860 coefficient equalities, yet completes in 16.22 seconds with lower bound 0.9988. The Neural comparison is rejected by its declared Bernstein resource budget. The quadratic c-SBC candidate is verified with lower bound 0.8002 but requires 298.20 seconds.

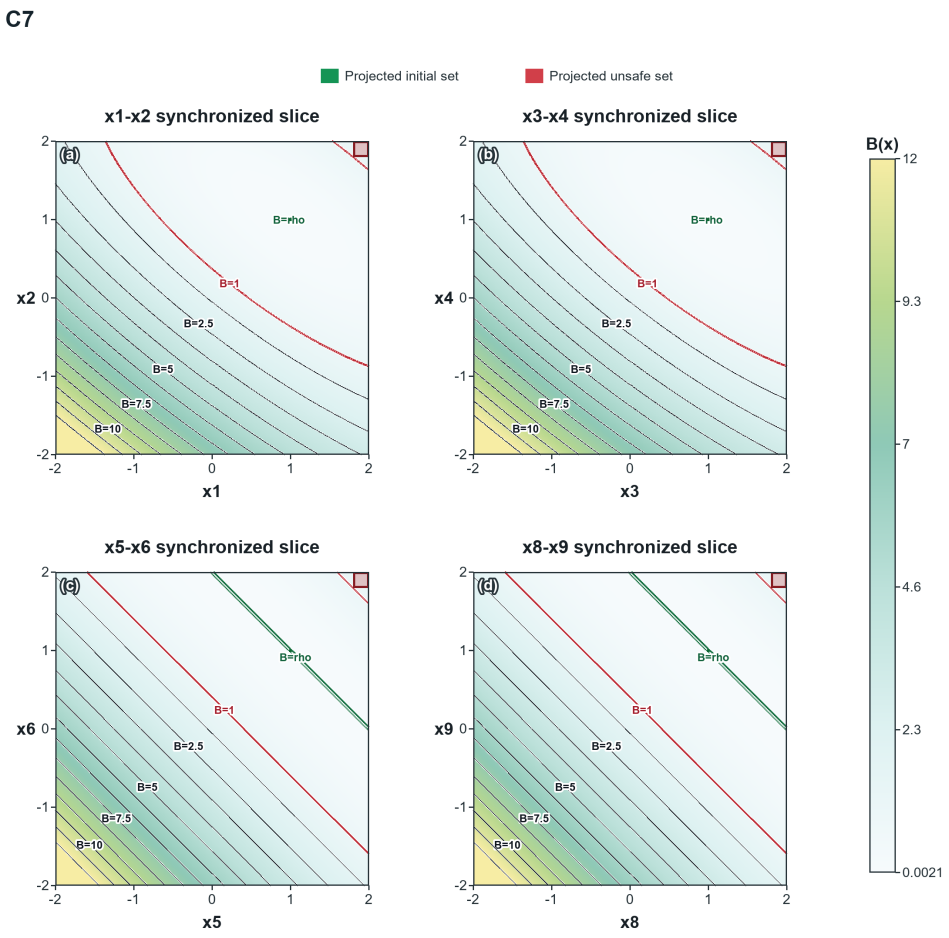


Figure 9: Main-method quadratic barrier for paper C_7. Four synchronized coordinate-pair slices summarize the nine-dimensional certificate, including a downstream slice containing the stochastic state x_9.

A.9.8 BENCHMARK C_8

The eighth case is

$$f(x) = \begin{bmatrix} 1 + (x_2 + x_3)/100 \\ x_3 \\ -10(x_2 - x_2^3/6) - x_2 \end{bmatrix}. \tag{146}$$

Its domain is $[-0.3, 0.3]^3$, while

$$\Theta = [-0.3, 0] \times [-0.2, 0.3]^2, \tag{147}$$

$$\Xi = [-0.2, -0.15] \times [-0.3, -0.25]^2. \tag{148}$$

After filtering, 74 training and 23 test transitions remain. The proposed quartic barrier gives 0.6891. The sextic proposed candidate fails its posterior check and is not selected. Gaussian Neural selects degree four with 0.5061, whereas multi-candidate c-SBC selects degree six with 0.8174 at a much higher 209.12-second cost.

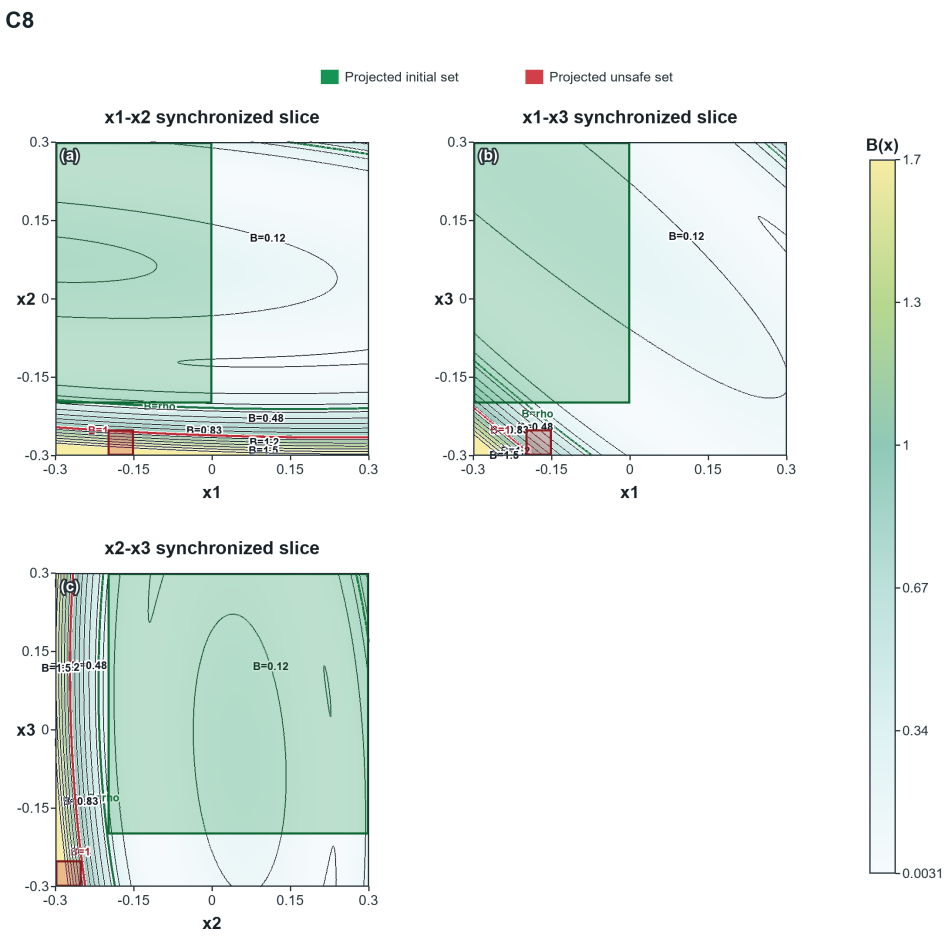


Figure 10: Main-method quartic barrier for paper C_8. The three pairwise slices visualize the stored three-state polynomial and the projected initial/unsafe boxes on the compact domain $[-0.3, 0.3]^3$.

A.9.9 BENCHMARK C_9

The nineth case is

$$\begin{aligned} f_1 &= 1 + (x_2 + 2x_3 + x_4)/100, \\ f_2 &= x_3, \\ f_3 &= -10(x_2 - x_2^3/6) - x_2, \\ f_4 &= x_5, \\ f_5 &= -10(x_4 - x_4^3/6) - x_2. \end{aligned} \tag{149}$$

The domain is $[-0.3, 0.3]^5$ and

$$\Theta = [-0.3, 0] \times [-0.2, 0.3]^4, \tag{150}$$

$$\Xi = [-0.2, -0.15] \times [-0.3, -0.25]^4. \tag{151}$$

The split contains 106 training and 18 test transitions. The learned diffusion covariance rate is 0.009617 and the Gaussian rate is 0.001080. The proposed quartic certificate has 11,004 scalar PSD entries and yields 0.6895 in 23.67 seconds. Both comparison methods select quartic barriers; Neural gives 0.4134 in 265.35 seconds and c-SBC gives 0.6379 in 290.34 seconds. All sextic programs are rejected by their declared resource budgets.

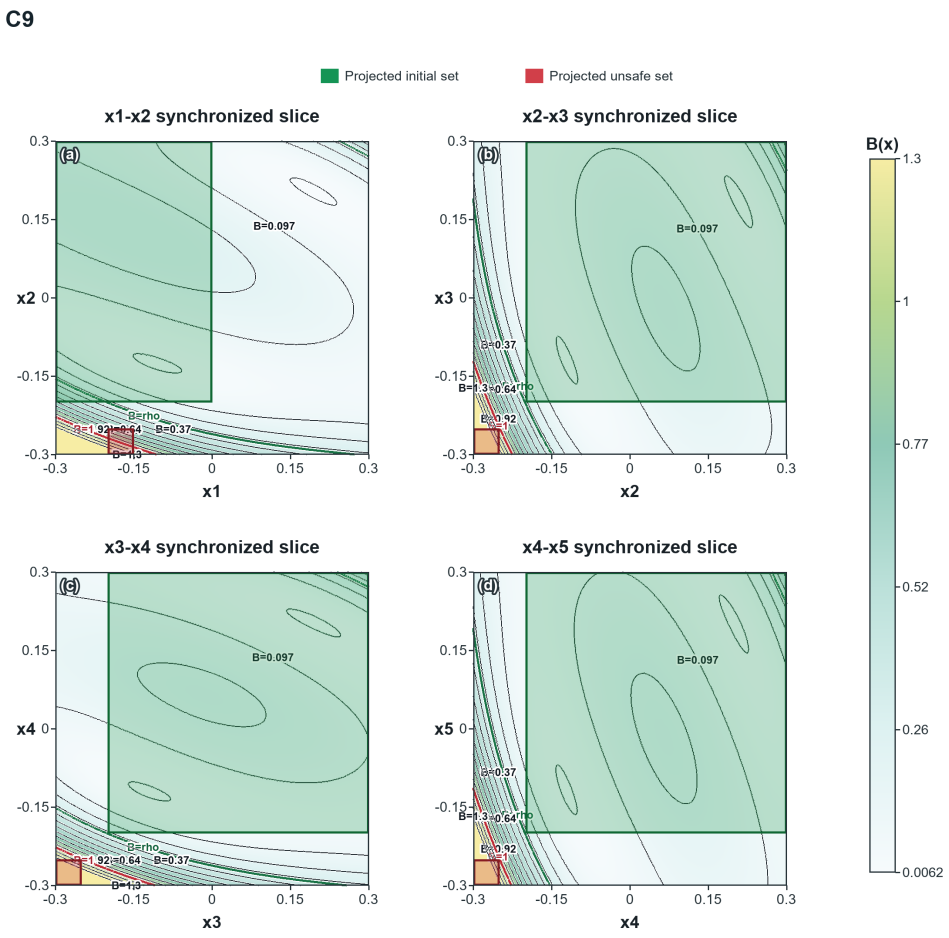


Figure 11: Main-method quartic barrier for paper C_9. The four synchronized slices span adjacent state pairs and include the stochastic coordinate x_5. The plotted contours are evaluations of the retained five-dimensional polynomial certificate.

A.10 Discussion an Limitations

The degree scans reveal a trade-off between certificate expressiveness and computational complexity. While quadratic barriers are sufficient for some benchmarks, higher-degree certificates are required to capture more complex barrier geometries in others. However, the number of polynomial basis terms grows combinatorially with both the system dimension and barrier degree, with additional complexity introduced by SOS Gram matrices and localizing multipliers. Consequently, higher-degree verification may become resource-limited for higher-dimensional systems, motivating the selection of the best successfully verified degree rather than fixing a common degree across all benchmarks. The reported verification time covers candidate construction, optimization, and posterior verification, but excludes diffusion-model training. Finally, SOS solutions are accepted only after posterior numerical checks on polynomial identities, Gram-matrix eigenvalues, and sampled certificate conditions, thereby reducing the risk of reporting numerically spurious certificates.