SYNC: SAFETY-AWARE NEURAL CONTROL FOR STABILIZING STOCHASTIC DELAY-DIFFERENTIAL EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Stabilization of the systems described by *stochastic delay*-differential equations (SDDEs) under preset conditions is a challenging task in control community. Here, to achieve this task, we leverage neural networks to learn control policies using the information of the controlled systems in some prescribed regions. Specifically, two learned control policies, i.e., the neural deterministic controller (NDC) and the neural stochastic controller (NSC), work effectively in that the learning procedures rely on, respectively, the well-known LaSalle-type theorem and the newly-established theorem for guaranteeing the stochastic stability in SDDEs. We theoretically investigate the performance of the proposed controllers in terms of convergence time and energy cost. More practically and significantly, we improve our learned control policies through considering the situation where the controlled trajectories only evolve in some specific safety set. The practical validity of such control policies restricted in safety set is attributed to the theory that we further develop for safety and stability guarantees in SDDEs using the stochastic control barrier function and the spatial discretization. We call this control as SYNC (**S**afet**Y**-aware **N**eural **C**ontrol). The efficacy of all the articulated control policies, including the SYNC, is demonstrated systematically by using representative control problems.

1 INTRODUCTION

Stochastic delay-differential equations (SDDEs) (Mao, 1996; Lin & He, 2005; Sun & Cao, 2007; Guo et al., 2016) have been widely applied to characterize the complex dynamical behavior emergent in real-world systems with dependence on the current state, the past state, and the noise. Efficiently controlling these systems is a long-standing and crucial problem, with the consequent emphasis being placed on the design of control policies and analysis of stability in SDDEs. Traditional control methods in stochastic settings have been fully developed in the convex optimization frameworks using the control Lyapunov stability theory, e.g. the quadratic programming (QP) (Fan et al., 2020; Sarkar et al., 2020). These methods cannot provide the analytical form of feedback controllers and own a high computational cost, requiring solving QP problems at each iteration step. To overcome these difficulties, utilizing neural networks (NNs) to automatically design controllers becomes one of the mainstream approaches in recent years (Zhang et al., 2022; Chang et al., 2019). However, existing machine-learning-based methods either focus on controlling systems without time-delay or aim at learning the control Lyapunov function instead of the control policy (Khansari-Zadeh & Billard, 2014). All these therefore motivate us to design neural controllers for general nonlinear SDDEs.

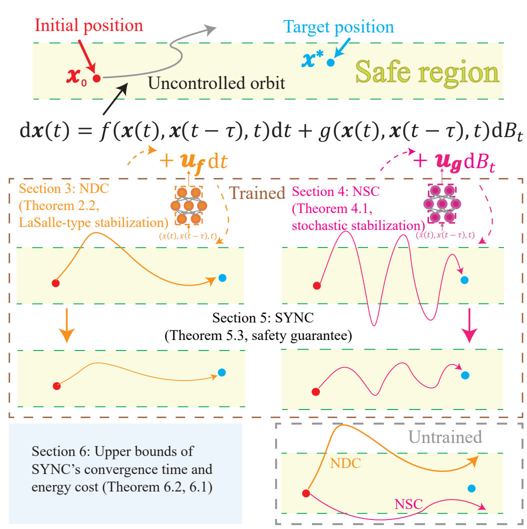

Figure 1: Overall work flow.

The safety verification of controlled systems plays an important role in many branches of cybernetics and industry. For example, with the safety verification, one can reduce a significant economic burden and loss of life (Ames et al., 2016; Wang et al., 2016). In particular, the dominant framework for safety control in stochastic settings is the use of stochastic control barrier function (SCBF) (Clark, 2019; 2021; Santoyo et al., 2021). The core idea of designing a candidate SCBF is that its value tends to explode as the system's state leaves the safe region, implying a safety guarantee as long as one could design a controller such that the SCBF is always finite within the controlled time duration. Unfortunately, the existing theories of SCBF either require a lot of inequality constraints or are limited in handling systems without any time delay.

In this paper, we utilize neural networks (NNs) to learn control policies for SDDEs based on the corresponding stability theories. Additionally, we develop a simplified SCBF theory for SDDEs and then use it to construct the neural controller with safety guarantee, named SYNC. All these control policies are intuitively depicted in Figure 1. The major contributions of this paper include:

- designing a novel and practical framework of neural deterministic control based on the existing LaSalle-Type stability theory,
- proposing a simplified stability theorem and designing the second novel neural stochastic control framework that can benefit from noise according to this theorem,
- establishing an SCBF theory for SDDEs as well as a theory of safety guarantee and stability guarantee using neural network settings,
- providing theoretical estimation for proposed neural controller in terms of convergence time and energy cost based on the developed theory of safety and stability guarantees, and
- demonstrating the efficacy of the proposed neural control methods through numerical comparisons with the typical existing control methods on several representative physical systems.

2 Preliminaries

To begin with, we consider the SDDE in a general form of

$$\mathrm{d}\boldsymbol{x}(t) = F(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)\mathrm{d}t + G(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)\mathrm{d}B_t,\ t \geq 0,\ \tau > 0,\ \boldsymbol{x}(t) \in \mathbb{R}^d, \quad (1)$$

where $\boldsymbol{x}(t) = \xi(t) \in C_{\mathcal{F}_0}([-\tau, 0]; \mathbb{R}^d)$ is the initial function, the drift term $F : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ and the diffusion term $G : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^{d\times r}$ are Borel-measurable functions, and B_t is a standard r-dimensional (r-D) Brownian motion defined on probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ with a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ satisfying the regular conditions. Without loss of generality, we assume that $F(\mathbf{0}, \mathbf{0}, t) = \mathbf{0}$ and $G(\mathbf{0}, \mathbf{0}, t) = \mathbf{0}$. This assumption guarantees that the zero solution $\boldsymbol{x}(t) \equiv \mathbf{0}$ with $t \geq 0$ is an equilibrium of Eq. (1). Additionally, the following notations and assumptions are used throughout the paper.

Assumption 2.1 *Assume that Eq. (1) has a unique solution $\boldsymbol{x}(t, \xi)$ on $t \geq 0$ for any $\xi \in C_{\mathcal{F}_0}([-\tau, 0]; \mathbb{R}^d)$ and that, for every integer $n \geq 1$, there is a number $K_n > 0$ such that*

$$\|F(\boldsymbol{x}, \boldsymbol{y}, t)\| \vee \|G(\boldsymbol{x}, \boldsymbol{y}, t)\|_{\mathrm{F}} \leq K_n,$$

for any $(\boldsymbol{x}, \boldsymbol{y}, t) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+$ with $\|\boldsymbol{x}\| \vee \|\boldsymbol{y}\| \leq n$, where $\|\cdot\|$ denotes the L^2-norm and $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, i.e. $\|G(\boldsymbol{x}, \boldsymbol{y}, t)\|_{\mathrm{F}}^2 = \sum_{i=1}^{d}\sum_{j=1}^{r} G_{ij}(\boldsymbol{x}, \boldsymbol{y}, t)^2$.

Definition 2.1 *(Derivative Operator) Define the differential operator $\mathcal{L}$ associated with Eq. (1) by*

$$\mathcal{L} \triangleq \frac{\partial}{\partial t} + \sum_{i=1}^{d} F_i(\boldsymbol{x}, \boldsymbol{y}, t)\frac{\partial}{\partial x_i} + \frac{1}{2}\sum_{i,j=1}^{d} [G(\boldsymbol{x}, \boldsymbol{y}, t)G^{\top}(\boldsymbol{x}, \boldsymbol{y}, t)]_{ij}\frac{\partial^2}{\partial x_i \partial x_j}.$$

According to the above definition of the derivative operator, an operation of $\mathcal{L}$ on the function $V \in C^{2,1}(\mathbb{R}^d \times \mathbb{R}_+; \mathbb{R})$ yields:

$$\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}, t) = V_t(\boldsymbol{x}, t) + \nabla V(\boldsymbol{x}, t)^{\top} F(\boldsymbol{x}, \boldsymbol{y}, t) + \frac{1}{2}\mathrm{Tr}\left[G^{\top}(\boldsymbol{x}, \boldsymbol{y}, t)\mathcal{H}V(\boldsymbol{x}, t)G(\boldsymbol{x}, \boldsymbol{y}, t)\right]. \quad (2)$$

Here, V_t, ∇V and $\mathcal{H}V$ represent, respectively, the time derivative, the gradient and the Hessian matrix of V. Notably, the following LaSalle-type stability theorem will be crucial to the establishment of our partial results.

Theorem 2.2 *(Mao, 2002) Suppose that Assumptions 2.1 holds. Assumes there are functions* $V \in C^{2,1}(\mathcal{X} \times \mathbb{R}_+; \mathbb{R}_+)$, $\gamma \in L^1(\mathbb{R}_+; \mathbb{R}_+)$, *and* $w_1, w_2 \in C(\mathcal{X}; \mathbb{R}_+)$ *such that* $\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}, t) \leq \gamma(t) - w_1(\boldsymbol{x}) + w_2(\boldsymbol{y})$, $w_1(\boldsymbol{x}) \geq w_2(\boldsymbol{x})$, *and* $\lim_{\|\boldsymbol{x}\|\to\infty} \inf_{0\leq t\leq\infty} V(\boldsymbol{x}, t) = \infty$. *Here,* $\mathcal{X} \subset \mathbb{R}^d$ *is the state space. Then,* $\mathrm{Ker}(w_1 - w_2) \neq \varnothing$ *and* $\lim_{t\to\infty} \mathrm{dist}(\boldsymbol{x}(t, \xi), \mathrm{Ker}(w_1 - w_2)) = 0$ *a.s., where* $\mathrm{Ker}(w_1 - w_2) \triangleq \{\boldsymbol{x} : w_1(\boldsymbol{x}) - w_2(\boldsymbol{x}) = 0\}$, $\mathrm{dist}(x, K) \triangleq \inf_{y\in K} \|x - y\|$ *for a set* $K \subseteq \mathbb{R}^d$, *and a.s. stands for the abbreviation of almost surely.*

Problem Statement We assume that the zero solution of the following SDDE:

$$\mathrm{d}\boldsymbol{x}(t) = f(\boldsymbol{x}, \boldsymbol{x}(t-\tau), t)\mathrm{d}t + g(\boldsymbol{x}, \boldsymbol{x}(t-\tau), t)\mathrm{d}B_t \tag{3}$$

is unstable, i.e. $\lim_{t\to\infty} \boldsymbol{x}(t;\xi) \neq \boldsymbol{0}$ on some set of positive measures. We aim to stabilize the zero solution using control based on neural networks (NNs). In other words, our goal is to leverage the NNs to design an appropriate controller $\boldsymbol{u} = (\boldsymbol{u}_f, \boldsymbol{u}_g)$ with $\boldsymbol{u}_f(\boldsymbol{0}, \boldsymbol{0}, t) = \boldsymbol{u}_g(\boldsymbol{0}, \boldsymbol{0}, t) = \boldsymbol{0}$ such that the controlled system

$$\mathrm{d}\boldsymbol{x} = [f + \boldsymbol{u}_f(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)]\mathrm{d}t + [g + \boldsymbol{u}_g(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)]\mathrm{d}B_t \tag{4}$$

is steered to the zero solution. We call $\boldsymbol{u}_f : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ as deterministic control while we call $\boldsymbol{u}_g : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^{d\times r}$ as stochastic control, since they are integrated with $\mathrm{d}t$ and $\mathrm{d}B_t$, respectively. The major difficulty of this problem comes from the non-Markovian property of SDDEs. As such, we cannot apply the Markov decision process (MDP)-based methods, such as the reinforcement learning, to control SDDEs. Majority of existing works prefer to learning deterministic control and often regard the noise as a negative ingredient that may destroy the natural dynamics of f. In what follows, we not only show that the deterministic control can achieve stabilization in a probability sense, but also that elaborately-designed stochastic control can make the same stabilization. This therefore yields two frameworks, viz., the neural deterministic control (Section 3) and the neural stochastic control (Section 4). We make all our code and data available at `https://anonymous.4open.science/r/SYNC-35E8`.

3 Neural Deterministic Control

In this section, we propose neural deterministic controller (NDC) based on the Theorem 2.2 to stabilize system (3). Heuristically, we construct neural network form auxiliary functions and control function, and integrate the sufficient conditions in theorem into the loss function to find the neural controller that satisfies the expected conditions. However, the NDC can neither be used to find stochastic controllers nor rigorously satisfy the expected stability conditions. These problems will be addressed in Section 4 and 5.

3.1 Method: Learning Control and Auxiliary Functions

The core idea of our method is base on using Theorem 2.2, that is, once we construct the auxiliary functions $V,\ \gamma,\ w_1,\ w_2$ and the neural controller $\boldsymbol{u}$ to meet all the conditions assumed in Theorem 2.2 for the controlled system (4), the solution $\boldsymbol{x}(t;\xi)$ converges to the $\mathrm{Ker}(w_1 - w_2)$. In particular, if we set $\mathrm{Ker}(w_1 - w_2) = \{\boldsymbol{0}\}$, the unstable zero solution of the control-free system (3) can be stabilized. To this end, we first provide appropriate constructions of NNs to learn these candidate functions. Thus, we design the explicit form of loss function in the learning step.

Auxiliary Function We employ a multi-layer feedforward neural network, denoted by $\mathbf{NN}(\cdot;\theta)$, to design all the functions. Precisely, θ_1 is the parameter vector of the positive function $V(\boldsymbol{x}, t; \theta_1)$, and the L_2 term $\|\boldsymbol{x}\|^2$ is added to guarantee $\lim_{\|\boldsymbol{x}\|\to\infty} \inf_{0\leq t<\infty} V(\boldsymbol{x}, t; \theta_1) = \infty$, that is

$$V(\boldsymbol{x}, t; \theta_V) = \mathbf{NN}(\boldsymbol{x}, t; \theta_V)^2 + \varepsilon\|\boldsymbol{x}\|^2, \ \ \varepsilon > 0. \tag{5}$$

In our framework, it requires $V \in C^{2,1}(\mathbb{R}^d \times \mathbb{R}_+)$. We therefore use a C^2 activation function for an NN, such as the hyperbolic tangent function, $\mathrm{Tanh}(\cdot)$. We further discuss the impact of the L_2 term in Appendix A.1.3. In order to design an integrable positive function $\gamma(t)$ with the NN, we use an activation function with at most linear growth such as ReLU and multiply an exponential decay factor to the output of the NN, that is

$$\gamma(t; \theta_\gamma) = \exp(-ct) \cdot \mathbf{NN}(t; \theta_\gamma)^2, \ \ c > 0. \tag{6}$$

For simplicity, we design $w(\boldsymbol{x}, \theta_w) = \mathbf{NN}(\boldsymbol{x}; \theta_w)^2$ as a positive function. Additionally, we set

$$w_2 = w, \;\; w_1 = w + p(x), \;\; p \geq 0, \;\; \mathrm{Ker}(p) = \{\mathbf{0}\}. \tag{7}$$

Deterministic Control Function We first consider the deterministic control, i.e. $\boldsymbol{u} = (\boldsymbol{u}_f, \mathbf{0})$. To guarantee the same zero solution of the control-free system (3) and the controlled system (4), the NDC $\boldsymbol{u}_f : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}^d$ should satisfy $\boldsymbol{u}_f(\mathbf{0}, \mathbf{0}, t) = \mathbf{0}$. One feasible way to meet such a condition is to set $\boldsymbol{u}_f(\boldsymbol{x}, \boldsymbol{y}, t) = \mathbf{NN}(\boldsymbol{x}, \boldsymbol{y}, t; \theta_f) - \mathbf{NN}(\mathbf{0}, \mathbf{0}, t; \theta_f)$ or $\boldsymbol{u}_f(\boldsymbol{x}, \boldsymbol{y}, t) = \mathrm{diag}(\boldsymbol{x})\mathbf{NN}(\boldsymbol{x}, \boldsymbol{y}, t; \theta_f)$. Here, $\mathrm{diag}(\boldsymbol{x})$ is a diagonal matrix with x_i as its i-th diagonal element.

Loss Function Once the learned functions V, γ, w_1, w_2 and $\boldsymbol{u}$ with the coefficient functions, $f_{\boldsymbol{u}} \triangleq f + \boldsymbol{u}$ and g, in the controlled system (4), meet all the conditions assumed in Theorem 2.2, the stability of zero solution is naturally assured. To achieve this, we demand a suitable loss function to evaluate the likelihood that those conditions are satisfied. It can be seen from our construction that the only condition needed to be satisfied is $\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}, t) \leq \gamma(t) - w_1(x) + w_2(y)$. Hence, we define LaSalle's loss function for the controlled system (4) as follows.

Definition 3.1 (LaSalle's Loss) Consider the above parameterized candidate functions V, γ, w_1, w_2 and a controller $\boldsymbol{u}_f$ for the controlled system (4). Then, LaSalle's loss is defined as

$$L_{N,\varepsilon,c,p}(\boldsymbol{\theta}_V, \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_w, \boldsymbol{\theta}_f) \;\; = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, \mathcal{L}V(\boldsymbol{x}_i, \boldsymbol{y}_i, t_i) - \gamma(t_i) + w_1(\boldsymbol{x}_i) - w_2(\boldsymbol{y}_i)\right), \tag{8}$$

where $\{\boldsymbol{x}_i, \boldsymbol{y}_i, t_i\}_{i=1}^N$ are sampled from some distribution μ on $\mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+$.

In summary, the developed NDC framework is shown in Algorithm 1 in Appendix A.3.1.

Remark 3.1 *The proposed NDC framework can be easily applied to the autonomous SDDE:* $\mathrm{d}\boldsymbol{x}(t) = f(\boldsymbol{x}, \boldsymbol{x}(t-\tau))\mathrm{d}t + g(\boldsymbol{x}, \boldsymbol{x}(t-\tau))\mathrm{d}B_t$. *In particular, one can simply consider the autonomous auxiliary function V and the control function, and set $\gamma(t) = 0$. For sample distribution $\mu(\Omega)$, here we select the uniform distribution on a sufficiently large and closed region Ω as used in (Han et al., 2016; Chang et al., 2019), and we include further analyses for the impact of μ in Appendix A.2.1.*

3.2 Numerical and Analytical Investigations

Comparison Studies Recent works on controlling time-delayed systems mainly focus on elaborately designing the analytical form of control to satisfy the conditions in the LaSalle-Type Theorem 2.2 (Lin & He, 2005; Xu et al., 2014), or simultaneously designing control and the Lyapunov function to satisfy the conditions based on the Lyapunov theory (Yu & Cao, 2007). It should be noted that all these methods require a delicate design of functions for specific dynamics, and thus are limited in practical application for controlling general time-delayed systems. However, our neural method leverages NNs to automatically learn the control policies, and can be applied in any kinds of time-delayed systems with stochastic settings. In Figure 3, we numerically compare the NDC and a baseline, the linear control (LC) proposed in (Lin & He, 2005), on a noised driving-response Chua's circuit. Here, Chua's circuit is a three dimensional autonomous dynamical system with a unique nonlinear element, producing typical chaotic dynamics (Matsumoto, 1984). In the simulation, we show that the NDC can find the neural control for the response system $\boldsymbol{y} = (y_1, y_2, y_3)$ with the autonomous and even the nonautonomous time-delay noise. Actually, the nonautonomous time-delay noise was not considered in (Lin & He, 2005). The simulation configurations are described in Appendix A.3.4.

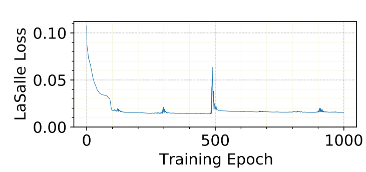

Figure 2: Training loss for the 1-D SDDE.

Failure in Finding Stochastic Control As we can see that the NDC performs well, a natural idea is to utilize the noise part to achieve the stabilization of the SDDE (3). To explore this idea, we adopt the same NN of $\boldsymbol{u}_f$, design $\boldsymbol{u}_g = \mathbf{NN}(\boldsymbol{x}, \boldsymbol{y}, t; \boldsymbol{\theta}_g)$, and train its parameters $\boldsymbol{\theta}_g$ with LaSalle's loss (8). However, in Figure 2, we show that the loss cannot converge to zero in controlling a simple 1-D toy system via the stochastic controller $\boldsymbol{u}_g$: $\mathrm{d}x(t) = [x(t) + x(t-\tau)]\mathrm{d}t + [x(t-\tau) + u_g(x(t), x(t-\tau); \theta_g)]\mathrm{d}B_t$. Actually, this phenomena can be analytically explained. Notice that $\boldsymbol{\theta}_g$ arises in loss function as a quadratic term $l(\boldsymbol{\theta}_g) = \frac{1}{2}\mathrm{Tr}[\boldsymbol{u}_g^\top \mathcal{H}V\boldsymbol{u}_g]$ according to

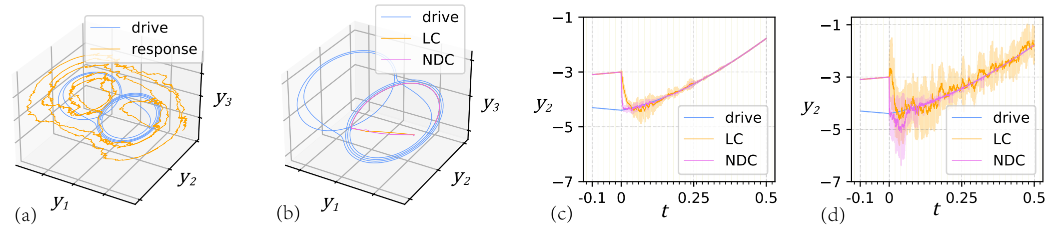

Figure 3: (a) The original driving-response model, (b) the controlled orbits under LC and NDC, (c) the time trajectory of y_2 with autonomous noise, and (d) the nonautonomous noise. The solid lines are obtained through averaging the 10 sampled trajectories, while the shaded areas stand for the standard errors.

Eq. (2), the sign of this term depends on the convexity of V, i.e. the maximum eigenvalue's sign of $\mathcal{H}V$. Nevertheless, the positive function V with $\lim_{\|\boldsymbol{x}\|\to\infty} V(\boldsymbol{x},t)=\infty$ implies $l(\boldsymbol{\theta}_g)\geq 0$ for most of time. Hence, when we minimize $l(\boldsymbol{\theta}_g)\geq 0$ in the training procedure, the ideal case $l(\boldsymbol{\theta}_g)=0$ is equivalent to $\boldsymbol{u}_g=0$. This indicates that we are unable to learn a stochastic controller under LaSalle's loss (8) satisfying the sufficient conditions assumed in Theorem 2.2.

4 Neural Stochastic Control

To find the neural stochastic controller (NSC), we provide the following theoretical result on stabilization of general stochastic functional differential equations (SFDEs) with the proof provided in Appendix A.1.4. The major idea is to construct sufficient condition for stability that makes the diffusion term contribute a negative number, contrary to the Theorem 2.2.

Theorem 4.1 *(Stochastic Stabilization) Consider the SFDE* $\mathrm{d}\boldsymbol{x}(t)=F(\boldsymbol{x}_t,t)\mathrm{d}t+G(\boldsymbol{x}_t,t)\mathrm{d}B(t)$, *with* F,G *being locally Lipschitzian functions,* $F(\boldsymbol{0},t)=\boldsymbol{0}$, *and* $G(\boldsymbol{0},t)=\boldsymbol{0}$. *For every* $M>0$, *assume that* $\min_{\|\boldsymbol{x}_t(0)\|=M}\|\boldsymbol{x}_t(0)^\top G(\boldsymbol{x}_t,t)\|>0$. *If there exists a number* $\alpha\in(0,1)$ *such that*

$$\|\boldsymbol{x}_t(0)\|^2(2\langle\boldsymbol{x}_t(0),F(\boldsymbol{x}_t,t)\rangle+\|G(\boldsymbol{x}_t,t)\|_{\mathrm{F}}^2)-(2-\alpha)\|\boldsymbol{x}_t(0)^\top G(\boldsymbol{x}_t,t)\|^2\leq 0, \tag{9}$$

for $\boldsymbol{x}_t\in C([-\tau,0],\mathcal{X})$, *where* $\boldsymbol{x}_t(s)=\boldsymbol{x}(t+s)$ *for* $s\in[-\tau,0]$ *and* $\mathcal{X}$ *is the state space. Then, the solution of the SFDE satisfies* $\lim_{t\to\infty}\boldsymbol{x}(t;\xi)=\boldsymbol{0}$ *a.s. for any* $\xi\in C_{\mathcal{F}_0}([-\tau,0];\mathbb{R}^d)$.

Remark 4.2 *The SFDE in Theorem 4.1 is formulated in a very general type, including the SDDE* $\mathrm{d}\boldsymbol{x}(t)=F(\boldsymbol{x}(t),\boldsymbol{x}(t-\tau_1),\cdots,\boldsymbol{x}(t-\tau_q),t)\mathrm{d}t+G(\boldsymbol{x}(t),\boldsymbol{x}(t-\tau_1),\cdots,\boldsymbol{x}(t-\tau_q),t)\mathrm{d}B_t$ *with* $\tau_1<\tau_2<\cdots<\tau_q\in[0,\tau]$. *This indicates that our framework can be generalized to stabilize the SDDEs with multiple delays and even more general SFDEs as well.*

In light of Theorem 4.1, we establish a more general framework for learning a neural controller of system (4) with the form $\boldsymbol{u}=(\boldsymbol{u}_f,\boldsymbol{u}_g)$ designed in the same NN architecture as the one used in the NDC framework. We focus on stochastic control with $\boldsymbol{u}_f=\boldsymbol{0}$ and provide more control combinations in Appendix A.3.3, whereas the loss function is differently designed as follows.

Definition 4.1 *(*Asymptotic Loss) Utilize the notations set in Definition 3.1 and $g_{\boldsymbol{u}}=g+\boldsymbol{u}_g$. The loss function for the controlled system (4) with the controller $\boldsymbol{u}$ is defined as:

$$L_{\mu,\alpha}(\boldsymbol{\theta})=\frac{1}{N}\sum_{i=1}^{N}\left[\max\left(0,(\alpha-2)\|\boldsymbol{x}_i{}^\top g_{\boldsymbol{u}}(\boldsymbol{x}_i,\boldsymbol{y}_i,t_i)\|^2+\|\boldsymbol{x}_i\|^2(2\langle\boldsymbol{x}_i,f(\boldsymbol{x}_i,\boldsymbol{y}_i,t_i)\rangle+\|g_{\boldsymbol{u}}(\boldsymbol{x}_i,\boldsymbol{y}_i,t_i)\|_{\mathrm{F}}^2)\right)\right], \tag{10}$$

where $\boldsymbol{\theta}=(\boldsymbol{\theta}_f,\boldsymbol{\theta}_g)$. Akin to Definition 3.1, we use the empirical loss function for training.

Here, α is an adjustable parameter, which is related to the convergence rate and the control energy. We further discuss the design of the asymptotic loss in Appendix A.2.2 and numerically investigate the role of α in Appendix A.4.1. We summarize the framework in Algorithm 3 in Appendix A.3.1. And we further compare the computational complexity in Appendix A.3.2.

4.1 Experiments of the Combination Methods

We compare our neural control methods on a noise-perturbed kinematic bicycle model for car-like vehicles (Rajamani, 2011) in terms of the convergence time and the energy cost, which are two important indexes to measure the quality of a controller (Yan et al., 2012; Li et al., 2017; Sun et al., 2017).

To quantify the energy cost in the control process, we first denote by $\tau_\epsilon \triangleq \inf\{t > 0 : \|\boldsymbol{x}(t)\| = \epsilon\}$ the stopping time and then by $\mathcal{E}_\epsilon \triangleq \mathbb{E}\left[\int_0^{\tau_\epsilon} \left(\|\boldsymbol{u}_f\|^2 + \|\boldsymbol{u}_g\|^2\right) \mathrm{d}t\right]$ the energy cost. We approximate this expectation value by the empirical value as $\frac{1}{N}\sum_{i=1}^{N} \int_0^{\tau_\epsilon^i} \left(\|\boldsymbol{u}_f^i\|^2 + \|\boldsymbol{u}_g^i\|^2\right) \mathrm{d}t$ through the Monte Carlo sampling. We show the results in Figure 4 and in Table 1 as well. Table 1 includes the training time (Tt), empirical energy cost $\mathcal{E}_{0.001}$, nearest distance (Nd) between the bicycle and target position, and empirical expectation $\mathbb{E}[\tau_{0.001}]$ for different methods. We include more experimental details in Appendix A.3.5. We can see that the ranking of the comprehensive performance is NSC $>$ NDC $>$ QP. This means that we can really benefit from introducing noise in control protocol. This is reasonable because, when we regard the energy cost as an objective function for minimization, the randomness is more likely to lead this functional to the shortest path, akin to the common case where the stochastic gradient descent outperforms the full-batch gradient descent. We show the NSC can enlarge the region of attraction of the 100-D gene regulatory networks in Appendix A.4.2.

Table 1: Results on kinematic bicycle model.

	Tt	$\mathcal{E}_{0.001}$	Nd	$\mathbb{E}[\tau_{0.001}]$
NDC	1028.81s	102.17	6.3e-4	1.81
NSC	**59.80**s	**62.10**	4.0e-7	0.29
QP	-	-	0.016	>5

Uncontrollable Fluctuation Although the neural stochastic method we propose outperforms the control methods including the deterministic control, obviously observed is a disadvantage that the method can cause uncontrollable fluctuation due to the stochasticity. However, we always want to bound this perturbation in practical application owing to physical and engineering restrictions in real world. We tackle this safety guarantee problem for our methods in Section 5.

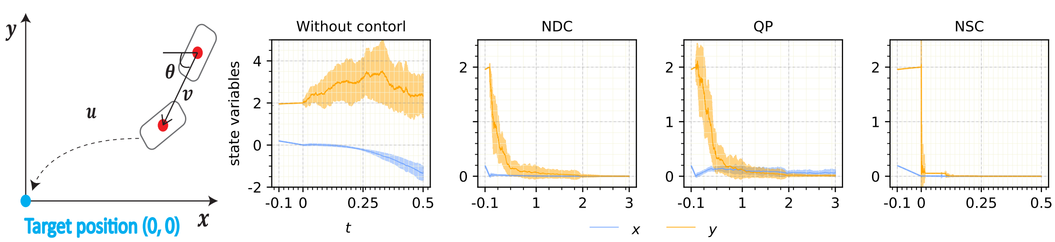

Figure 4: (Left) A schematic diagram of the kinematic bicycle model. (Right) Time trajectories of the state variables x, y of the kinematic bicycle under different control cases. The solid lines are obtained through averaging the 10 sampled trajectories, while the shaded areas stand for the standard errors.

5 Safety Guarantee for SDDEs

In this section, we study the safety and stability guarantees for the SYNC framework. Based on the stochastic control barrier functions, we establish an analytical result on the safety guarantee problem for SDDEs, which guarantees that the process $\boldsymbol{x}(t;\xi)$ satisfies the safety constraint, i.e., $\boldsymbol{x}(t;\xi) \in \mathrm{int}(\mathcal{C})$ for all t with the initial value $\xi(0) \in \mathrm{int}(\mathcal{C})$. Here, $\mathcal{C} = \{\boldsymbol{x} : h(\boldsymbol{x}) \geq 0\}$ is a compact set and the local Lipschitz function h: $\mathbb{R}^d \to \mathbb{R}$ is called a stochastic control barrier function (SCBF). Inspired by (Lechner et al., 2022), we prove that the safety and stability conditions for NN form functions can be guaranteed through a stronger condition on finite samples. We include the analytical proofs for all the results in Appendix A.1.

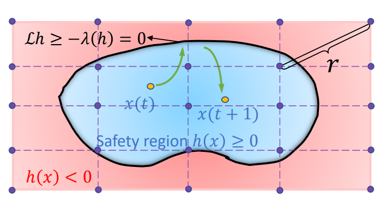

Figure 5: Diagram of the safety guarantee. We check the safety condition on discretization points with mesh r.

Definition 5.1 *A continuous function $\alpha : (-b, +\infty) \to (-\infty, +\infty)$ is said to be of an extended class-$\mathcal{K}$ function for some $b > 0$ if it is strictly increasing and $\alpha(0) = 0$.*

Baseline We extend the recent results on stochastic control barrier functions in SDEs (Clark, 2019) to the SDDEs and summarize the results in Proposition 5.1. With this proposition and Theorem 2.2, the traditional deterministic control methods based on the Quadratic Program (QP) in (Fan et al., 2020; Sarkar et al., 2020) can be applied to test on the SDDEs. We use this QP method as the baseline and the specific algorithm is shown in Appendix A.3.1. We also take the classic MPC method as the baseline.

Proposition 5.1 *Let the function $\mathcal{B}$: $\mathbb{R}^d \to \mathbb{R}$ be locally Lipschitz and twice-differentiable on $\mathrm{int}(\mathcal{C})$. If there exist three extended class-$\mathcal{K}$ functions $\alpha_{1,2,3}(\boldsymbol{x})$ such that $[\alpha_1(h(\boldsymbol{x}))]^{-1} \le \mathcal{B}(\boldsymbol{x}) \le [\alpha_2(h(\boldsymbol{x}))]^{-1}$, and $\mathcal{L}\mathcal{B}(\boldsymbol{x}, \boldsymbol{y}, t) \le \alpha_3(h(\boldsymbol{x}))$ for the SDDE in (1). Then, $\mathbb{P}\big(\boldsymbol{x}(t) \in \mathrm{int}(\mathcal{C})\big) = 1$ for all t, provided with $\boldsymbol{x}(0) \in \mathrm{int}(\mathcal{C})$.*

A natural idea is to integrate Proposition 5.1 into our proposed neural control framework, but the main drawback in the usage of this proposition is that $\mathcal{B}(\boldsymbol{x})$ is unbounded on $\mathcal{C}$, lacking Lipschitz continuity. This drawback makes it impossible to fulfill the expected conditions only through numerical verification on finite samples. To conquer the difficulty, we propose the following theorem for safety guarantee, which, we believe, is a significantly promotion to the existing barrier function theory.

Theorem 5.2 *For the SDDE specified in (1), where F and G satisfy locally Lipschitz condition and locally linear growth condition, if there exists an extended class-$\mathcal{K}$ function $\lambda(x)$ such that $\mathcal{L}h \ge -\lambda \circ h$ for $\boldsymbol{x} \in \mathcal{D}$, where $\circ$ represents the function composition, $\mathcal{D}$ is compact and $\mathcal{C} \subset \mathcal{D}$. Then, the solution satisfies $\mathbb{P}(\boldsymbol{x}(t;\xi) \in \mathrm{int}(\mathcal{C})) = 1$ for any $\xi \in C_{\mathcal{F}_0}([-\tau, 0]; \mathbb{R}^d)$ with $\xi(0) \in \mathrm{int}(\mathcal{C})$.*

Discretization and Safety Guarantee. Based on the Theorem 5.2, we can construct a neural candidate class-$\mathcal{K}$ function λ and combine it with the NDC and NSC to learn a safe controller, where the candidate λ is required to satisfy the condition assumed in Theorem 5.2. However, the main difficulty is to guarantee the condition for every point $\boldsymbol{x} \in \mathcal{D}$, since, in practice, we can basically guarantee this condition on a finite number of training data $\tilde{\mathcal{D}}$ with $\tilde{\mathcal{D}}$ being a discretization of $\mathcal{D}$. Surprisingly, the following theorem suggests that we only need to check a slightly stronger condition on a finite number of states in $\tilde{\mathcal{D}}$ in order to establish the safety guarantee on the whole $\mathcal{D}$.

Theorem 5.3 *Let $M = \mathcal{M}(F, G, h, \lambda, \mathcal{D})$ be the maximum of the Lipschitz constants of $\mathcal{L}h$ and $\lambda \circ h$ on $\mathcal{D}$. Also let r be the mesh size of $\tilde{\mathcal{D}}$. Thus, for each $\boldsymbol{x} \in \mathcal{D}$, there exists $\tilde{\boldsymbol{x}} \in \tilde{\mathcal{D}}$ such that $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2 < r$. Suppose there exists a non-negative constant $\delta \le Mr$ such that*

$$-\mathcal{L}h - \lambda \circ h + 4Mr \le \delta, \ \forall \boldsymbol{x} \in \tilde{\mathcal{D}}. \tag{11}$$

Then, λ satisfies the safety condition specified in Theorem 5.2.

Remark 5.4 *Here, the non-negative δ is regarded as the tolerance error in the training stage. So, practically, we terminate the training until the safety loss is smaller than Mr.*

Construct Neural Networks with Bounded Lipschitz Constant. We can define the loss function for safety in the manner as the left hand side in (11). However, M depends on the Lipschitz constants of the NN functions λ and $\boldsymbol{u}$, which probably makes it complex and difficult to train the loss function. To simplify the loss function, we construct the NNs with bounded Lipschitz constants for λ and $\boldsymbol{u}$. Specifically, we add the spectral normalization for the neural control function to constrain its Lipschitz constant lower than 1 (Miyato et al., 2018; Yoshida & Miyato, 2017). We apply the monotonic NNs to construct the candidate extended class-$\mathcal{K}$ function as $\lambda_{\boldsymbol{\theta}_\lambda}(x) = \int_0^x q_{\boldsymbol{\theta}_\lambda}(s)\mathrm{d}s$, where $q_{\boldsymbol{\theta}_\lambda}(\cdot)$, the output of the NNs, is definitely positive (Wehenkel & Louppe, 2019). To constrain the Lipschitz constant of $\lambda_{\boldsymbol{\theta}_\lambda}$, we modify the integral formula as $\lambda_{\boldsymbol{\theta}_\lambda}(x) = \int_0^x \min\{q_{\boldsymbol{\theta}_\lambda}(s), M_\lambda\}\mathrm{d}s$, where M_λ is a predefined hyperparameter. Thus, the Lipschitz constant of $\lambda_{\boldsymbol{\theta}_\lambda}$ is smaller than M_λ. Therefore, we can calculate M from the considered functions and M_λ. Other Lipschitz regularization methods can be applied in our framework (Gouk et al., 2021; Liu et al., 2022) as well. We define the loss function for safety guarantee of the controlled system (4) as follows:

$$L_{\tilde{\mathcal{D}}, M_\lambda}(\boldsymbol{\theta}, \boldsymbol{\theta}_\lambda) = \frac{1}{|\tilde{\mathcal{D}}|^2} \sum_{(\boldsymbol{x}, \boldsymbol{y}) \in \tilde{\mathcal{D}} \times \tilde{\mathcal{D}}} \max\left\{0, -\mathcal{L}h(\boldsymbol{x}, \boldsymbol{y}) - \lambda_{\boldsymbol{\theta}_\lambda}(h(\boldsymbol{x})) + 4Mr\right\}, \tag{12}$$

and we can terminate the training process once $L_{\tilde{\mathcal{D}}, M_\lambda}(\boldsymbol{\theta}, \boldsymbol{\theta}_\lambda)$ is less than Mr.

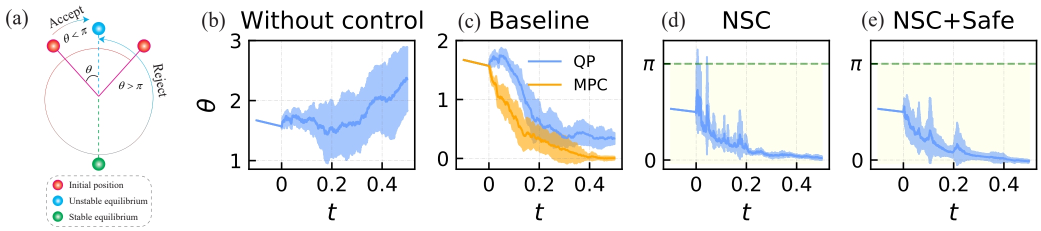

Figure 6: Schematic diagram of inverted pendulum task (a). The θ component of the original system (b), under baseline control (c), under NSC (d), and under our proposed safe control (e). The solid lines are obtained through averaging the 5 sampled trajectories, while the shaded areas stand for the standard errors.

From Safety Guarantee to Stability Guarantee. Akin to the safety guarantee, we provide the stability guarantee for the candidate neural control functions satisfying the condition in Theorems 2.2 and 4.1. However, both theorems require their conditions to be valid for every point $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$, while, in practice, it is impossible to obtain a finite discretization or a bounded Lipschitz constant on the unbounded $\mathcal{X}$. Ingeniously, this difficulty can be conquered with the help of safety guarantee since the safety condition restricts $\mathcal{X} \subset \mathcal{D}$ where $\mathcal{D}$ is compact. As such, we can establish theoretical results on stability guarantee for NDC and NSC in a similar manner as that in Theorem 5.3. We thus summarize all these results in Appendix A.1.8.

We test the proposed safe control method to suppress the fluctuations emergent in the control process on the task of controlling noise-perturbed inverted pendulum with time-delay. This control task is a standard nonlinear control problem for testing different control methods (Anderson, 1989; Huang & Huang, 2000). We apply the safe control method to steer the system to the upright position without rotating a semi-circle, i.e. $|\theta| \leq \pi$. The results are shown in Figure 6 and the experimental details are provided in Appendix A.3.6. It is observed that the safe control method significantly outperms the baseline and the stochastic control method in terms of stabilization and safety guarantee.

6 Theoretical Results for NDC and NSC

We have mentioned the stopping time and the energy cost in section 4.1 and numerically compare the proposed neural controllers with these indexes. These two indexes are the classic factors to measure the performance of the controller (Sun et al., 2017). From the construction in Section 5, we circumscribe the Lipschitz constant $k_{\boldsymbol{u}}$ of the control function. Based on the safety and stability guarantee, the neural controller thus satisfies the conditions assumed in Theorems 2.2 and 4.1.

Then, we have the following two theorems and include their proofs in Appendices A.1.9 and A.1.10.

Theorem 6.1 (Estimation for NDC) Consider the SDDE with NDC controller as

$$\mathrm{d}\boldsymbol{x}(t) = (f(\boldsymbol{x}, \boldsymbol{x}(t-\tau)) + \boldsymbol{u}_f(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau))\mathrm{d}t + g(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau))\mathrm{d}B_t, \;\; \boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^d,$$

where $\|f(\boldsymbol{x},\boldsymbol{y}) - f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\| \vee \|\boldsymbol{u}_f(\boldsymbol{x},\boldsymbol{y}) - \boldsymbol{u}_f(\bar{\boldsymbol{x}},\bar{\boldsymbol{y}})\| \leq L(\|\boldsymbol{x}-\bar{\boldsymbol{x}}\| + \|\boldsymbol{y}-\bar{\boldsymbol{y}}\|)$. Assume that the controlled system satisfies the conditions assumed in Theorem 2.2 and Remark 3.1 with $\mathrm{Ker}(w_1 - w_2) = \mathbf{0}$. Denote by $\eta_\varepsilon = \inf\{t > 0 : \|\boldsymbol{x}(t)\| = \varepsilon\}$ the stopping time and by $\mathcal{E}(\eta_\varepsilon, T) = \mathbb{E}[\int_0^{\eta_\varepsilon \wedge T} \|\boldsymbol{u}(\boldsymbol{x}(s), \boldsymbol{x}(s-\tau))\|^2 \mathrm{d}s]$ the corresponding energy cost in the control process with $\epsilon < \|\boldsymbol{x}_0\|$. Thus, using the same notations in Theorem 2.2, we have

$$\begin{cases} \mathbb{E}[\eta_\epsilon] \leq T_\epsilon = \dfrac{V(\boldsymbol{x}_0) - \min_{\|\boldsymbol{x}\|=\varepsilon} V(\boldsymbol{x}) + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s}{\min_{\|\boldsymbol{x}\|\geq\varepsilon}(w_1(\boldsymbol{x}) - w_2(\boldsymbol{x}))}, \\ \mathcal{E}(\eta_\epsilon, T_\epsilon) \leq \dfrac{k_{\boldsymbol{u}}^2 C_0}{2(L^2+L+k_{\boldsymbol{u}})}\left[\exp\left(4(L^2+L+k_{\boldsymbol{u}})T_\varepsilon\right) - 1\right] + \displaystyle\int_{-\tau}^0 k_{\boldsymbol{u}}^2 \xi^2(s)\mathrm{d}s. \end{cases}$$

where $C_0 = \|\boldsymbol{x}_0\|^2 + (2L^2 + L + k_{\boldsymbol{u}}) \int_{-\tau}^0 \xi(s)^2 \mathrm{d}s$ and $\xi \in C[-\tau, 0]$ is the initial data.

Theorem 6.2 (Estimation for NSC) Consider the SDDE with NSC controller as

$$\mathrm{d}\boldsymbol{x}(t) = f(\boldsymbol{x}, \boldsymbol{x}(t-\tau))\mathrm{d}t + (g(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau)) + \boldsymbol{u}_g(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau))\mathrm{d}B_t, \;\; \boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathbb{R}^d,$$

where f, g are the same as those in Theorem 5.2. Assume that the controlled system satisfies the conditions assumed in Theorem 4.1. Using the same notations in Theorem 4.1, if the term in (9) further satisfies $\max_{\|\boldsymbol{x}_t(0)\|\geq\varepsilon} \|\boldsymbol{x}_t(0)\|^{\alpha-4}(\|\boldsymbol{x}_t(0)\|^2(2\langle \boldsymbol{x}_t(0), f(\boldsymbol{x}_t)\rangle + \|G(\boldsymbol{x}_t)\|_{\mathrm{F}}^2) - (2-\alpha)\|\boldsymbol{x}_t(0)^\top G(\boldsymbol{x}_t)\|^2) = -\delta_\varepsilon < 0$ with $G = g + \boldsymbol{u}_g$, we have

$$\begin{cases} \mathbb{E}[\eta_\epsilon] \leq T_\epsilon = \dfrac{2(\|\boldsymbol{x}_0\|^\alpha - \varepsilon^\alpha)}{\alpha \cdot \delta_\varepsilon}, \\ \mathcal{E}(\eta_\epsilon, T_\epsilon) \leq \dfrac{k_{\boldsymbol{u}}^2 C_1}{2(2L^2 + L + k_{\boldsymbol{u}}^2)} \left[\exp\left(4(2L^2 + L + k_{\boldsymbol{u}}^2)T_\varepsilon\right) - 1\right] + \displaystyle\int_{-\tau}^0 k_{\boldsymbol{u}}^2 \xi^2(s)\mathrm{d}s. \end{cases}$$

where $C_1 = \|\boldsymbol{x}_0\|^2 + (4L^2 + L + 2k_{\boldsymbol{u}}^2)\int_{-\tau}^0 \xi(s)^2\mathrm{d}s$ and $\xi \in C[-\tau, 0]$ is the initial data.

We can design the NN's structure according to these theoretical results. Here we only analyse T_ε because the energy cost $\mathcal{E}(\eta_\varepsilon, T_\varepsilon)$ explicitly depends on T_ε. First, the upper bound for $\mathbb{E}[\eta_\varepsilon]$ of NDC implies that the convergence time decreases as the slope of $w = w_1 - w_2$ near the origin grows due to the fact that $w^{'}(\mathbf{0}) \approx w(\varepsilon)/\varepsilon$, and the same effect is valid for V. Hence, to accelerate the control process, we can construct $w_{1,2}$ and V as the NNs with a steeper slope at the origin, and thus reduce the upper bound of the energy cost using NDC. Second, the time upper bound for NSC is directly related to the hyperparameter α in the training period, so we choose $\alpha^* = \arg\min_\alpha(\|\boldsymbol{x}_0^\alpha - \varepsilon^\alpha\|)/\alpha$ to obtain the optimal NSC controller with the least upper bound of convergence time and energy cost. We numerically investigate the impact of α in Appendix A.4.1.

7 Related Works

Stability Theory of SDDEs. The early endeavors to develop the stability theory for SDDEs were attributed to (Mao, 1999; 2002) inspired by LaSalle's theory (LaSalle, 1968). The subsequent developments have been systematically and fruitfully achieved in the last twenty years in the control community Appleby (2003); Song et al. (2014); Liu et al. (2016); Zhu (2018); Peng et al. (2021). These works reveal the positive effect of multiplicative noise to the stochastic dynamics with delays, and motivate us to develop *only* neural stochastic control to stabilize dynamical systems .

Finding Stabilization Controller. Traditional control methods focus on transforming control criteria, such as the control Lyapunov functions (CLFs), into the QP (Fan et al., 2020; Sarkar et al., 2020) or the semi-definite planning (SDP) problems (Henrion & Garulli, 2005; Jarvis-Wloszek et al., 2003; Parrilo, 2000) to find optimal control iteratively. These methods have high computational complexity since they cannot give the closed form of the control. Hence, machine-learning based control methods have been introduced to improve the generalization and efficiency of the original convex optimal problems (Khansari-Zadeh & Billard, 2014; Ravanbakhsh & Sankaranarayanan, 2019; Gurriet et al., 2018). However, all the existing learning methods consider dynamics without time-delay (Wagener et al., 2019; Williams et al., 2018; Chang et al., 2019; Zhang et al., 2022).

Theory and Application of Control Barrier Function The barrier function method has been extensively researched in the problem of safety verification of controlled dynamics (Prajna & Jadbabaie, 2004; Jankovic, 2018; Prajna et al., 2004; Clark, 2019; 2021). Existing works for constructing barrier functions in application typically based on quadratic programming (Ames et al., 2014; 2016; Khojasteh et al., 2020; Fan et al., 2020). Machine learning methods have also be introduced in safe control fields in (Robey et al., 2020; Dean et al., 2020; Taylor et al., 2020).

8 Discussion

We heuristically design two kinds of neural controllers for SDDEs based on the classic LaSalle-type stabilization theory and the newly proposed stochastic stabilization theorem. To assure the controlled trajectories can stay in the safety region, we cultivate the safety guarantee theorem through the SCBF and the discretization techniques. Since the state space of the controlled SDDEs with safety guarantee is bounded by the compact safety region, we can similarly deduce the stability guarantee theorem for neural controllers through spatial discretization. Furthermore, we theoretically and numerically investigate the neural controllers' performance in terms of convergence time and energy cost. The proposed neural controllers with safety and stability guarantee are summarized as SYNC, which significantly simplify the process of control design and have extensive potential in different control fields, such as financial engineering (Zhou & Li, 2000).

REFERENCES

Uri Alon. *An introduction to systems biology: design principles of biological circuits*. Chapman and Hall/CRC, 2006.

Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pp. 6271–6278. IEEE, 2014.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.

Charles W Anderson. Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9(3):31–37, 1989.

John AD Appleby. Stabilisation of functional differential equations by noise. *Systems & Control Letters*, 2003.

Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 3245–3254, 2019.

Andrew Clark. Control barrier functions for complete and incomplete information stochastic systems. In *2019 American Control Conference (ACC)*, pp. 2928–2935. IEEE, 2019.

Andrew Clark. Control barrier functions for stochastic systems. *Automatica*, 130:109688, 2021.

Sarah Dean, Andrew J Taylor, Ryan K Cosner, Benjamin Recht, and Aaron D Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. *arXiv preprint arXiv:2010.16001*, 2020.

David D Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Bayesian learning-based adaptive control for safety critical systems. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 4093–4099. IEEE, 2020.

Marco Gallieri, Seyed Sina Mirrazavi Salehian, Nihat Engin Toklu, Alessio Quaglino, Jonathan Masci, Jan Koutník, and Faustino Gomez. Safe interactive model-based learning. *arXiv preprint arXiv:1911.06556*, 2019.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Qian Guo, Xuerong Mao, and Rongxian Yue. Almost sure exponential stability of stochastic differential delay equations. *SIAM Journal on Control and Optimization*, 54(4):1919–1933, 2016.

Thomas Gurriet, Andrew Singletary, Jacob Reher, Laurent Ciarletta, Eric Feron, and Aaron Ames. Towards a framework for realizable safety critical control through active set invariance. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 98–106. IEEE, 2018.

Jiequn Han et al. Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*, 2016.

Didier Henrion and Andrea Garulli. *Positive Polynomials in Control*, volume 312. Springer Science & Business Media, 2005.

Shiuh-Jer Huang and Chien-Lo Huang. Control of an inverted pendulum using grey prediction model. *IEEE Transactions on Industry Applications*, 36(2):452–458, 2000.

Mrdjan Jankovic. Control barrier functions for constrained control of linear systems with input delay. In *2018 annual American control conference (ACC)*, pp. 3316–3321. IEEE, 2018.

Zachary Jarvis-Wloszek, Ryan Feeley, Weehong Tan, Kunpeng Sun, and Andrew Packard. Some controls applications of sum of squares programming. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, volume 5, pp. 4676–4681. IEEE, 2003.

Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 2012.

S Mohammad Khansari-Zadeh and Aude Billard. Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6):752–765, 2014.

Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. Probabilistic safety constraints for learned high relative degree system dynamics. In *Learning for Dynamics and Control*, pp. 781–792. PMLR, 2020.

J Zico Kolter and Gaurav Manek. Learning stable deep dynamics models. *Advances in Neural Information Processing Systems*, 32:11128–11136, 2019.

Joseph P LaSalle. Stability theory for ordinary differential equations. *Journal of Differential equations*, 4(1):57–65, 1968.

Mathias Lechner, Đorđe Žikelić, Krishnendu Chatterjee, and Thomas A Henzinger. Stability verification in stochastic control systems via neural network supermartingales. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7326–7336, 2022.

Aming Li, Sean P Cornelius, Y-Y Liu, Long Wang, and A-L Barabási. The fundamental advantages of temporal networks. *Science*, 358(6366):1042–1046, 2017.

Wei Lin and Yangbo He. Complete synchronization of the noise-perturbed chua's circuits. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 15(2):023705, 2005.

Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. Learning smooth neural functions via lipschitz regularization. *arXiv preprint arXiv:2202.08345*, 2022.

Liang Liu, Shen Yin, Lixian Zhang, Xunyuan Yin, and Huaicheng Yan. Improved results on asymptotic stabilization for stochastic nonlinear time-delay systems with application to a chemical reactor system. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):195–204, 2016.

X Mao. Robustness of exponential stability of stochastic differential delay equations. *IEEE Transactions on Automatic Control*, 41(3):442–447, 1996.

Xuerong Mao. Lasalle-type theorems for stochastic differential delay equations. *Journal of mathematical analysis and applications*, 236(2):350–369, 1999.

Xuerong Mao. A note on the lasalle-type theorems for stochastic differential delay equations. *Journal of mathematical analysis and applications*, 268(1):125–142, 2002.

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

Takashi Matsumoto. A chaotic attractor from chua's circuit. *IEEE Transactions on Circuits and Systems*, 31(12):1055–1058, 1984.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Pablo A Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. California Institute of Technology, 2000.

Dongxue Peng, Xiaodi Li, R Rakkiyappan, and Yanhui Ding. Stabilization of stochastic delayed systems: Event-triggered impulsive control. *Applied Mathematics and Computation*, 401:126054, 2021.

Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*, pp. 477–492. Springer, 2004.

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.

Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

Hadi Ravanbakhsh and Sriram Sankaranarayanan. Learning control lyapunov functions from counterexamples and demonstrations. *Autonomous Robots*, 43(2):275–307, 2019.

Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717–3724. IEEE, 2020.

Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. A barrier function approach to finite-time stochastic system verification and control. *Automatica*, 125:109439, 2021.

Meenakshi Sarkar, Debasish Ghose, and Evangelos A Theodorou. High-relative degree stochastic control lyapunov and barrier functions. *arXiv preprint arXiv:2004.03856*, 2020.

AN Shiryaev. Theory of martingales, 1989.

Bo Song, Ju H Park, Zheng-Guang Wu, and Xuchao Li. New results on delay-dependent stability analysis and stabilization for stochastic time-delay systems. *International Journal of Robust and Nonlinear Control*, 24(16):2546–2559, 2014.

Yong-Zheng Sun, Si-Yang Leng, Ying-Cheng Lai, Celso Grebogi, and Wei Lin. Closed-loop control of complex networks: A trade-off between time and energy. *Physical Review Letters*, 119(19): 198301, 2017.

Yonghui Sun and Jinde Cao. Adaptive synchronization between two different noise-perturbed chaotic systems with fully unknown parameters. *Physica A: statistical mechanics and its applications*, 376:253–265, 2007.

Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control*, pp. 708–717. PMLR, 2020.

Nolan Wagener, Ching-An Cheng, Jacob Sacks, and Byron Boots. An online learning approach to model predictive control. *arXiv preprint arXiv:1902.08967*, 2019.

Li Wang, Aaron D Ames, and Magnus Egerstedt. Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 2659–2664. IEEE, 2016.

Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

Antoine Wehenkel and Gilles Louppe. Unconstrained monotonic neural networks. *Advances in neural information processing systems*, 32, 2019.

Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.

Yuhua Xu, Yuling Wang, Wuneng Zhou, and Jian'an Fang. Stochastic complex networks synchronize to the limit set with adaptive controller and adaptive delay. *Mathematical Methods in the Applied Sciences*, 37(15):2290–2296, 2014.

Gang Yan, Jie Ren, Ying-Cheng Lai, Choy-Heng Lai, and Baowen Li. Controlling complex networks: How much energy is needed? *Physical Review Letters*, 108(21):218703, 2012.

Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, pp. 022022. IOP Publishing, 2019.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

Wenwu Yu and Jinde Cao. Adaptive synchronization and lag synchronization of uncertain dynamical system with time delay based on parameter identification. *Physica A: Statistical Mechanics and its Applications*, 375(2):467–482, 2007.

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Neural stochastic control. *arXiv preprint arXiv:2209.07240*, 2022.

Xun Yu Zhou and Duan Li. Continuous-time mean-variance portfolio selection: A stochastic lq framework. *Applied Mathematics and Optimization*, 42(1):19–33, 2000.

Quanxin Zhu. Stabilization of stochastic nonlinear delay systems with exogenous disturbances and the event-triggered feedback control. *IEEE Transactions on Automatic Control*, 64(9):3764–3771, 2018.

A Appendix

A.1 Proofs and Derivations

A.1.1 Notations and Preliminaries

In this section, we introduce some basic definitions and notations and then provide the proofs of the theoretical results.

Notations. Throughout the paper, we employ the following notation. Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ be a complete probability space with a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ satisfying the usual conditions (i.e. it is increasing and right continuous while $\mathcal{F}_0$ contains all $\mathbb{P}$-null sets). Let $B_t = (B_1(t), \cdots, B_r(t))^\top$ be a r-dimensional (r-D) Brownian motion defined on the probability space, where $\top$ denote the transpose of a vector or matrix. If x, y are real numbers, then $x \wedge y$ denotes the minimum of x and y. Let $\|\boldsymbol{x}\|$ denote the L^2 norm of a vector and $\|A\|_{\mathrm{F}}$ denote the Frobenius norm of a matrix A. Let $\langle \boldsymbol{x}, \boldsymbol{y}\rangle$ be the inner product of vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$. For a function $f(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}$, let Ker$f$ denote the zero solutions of $f(\boldsymbol{x})$, that is, Ker$f = \{\boldsymbol{x} : f(\boldsymbol{x}) = 0\}$. For the two sets A, B, let $A \subset B$ denote that A is covered in B.

Definition A.1 *(Martingale) The stochastic process X_t on $t \geq 0$ is called a martingale (sub-martingale) on probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ with a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ satisfying the usual conditions, if the following two conditions hold: (1) X_t is $\mathcal{F}_t-$measurable for any $t \geq 0$; (2) $\mathbb{E}[X_t|\mathcal{F}_s] = X_s (\mathbb{E}[X_t|\mathcal{F}_s] = X_s)$ for any $t > s \geq 0$.*

Definition A.2 *(Stopping Time) Given probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ and a mapping $\tau : \Omega \to [0, \infty)$, we call τ an $\{\mathcal{F}_t\}_{t\geq 0}$ stopping time, for any $t \geq 0$, $\tau \leq t \in \mathcal{F}_t$,*

Definition A.3 *(Local Martingale) The stochastic process $X_t,\ t \geq 0$ is called a local martingale, if there exists a family of stopping times $\{\tau_n\}_{n\in\mathbb{Z}_+}$ such that $\lim_{n\to\infty} \tau_n = \infty,\ a.s.$ and $\{X_{t\wedge\tau_n}\}_{n\in\mathbb{Z}_+}$ is a martingale.*

Definition A.4 *(Itô's Process) Let B_t be a d-dimensional Brownian motion on probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$. A ($d$-dimensional) Itô's process is a stochastic process X_t on $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbb{P})$ in the form of*

$$X_t = X_0 + \int_0^t u(s, w)\mathrm{d}s + \int_0^t \mathrm{d}v(s, w)B_s \left(\Leftrightarrow \mathrm{d}X_t = u(t, w)\mathrm{d}t + v(t, w)\mathrm{d}B_t\right),$$

where u and v satisfy the constraints as follows:

$$\mathbb{P}\left[\int_0^t \|v(s, w)\|^2 \mathrm{d}s < \infty \text{ for all } t \geq 0\right] = 1,$$
$$\mathbb{P}\left[\int_0^t \|u(s, w)\| \mathrm{d}s < \infty \text{ for all } t \geq 0\right] = 1.$$

Definition A.5 *(Itô's Formula) Let X_t be a d-dimensional Itô's process given by*

$$\mathrm{d}X_t = u\mathrm{d}t + v\mathrm{d}B_t.$$

Let $f(t, \boldsymbol{x}) \in C^{1,2}([0, \infty) \times \mathbb{R}^d)$. Then, $Y_t = f(t, X_t)$ is an Itô's process as well, satisfying

$$\mathrm{d}Y_t = \frac{\partial h}{\partial t}(t, X_t)\mathrm{d}t + \nabla_{\boldsymbol{x}} f(t, X_t) \cdot \mathrm{d}X_t + \frac{1}{2}\mathrm{Tr}(\mathrm{d}X_t^\top \mathrm{Hess}f(t, X_t)\mathrm{d}X_t).$$

We further denote by $\|\cdot\|$ the L^2-norm for any given vector in $\mathbb{R}^d$. Denote by $|\cdot|$ the absolute value of a scalar number or the modulus length of a complex number number. For $A = (a_{ij})$, a matrix of dimension $d \times r$, denote by $\|A\|_{\mathrm{F}}^2 = \sum_{i=1}^d \sum_{j=1}^r a_{ij}^2$ the Frobenius norm.

A.1.2 Useful Lemmas

The following results will be of great use in the proofs of our main theorems.

Lemma A.1 *(Shiryaev, 1989) Let A_1 and A_2 be non-decreasing processes a.s., let Z be a non-negative semi-martingale with $\mathbb{E}(Z) < \infty$, let M be a local martingale, and*

$$Z(t) = Z(0) + A_1(t) - A_2(t) + M(t), \ \ t \geq 0.$$

Then $\{w : \lim_{t\to\infty} A_1(t) < \infty\} \subseteq \{w : \lim_{t\to\infty} Z(t) < \infty\} \bigcap \{w : \lim_{t\to\infty} A_2(t) < \infty\}, a.s.$

Lemma A.2 *(Karatzas & Shreve, 2012) Let X_t be non-negative submartingale, $[r, s]$ be any subinterval of $[0, \infty)$ and $\lambda > 0$. Then*

$$\lambda \mathbb{P}\left(\sup_{r\leq t\leq s} X_t \geq \lambda\right) \leq \mathbb{E}(X_s).$$

A.1.3 Drawbacks of L^2 regularization in V

Adding L^2 regularization to objective functions is a classical operation to avoid over-fitting (Ying, 2019) and guarantee the positive definiteness (Gallieri et al., 2019). However, the explicit form $\varepsilon\|\boldsymbol{x}\|^2$ may fail in learning an effective neural control as this function cannot be the candidate V function in some cases (Zhang et al., 2022). The following example illustrates this point.

Example A.3 *Consider a 2-D SDDE as follows:*

$$\mathrm{d}x_1(t) = x_2(t)\mathrm{d}t + \frac{1}{2}x_1(t-1)\mathrm{d}B_1(t), \ \mathrm{d}x_2(t) = [-2x_1(t) - x_2(t)]\mathrm{d}t + x_1(t)\mathrm{d}B_2(t)$$

the solution of this system is validated to satisfy $\lim_{t\to\infty} \boldsymbol{x}(t;\xi) = \mathbf{0}$ $a.s.$ with any initial data $\xi \in C_{\mathcal{F}_0}([-1, 0]; \mathbb{R}^2)$; however, $k\|\boldsymbol{x}\|^2$ for any $k \in \mathbb{R}_+$ cannot be a useful auxiliary V function to identify the sufficient conditions in Theorem 2.2.

Proof. On one hand, we select as $V(\boldsymbol{x}) = k\|\boldsymbol{x}\|^2 \equiv k(x_1^2 + x_2^2)$ with $k > 0$, an undetermined coefficient. We thus get $\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}) = k(x_1^2 - 2x_1x_2 - 2x_2^2 + y_1^2/4)$ and to satisfy $\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}) \leq -w_1(\boldsymbol{x}) + w_2(\boldsymbol{y})$, the following inequalities must hold

$$k(x_1^2 - 2x_1x_2 - 2x_2^2) \leq -w_1(\boldsymbol{x}), \ \ y_1^2/4 \leq w_2(\boldsymbol{y}), \ \ w_1(\boldsymbol{x}) \geq w_2(\boldsymbol{x}) \geq 0.$$

Then we have

$$x_1^2 \leq w_2(\boldsymbol{x}) \leq w_1(\boldsymbol{x}) \leq 2x_2^2 + 2x_1x_2 - x_1^2, \quad \Leftrightarrow \quad x_1^2 \leq x_2^2 + x_1x_2, \ \ \forall (x_1, x_2)^\top \in \mathbb{R}^2$$

which is impossible. Hence, the above form of V cannot guarantee the sufficient conditions for the stability of the zero solution in Theorem 2.2.

On the other hand, we set as $\hat{V}(\boldsymbol{x}) = \frac{5}{2}x_1^2 + x_1x_2 + x_2^2$, and then we obtain

$$\mathcal{L}\hat{V}(\boldsymbol{x}, \boldsymbol{y}) = -\frac{3}{2}x_1^2 - x_2^2 + \frac{5}{8}y_1^2$$

As we choose $w_1(\boldsymbol{x}) = \frac{3}{2}x_1^2 + x_2^2$ and $w_2(\boldsymbol{x}) = \frac{5}{8}x_1^2$, we have $\mathrm{Ker}(w_1 - w_2) = \{(0, 0)^\top\}$ and $\mathcal{L}\hat{V}(\boldsymbol{x}, \boldsymbol{y}) \leq -w_1(\boldsymbol{x}) + w_2(\boldsymbol{y})$. So all the conditions in Theorem 2.2 are satisfied. Therefore, the property of $\lim_{t\to\infty} \boldsymbol{x}(t;\xi) = \mathbf{0}$ is assured. This example particularly indicates that regularization terms need delicate design and fine-tune in applications.

A.1.4 Proof of Theorem 4.1

To begin with, for any $\phi, \varphi \in C([-\tau, 0]; \mathbb{R}^d), \ t \in [0, T]$ with $\|\phi - \varphi\| \leq K$ we have

$$\|F(\phi, t) - F(\varphi, t)\| \vee \|G(\phi, t) - G(\varphi, t)\|_{\mathrm{F}} \leq K\|\phi - \varphi\|,$$

according to the locally Lipschitz condition. Notice that $F(\mathbf{0}, t) = \mathbf{0}$ and $G(\mathbf{0}, t) = \mathbf{0}$. Then we can deduce the following locally linear growth upper bound

$$\|F(\phi, t)\| \vee \|G(\phi, t)\|_{\mathrm{F}} \leq K.$$

Thus, a unique continuous solution $\boldsymbol{x}(t;\xi)$ almost surely exists for the SFDE in Theorem 4.1 (Mao, 2007). This means the positive quadratic process $\|\boldsymbol{x}(t)\|^2$ is well-defined. For the simplicity of the symbols, we write $F(\boldsymbol{x}_t,t),\ G(\boldsymbol{x}_t,t)$ as $F,\ G$, respectively. Applying Itô's formula to $\|\boldsymbol{x}(t)\|^2$ we have,

$$\begin{aligned}\mathrm{d}\|\boldsymbol{x}\|^2 &= 2\boldsymbol{x}^\top \mathrm{d}\boldsymbol{x} + \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}\\ &= 2\boldsymbol{x}^\top (F\mathrm{d}t + G\mathrm{d}B_t) + \|G\|_{\mathrm{F}}^2\mathrm{d}t\end{aligned}$$

For $\alpha \in (0,1)$, applying Itô's formula to $\|\boldsymbol{x}\|^\alpha = (\|\boldsymbol{x}\|^2)^{\alpha/2}$ we have

$$\begin{aligned}\mathrm{d}\|\boldsymbol{x}\|^\alpha &= \frac{\alpha}{2}(\|\boldsymbol{x}\|^2)^{\frac{\alpha}{2}-1}\mathrm{d}\|\boldsymbol{x}\|^2 + \frac{\alpha}{4}(\frac{\alpha}{2}-1)(\|\boldsymbol{x}\|^2)^{\frac{\alpha}{2}-2}\mathrm{d}\|\boldsymbol{x}\|^2\mathrm{d}\|\boldsymbol{x}\|^2\\ &= \frac{\alpha}{2}(\|\boldsymbol{x}\|^2)^{\frac{\alpha}{2}-1}\left[2\boldsymbol{x}^\top(F\mathrm{d}t + G\mathrm{d}B_t) + \|G\|_{\mathrm{F}}^2\mathrm{d}t\right] + \alpha(\frac{\alpha}{2}-1)(\|\boldsymbol{x}\|^2)^{\frac{\alpha}{2}-2}\|\boldsymbol{x}^\top G\|^2\mathrm{d}t\\ &= \frac{\alpha}{2}\|\boldsymbol{x}\|^{\alpha-4}\left[\|\boldsymbol{x}\|^2\left(2\boldsymbol{x}^\top F + \|G\|_{\mathrm{F}}^2\right) - (2-\alpha)\|\boldsymbol{x}^\top G\|^2\right]\mathrm{d}t + \alpha\|\boldsymbol{x}\|^{\alpha-2}\boldsymbol{x}^\top G\mathrm{d}B_t\end{aligned}$$

Next we let

$$\begin{aligned}&A_1(t) = 0,\\ &A_2(t) = -\int_0^t \frac{\alpha}{2}\|\boldsymbol{x}(s)\|^{\alpha-4}\left[\|\boldsymbol{x}(s)\|^2\left(2\boldsymbol{x}(s)^\top F + \|G\|_{\mathrm{F}}^2\right) - (2-\alpha)\|\boldsymbol{x}(s)^\top G\|^2\right]\mathrm{d}s,\\ &M(t) = \int_0^t \alpha\|\boldsymbol{x}(s)\|^{\alpha-2}\boldsymbol{x}(s)^\top G\mathrm{d}B_s\end{aligned}$$

Then A_2 is a non-decreasing process and M is a local martingale. Hence, combining the Lemma A.1 with the following formula,

$$\|\boldsymbol{x}(t)\|^\alpha = \|\boldsymbol{x}(0)\|^\alpha + A_1(t) - A_2(t) + M(t),$$

we have

$$\lim_{t\to\infty}\|\boldsymbol{x}(t)\|^\alpha < \infty,\ \ \lim_{t\to\infty}A_2(t) < \infty,\ \ a.s.$$

Notice that $M(t) = \|\boldsymbol{x}(t)\|^\alpha - \|\boldsymbol{x}(0)\|^\alpha + A_2(t)$, so we have $\lim_{t\to\infty}M(t) < \infty,\ a.s.$ Then we claim that $\lim_{t\to\infty}\boldsymbol{x}(t) = \mathbf{0},\ a.s.$, otherwise, there exits a set Ω_0 with positive measure such that,

$$\lim_{t\to\infty}\boldsymbol{x}(t;\xi)(w) = \boldsymbol{x}^w,\ \ \|\boldsymbol{x}^w\| = k_w > 0,\ \ \forall w \in \Omega_0$$

Since local martingale $M(t)$ exists finite limit $\lim_{t\to\infty}M(t)$ almost surely, the quadratic variation process $\langle M\rangle(t)$ also possesses a finite limit almost surely, where

$$\langle M\rangle(t) = \int_0^t \alpha^2\|\boldsymbol{x}(s)\|^{2\alpha-4}\|\boldsymbol{x}(s)^\top G\|^2\mathrm{d}s.$$

Thus, there exists a set $\Omega_1 \subset \omega_0,\ \mathbb{P}(\Omega_1) > 0$, such that,

$$\lim_{t\to\infty}\langle M\rangle(t)(w) = \int_0^\infty \alpha^2\|\boldsymbol{x}(s)(w)\|^{2\alpha-4}\|\boldsymbol{x}(s)(w)^\top G\|^2\mathrm{d}s < \infty,\ \ \forall w \in \Omega_1,$$

which further implies

$$\int_0^\infty \|\boldsymbol{x}(s)(w)^\top G\|^2\mathrm{d}s = \int_0^\infty \|\boldsymbol{x}_s(0)(w)^\top G(\boldsymbol{x}_s(w),t)\|^2\mathrm{d}s < \infty,\ \ \forall w \in \Omega_1,$$

However, according to the condition $\min_{\|\boldsymbol{x}_t(0)\|=M}\|\boldsymbol{x}_t(0)^\top G(\boldsymbol{x}_t,t)\| > 0,\ \forall M > 0$, we have

$$\liminf_{t\to\infty}\|\boldsymbol{x}_t(0)(w)^\top G(\boldsymbol{x}_t(w),t)\|^2 = \|(\boldsymbol{x}^w)^\top G(\boldsymbol{x}_t \equiv \boldsymbol{x}^w, t)\|^2 > 0,$$

which contradicts the integral $\int_0^\infty \|\boldsymbol{x}_s(0)(w)^\top G(\boldsymbol{x}_s(w),t)\|^2\mathrm{d}s < \infty$. Therefore, $\mathbb{P}(\Omega_0) = 0$ and $\lim\boldsymbol{x}(t;\xi) = \mathbf{0},\ a.s.$

A.1.5 VALIDATION OF PROPOSITION 5.1

The ideas in the proof are the same as that in (Clark, 2019) and here we validate the results for SDDEs.

First, notice that the barrier funcition $\mathcal{B}(\boldsymbol{x})$ with $\frac{1}{\alpha_1(h(\boldsymbol{x}))} \leq \mathcal{B}(\boldsymbol{x}) \leq \frac{1}{\alpha_2(h(\boldsymbol{x}))}$ is continuous on $\mathrm{int}(\mathcal{C})$ and becomes $+\infty$ at the boundary $\partial(\mathcal{C})$. Since each sample path of $\boldsymbol{x}(t)$ is continuous, each sample path of $\mathcal{B}(\boldsymbol{x}(t))$ is continuous. Then, the safety property for any trajectory initiated from $\mathrm{int}(\mathcal{C})$, i.e. $\boldsymbol{x}(t) \in \mathrm{int}(\mathcal{C}),\ \forall t \geq 0$, is equivalent to $\mathcal{B}(\boldsymbol{x}(t)) < \infty,\ \forall t \geq 0$. So we only need to prove that the state trajectory of the barrier function $\mathcal{B}$ is bounded provided with $B(\boldsymbol{x}(0)) < \infty$, i.e.

$$\sup_{t\in[0,\infty)} \mathcal{B}(\boldsymbol{x}(t)) < \infty,\ \ a.s.$$

Due to the continuity of the sample path for the barrier function, we only need to prove

$$\mathbb{P}\big[\sup_{t\in[0,\infty)} \mathcal{B}(\boldsymbol{x}(t)) = \infty\big] = 0 \Leftrightarrow \mathbb{P}\big[\sup_{t\in[0,s]} \mathcal{B}(\boldsymbol{x}(t)) = \infty\big] < \delta,\ \ \forall s, \delta > 0.$$

Now, we fix any $s,\ \delta > 0$ and find a suitable $K = K(s, \delta)$ such that

$$\mathbb{P}\big[\sup_{t\in[0,s]} \mathcal{B}(\boldsymbol{x}(t)) > K\big] < \delta,$$

Next, denote $L = \mathcal{B}(\boldsymbol{x}(0))$ and define the stopping times as follows:

$$\begin{aligned}
&\eta_0 = 0,\ \zeta_0 = \inf\{t : \mathcal{B}(\boldsymbol{x}_t) < L\},\\
&\eta_i = \inf\{t : \mathcal{B}(\boldsymbol{x}_t) > L,\ t > \zeta_{i-1}\},\ i = 1, 2, \cdots,\\
&\zeta_i = \inf\{t : \mathcal{B}(\boldsymbol{x}_t) < L,\ t > \eta_i\},\ i = 1, 2, \cdots.
\end{aligned}$$

Then we define a new process $\tilde{\mathcal{B}}$ as follows:

$$\tilde{\mathcal{B}}(t) = L + \sum_{i=1}^{\infty}\left[\int_{\eta_i\wedge t}^{\zeta_i\wedge t} \alpha_3(\alpha_2^{-1}(\frac{1}{L}))\mathrm{d}r + \int_{\eta_i\wedge t}^{\zeta_i\wedge t} \nabla\mathcal{B}(\boldsymbol{x}(r))^\top G\mathrm{d}B_r\right].$$

It can be seen that $\tilde{\mathcal{B}}$ is a submartingale since $\tilde{\mathcal{B}}$ is the summaiton of a positive increasing process and martingale. By Itô's formula, we have

$$\mathcal{B}(\boldsymbol{x}(t)) = L + \int_0^t \mathcal{L}\mathcal{B}(\boldsymbol{x}(r), \boldsymbol{x}(r-\tau))\mathrm{d}r + \int_0^t \nabla\mathcal{B}(\boldsymbol{x}(r))^\top G\mathrm{d}B_r.$$

We claim that $\mathcal{B}(\boldsymbol{x}(t)) \leq \tilde{\mathcal{B}}(t)$ and the proof is by induction. Notice that we have the following equalities by definition,

$$\tilde{\mathcal{B}}(0) = \mathcal{B}(\boldsymbol{x}(\eta_i)) = \mathcal{B}(\boldsymbol{x}(\zeta_i)) = L,\ i = 0, 1, \cdots,$$

For any $t \in (\eta_i, \zeta_i]$ we have

$$\begin{aligned}
\tilde{\mathcal{B}}(t) &= \tilde{\mathcal{B}}(\eta_i) + \int_{\eta_i}^t \alpha_3(\alpha_2^{-1}(\frac{1}{L}))\mathrm{d}r + \int_{\eta_i}^t \mathcal{B}(\boldsymbol{x}(r))^\top G\mathrm{d}B_r\\
\mathcal{B}(\boldsymbol{x}(t)) &= \mathcal{B}(\eta_i) + \int_{\eta_i}^t \mathcal{L}\mathcal{B}(\boldsymbol{x}(r), \boldsymbol{x}(r-\tau))\mathrm{d}r + \int_{\eta_i}^t \nabla\mathcal{B}(\boldsymbol{x}(r))^\top G\mathrm{d}B_r.
\end{aligned} \tag{13}$$

By induction we have $\mathcal{B}(\boldsymbol{x}(\eta_i)) \leq \tilde{\mathcal{B}}(\eta_i)$, and when $t \in [\eta_i, \zeta_i]$ we have $\mathcal{B}(\boldsymbol{x}(t)) \geq L$, which further indicates that,

$$\mathcal{L}\mathcal{B} \leq \alpha_3(h(\boldsymbol{x})) \leq \alpha_3(\alpha_2^{-1}(\frac{1}{\mathcal{B}(\boldsymbol{x}(t))})) \leq \alpha_3(\alpha_2^{-1}(\frac{1}{L})) \tag{14}$$

Combining Eqs. (13) and (14), we have $\mathcal{B}(\boldsymbol{x}(t)) \leq \tilde{\mathcal{B}}(t)$ on $[\eta_i, \zeta_i]$. Next, for $t \in (\zeta_i, \eta_{i+1}]$, we have

$$\tilde{\mathcal{B}}(t) = \tilde{\mathcal{B}}(\zeta_i) \geq \mathcal{B}(\boldsymbol{x}(\zeta_i)) = L \geq \mathcal{B}(\boldsymbol{x}(t)),$$

which completes the proof of the claim. Notice that

$$\mathbb{E}[\tilde{\mathcal{B}}(s)] = L + \sum_{i=1}^{\infty}\int_{\eta_i\wedge s}^{\zeta_i\wedge s} \alpha_3(\alpha_2^{-1}(\frac{1}{L}))\mathrm{d}r \leq L + \int_0^s \alpha_3(\alpha_2^{-1}(\frac{1}{L}))\mathrm{d}r = L + s\alpha_3(\alpha_2^{-1}(\frac{1}{L}))$$

Then, set $K(s,\delta) = \dfrac{\delta}{2(L + s\alpha_3(\alpha_2^{-1}(\frac{1}{L})))}$. Apply Lemma A.2 to $\tilde{\mathcal{B}}$ yields:

$$\mathbb{P}(\sup_{t\in[0,s]} \tilde{\mathcal{B}}(t) > K) \leq \frac{1}{K}\mathbb{E}[\tilde{\mathcal{B}}(s)] = \frac{\delta}{2} < \delta.$$

Thus, we have

$$\mathbb{P}(\sup_{t\in[0,s]} \mathcal{B}(\boldsymbol{x}(t)) > K) \leq \mathbb{P}(\sup_{t\in[0,s]} \tilde{\mathcal{B}}(t) > K) < \delta.$$

Based on the arbitrariness of $s,\ \delta$, we have $\mathbb{P}[\sup_{t\in[0,\infty)} \mathcal{B}(\boldsymbol{x}(t)) = \infty] = 0$, which completes the whole proof.

A.1.6 Proof of Theorem 5.2

Notice that each sample path of $\boldsymbol{x}(t)$ is continuous and $h(\boldsymbol{x})$ is also continuous. This implies that $h(\boldsymbol{x}(t)) > 0 \iff \boldsymbol{x}(t) \in \mathrm{int}(\mathcal{C})$. Now we prove $h(\boldsymbol{x}(t)) > 0$ $a.s.$ with initial $h(\boldsymbol{x}(0)) > 0$, which is equivalent to $\tau = \infty$ $a.s.$, where stopping time $\tau = \inf\{t \geq 0 : h(\boldsymbol{x}(t)) = 0\}$. we prove it by contradiction. If $\tau = \infty$ $a.s.$ was false, then we can find a pair of constants $T > 0$ and $M \gg 1$ for $\mathbb{P}(B) > 0$, where

$$B = \{w \in \Omega : \tau < T \text{ and } \|\boldsymbol{x}(t)\|_2 \leq M, \forall 0 \leq t \leq T\}.$$

But, by the standing hypotheses, there exists a positive constant K_M such that

$$\lambda(x) \leq K_M x,\ \forall |x| \leq \sup_{\|\boldsymbol{x}\|_2 \leq M} h(\boldsymbol{x}) < \infty.$$

Then, for $w \in B$ and $t \leq T$,

$$\lambda(h(\boldsymbol{x}(t))) \leq K_M h(\boldsymbol{x}(t)).$$

Now, for any $\varepsilon \in (0, h(\boldsymbol{x}(0)))$, define the stopping time

$$\tau_\varepsilon = \inf\{t \geq 0 : h(\boldsymbol{x}(t)) \notin (\varepsilon, h(\boldsymbol{x}(0))\}.$$

By Itô's formula,

$$\begin{aligned} \mathrm{d}h(\boldsymbol{x}) &= \mathcal{L}h\mathrm{d}t + \nabla h^\top G\mathrm{d}B_t \\ &\geq -\lambda(h)\mathrm{d}t + \nabla h^\top G\mathrm{d}B_t \end{aligned}$$

Take expectation on both sides with respect to τ_ε on set B,

$$\begin{aligned} \mathbb{E}[h(\boldsymbol{x}(\tau_\varepsilon \wedge t))\mathbb{1}_B] &\geq h(\boldsymbol{x}(0)) - \int_0^t \mathbb{E}[\lambda(h(\boldsymbol{x}(\tau_\varepsilon \wedge s)))\mathbb{1}_B]\mathrm{d}s \\ &\geq h(\boldsymbol{x}(0)) - \int_0^t \mathbb{E}[K_M h(\boldsymbol{x}(\tau_\varepsilon \wedge s))\mathbb{1}_B]\mathrm{d}s \end{aligned}$$

By Gronwall's inequality,

$$\mathbb{E}[h(\boldsymbol{x}(\tau_\varepsilon \wedge t))\mathbb{1}_B] \geq Le^{-K_M(\tau_\varepsilon \wedge t)\mathbb{P}(B)}$$

Note that for $w \in B$, $\tau_\varepsilon \leq T$ and $h(\boldsymbol{x}(\tau_\varepsilon)) = \varepsilon$. The above inequality therefore implies that

$$\mathbb{E}[\varepsilon\mathbb{P}(B)] = \varepsilon\mathbb{P}(B) \geq h(\boldsymbol{x}(0))e^{-K_M(\tau_\varepsilon \wedge T)\mathbb{P}(B)} \geq h(\boldsymbol{x}(0))e^{-K_M T\mathbb{P}(B)}$$

Letting $\varepsilon \to 0$ yields that $0 \geq h(\boldsymbol{x}(0))e^{-K_M T\mathbb{P}(B)}$, but this contradicts the definition of B and $\mathbb{P}(B) > 0$.
The proof is complete.

A.1.7 Proof of Theorem 5.3

From the condition we know that $\mathcal{L}h + \lambda \circ h - 3Mr \geq 0$. For any $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$, there exists $\tilde{\boldsymbol{x}},\ \tilde{\boldsymbol{y}} \in \tilde{\mathcal{D}}$ s.t.

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2 \leq r,\ \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2 \leq r.$$

By interpolation,

$$\begin{aligned} \mathcal{L}h(\boldsymbol{x}, \boldsymbol{y}) - \lambda(h(\boldsymbol{x})) &= \mathcal{L}h(\boldsymbol{x}, \boldsymbol{y}) - \mathcal{L}h(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) + \mathcal{L}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) + \lambda(h(\tilde{\boldsymbol{x}})) + \lambda(h(x)) - \lambda(h(\tilde{\boldsymbol{x}})) \\ &\geq -\|\mathcal{L}h(\boldsymbol{x}, \boldsymbol{y}) - \mathcal{L}h(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})\| + \mathcal{L}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) + \lambda(h(\tilde{\boldsymbol{x}})) - 3Mr - \|\lambda(h(x)) - \lambda(h(\tilde{\boldsymbol{x}}))\| + 3Mr \\ &\geq -M(2\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| + \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|) + 3Mr \geq 0. \end{aligned}$$

The proof is complete.

A.1.8 Theorems and Proofs in Stability Guarantee

Based on the safety guarantee, the state space $\mathcal{X}$ is restricted in the compact set $\mathcal{D}$, so we need to ensure the validity of the conditions assumed in Theorem 2.2 and Theorem 4.1 through the discretization on $\mathcal{D}$. We summarize the analytical results as follows.

Theorem A.4 *(Stability guarantee for NSC) For the stochastic functional differential equation* $\mathrm{d}\boldsymbol{x}(t) = F(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau))\mathrm{d}t + G(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau))\mathrm{d}B(t)$, *with* F, G *satisfying locally Lipschitz condition and locally linear growth condition. Let* $M = \mathcal{M}(F, G, \mathcal{D})$ *be the maximum of the Lipschitz constants of* $\|\boldsymbol{x}\|^2(2\langle \boldsymbol{x}, F\rangle + \|G(\boldsymbol{x})\|_{\mathrm{F}}^2)$ *and* $\boldsymbol{x}^\top G(\boldsymbol{x})$ *on* $\mathcal{D}$. *Suppose that there exists a non-negative constant* $\delta \le Mr$ *such that*

$$\|\boldsymbol{x}\|^2(2\langle \boldsymbol{x}, F\rangle + \|G\|_{\mathrm{F}}^2) - (2-\alpha)\|\boldsymbol{x}^\top G\|^2 + (7-2\alpha)Mr \le \delta,\ \forall \boldsymbol{x}, \boldsymbol{y} \in \tilde{\mathcal{D}}. \tag{15}$$

Then, under the safety condition in Theorem 5.3, the solution satisfies $\lim_{t\to\infty} \boldsymbol{x}(t;\xi) = \mathbf{0}$ *a.s. for any* $\xi \in C_{\mathcal{F}_0}([-\tau, 0]; \mathbb{R}^d)$.

Proof. Analogous to the proof performed in Appendix A.1.7, we prove the results by the interpolation method. For any $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$, there exists $\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \in \tilde{\mathcal{D}}$ such that

$$\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_2 \le r,\ \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_2 \le r.$$

Then we have

$$\begin{aligned}
&\|\boldsymbol{x}\|^2(2\langle \boldsymbol{x}, F\rangle + \|G\|_{\mathrm{F}}^2) - (2-\alpha)\|\boldsymbol{x}^\top G(\boldsymbol{x})\|^2 \\
&\quad = \|\boldsymbol{x}\|^2(2\langle \boldsymbol{x}, F(\boldsymbol{x},\boldsymbol{y})\rangle + \|G(\boldsymbol{x},\boldsymbol{y})\|_{\mathrm{F}}^2) - \|\tilde{\boldsymbol{x}}\|^2(2\langle \tilde{\boldsymbol{x}}, F(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\rangle + \|G(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\|_{\mathrm{F}}^2) \\
&\quad + \|\tilde{\boldsymbol{x}}\|^2(2\langle \tilde{\boldsymbol{x}}, F(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\rangle + \|G(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\|_{\mathrm{F}}^2) - (2-\alpha)\|\tilde{\boldsymbol{x}}^\top G(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\|^2 \\
&\quad + (2-\alpha)\|\tilde{\boldsymbol{x}}^\top G(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\|^2 - (2-\alpha)\|\boldsymbol{x}^\top G(\boldsymbol{x},\boldsymbol{y})\|^2 \\
&\quad \le M(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| + \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|) + \delta - (7-2\alpha)Mr + (2-\alpha)M(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| + \|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|) \\
&\quad \le 2Mr - (6-2\alpha)Mr + 2(2-\alpha)r \le 0.
\end{aligned}$$

Hence, the stability condition assumed in Theorem 4.1 is satisfied on $\mathcal{D}$. Under the safety condition, the state space satisfies $\mathcal{X} \subset \mathcal{D}$, so the conclusion follows directly from Theorem 4.1. This therefore completes the proof.

Theorem A.5 *(Stability guarantee for NDC) Consider the system the same as the one considered in Theorem A.4. Let* $M = \mathcal{M}(F, G, V, w_1, w_2, \mathcal{D})$ *be the maximum of the Lipschitz constants of* $\mathcal{L}V$, w_1 *and* w_2 *on* $\mathcal{D}$. *Suppose that there exists a non-negative constant* $\delta \le Mr$ *such that*

$$\mathcal{L}V(\boldsymbol{x}, \boldsymbol{y}) + w_1(\boldsymbol{x}) - w_2(\boldsymbol{y}) + 5Mr \le \delta,\ \forall \boldsymbol{x}, \boldsymbol{y} \in \tilde{\mathcal{D}}. \tag{16}$$

Then, under the safety condition in Theorem 5.3, the condition in Theorem 2.2 is satisfied within $\mathcal{X} \subset \mathcal{D}$.

Proof. The proof is the same as the proof for Theorem A.4.

Remark A.6 *Here, we consider the autonomous case for SDDE because we cannot discretize the unbounded time domain with a finite number of points. However, some non-autonomous system could be transformed into a higher-dimensional autonomous systems, which makes our theory more practically useful for a broader range of systems.*

Based on the above two theorems about stability guarantee, we can redesign the stability loss for NDC and NSC under the safety guarantee with Eqs. (15) and (16). The definition is the same as Eq. (12) and we summarize them as follows.

Loss for stability guarantee in NDC:

$$L_{\tilde{\mathcal{D}},\varepsilon,c,p}(\boldsymbol{\theta}_V, \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_w, \boldsymbol{\theta}_f) = \frac{1}{|\tilde{\mathcal{D}}|^2} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\tilde{\mathcal{D}}\times\tilde{\mathcal{D}}} \max\left(0, \mathcal{L}V(\boldsymbol{x},\boldsymbol{y}) + w_1(\boldsymbol{x}) - w_2(\boldsymbol{y}) + 5Mr\right), \tag{17}$$

Loss for stability guarantee in NSC:

$$L_{\tilde{\mathcal{D}},\alpha}(\boldsymbol{\theta}) = \frac{1}{|\tilde{\mathcal{D}}|^2} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\tilde{\mathcal{D}}\times\tilde{\mathcal{D}}} \max\big(0, (\alpha-2)\|\boldsymbol{x}^\top g_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{y})\|^2 \\ +\|\boldsymbol{x}\|^2(2\langle \boldsymbol{x}, f(\boldsymbol{x},\boldsymbol{y})\rangle + \|g_{\boldsymbol{u}}(\boldsymbol{x},\boldsymbol{y})\|_{\mathrm{F}}^2) + (7-2\alpha)Mr\big), \tag{18}$$

During the training stage, we terminate the training process once the above loss is less than Mr.

A.1.9 Proof of Theorem 6.1

First, we prove the estimation for $\mathbb{E}[\eta_\varepsilon]$. Applying Itô's formula to $V(\boldsymbol{x})$ yields:

$$V(\boldsymbol{x}(t)) = V(\boldsymbol{x}_0) + \int_0^t \mathcal{L}V(\boldsymbol{x}(s))\mathrm{d}s + \int_0^t \nabla V(\boldsymbol{x}(s)) \cdot g(\boldsymbol{x}(s), \boldsymbol{x}(s-\tau))\mathrm{d}B_s$$

$$\begin{aligned}\int_0^t \mathcal{L}V(\boldsymbol{x}(s))\mathrm{d}s &\le \int_0^t [-w_1(\boldsymbol{x}(s)) + w_2(\boldsymbol{x}(s-\tau))]\mathrm{d}s \\ &= -\int_0^t w_1(\boldsymbol{x}(s))\mathrm{d}s + \int_{-\tau}^{t-\tau} w_2(\boldsymbol{x}(s))\mathrm{d}s \\ &\le -\int_0^t [w_1(\boldsymbol{x}(s)) - w_2(\boldsymbol{x}(s))]\mathrm{d}s + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s\end{aligned}$$

Substituting t with the stopping time η_ε and taking expectation on both sides, we have

$$\mathbb{E}[V(\boldsymbol{x}(\eta_\varepsilon))] \le \mathbb{E}[V(\boldsymbol{x}_0)] + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s - \int_0^{\eta_\varepsilon} [w_1(\boldsymbol{x}(s)) - w_2(\boldsymbol{x}(s))]\mathrm{d}s.$$

From $\|\boldsymbol{x}(\tau_\varepsilon)\| = \varepsilon < \|\boldsymbol{x}_0\|,\ w_1 \ge w_2$ and $\|\boldsymbol{x}(t)\| \ge \varepsilon,\ t \le \eta_\varepsilon$, it follows that

$$\begin{aligned}&\mathbb{E}\int_0^{\eta_\varepsilon} [w_1(\boldsymbol{x}(s)) - w_2(\boldsymbol{x}(s))]\mathrm{d}s \le \mathbb{E}[V(\boldsymbol{x}_0) - V(\boldsymbol{x}(\eta_\varepsilon))] + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s, \\ &\to \mathbb{E}\int_0^{\eta_\varepsilon} \min_{\boldsymbol{x}(s)\ge\varepsilon}[w_1(\boldsymbol{x}(s)) - w_2(\boldsymbol{x}(s))]\mathrm{d}s \le \mathbb{E}[V(\boldsymbol{x}_0) - \min_{\|\boldsymbol{x}\|=\varepsilon} V(\boldsymbol{x})] + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s, \\ &\to \mathbb{E}[\eta_\varepsilon] \le \frac{V(\boldsymbol{x}_0) - \min_{\|\boldsymbol{x}\|=\varepsilon} V(\boldsymbol{x}) + \int_{-\tau}^0 w_2(\xi(s))\mathrm{d}s}{\min_{\|\boldsymbol{x}\|\ge\varepsilon}(w_1(\boldsymbol{x}) - w_2(\boldsymbol{x}))} \triangleq T_\varepsilon\end{aligned}$$

Notice that NN control satisfies $\boldsymbol{u}(\mathbf{0},\mathbf{0}) = \mathbf{0}$. Thus, under the Lipschitz condition, we have $\|\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y})\| \le k_{\boldsymbol{u}}\|(\boldsymbol{x},\boldsymbol{y})\| \le k_{\boldsymbol{u}}\|\boldsymbol{x}\| + k_{\boldsymbol{u}}\|\boldsymbol{y}\|$. From Itô's formula for $\|\boldsymbol{x}\|^2$, we have

$$\|\boldsymbol{x}(t)\|^2 - \|\boldsymbol{x}(0)\|^2 = \int_0^t (2\langle \boldsymbol{x}, f + \boldsymbol{u}_f\rangle + \|g\|^2)\mathrm{d}s + \int_0^t 2\langle \boldsymbol{x}(s), g(\boldsymbol{x}(s), \boldsymbol{x}(s-\tau))\mathrm{d}B_s\rangle \tag{19}$$

According to the Lipschitz conditions for $f, g, \boldsymbol{u}_f$, we have

$$\begin{aligned}&\int_0^t (2\langle \boldsymbol{x}, f + \boldsymbol{u}_f\rangle + \|g\|^2)\mathrm{d}s \\ &\quad\le \int_0^t 2\|\boldsymbol{x}\|[(L + k_{\boldsymbol{u}})(\|\boldsymbol{x}(s)\| + \|\boldsymbol{x}(s-\tau)\|)] + 2L^2(\|\boldsymbol{x}(s)\|^2 + \|\boldsymbol{x}(s-\tau)\|^2)\mathrm{d}s, \\ &\quad\le \int_0^t (2L^2 + 3L + 3k_{\boldsymbol{u}})\|\boldsymbol{x}(s)\|^2 + (2L^2 + L + k_{\boldsymbol{u}})\|\boldsymbol{x}(s-\tau)\|^2\mathrm{d}s, \\ &\quad\le \int_0^t (2L^2 + 3L + 3k_{\boldsymbol{u}})\|\boldsymbol{x}(s)\|^2\mathrm{d}s + \int_{-\tau}^{t-\tau} (2L^2 + L + k_{\boldsymbol{u}})\|\boldsymbol{x}(s)\|^2\mathrm{d}s, \\ &\quad\le \int_0^t 4(L^2 + L + k_{\boldsymbol{u}})\|\boldsymbol{x}(s)\|^2\mathrm{d}s + \int_{-\tau}^0 (2L^2 + L + k_{\boldsymbol{u}})\|\xi(s)\|^2\mathrm{d}s.\end{aligned}$$

Thus, taking the expectation on both sides in Eq.(19) along the time interval $[0, t\wedge\eta_\varepsilon]$ gives

$$\begin{aligned}\mathbb{E}[\|\boldsymbol{x}(t\wedge\eta_\epsilon)\|^2] &\le C_0 + \mathbb{E}\int_0^{t\wedge\eta_\epsilon} 4(L^2+L+k_{\boldsymbol{u}})\|\boldsymbol{x}(s)\|^2\mathrm{d}s\\ &= C_0{+}4(L^2+L+k_{\boldsymbol{u}})\int_0^t \mathbb{E}[\|\boldsymbol{x}(s)\|^2\mathbb{1}_{\{s<\eta_\epsilon\}}]\mathrm{d}s,\end{aligned}$$

where $C_0 = \|\boldsymbol{x}_0\|^2 + \int_{-\tau}^0 (2L^2+L+k_{\boldsymbol{u}})\|\xi(s)\|^2\mathrm{d}s$ .Then we have

$$\begin{aligned}&\mathbb{E}[\|\boldsymbol{x}(t)\|^2\mathbb{1}_{\{t<\tau_\epsilon\}}] \le \mathbb{E}[\|\boldsymbol{x}(t\wedge\tau_\epsilon)\|^2]\\ \le& C_0 + 4(L^2+L+k_{\boldsymbol{u}})\int_0^t \mathbb{E}[\|\boldsymbol{x}(s)\|^2\mathbb{1}_{\{s<\tau_\epsilon\}}]\mathrm{d}s.\end{aligned}$$

Now, applying Gronwall's inequality, we get

$$\mathbb{E}[\|\boldsymbol{x}(t)\|^2\mathbb{1}_{\{t<\eta_\epsilon\}}] \le C_0 e^{4(L^2+L+k_{\boldsymbol{u}})t}.$$

Finally, we have

$$\begin{aligned}\mathcal{E}(\tau_\varepsilon, T_\varepsilon) &= \mathbb{E}\left(\int_0^{\tau_\varepsilon\wedge T_\varepsilon} \|\boldsymbol{u}(\boldsymbol{x}(s),\boldsymbol{x}(s-\tau))\|^2\mathrm{d}s\right)\\ &\le \mathbb{E}\left(\int_0^{\tau_\varepsilon\wedge T_\varepsilon} k_{\boldsymbol{u}}^2(\|\boldsymbol{x}(s)\|^2+\|\boldsymbol{x}(s-\tau)\|^2)\mathrm{d}s\right)\\ &\le \mathbb{E}\left(\int_0^{\tau_\varepsilon\wedge T_\varepsilon} 2k_{\boldsymbol{u}}^2\|\boldsymbol{x}(s)\|^2\mathrm{d}s + \int_{-\tau}^0 \|\xi(s)\|^2)\mathrm{d}s\right)\\ &\le 2k_{\boldsymbol{u}}^2\int_0^{T_\varepsilon} \mathbb{E}[\|\boldsymbol{x}(s)\|^2\mathbb{1}_{\{s<\tau_\epsilon\}}]\mathrm{d}s + \int_{-\tau}^0 \|\xi(s)\|^2)\mathrm{d}s\\ &\le \frac{k_{\boldsymbol{u}}^2 C_0}{2(L^2+L+k_{\boldsymbol{u}})}\left[\exp\left(4(L^2+L+k_{\boldsymbol{u}})T_\varepsilon\right)-1\right] + \int_{-\tau}^0 \|\xi(s)\|^2)\mathrm{d}s,\end{aligned}$$

which completes the proof.

A.1.10 Proof of Theorem 6.2

First, we prove the estimation for $\mathbb{E}[\eta_\varepsilon]$. From the arguments presented in Appendix A.1.4, we have

$$\begin{aligned}\|\boldsymbol{x}(t)\|^\alpha &= \|\boldsymbol{x}(0)\|^\alpha + \int_0^t \frac{\alpha}{2}\|\boldsymbol{x}\|^{\alpha-4}q(\boldsymbol{x}(s),\boldsymbol{x}(s-\tau))\mathrm{d}s + \int_0^t \alpha\|\boldsymbol{x}\|^{\alpha-2}\langle\boldsymbol{x}, g+\boldsymbol{u}_g\mathrm{d}B_s\rangle,\\ q(\boldsymbol{x}(s),\boldsymbol{x}(s-\tau)) &= (\|\boldsymbol{x}(s)\|^2(2\langle\boldsymbol{x}(s), f(\boldsymbol{x}(s),\boldsymbol{x}(s-\tau))\rangle\\ &\quad + \|(g+\boldsymbol{u}_g)(\boldsymbol{x}(s),\boldsymbol{x}(s\tau))\|_{\mathrm{F}}^2) - (2-\alpha)\|\boldsymbol{x}(s)^\top(g+\boldsymbol{u}_g)(\boldsymbol{x}(s),\boldsymbol{x}(s-\tau))\|^2).\end{aligned} \tag{20}$$

From the condition in Theorem 6.2, we have

$$\max_{\|\boldsymbol{x}(s)\|\ge\varepsilon} \frac{q(\boldsymbol{x}(s),\boldsymbol{x}(s))}{\|\boldsymbol{x}(s)\|^{4-\alpha}} \le -\delta_\varepsilon.$$

Noticing $\|\boldsymbol{x}(t)\|\ge\varepsilon,\ t\le\eta_\varepsilon$, setting t as η_ε, and taking expectation in (20), we have

$$\varepsilon^\alpha \le \|\boldsymbol{x}_0\|^\alpha - \frac{\alpha}{2}\delta_\varepsilon\mathbb{E}[\tau_\varepsilon]$$

Then we have

$$\mathbb{E}[\tau_\varepsilon] \le \frac{2(\|\boldsymbol{x}_0\|^\alpha - \varepsilon^\alpha)}{\alpha\cdot\delta_\varepsilon} \triangleq T_\varepsilon.$$

The estimation of the energy cost is just the same as that in Appendix A.1.9.

A.2 Limitations and Analysis

A.2.1 Sample Distribution

For sample distribution $\mu = \mu(\Omega)$ in NDC and NSC frameworks without safe guarantee, we empirically select a large enough closed domain Ω around the target point and uniformly sample N points in Ω as our training data. Theoretically, this sample methods can not guarantee the sufficient conditions for stability be satisfied everywhere in the domain where the system evolves even though the loss is low (or zero). In candid, the reasonable selection for sample distribution need further investigation and in this paper we do not focus on this direction due to the good numerically performance of our neural frameworks. Here we provide an explanation for the validity of our numerical experiments: the low loss implies that the LaSalle-Type or Asymptotic stability conditions are satisfied in Ω, and this may force the state trajectories initiated from Ω to the zero solution. In this way, the system will still evolve in Ω and the trained stability conditions are still effective.

However, this problem can be naturally solved when we consider safety guarantee. The condition for safety guarantee can be approximately satisfied once barrier loss is low (or zero) on training data sampled from $\mathrm{int}(\mathcal{C})$, thus the system will evolve in $\mathrm{int}(\mathcal{C})$. Now we only need the stability conditions are satisfied in $\mathrm{int}(\mathcal{C})$, so it is enough to set the data sample distribution μ for LaSalle or Asymptotic loss as uniform distribution on $\mathrm{int}(\mathcal{C})$.

A.2.2 Design of Asymptotic Loss

We omit the condition $\min_{\boldsymbol{x}_t(0)=M} \|\boldsymbol{x}_t(0)^\top G(\boldsymbol{x}_t, t)\| > 0$ when we construct the Asymptotic loss 4.1 because the NSC performs well in numerical experiments. In case of $G(\boldsymbol{x}_t, t) = G(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)$, this condition requires the output of the NN- $G(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\theta})$ locates outside the orthogonal space of $\boldsymbol{x}$. The projection operator in (Kolter & Manek, 2019) can be introduced to design our NSC to locally satisfy this condition. However, how to design a NN to globally satisfy this condition is a challenging direction that needs further investigations. For completeness, we check whether the condition is satisfied or not on the train data, and we show the results in Table 2.

Table 2: The test results for the learned control policies in the second framework. The minimum norm represents $\min_{i=1,\cdots,N} \|\boldsymbol{x}_i^\top \boldsymbol{u}_g(\boldsymbol{x}_i, \boldsymbol{y}_i)\|$ on the train data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1,\cdots,N}$, where $\boldsymbol{u}_g$ is the corresponding diffusion term in the controlled dynamics. We use (1) and (2) to denote the case in 2-D kinematic bicycle and inverted pendulum, respectively.

	NSC (1)	NSC+D (1)	NSC+M (1)	NSC (2)	NSC+Safe (2)
Minimum norm	523.7	0.2788	304.6	0.1381	0.1313
Condition satisfied?	Yes	Yes	Yes	Yes	Yes

A.3 Experimental Configurations

In this section, we provide the detailed descriptions for the experimental configurations of the control problems in the main text. The computing device that we use for calculating our examples includes a single i7-10870 CPU with 16GB memory, and we train all the parameters with Adam optimizer until the loss function is below the given training error δ.

A.3.1 Algorithms

We summarize the Algorithms of NDC and NSC as follows, we mark the part corresponding to safety in blue,

Algorithm 1: Neural Deterministic Control

Input: Data $\{\boldsymbol{x}_i, \boldsymbol{y}_i, t_i\}_{i=1}^n$ sampled from $\mu(\Omega)$, iteration step m, learning rate β, training error δ, coefficient functions f and g, initial parameters $\boldsymbol{\theta}_0$, and $\varepsilon,\ c,\ p(\boldsymbol{x})$ used in Eq.(5)(6)(7),
$\boldsymbol{\theta} = (\boldsymbol{\theta}_V, \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_w, \boldsymbol{\theta}_f)$ or $\boldsymbol{\theta} = (\boldsymbol{\theta}_V, \boldsymbol{\theta}_\gamma, \boldsymbol{\theta}_w, \boldsymbol{\theta}_f, \boldsymbol{\theta}_\lambda)$ and M_λ with safety guarantee.
Output: Controller $\boldsymbol{u}_f(\boldsymbol{x}_i, \boldsymbol{y}_i, t_i)$ and auxiliary function $V(\boldsymbol{x}_i, t_i), \gamma(t_i), w_1(\boldsymbol{x}_i), w_2(\boldsymbol{y}_i)$ in the form of Eq.(5)(6)(7). And candidate barrier function $B(\boldsymbol{x}_i)$ with safety guarantee.
for $r = 0$ **to** $m-1$ **do**
  Compute $V_t(\boldsymbol{x}_i, t_i), \nabla V(\boldsymbol{x}_i, t_i), \mathcal{H}V(\boldsymbol{x}_i, t_i)$, $\nabla B(\boldsymbol{x}_i), HB(\boldsymbol{x}_i)$ with safety guarantee
  $i = 1, \cdots, n$
  Compute LaSalle loss: $L(\boldsymbol{\theta}_r, \boldsymbol{u}_r)$ from Eq.(8) and plus Eq.(12) with safety guarantee.
  $\boldsymbol{\theta}_{r+1} = \boldsymbol{\theta}_r - \beta \cdot \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_r)$ ▷ Update parameters
  if $L(\boldsymbol{\theta}_{r+1}) \leq \delta$ **then**
    break

Algorithm 2: Neural Stochastic Control

Input: Data $\{\boldsymbol{x}_i, \boldsymbol{y}_i, t_i\}_{i=1}^n$ sampled from $\mu(\Omega)$, parameter $\alpha \in (0,1)$ used in Eq.(10), and all other parameters, m, β, δ, f, and g, defined in the same manner as those in Algorithm 1.
$\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_g)$ or $\boldsymbol{\theta} = (\boldsymbol{\theta}_f, \boldsymbol{\theta}_g, \boldsymbol{\theta}_\lambda)$ and M_λ with safety guarantee.
Output: Controller $\boldsymbol{u}(\boldsymbol{x}_i, \boldsymbol{y}_i, t_i)$, and candidate barrier function $B(\boldsymbol{x}_i)$ with safety guarantee.
for $r = 0$ **to** $m-1$ **do**
  Compute loss function: $L(\boldsymbol{\theta}_r)$ from (10) and plus Eq.(12) with safety guarantee.
  $\boldsymbol{\theta}_{r+1} = \boldsymbol{\theta}_r - \beta \cdot \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}_r)$ ▷ Update parameters
  if $L(\boldsymbol{u}_{r+1}) \leq \delta$ **then**
    break

In the above two algorithms, We can replace the stability loss in NDC and NSC with Eq. 17 and Eq. 18 when we want to obtain the stability guarantee. We extend the QP methods in (Sarkar et al., 2020; Fan et al., 2020) to the controlled SDDEs $\mathrm{d}\boldsymbol{x}(t) = [f(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t) + u(t)]\mathrm{d}t + g(\boldsymbol{x}(t), \boldsymbol{x}(t-\tau), t)\mathrm{d}B_t$ as follows, and similarly the blue parts only appear when we consider safety guarantee.

Algorithm 3: Baseline QP Control

Input parameters: Relaxation coefficients p_1, p_2, Lyapunov exponent ε and coefficient γ of linear class-$\mathcal{K}$ function.
Objective function: $\boldsymbol{u}^* = \arg\min \|\boldsymbol{u}\|^2 + p_1 d_1^2 + p_2 d_2^2$,

Constraints: $\mathcal{L}V + \dfrac{1}{\varepsilon}V \leq d_1,$ ▷ Control Lyapunov function

$$V(\boldsymbol{x}, t) = \frac{1}{2}\|\boldsymbol{x}\|^2,$$

$\mathcal{L}\mathcal{B} - \gamma h(\boldsymbol{x}) \leq d_2,$ ▷ Control barrier function

$$\mathcal{B} = \frac{1}{h(\boldsymbol{x})}.$$

A.3.2 Analysis for Computational Complexity

Computational Complexity of NDC Although the NDC outperforms those traditional control methods in terms of flexibility, convergence rate and generalization ability, the training for NDC is not efficient. The major reason is that we should compute the Hessian matrix $\mathcal{H}V$. The computational complexity of this operator is $\mathcal{O}((mn)^2)$ for batch size $= m$ on n-D dynamics, which is extremely time-consuming on controlling high dimensional systems with large amounts of data.

Computational Complexity of NSC The NSC framework is computationally efficient because we only need the tensor operation in the training process. The computational complexity of this procedure is $\mathcal{O}(mn)$ for batch size $= m$ on n-D dynamics, which is significantly faster than the NDC framework in high dimensional tasks.

From the above investigations, we suggest that, when the task is safe-critical, the deterministic control is recommended. This is because the noise in the stochastic control can definitely bring uncertainty to the system, which may impact the efficacy of the safety guarantee. However, the deterministic control can suppress the influence of stochasticity.

A.3.3 Variants of NSC

We can use different combinations in the current framework, viz., the neural stochastic control (NSC) $\boldsymbol{u} = (\boldsymbol{0}, \boldsymbol{u}_g)$, the neural deterministic control in this framework (NSC+D) $\boldsymbol{u} = (\boldsymbol{u}_f, \boldsymbol{0})$, and the neural mixed control (NSC+M) $\boldsymbol{u} = (\boldsymbol{u}_f, \boldsymbol{u}_g)$.

A.3.4 Chua's Model in Section 3

The driving system is,

$$\begin{aligned}
\dot{x}_1 &= a[x_2 - x_1 - q(x_1)],\\
\dot{x}_2 &= b[x_1 - x_2] + cx_3,\\
\dot{x}_3 &= -dx_2,\\
q(x) &= m_0 x + \frac{1}{2}(m_1 - m_0)(|x+B| - |x-B|).
\end{aligned}$$

The response system is perturbed by uncorrelated noise,

$$\begin{aligned}
\mathrm{d}y_1 &= a[y_2 - y_1 - q(y_1)]\mathrm{d}t + g_1(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_1(t),\\
\mathrm{d}y_2 &= [b(y_1 - y_2) + cy_3]\mathrm{d}t + g_2(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_2(t),\\
\mathrm{d}y_3 &= -dy_2 + g_3(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_3(t),
\end{aligned}$$

where $\boldsymbol{z} = (z_1, z_2, z_3)^\top = (x_1 - y_1, x_2 - y_2, x_3 - y_3)^\top$, $\boldsymbol{z}_\tau(t) = \boldsymbol{z}(t-\tau)$, and the control goal is finding deterministic control u that can completely synchronize the response system to the driving system, that is $\boldsymbol{z} = 0$. The SDDEs of variation $\boldsymbol{z}$ is

$$\begin{aligned}
\mathrm{d}z_1 &= a[z_2 - z_1 - (p(x_1) - p(y_1))]\mathrm{d}t + g_1(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_1(t),\\
&\leq \big(a[z_2 - z_1] + |a|\max(|m_0|, |m_1|)|z_1|\big)\mathrm{d}t + g_1(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_1(t),\\
\mathrm{d}z_2 &= [b(z_1 - z_2) + cz_3]\mathrm{d}t + g_2(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_2(t),\\
\mathrm{d}z_3 &= -dz_2\mathrm{d}t + g_3(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_3(t),
\end{aligned}$$

and we denote the above equations as $\mathrm{d}\boldsymbol{z} \leq f(\boldsymbol{z})\mathrm{d}t + g(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_t$. So we only need to find deterministic control for the corresponding master equation $\mathrm{d}\boldsymbol{z} = [f(\boldsymbol{z}) + \boldsymbol{u}(\boldsymbol{z}, \boldsymbol{z}_\tau, t)]\mathrm{d}t + g(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\mathrm{d}B_t$. Here we set $a = 7, b = 0.35, c = 0.5, d = 7, m_0 = -1/7, m_1 = -40/7, B = 1, \tau = 0.1$, initial value for driving system is $\xi_{\boldsymbol{x}}(t) = (1.5 - \sin(t), -4.4 - \sin(t), 0.15 - \sin(t))^\top$, initial value for response system is $\xi_{\boldsymbol{y}}(t) = (15 + \exp(t), -4 + \exp(t), 1.5 + \exp(t))^\top$, and we consider two cases of diffusion terms

Autonomous Diffusion Term We set $g(\boldsymbol{z}, \boldsymbol{z}_\tau, t)$ as follows,

$$g_1(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = g_2(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = g_3(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = \sum_{i=1}^{3}[\sin(2z_i) - \sin(z_{\tau,i})],$$

which satisfies the condition $\|g_i(\boldsymbol{z}, \boldsymbol{z}_\tau, t)\|^2 \leq q_i\|\boldsymbol{z}\|^2 + r_i\|\boldsymbol{z}_\tau\|^2$ for some positive numbers $q_i, r_i, i = 1, 2, 3$ in (Lin & He, 2005). For finding the autonomous neural control $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{y})$ in this case, we sample 10000 data $(\boldsymbol{x}, \boldsymbol{y})$ from uniform distribution $\mathcal{U}([-50, 50]^6)$. We parameterize the functions $V(\boldsymbol{x})$ as $3 \times 12 \times 1$ NN with **Tanh** activation, $w(\boldsymbol{x})$ as $3 \times 6 \times 6 \times 1$ NN with **ReLU** activation, $\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{y})$ as $6 \times 24 \times 24 \times 3$ NN with **ReLU**. We set $\varepsilon = 1\text{e-}4$, $p(\boldsymbol{x}) = \varepsilon\exp(-\frac{1}{\|\boldsymbol{x}\|^2})$, $\delta = 1\text{e-}3$ and we train the parameters with learning rate (lr) 0.05 for 200 steps for 20 batches with batch size 1000.

Nonautonomous Diffusion Term Now we set time-dependent $g(\boldsymbol{z}, \boldsymbol{z}_\tau, t)$ as

$$g_1(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = g_2(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = g_3(\boldsymbol{z}, \boldsymbol{z}_\tau, t) = 5(1+t)\sum_{i=1}^{3}[\sin(2z_i) - \sin(z_{\tau,i})],$$

which obviously contradicts the condition $\|g_i(\boldsymbol{z},\boldsymbol{z}_\tau,t)\|^2 \leq q_i\|\boldsymbol{z}\|^2 + r_i\|\boldsymbol{z}_\tau\|^2$. Hence, the response system can not be synchronized to the driving system by the linear control in (Lin & He, 2005). Now we learn the nonautonomous neural control $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y},t)$ in this case, we sample 10000 data $(\boldsymbol{x},\boldsymbol{y},t)$ from uniform distribution $\mathcal{U}([-30,30]^6\times[0,10])$. We parameterize the functions $V(\boldsymbol{x},t)$ as $4\times16\times1$ NN with **Tanh** activation, $\gamma(t)$ as $1\times6\times6\times1$ NN with **ReLU** multiplied by e^{-t}, $\boldsymbol{u}(\boldsymbol{x},\boldsymbol{y},t)$ as $7\times24\times24\times3$ NN with **ReLU**. The others are the same as those in autonomous diffusion term case. Finally we show the controlled trajectories for y_1, y_3 in Figure 7 as complement. The system is simulated with Euler–Maruyama numerical scheme, and the random seeds are set as $\{20\times i, i=0,\cdots,9\}$.

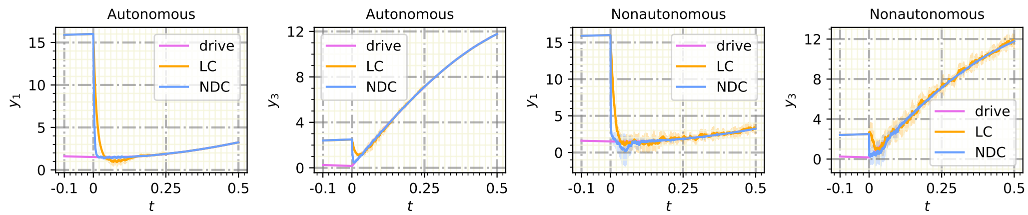

Figure 7: (a) Time trajectories of y_1 in autonomous case, (b) time trajectories of y_3 in autonomous case, (c) time trajectories of y_1 in nonautonomous case, (d) time trajectory of y_3 in nonautonomous case.

A.3.5 2-D KINEMATIC BICYCLE MODEL IN SECTION 4

Here we model the state of the common noise-perturbed kinematic bicycle system as $\boldsymbol{x} = (x,y,\theta,v)^\top$, where x,y are the coordinate positions in phase plane, θ is the heading, v is the velocity. The dynamic is as follows,

$$\begin{aligned}
&\mathrm{d}x(t) = v(t)\cos\theta(t)\mathrm{d}t + x(t-\tau)\mathrm{d}B_t\\
&\mathrm{d}y(t) = v(t)\sin\theta(t)\mathrm{d}t + y(t-\tau)\mathrm{d}B_t\\
&\mathrm{d}\theta(t) = v(t)\mathrm{d}t\\
&\mathrm{d}v(t) = (x(t)^2 + y(t)^2)\mathrm{d}t
\end{aligned}$$

The initial value is $\xi = ((2+\dfrac{t}{3})\cos(\dfrac{\pi}{2}+t), (2+\dfrac{t}{3})\sin(\dfrac{\pi}{2}+t), \pi/2+t, 2+t/3)^\top,\ t\in[-0.1,0]$. For training deterministic control $\boldsymbol{u}_f$ and stochastic control $\boldsymbol{u}_g$ under different frameworks, we sample 2000 data from $\mathcal{U}([-10,10]^8)$, we construct the NNs as follows.

NDC We parameterize $V(\boldsymbol{x})$ as $4\times16\times1$ NN with **Tanh** activation, $w(\boldsymbol{x})$ as $4\times8\times8\times1$ NN with **ReLU** activation, $\boldsymbol{u}_f(\boldsymbol{x},\boldsymbol{y})$ as $8\times32\times32\times4$ NN with **ReLU**. We set $\varepsilon=$ 1e-4, $p(\boldsymbol{x}) = \varepsilon\exp(-\frac{1}{\|\boldsymbol{x}\|^2})$, $\delta=$ 5e-4. We train the parameters with $\mathrm{lr}=0.05$ for 200 steps.

NSC We parameterize both $\boldsymbol{u}_f(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}_f)$ and $\boldsymbol{u}_g(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\theta}_g)$ as $8\times32\times32\times4$ NNs with **ReLU** activation, and we set $\alpha=0.8,\ \delta=$ 1e-8. We train the $\boldsymbol{\theta}=(\boldsymbol{\theta}_f,\boldsymbol{0})$, $\boldsymbol{\theta}=(\boldsymbol{\theta}_f,\boldsymbol{\theta}_g)$, $\boldsymbol{\theta}=(\boldsymbol{0},\boldsymbol{\theta}_g)$ in NSC(+D), NSC(+M), NSC cases, respectively, for 200 steps until the Asymptotic loss is below δ.

We use Euler–Maruyama numerical scheme to simulate the system without and with control, and the random seeds are set as $\{20\times i, i=0,\cdots,9\}$. For QP method without safety guarantee, we set $p_1=10, p_2=0, \varepsilon=0.2, \gamma=5$. We provide a more comprehensive comparison in Table 3.

A.3.6 INVERTED PENDULUM IN SECTION 5

The pendulum can be written as a system with two state variables: θ, the angle deviating from the vertical position, and $\dot{\theta}$, the angular velocity. Denote the 2-D state variable by $\boldsymbol{x}=(\theta,\dot{\theta})\triangleq(x,y)$

Table 3: Results on kinematic bicycle model.

	Tt	$\mathcal{E}_{0.001}$	Nd	$\mathbb{E}[\tau_{0.001}]$
NDC	1028.81s	102.17	6.3e-4	1.81
NSC	**59.80**s	**62.10**	4.0e-7	0.29
NSC+D	61.56s	529.23	4.3e-6	0.25
NSC+M	82.11s	196.21	**6.0**e-**8**	**0.19**
QP	-	-	0.016	> 5

and we have the following noise retarded equations

$$\begin{aligned} \mathrm{d}x(t) &= y(t)\mathrm{d}t + \sin x(t-\tau)\mathrm{d}B_t, \\ \mathrm{d}y(t) &= \left[\frac{g}{l}\sin x(t) - \frac{b}{ml^2}y(t)\right]\mathrm{d}t + \sin y(t-\tau)\mathrm{d}B_t \end{aligned} \tag{21}$$

The initial value is $\xi = (\frac{\pi}{2} - t, -1 + \frac{t}{3}),\ t \in [-0.1, 0]$. For training stochastic control $\boldsymbol{u}_g$ without and with safety guarantee $|\theta| \leq 2\pi$, we sample 1000 data from $\mathcal{U}([-5,5]^4) \subset ([-2\pi, 2\pi]^4)$ to accelerate the convergence of barrier loss. We construct the NNs as follows.

NSC We parameterize $\boldsymbol{u}_g(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_g)$ as $4 \times 16 \times 16 \times 2$ NN with **ReLU** activation. We set $\alpha = 0.8,\ \delta =$ 1e-5, and train the parameters $\boldsymbol{\theta}_g$ for 200 steps with $\mathrm{lr} = 0.05$.

NSC+Safety Based on the above constructions in NSC, we set $h(\theta, \dot{\theta}) = \pi^2 - \theta^2$, and set the class-$\mathcal{K}$ function λ as a parameterized UMNN (Wehenkel & Louppe, 2019):

$$\lambda(x) = \int_0^x \min(M_\lambda, q_{\boldsymbol{\theta}_\lambda})\mathrm{d}s,$$

where $q_{\boldsymbol{\theta}_\lambda} > 0$ and M_λ is the hyperparameter to control the Lipschitz constant of λ. We use $1 \times 10 \times 10 \times 1$ NN with **ELU** to parameterize $q_{\boldsymbol{\theta}_\lambda}$. We train the parameters $(\boldsymbol{\theta}_g, \boldsymbol{\theta}_\lambda)$ simultaneously for 2000 steps with $\mathrm{lr} = 0.05$.

Discretization and Lipschitz constant. We use the square domain $\mathcal{D} = [-\pi, \pi]^2$ to cover the safety region and use `torch.linspace` to discretize this domain on each dimension with interval r. Then we obtain the Lipschitz in Eq. (11) as follows:

$$\begin{aligned} M &= \max(2\pi M_\lambda, 2(L_f\pi + M_f) + (L_g+1)(\pi + M_g)) \\ L_f &= 1 + g/l + b/ml^2,\ M_f = g/l + (1 + b/ml^2)\pi, \\ L_g &= 2,\ M_g = 2, \end{aligned}$$

which directly follow from the calculation of the concrete form of Eq. (21). Moreover, the Lipschitz constant of control $\boldsymbol{u}$ is less than 1 and the Lipschitz constant of λ is less than M_λ. For simplicity, we set $g = m = l = b = 1$, $M_\lambda = 200$, and $r = 0.05$.

MPC Following the standard setting in (Camacho & Alba, 2013), we set the horizon in MPC rollout process as $N = 10$, with the constraints $|x| \leq 2\pi,\ (u_1^2(k) + u_2^2(k)) \leq 100,\ (x_0, y_0) = \theta, \dot{\theta},\ x_{k+1} = x_k + \delta t(y + u_1(k)), y_{k+1} = y_k + \delta t(g/l\sin(x_k) - b/(ml^2)y_k + u_2(k)), k = 1, \cdots, N$. The objective function is set as $x^2(N) + y^2(N)$. Here δ is the step in Euler simulation.

In Euler–Maruyama numerical simulation, we pick random seeds that the state trajectories under NSC control cross the safety boundary $|\theta| = 2\pi$, and we test the performance of NSC(+Safety) control on these same random seeds $\{1, 4, 79, 80, 81\}$. For baseline QP method with safety guarantee, we set $p_1 = 20, p_2 = 20, \varepsilon = 0.2, \gamma = 5.0$.

A.4 More Experiments

A.4.1 Influence of α

We test the performances of the NSC in the stabilization of system (21) for different values of α, where the values of α are equally spaced in $[0, 1]$. To this end, we construct the stochastic control $\boldsymbol{u}_g$ as $4 \times 16 \times 16 \times 2$ NN with **ReLU** activation. We sample 300 points from $\mathcal{U}([-5, 5]^4)$ as the training data. For each α and the corresponding NSC, we sample 10 controlled trajectories along the time interval $[-0.1, 0.5]$. We depict the average final position of the variable $\theta(t)$ over the 10 sampled trajectories and the cin Figure 8. Clearly, the control efficacy tends to be better and better with an increase of α.

Moreover, we select three values, $\{0.1, 0.5, 0.9\}$, for α to specifically compare the stabilization performance with baseline QP method on 12 random seeds. As clearly shown in Figure 9, the performance of the correspondingly-constructed control function $\boldsymbol{u}_g$ becomes stronger as the value of α increases, and all the neural control outperform the baseline.

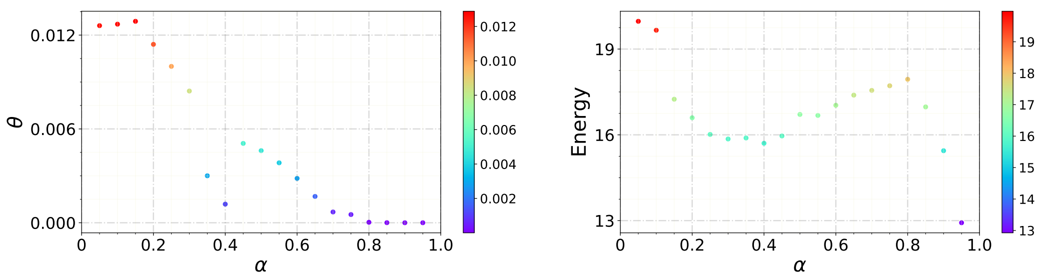

Figure 8: (Left) The convergence positions of $\theta(t)$ in the controlled system (21) for different values of α selected from $\{0.05, 0.1, 0.15, \cdots, 0.95\}$, (Right) the corresponding energy cost in the control process. Here in the simulations, the time for the convergence position is set at $t = 0.5$, and the convergence position for each α is obtained through averaging the quantities of the 10 sampled trajectories with random seeds $\{0, 1, \cdots, 11\}$

A.4.2 Controlling the Gene Regulatory Networks

Here, we show that our proposed frameworks of neural control can perform well in controlling high-dimensional dynamics. We consider the Michaelis-Menten equation (Alon, 2006), which is applied to the gene regulatory networks and governs the concentration of substrates as

$$\dot{x}_i = -cx_i^a + \sum_{j=1}^{N} W_{ij} \frac{x_j^b}{1 + x_j^b},\ i = 1, \cdots, N, \tag{22}$$

where a, b, c are positive parameters. We focus on cooperative interactions in which the nodes x_i positively contribute to each other's activity, i.e. $W_{ij} \geq 0$. We fix $a = c = 1, b = 2, N = 100$ and generate adjacent matrix as small world network (Watts & Strogatz, 1998), and assign values to the edged positions according to a distribution $\mathcal{U}([0, 2])$. The systems exhibits an active state $\boldsymbol{x}_1^*$ in which all $x_{1,i} > 0$, and an inactive state $\boldsymbol{x}_0^*$ in which all $x_{0,i} = 0$. Hence, the following noise-perturbed dynamic has the same equilibrium states as the Eq. (22).

$$\mathrm{d}x_i = \big(- x_i + \sum_{j=1}^{100} W_{ij} \frac{x_j^2}{1 + x_j^2} \big)\mathrm{d}t + \sin(\frac{x_i}{x_{1,i}}\pi)\mathrm{d}B_t,\ i = 1, \cdots, 100, \tag{23}$$

The domain of attraction of $\boldsymbol{x}_1^*$ is larger than that of $\boldsymbol{x}_0^*$, as shown in Figure 10. We now use our NSC to enlarge the attraction region of $\boldsymbol{x}_0^*$, that is, any trajectory initiated from the domain of attraction of $\boldsymbol{x}_1^*$ will be stabilized to $\boldsymbol{x}_0^*$.

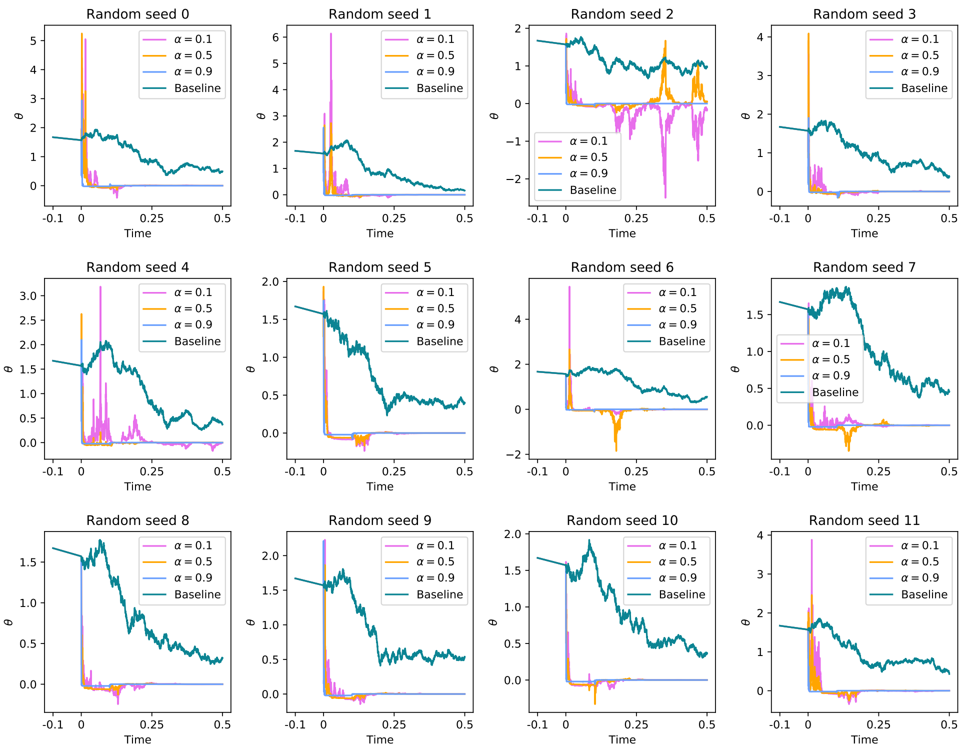

Figure 9: Comparison of control results of the controlled system (21) for the hyperparameter α in the NSC taking values from $\in \{0.1, 0.5, 0.9\}$, respectively, and baseline QP control.

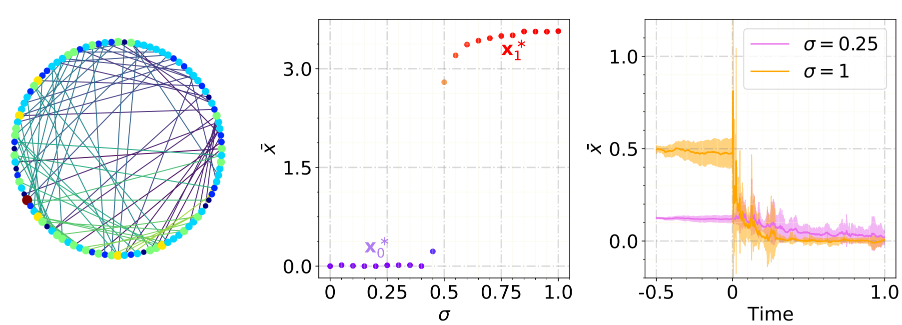

Figure 10: (Left) The schematic diagram of weighted small world network W in Eq. (22). (Middle) Average activity $\bar{\boldsymbol{x}}$ versus initial data $\mathrm{d}\xi(t) = \sigma \mathrm{d}B_t$ on $[-0.5, 0]$ with $\xi(-0.5) \sim \mathcal{U}([0, \sigma])$ in Eq. (23). (Right) Controlled average activity along time with ξ initiated from attraction region of $\boldsymbol{x}_0^*$ and $\boldsymbol{x}_1^*$, respectively, $\sigma = 0.25$ and $\sigma = 1$. The results are sampled with 10 random seeds $\{0, 1, \cdots, 9\}$.

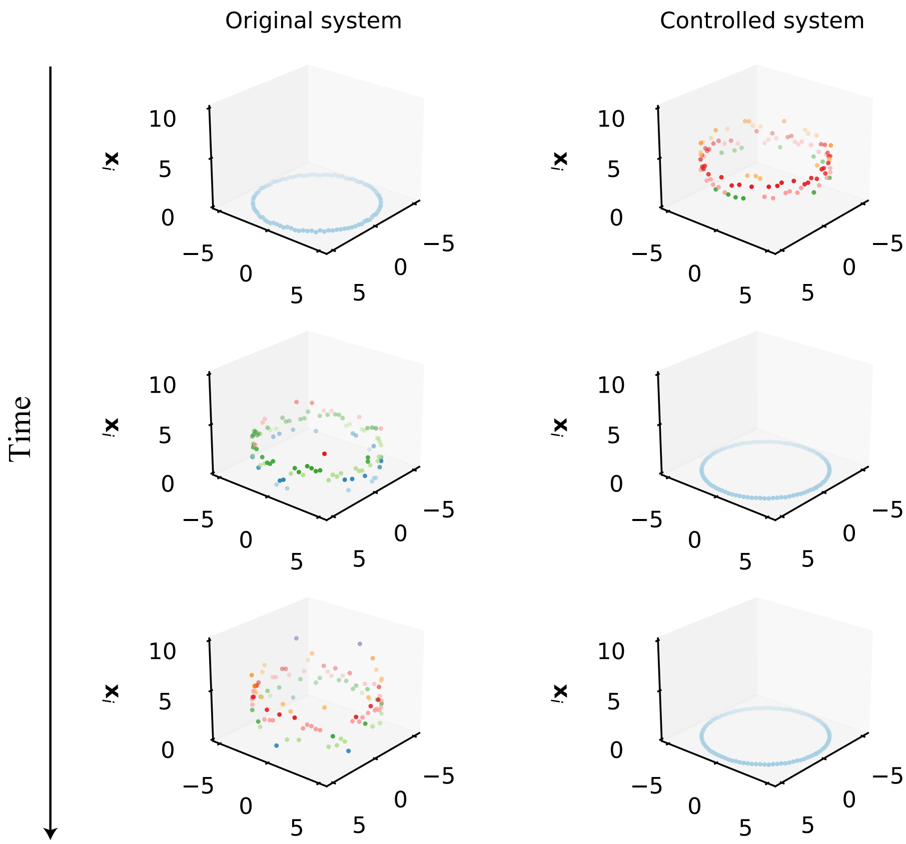

Figure 11: Time evolution of the gene regulatory networks. The state variables in the original system are activated near the inactive state $\boldsymbol{x}_0^*$ (left), and the active state $\boldsymbol{x}_1^*$ can be suppressed to the inactive state in the controlled system (right).