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 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. The two learned control policies, the neural deterministic controller (NDC) and the neural stochastic controller (NSC), work effectively because the learning procedures use, 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 NDC and NSC 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 can only evolve in some specific safety set. Such successful stabilization based on neural networks restricted in safety set is attributed to our further developed theory for safety and stability guarantee of SDDEs using the stochastic control barrier function and spatial discretization, and we name it 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 leveraged 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 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), promoting us to design neural controllers for general nonlinear SDDEs.

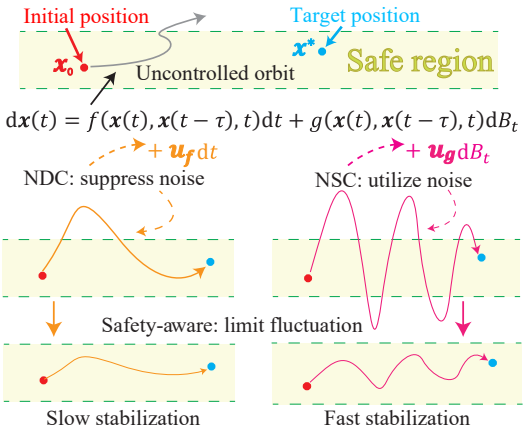

Figure 1: Sketches of SYNC. Both the NDC and NSC can stabilize the SDDEs to the target unstable equilibrium x^*. The safety-aware controlled state trajectories are restricted in the safe region.

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 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 are listed as follows

- 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,
- cultivating a SCBF theory for SDDEs and providing analytical results of the safety guarantee and stability guarantee in neural network settings,
- providing theoretical estimation for proposed neural controller in terms of convergence time and energy cost based on the safety and stability guarantee, 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 of the following general form:

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

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 $F(\mathbf{0}, \mathbf{0}, t) = \mathbf{0},\ G(\mathbf{0}, \mathbf{0}, t) = \mathbf{0}$ to guarantee that $\boldsymbol{x}(t) \equiv \mathbf{0},\ t \geq 0$ is an equilibrium of Eq. (1). For simplicity, the following notations and assumptions are used throughout the paper.

Assumption 2.1 *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 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, $V_t,\ \nabla V$ 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 used in the establishment of our part of the 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. $\lim_{t \to \infty} \boldsymbol{x}(t; \xi) \neq \mathbf{0}$ on some set of positive measures. We aim to stabilize the zero solution with 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 $\boldsymbol{u}_f(\mathbf{0}, \mathbf{0}, t) = \boldsymbol{u}_g(\mathbf{0}, \mathbf{0}, t) = \mathbf{0}$ 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 stabilized 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 and $\boldsymbol{u}_g : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^{d \times r}$ 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, hence we cannot apply MDP (Markov decision process)-based methods, such as reinforcement learning, to control SDDEs. Existing works prefer to learning deterministic control and regard the noise as a negative part that may destroy the natural dynamics of f. In what follows, we not only show that the deterministic control can achieve stabilization, but also that it can be done by elaborately-designed stochastic control, yielding two frameworks, 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 a framework of neural deterministic control (NDC) based on Theorem 2.2 to stabilize system (3) and provide experimental comparisons with traditional control methods. In addition, we analytically and experimentally investigate that the framework cannot be directly generalized to the stochastic control version. Such a problem will be addressed in the next section.

3.1 Method: Learning Control and Auxiliary Functions

The core idea of our method is that once we construct the auxiliary functions V, γ, w_1, w_2 and the neural controller $\boldsymbol{u}$, resulting in the controlled system (4) satisfying all the conditions assumed in Theorem 2.2, then the solution $\boldsymbol{x}(t; \xi)$ converges to the $\mathrm{Ker}(w_1 - w_2)$. In particular, if we set $\mathrm{Ker}(w_1 - w_2) = \{\mathbf{0}\}$, the unstable zero solution of the control-free system (3) can be stabilized. To this end, we first provide the reasonable 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. To be concrete, 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 choose C^2 activation function of an NN, such as the hyperbolic tangent function, $\mathrm{Tanh}(\cdot)$. We further discuss the influence of the L_2 term in Appendix A.1.3. In order to design an integrable positive function $\gamma(t)$ with an 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

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

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 its i-th diagonal element being x_i.

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

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

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 large enough closed region* Ω *as used in (Han et al., 2016; Chang et al., 2019), and we provide further analysis about the influence of* μ *in Appendix A.2.1.*

3.2 Numerical and Analytical Investigations

Comparison with Existing Methods 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 neural networks to automatically learn the control policies, and can be easily applied in any kinds of time-delayed systems having 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 third-order autonomous dynamical system with only one nonlinear element, producing typical chaotic dynamics (Matsumoto, 1984). In the experiment, 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 while the nonautonomous time-delay noise is beyond the concern in (Lin & He, 2005). The configuration of the experiment is described in Appendix A.3.4.

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 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 $l(\boldsymbol{\theta}_g) \geq 0$ at most time. Hence, when we minimize $l(\boldsymbol{\theta}_g) \geq 0$ in the training procedure, the ideal case $l(\boldsymbol{\theta}_g) = 0$ is

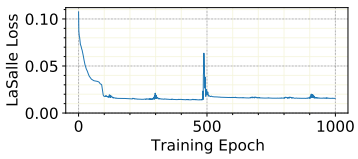

Figure 2: Training loss for the 1-D SDDE.

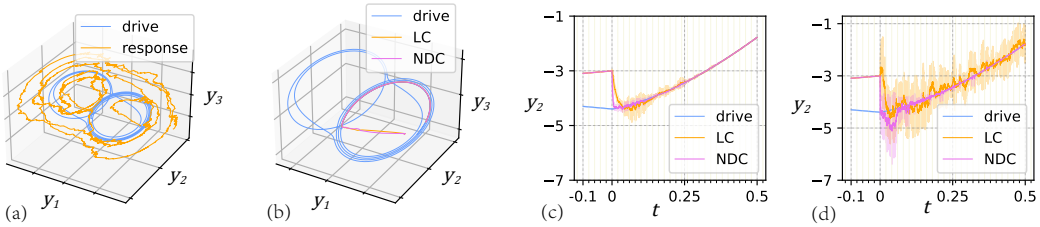

Figure 3: (a) The original driving-response model, (b) the controlled orbits under LC and NDC, (c) the time trajectory of y_2 with autonomous noise , and (d) nonautonomous noise. The solid lines are obtained through averaging the 10 sampled trajectories, while the shaded areas stand for the standard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

equivalent to $\boldsymbol{u}_g = 0$, which means 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 address the illustrated problems above, we design a new concise framework for fast learning a stochastic controller, called the neural stochastic control (NSC). To this end, we first 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 here is to construct the stability condition that makes the diffusion term contribute a negative number, contrary to the Theorem 2.2.

Theorem 4.1 *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,* $F(\mathbf{0}, t) = \mathbf{0}$*, and* $G(\mathbf{0}, t) = \mathbf{0}$*. 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* $\lim_{t\to\infty} \boldsymbol{x}(t; \xi) = 0$ *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 $\boldsymbol{u}_f = \mathbf{0}$ and provide more control combinations in Appendix A.3.3, whereas the loss function is differently designed as follows.

Definition 4.1 *(*Asymptotic Loss) Utilizing the notations 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 the stopping time by $\tau_\epsilon \triangleq \inf\{t > 0 : \|\boldsymbol{x}(t)\| = \epsilon\}$ and then denote the energy cost by $\mathcal{E}_\epsilon \triangleq \mathbb{E}\left[\int_0^{\tau_\epsilon} \left(\|\boldsymbol{u}_f\|^2 + \|\boldsymbol{u}_g\|^2\right) \mathrm{d}t\right]$. 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 Table 1. Table 1 includes the training time (Tt), empirical energy cost $\mathcal{E}_{0.001}$, nearest distance (Nd) between the bicycle and target position, empirical expectation $\mathbb{E}[\tau_{0.001}]$ of different methods. We provide more experimental details in Appendix A.3.5. We can see that the ranking of the comprehensive performance is that NSC is greater than NDC than QP. This means that we can really benefit from introducing noise as control. An explanation for this phenomena is that when we regard energy cost as objective function to be minimized, randomness can lead this functional to the shortest path quicker than the deterministic control, just like stochastic gradient descent outperforms the full-batch gradient descent. We show the NSC can enlarge the region of attraction of 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 deterministic control, there exists an obvious 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.

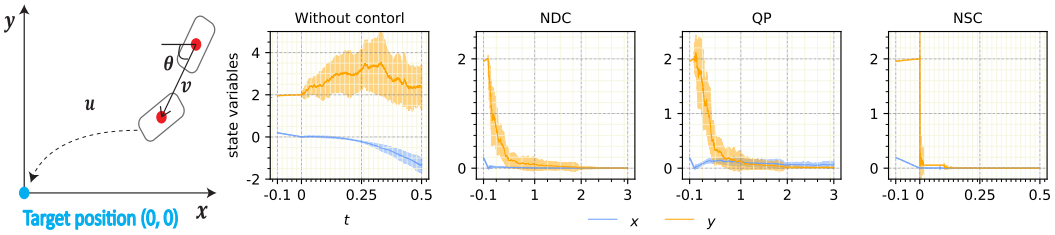

Figure 4: (Left) A schematic diagram of the kinematic bicycle model. (Right) Time trajectories of the state variables x, y of the kinematic bicycle under different control cases. The solid lines are obtained through averaging the 10 sampled trajectories, while the shaded areas stand for the standard 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 derive the theoretical result of 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 $\xi(0) \in \mathrm{int}(\mathcal{C})$, where $\mathcal{C}$ is a compact set and defined by a local Lipschitz function $h\colon \mathbb{R}^d \to \mathbb{R}$ as $\mathcal{C} = \{\boldsymbol{x} : h(\boldsymbol{x}) \geq 0\}$, and h is called a stochastic control barrier function (SCBF). Inspired by (Lechner et al., 2022), we prove that the safety and stability conditions for neural network form functions can be guaranteed through a stricter condition on finite samples. We include proofs of all the theoretical results in Appendix A.1.

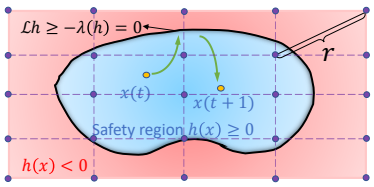

Figure 5: Diagram of the safety guarantee. We check the safety condition on discretization points with mesh r.

Definition 5.1 *A continuous function $\alpha : (-b, +\infty) \to (-\infty, +\infty)$ is said to belong to the extended class-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 easily applied to the SDDEs. We use this QP method as the baseline and the specific algorithm is shown in Appendix A.3.1. We further use 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 exists extended class-K functions $\alpha_1(\boldsymbol{x})$, $\alpha_2(\boldsymbol{x})$ and $\alpha_3(\boldsymbol{x})$ such that, $\frac{1}{\alpha_1(h(\boldsymbol{x}))} \leq \mathcal{B}(\boldsymbol{x}) \leq \frac{1}{\alpha_2(h(\boldsymbol{x}))}$, and $\mathcal{L}\mathcal{B}(\boldsymbol{x}, \boldsymbol{y}, t) \leq \alpha_3(h(\boldsymbol{x}))$ for the SDDE (1). Then for all t, $\mathbb{P}\big(\boldsymbol{x}(t) \in \mathrm{int}(\mathcal{C})\big) = 1$, provided that $\boldsymbol{x}(0) \in \mathrm{int}(\mathcal{C})$.*

A natural idea is to integrate Proposition 5.1 into our proposed neural framework, but the main flaw of this proposition is that $\mathcal{B}(\boldsymbol{x})$ is unbounded on $\mathcal{C}$ and lacks Lipschitz continuity, which makes it impossible to guarantee the expected conditions through verification on finite samples. To conquer the difficulty, we propose the following theorem for safety guarantee, which is a significantly novel contribution to the existing barrier function theory.

Theorem 5.2 *For 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, if there exists an extended class-K function $\lambda(x)$ such that $\mathcal{L}h \geq -\lambda \circ h$ 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 of this equation 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-K function λ and combine it with the NDC and NSC to learn a safe controller, where the candidate λ needs to satisfy the expected condition in Theorem 5.2. However, the main difficulty is to guarantee the condition for every point $\boldsymbol{x} \in \mathcal{D}$, since we can basically guarantee this condition on finite training data $\tilde{\mathcal{D}}$ with $\tilde{\mathcal{D}}$ being a discretization of $\mathcal{D}$. However, the following theorem shows that it is sufficient to check a slightly stricter condition on states in $\tilde{\mathcal{D}}$ 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 $\mathcal{D}$. r denotes the mesh size of $\tilde{\mathcal{D}}$, implying that 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 some non-negative constant $\delta \leq Mr$ such that*

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

Then λ satisfies the safety condition, i.e. the condition in Theorem 5.2.

Remark 5.4 *Here the non-negative δ is regarded as the tolerance error in training stage, that is, we can stop 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 as the left hand side in Eq. (11). However, the Lipschitz constant M depends on the Lipschitz constants of the neural network functions λ and $\boldsymbol{u}$,which may cause the loss function very complex and difficult to train. To simplify the loss function, we construct the neural networks with bounded Lipschitz constants for λ and $\boldsymbol{u}$. Specifically, we add the spectral normalization for neural control function to constrain its Lipschitz constant lower than 1 (Miyato et al., 2018; Yoshida & Miyato, 2017). We apply the monotonic neural networks to construct the candidate extended class-K function as $\lambda_{\boldsymbol{\theta}_\lambda}(x) = \int_0^x q_{\boldsymbol{\theta}_\lambda}(s)\mathrm{d}s$, where the output of the neural network $q_{\boldsymbol{\theta}_\lambda}(\cdot)$ 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. Then the Lipschitz constant of $\lambda_{\boldsymbol{\theta}_\lambda}$ is smaller than M_λ. Then we can calculate the Lipschitz constant 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,

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

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

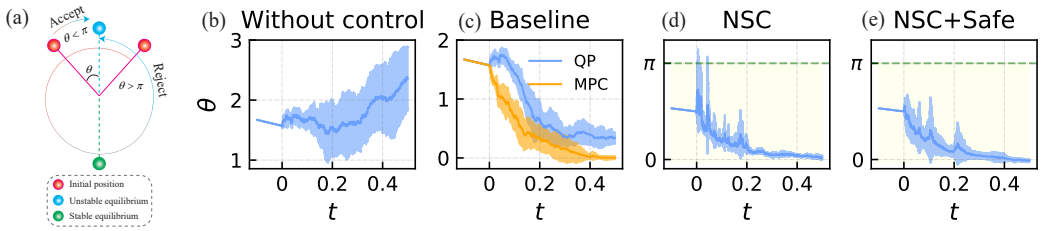

Figure 6: Schematic diagram of inverted pendulum task (a).The θ component of the original system (b), under baseline control (c), under NSC (d), under our proposed safe control (e). The solid lines are obtained through averaging the 5 sampled trajectories, while the shaded areas stand for the standard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

From Safety Guarantee to Stability Guarantee. Akin to the safety guarantee, we provide the stability guarantee for candidate neural control functions satisfying the condition in Theorems 2.2 and 4.1. The main difficulty is that both two theorems expect their conditions holds for each point $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$, but we can neither obtain a finite discretization nor a bounded Lipschitz constant on unbounded $\mathcal{X}$. Ingeniously, this difficulty can be conquered with the help of safety guarantee since the safety condition restrict $\mathcal{X} \subset \mathcal{D}$ and $\mathcal{D}$ is compact. We can obtain the similar theoretical results for stability guarantee of NDC and NSC as that in Theorem 5.3 and we summarized them in Appendix A.1.8.

We investigate the ability of the proposed safe control method to limit the fluctuation in the control process on the task of control noise-perturbed inverted pendulum with time-delay, which is a standard nonlinear control problem for testing different control methods (Anderson, 1989; Huang & Huang, 2000). We apply the safe 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 can bound the Lipschitz constant $k_{\boldsymbol{u}}$ of control function. Based on the safety and stability guarantee, the neural controller can satisfy the conditions in the Theorem 2.2 and Theorem 4.1. Then we have the following two theorems and we provide the proofs in Appendix A.1.9,A.1.10.

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

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

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 the controlled system satisfies the similar conditions in Theorem 2.2 and Remark 3.1 with $\mathrm{Ker}(w_1 - w_2) = \mathbf{0}$. Denote stopping time as $\eta_\varepsilon = \inf\{t > 0 : \|\boldsymbol{x}(t)\| = \varepsilon\}$, the corresponding energy cost in the control process with $\epsilon < \|\boldsymbol{x}_0\|$ as $\mathcal{E}(\eta_\varepsilon, T) = \mathbb{E}[\int_0^{\eta_\varepsilon \wedge T} \|\boldsymbol{u}(\boldsymbol{x}(s), \boldsymbol{x}(s-\tau))\|^2 \mathrm{d}s]$ and under 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

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

where f, g are the same as those in Theorem 5.2. Assume the controlled system satisfies the similar conditions in Theorem 4.1, under 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 neural network 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 decrease 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 holds for V. Hence, we can construct $w_{1,2}$ and V as neural networks with steeper slope at the origin to accelerate the control process, and thus reduce the upper bound of energy cost, for NDC. Secondly, the time upper bound of NSC is directly relates to the hyperparameter α in the training period, so we can 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 influence 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). In comparison, our work focuses much on learning control policies for more general dynamics with both time-delay and stochastic settings.

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 have proposed SYNC, a novel control family including two frameworks to learn control policies for nonlinear systems with time-delay in stochastic setting with safe guarantee. To this end, We prove two new theorems on stability and safety criteria for SDDEs that are more concise than the existing theories. We provide analytical results to show that we can obtain safety and stability guarantee for the neural controllers. The frameworks significantly simplify the process of control design, and we theoretically and numerically investigate the propose methods in terms of energy cost and convergence time with comparison of the existing methods. Furthermore, our frameworks can easily generalize to find control policies for SFDEs with the theorems we proved. Finally, we provide some limitations of the SYNC in Appendix A.2, which need in-depth investigation in future works.

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.

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.