Control of stochastic quantum dynamics by differentiable programming

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035004 https://doi.org/10.1088/2632-2153/abec22

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
30 December 2020

REVISED
19 February 2021

ACCEPTED FOR PUBLICATION
4 March 2021

PUBLISHED
22 April 2021

PAPER

Control of stochastic quantum dynamics by differentiable programming

Frank Schäfer[1,*], **Pavel Sekatski**[1,2], **Martin Koppenhöfer**[1,3], **Christoph Bruder**[1] and **Michal Kloc**[1,*]

[1] Department of Physics, University of Basel, Klingelbergstrasse 82, CH-4056 Basel, Switzerland
[2] Department of Applied Physics, University of Geneva, CH-1211 Geneva, Switzerland
[3] Pritzker School of Molecular Engineering, University of Chicago, Chicago, IL 60637, United States of America
[*] Authors to whom any correspondence should be addressed.

E-mail: frank.schaefer@unibas.ch **and** michal.kloc@unibas.ch

Keywords: scientific machine learning, quantum control, stochastic Schrödinger equation, homodyne detection, differentiable programming

Supplementary material for this article is available online

Abstract

Control of the stochastic dynamics of a quantum system is indispensable in fields such as quantum information processing and metrology. However, there is no general ready-made approach to the design of efficient control strategies. Here, we propose a framework for the automated design of control schemes based on differentiable programming. We apply this approach to the state preparation and stabilization of a qubit subjected to homodyne detection. To this end, we formulate the control task as an optimization problem where the loss function quantifies the distance from the target state, and we employ neural networks (NNs) as controllers. The system's time evolution is governed by a stochastic differential equation (SDE). To implement efficient training, we backpropagate the gradient information from the loss function through the SDE solver using adjoint sensitivity methods. As a first example, we feed the quantum state to the controller and focus on different methods of obtaining gradients. As a second example, we directly feed the homodyne detection signal to the controller. The instantaneous value of the homodyne current contains only very limited information on the actual state of the system, masked by unavoidable photon-number fluctuations. Despite the resulting poor signal-to-noise ratio, we can train our controller to prepare and stabilize the qubit to a target state with a mean fidelity of around 85%. We also compare the solutions found by the NN to a hand-crafted control strategy.

1. Introduction

The ability to precisely prepare and manipulate quantum degrees of freedom is a prerequisite for most applications of quantum mechanics in sensing, computation, simulation and general information processing. Many relevant tasks in this area can be formulated as optimal control problems, and therefore, *quantum control* is a rich and very active research field; see [1, 2] for two recent textbooks that provide a theoretical background and [3] for a recent review of important issues.

A typical goal of quantum control is to find a sequence of operations or parameter values (e.g., external field amplitudes) such that the quantum system under consideration is maintained in a certain target state or evolves in a desired fashion subject to additional boundary conditions (e.g., the fastest evolution for a prescribed maximum strength of the control fields). In the case of *feedback* control, the control sequence is determined based on a signal coming from the system [3, 4]. The control task and its boundary conditions are typically specified by a loss function. To convert the optimal control problem into an optimization task, one introduces a parametric ansatz for feedback schemes, also known as controllers, and explores the parameter space to minimize the loss function.

Reinforcement learning (RL) [5, 6] has been proposed as a suitable framework for the development of control strategies. In this framework, the controller (or *agent*) optimizes its strategy (the *policy*) based on a loss function (the *rewards*) obtained from repetitive interactions with the system to be controlled. In the context of quantum physics, RL has proven useful, e.g., for reducing the error rate in quantum gate synthesis [7], for autonomous preparation of Floquet-engineered states [8], and for optimal manipulation of many-qubit systems [9]. RL is a black-box setting, i.e., the agent has no prior knowledge about the structure of the system it interacts with, and has to develop its policy (explore the space of control parameters) by relying solely on its past interactions with the system. This makes RL very versatile, but requires many training episodes to find a good strategy.

Optimal control design is rarely performed using an (unknown) system in its original location. Instead, one trains the controller on a physical model of the real system. This means that rather than learning how to interact with an unknown environment, one actually starts with a lot of prior scientific knowledge about the system, namely its precise dynamic model. In the simplest cases, one can even solve the optimal control problem for this model analytically. Generally, the use of prior information leads to more data-efficient training [10, 11]. In the context of the present paper, we use a physical model of the system to efficiently compute the loss function's gradients with respect to the parameters of the control ansatz. Naturally, having access to the gradients of the loss function can streamline the optimal control design tremendously.

In the case of quantum control, the model consists of a quantum state space and a parametric equation of motion. The dynamics of the system can be solved for fixed values of the parameters of the model and of the controller. Usually, this is done numerically. Then, the most naive way to obtain the loss function's gradients is to solve the dynamics for a set of parameter values in the neighborhood of each point and use finite difference quotients. However, this method is infeasible if the number of parameters is large, it suffers from floating-point errors, and it may be numerically unstable. To circumvent these issues, automatic differentiation (AD) has been proposed as another approach to calculating gradients numerically [12, 13], a paradigm also known as differentiable programming (∂P) [14, 15]. By backpropagating the loss function's sensitivity through the numerical simulation, one can compute gradients with a similar time complexity to that of solving the system's dynamics [16]. Recently, these techniques have been merged with deep neural networks (NNs) as an ansatz for controllers [10, 17, 18]. This is possible because the training of deep NNs is based on stochastic gradient descent through a large parameter space and becomes efficient when used in conjunction with ∂P-compatible solvers for the system's dynamics.

In this work, we develop such a physics-informed RL framework based on ∂P and NNs to control the stochastic dynamics of a quantum system under continuous monitoring [2, 19]. Continuous measurements, such as photon counting and homodyne detection allow one to gain information on the random evolution of a dissipative quantum system [2]. This information can be used to estimate the state of the quantum system [20–22], to implement feedback protocols [23–27], to generate nonclassical states [28–31], and to implement teleportation protocols [32, 33]. Continuous homodyne detection can be realized experimentally in the microwave [34, 35] and optical regimes [20, 36]. The time evolution of a monitored quantum system is described by quantum trajectories, which are solutions of differential equations driven by a Lévy process.

To illustrate our framework, we focus on a qubit subjected to continuous homodyne detection [34, 37] described by a stochastic Schrödinger equation. We engineer a controller which provides a control scheme that fulfils a given state preparation task based on the measured homodyne current. This situation extends an earlier study [10], where it was demonstrated that ∂P can be efficiently used for quantum control in the context of (unitary) closed-system dynamics, i.e., when the dynamics follows an ordinary differential equation. The stochastic nature of the problem analyzed here renders the control task more challenging because the controller must adapt to the random evolution of the quantum state in each trajectory. Moreover, the instantaneous value of the homodyne current does not determine the actual state of the qubit. It is correlated only with the projection of the state onto the x-axis, and this signal is hidden in the noise which dominates the measured homodyne current [38, 39]. Thus, the information about the state of the qubit at a given time must be filtered out from the time series of measurement results.

This paper is organized as follows: In section 2, we describe the proposed setup of a qubit in a leaky cavity subjected to homodyne detection and derive the stochastic Schrödinger equation that describes its dynamics. We then discuss two ways to use the record of the homodyne detection signal in a feedback scheme to engineer a drive that can be applied to the qubit to perform a desired control task, e.g., state preparation or stabilization. We also introduce the concept of adjoint sensitivity methods that we use to efficiently compute gradients with respect to solutions of stochastic differential equations (SDEs). Section 3 describes the first feedback scheme in detail: here, we assume that the controller has direct access to the quantum state, e.g., through an appropriate filtering procedure applied to the measurement records. We compare three strategies, viz. a hand-crafted control scheme, one in which an NN continuously updates the control drive based on knowledge of the state, and a numerically less demanding one with a piecewise-constant control drive.

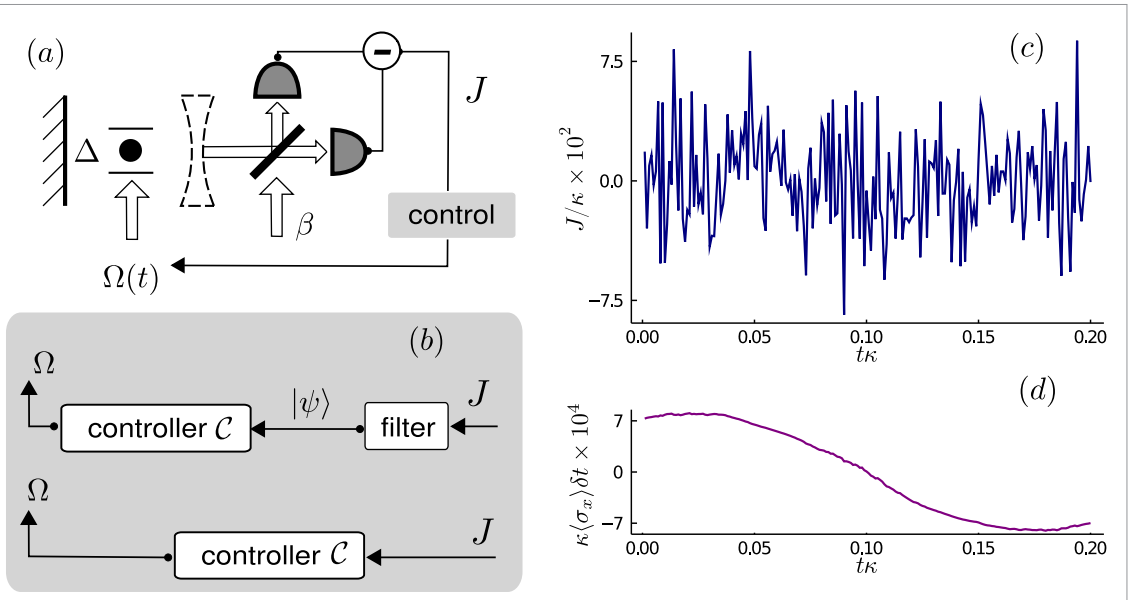

Figure 1. (a) Sketch of the setup under consideration. A qubit coupled to a leaky cavity is continuously monitored by homodyne detection. The radiation emitted by the qubit is mixed with a local oscillator laser with a complex amplitude β at a beamsplitter. The signal $J(t)$, which is the difference of the photocurrents of the two detectors, is fed to a control agent, which applies a drive $\Omega(t)$ to the qubit to generate and stabilize a target state. (b) Different structures of the controller are considered in sections 3 and 4, respectively. (c) Example of a typical homodyne detection signal $J(t)$. (d) The detected homodyne signal is proportional to the quadrature $\langle\sigma_x\rangle$, which is hidden in the noise of the measurement [note the different axis scales in (c) and (d)]. The data has been integrated over a measurement interval $\delta t = 10^{-3}\kappa^{-1}$. The parameters of the physical model are $\Delta = 20\kappa$, $\Omega = 2\kappa$.

Section 4 presents the second feedback scheme where the measured record of the homodyne current is directly fed to the NN representing the controller. In this setup, the NN must first learn how to filter the data to obtain information on the state of the system. Only after that can it propose an efficient control strategy. We conclude in section 5 and discuss potential future applications.

2. Theoretical background

2.1. A qubit under homodyne detection

We consider a driven two-level system with states $|g\rangle$ and $|e\rangle$. In a rotating frame, its Hamiltonian reads

$$H = \frac{\Delta}{2}\sigma_z + \frac{\Omega(t)}{2}\sigma_x, \tag{1}$$

where σ_x, σ_z are Pauli matrices, $\Omega(t)$ is the Rabi frequency of the driving laser, and $\Delta = \omega_{eg} - \omega_{\text{laser}}$ is the detuning between the qubit and the laser. The qubit can spontaneously decay into a photon field $a(t)$ via the interaction Hamiltonian:

$$H_{\text{int}} = \mathrm{i}\sqrt{\kappa}\left[\sigma_+ a(t) - \sigma_- a^\dagger(t)\right], \tag{2}$$

where κ is the decay rate. The field operators $a(t)$ and $a^\dagger(t)$ satisfy the commutation relation $[a(t), a^\dagger(t')] = \delta(t-t')$. We assume that the field is initially in the vacuum state, $\langle a^\dagger(t)a(t')\rangle = 0$. Physical examples of such a system include a two-level atom inside a leaky single-mode cavity that can be adiabatically eliminated or an artificial atom, e.g., a superconducting qubit, coupled to a waveguide. The radiation emitted from the two-level system is monitored using continuous homodyne measurement, as depicted in figure 1(a).

In appendix A (see also [2, 19]), we show that the evolution of the qubit is governed by a stochastic Schrödinger equation

$$\mathrm{d}|\tilde{\psi}\rangle = \mathrm{d}t\left\{-\mathrm{i}H - \frac{\kappa}{2}\sigma_+\sigma_- + J(t)\sigma_-\right\}|\tilde{\psi}\rangle, \tag{3}$$

where $|\tilde{\psi}\rangle$ denotes an unnormalized qubit state. The instantaneous value of the measured homodyne current $J(t)$ is a random variable satisfying

$$J(t) = \kappa\langle\sigma_x\rangle_{\psi(t)} + \sqrt{\kappa}\xi(t). \tag{4}$$

Here, $\langle\sigma_x\rangle_{\psi(t)}$ is the expectation value of σ_x at time t, and $\xi(t)$ is a stochastic white-noise term satisfying $\mathbb{E}[\xi(t)\xi(t')] \propto \delta(t-t')$, which stems from the shot noise of the local oscillator. Heuristically, $\xi(t)$ can be considered to be the derivative of a stochastic Wiener increment, $\xi(t) = \mathrm{d}W(t)/\mathrm{d}t$, such that the contribution of the noise to the current integrated over a short time interval $\mathrm{d}t$ is described by a Wiener process:

$$J(t)\mathrm{d}t = \kappa\langle\sigma_x\rangle_{\psi(t)}\mathrm{d}t + \sqrt{\kappa}\,\mathrm{d}W(t)\,. \tag{5}$$

The ensemble averages of the stochastic Wiener increment $\mathrm{d}W$ satisfy $\mathbb{E}[\mathrm{d}W(t)] = 0$ and $\mathbb{E}[\mathrm{d}W(t)^2] = \mathrm{d}t$. Several remarks are in order, to better understand the dynamics of the system.

First, equation (5) implies that the value of the current over a short interval $J(t)\mathrm{d}t$ contains very little information about the state $|\psi(t)\rangle$ of the qubit. This can be seen from the (heuristically stated) vanishing signal-to-noise ratio:

$$\frac{\kappa\langle\sigma_x\rangle_{\psi(t)}\mathrm{d}t}{\sqrt{\mathbb{E}[\kappa\,\mathrm{d}W(t)^2]}} = \frac{\kappa\langle\sigma_x\rangle_{\psi(t)}\mathrm{d}t}{\sqrt{\kappa\mathrm{d}t}} = \langle\sigma_x\rangle_{\psi(t)}\sqrt{\kappa\mathrm{d}t}. \tag{6}$$

If $\langle\sigma_x\rangle_{\psi(t)}$ were a constant signal, one could simply integrate the current $J(t)$ over a time interval longer than $1/\kappa$ to increase the signal-to-noise ratio. However, this is not possible because relaxation changes the state of the two-level system on the timescale $1/\kappa$. Thus, the low signal-to-noise ratio is an intrinsic feature of this homodyne detection scheme. This is illustrated in figures 1(c) and (d), where we show a simulation of the homodyne current $J(t)$ together with the respective value $\langle\sigma_x\rangle_{\psi(t)}$ for a single quantum trajectory.

Second, equation (3) shows that the infinitesimal time evolution and thus the quantum trajectory is fully determined by the record of the measured homodyne current $\boldsymbol{J}_t$ and the values of the applied drive $\boldsymbol{\Omega}_t$, which are vectors containing the respective values of $J(t)$ and $\Omega(t)$ from the starting time $t_0 = 0$ until time t. In appendix A.4, we derive a closed-form expression for the operator $D_t = D_t[\boldsymbol{J}_t, \boldsymbol{\Omega}_t]$, which gives the mapping:

$$\rho_t = \frac{D_t\rho_0 D_t^\dagger}{\mathrm{tr}[D_t\rho_0 D_t^\dagger]}, \tag{7}$$

between the states of the qubit at times $t_0 = 0$ and t. The operator $D_t[\boldsymbol{J}_t, \boldsymbol{\Omega}_t]$ can be interpreted as a filter determining the state of the qubit at time t from the values of the measured homodyne current and the applied drive.

Finally, equation (3) does not preserve the norm of the state $|\psi\rangle$, as indicated by the tilde. This is not a problem if one integrates equation (3) numerically, since one can renormalize the state after each time step. For analytical calculations, it is useful to add some correction terms (see appendix A.5 or [2, 19]) such that the norm of the state is preserved up to second order in $\mathrm{d}t$:

$$\mathrm{d}|\psi\rangle = K_{\psi(t)}\mathrm{d}t + M_{\psi(t)}\mathrm{d}W(t). \tag{8}$$

Here, the nonlinear drift and diffusion terms are

$$K_{\psi(t)} = -\mathrm{i}H|\psi\rangle + \frac{\kappa}{2}\left\{\langle\sigma_x\rangle_{\psi(t)}\sigma_- - \sigma_+\sigma_- - \frac{1}{4}\langle\sigma_x\rangle^2_{\psi(t)}\right\}|\psi\rangle\,, \tag{9}$$

$$M_{\psi(t)} = \sqrt{\kappa}\left\{\sigma_- - \frac{1}{2}\langle\sigma_x\rangle_{\psi(t)}\right\}|\psi\rangle\,. \tag{10}$$

Note that equation (8) is a stochastic Schrödinger equation in the Itô form with multiplicative scalar noise.

2.2. Feedback control overview

In our control protocol, the results of the homodyne detection measurement determine the drive $\Omega(t)$ to be applied to the qubit. We consider two different schemes which are outlined in figure 1(b).

In the first scheme, discussed in section 3, we filter the homodyne signal to extract the system's exact state $|\psi(t)\rangle$ at time t. Subsequently, the controller receives this state as an input and determines the drive $\Omega(t)$ to be applied next. Equations (7) and (A.33) give an explicit filtering procedure $D_t[\boldsymbol{J}_t, \boldsymbol{\Omega}_t]$ that determines the state of the qubit at time t from the records of homodyne measurements $\boldsymbol{J}_t$ and the drive $\boldsymbol{\Omega}_t$, which are known in the experiment. Since we train the agent on simulated trajectories, we know the system's state at each time step. Therefore, we can skip the filter in figure 1(b) and directly feed back the solution of the SDE solver at each time step to our controller. In other words, we assume perfect filtering at all times. Thus, this situation corresponds to the feedback used in [10] with the exception that deterministic evolution is replaced by an SDE. We will use this control scheme in section 3 to test different backpropagation methods and to compare different control strategies.

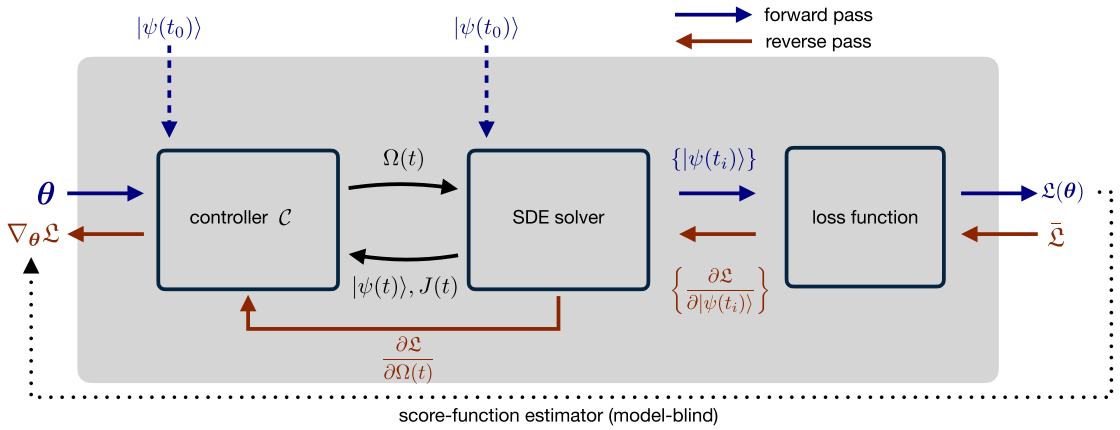

Figure 2. Workflow of the learning scheme (discussed in section 2.3) used to train the controller. In the forward pass, a controller, which is, in this work, implemented by a NN, maps the present quantum state $|\psi(t)\rangle$ (see section 3) or a measurement of the homodyne current $J(t)$ (see section 4) to a drive $\Omega(t)$. Then, an SDE is solved to determine the subsequent state and homodyne detection current $J(t)$. A loss function $\mathfrak{L}$ modeling the state preparation objective and possible constraints is evaluated based on a quantum trajectory, i.e., a sequence of states. In the backward pass, the gradient of the loss function with respect to the parameters $\boldsymbol{\theta}$ of the controller is evaluated by (adjoint) sensitivity methods (see section 2.4). This step incorporates physical knowledge of the system into the training process and is numerically more efficient than a model-blind gradient estimation. The gradient of the loss function with respect to the parameters of the controller is used to update the control strategy over a series of training epochs.

In the second scheme, discussed in section 4, the controller obtains the homodyne current record $\boldsymbol{J}_\tau(t)$ at time t measured over some time interval $[t-\tau, t]$. The NN forming the controller now has to simultaneously learn how to filter the signal from the noise and to predict the next action $\Omega(t)$. Such an implementation of the control protocol based only on $J(t)$ is a challenging task because the signal of the system quadrature $\langle\sigma_x\rangle_{\psi(t)}$ is hidden in the noise, as discussed in section 2.1.

2.3. Workflow

The learning scheme that controls the stochastic dynamics of the continuously monitored qubit based on ∂P consists of three building blocks, as outlined in figure 2: a parameterized controller $\mathcal{C}$, based on a NN, a model of the dynamics (expressed as an SDE), and a loss function.

At the beginning of each run, we initialize the system in an arbitrary state on the Bloch sphere,

$$|\psi(t_0)\rangle = \cos\left(\frac{\vartheta}{2}\right)|e\rangle + \sin\left(\frac{\vartheta}{2}\right)\mathrm{e}^{i\phi}|g\rangle\,, \tag{11}$$

where $|e\rangle$ and $|g\rangle$ are the excited and ground states of the qubit in the z-basis, respectively. To ensure that the controller performs optimally for any initial state, we sample the angles ϑ and ϕ uniformly from their intervals $[0, \pi]$ and $[0, 2\pi]$, respectively.

Depending on the chosen control scheme, see figure 1(b), the controller's input is either the quantum state $|\psi(t)\rangle$ (section 3) or the last homodyne detection record in the form of a vector $\boldsymbol{J}_\tau(t)$ gathered over some time interval $[t-\tau, t]$ (section 4). Moreover, in section 4, the controller also receives a vector $\boldsymbol{\Omega}_m(t)$ of the m last control actions applied prior to the time t.

The controller then maps this input to the next value of the drive, $\Omega(t)$. Given $|\psi(t)\rangle$ and $\Omega(t)$, we use the Runge–Kutta Milstein solver from the StochasticDiffEq.jl package [40–42] to calculate the next state $|\psi(t+\mathrm{d}t)\rangle$ according to the SDE (8). This loop between the control agent and the SDE solver is iterated for all time steps. We store the quantum states $|\psi(t_i)\rangle$ and the drive values $\Omega(t_i)$ at N uniformly spaced time steps $\{t_i\}_{i=1}^N$ so as to evaluate the loss function. Hereafter, the set $\{|\psi(t_i)\rangle, \Omega(t_i)\}$ is called the checkpoints.

We minimize a loss function of the form [10, 12, 13, 43]:

$$\mathfrak{L} = \sum_\mu c_\mu \mathfrak{L}_\mu, \tag{12}$$

where the terms $\mathfrak{L}_\mu$ encode case-specific objectives of the optimization process, see, e.g., [12]. Their relative importance can be controlled by the weights c_μ. We choose the weights c_μ empirically but, if necessary, they could also be tuned by means of hyperparameter optimization techniques [10]. To enforce high fidelity with respect to the target state over the whole control interval, we include in the loss function the average infidelity of the checkpoints $|\psi(t_i)\rangle$ with respect to the target state $|\psi_{\mathrm{tar}}\rangle$,

$$\mathfrak{L}_F = \frac{1}{N}\sum_{i=0}^{N}\left(1 - |\langle\psi(t_i)|\psi_{\text{tar}}\rangle|^2\right). \tag{13}$$

This form of the loss function also leads to a time-optimal performance of the controller. Focusing on specific time intervals, the last few steps, for example, the sum in equation (13) can be straightforwardly adjusted. We will consider the target state $|\psi_{\text{tar}}\rangle = |e\rangle$ in the following. In addition to $\mathfrak{L}_F$, we include the term

$$\mathfrak{L}_\Omega = \frac{1}{N}\sum_{i=0}^{N}|\Omega(t_i)|^2, \tag{14}$$

in the loss function to favor smaller amplitudes of the drive $\Omega(t_i)$. In section 4, we will find that this term is important to suppress the collapse of the NN toward a strategy where constant maximal pulses are applied during training.

At this stage, we have calculated a quantum trajectory and evaluated its value of the loss function. In the next step, we must update the control strategy to decrease the value of the loss function. The derivative of the loss function $\mathfrak{L}$ with respect to the parameters of the NN provides a meaningful update rule toward a better control strategy. Thus, we need to calculate the gradient $\nabla_{\boldsymbol{\theta}}\mathfrak{L}$. This can be computed efficiently using the sensitivity methods for SDEs discussed next.

2.4. Adjoint sensitivity methods

The loss function $\mathfrak{L} = \mathfrak{L}(\{|\psi(t_i)\rangle, \Omega(t_i)\})$, defined in equation (12), is a scalar function which explicitly depends on the checkpoints and implicitly on the parameters $\boldsymbol{\theta}$ of the controller. In contrast to score-function estimators [44–46], such as the REINFORCE algorithm [47], we will incorporate the physical model into the gradient computation, as outlined in figure 2.

AD is a powerful tool to evaluate the gradients of numeric functions at machine precision [48]. A key concept in AD is the computational graph [49], also known as the Wengert trace, which is a directed acyclic graph that represents the sequence of elementary operations that a computer program applies to its input values to calculate its output values. The nodes of the computational graph are the elementary computational steps of the program, known as *primitives*. The outcome of each primitive operation, called an *intermediate variable*, is evaluated in the *forward pass* through the graph.

In forward-mode AD, one associates the value of the derivative:

$$\dot{v}_{ji} = \frac{\partial v_j}{\partial \theta_i}, \tag{15}$$

with each intermediate variable v_j with respect to a parameter θ_i of interest. The derivatives $\dot{v}_{ji}$ are calculated together with the associated intermediate values v_j in the forward pass, i.e., the gradient is pushed forward through the graph. This procedure must be repeated for each parameter θ_i, therefore, forward-mode AD scales poorly, in terms of computation time, with an increasing number of parameters $\{\theta_i\}$.

In contrast, reverse-mode AD traverses the computation graph backward from the loss function to the parameters θ_i by defining an *adjoint process*:

$$\bar{v}_i = \frac{\partial \mathfrak{L}}{\partial v_i}, \tag{16}$$

which is the *sensitivity* of the loss function $\mathfrak{L}$ with respect to changes in the intermediate variable v_i. Reverse-mode AD is very efficient in terms of the number of input parameters because one needs just a single *backward pass* after the forward pass to obtain the gradient with respect to all parameters $\{\theta_i\}$. Thus, we always implement AD for the NN in the reverse mode. However, reverse-mode AD might be very expensive in terms of memory, because all the intermediate variables v_i from the forward pass need to be stored for the backward pass. Therefore, reverse-mode AD scales poorly in terms of memory if the number of steps and parameters of the SDE solver increases. Whether forward-mode or reverse-mode AD is more efficient to calculate gradients of the loss function depends on the details of the control loop shown in figure 2.

Algorithm 1: Piecewise constant

Input: $\psi(t_0)$, t_0, $\Omega(t_0) = 0$
Result: $\mathfrak{L}$
for $i = 0 : N-1$ **do**
  compute and store checkpoints:
  $\Omega(t_{i+1}) \leftarrow \mathrm{NN}_{\boldsymbol{\theta}}(\psi(t_i))$
  $\psi(t_{i+1}) \leftarrow \mathrm{solve}(\psi(t_i), \Omega(t_{i+1}))$ for SDE (8) in $[t_i, t_{i+1}]$
end
$\mathfrak{L} \leftarrow \mathrm{loss}(\{\psi(t_i), \Omega(t_i)\})$

Algorithm 2: Continuously updated

Input: $\psi(t_0)$, t_0, $\Omega(t_0) = 0$
Result: $\mathfrak{L}$
compute and store checkpoints:
$\{\psi(t_i), \Omega(t_i)\} \leftarrow \mathrm{solve}(\psi(t_0), \mathrm{NN}_{\boldsymbol{\theta}})$ for SDE (8) in $[t_0, t_N]$
$\mathfrak{L} \leftarrow \mathrm{loss}(\{\psi(t_i), \Omega(t_i)\})$

As we illustrate now, it is possible to combine different AD methods in different parts of the computational graph. In sections 3.3 and 4, we will use algorithm 1, where the drive $\Omega(t)$ is piecewise constant, i.e., a constant value of Ω is applied over N_{sub} successive time steps between two checkpoints. In this case, the memory consumption of the reverse-mode AD of the NN is moderate, since the number of parameters $\{\Omega(t_i)\}$ grows only with the number of checkpoints rather than with the number of time steps. In contrast, in the SDE solver, the evaluation of the time evolution between two checkpoints $\{|\psi(t_i)\rangle, \Omega(t_i)\}$ and $\{|\psi(t_{i+1}\rangle), \Omega(t_{i+1})\}$ only depends on the single parameter $\Omega(t_i)$, which makes forward-mode AD very efficient [50]. Therefore, we nest both methods and use an inner forward-mode AD for the SDE solver and an outer reverse-mode AD for the remaining parts, i.e., the NN and the computation of the loss function.

Restricting oneself to a piecewise-constant control drive, however, prevents the controller from reacting instantaneously to changes of the state. In order to implement a fast control loop, the controller has to be placed in the drift term of the SDE (8). Thus, the parameters entering the solver are the NN parameters $\boldsymbol{\theta}$, see algorithm 2. The *continuous adjoint sensitivity method* [51–53] circumvents the resulting memory issues by introducing a new primitive for the whole SDE in the backward pass of the code. This new primitive is determined by the solution of another SDE problem, the *adjoint SDE*. Formally, defining the *adjoint process* as

$$\boldsymbol{a}_{\psi}(t) = \nabla_{\psi(t)} \mathfrak{L}(\{|\psi(t_i)\rangle\}), \tag{17}$$

the adjoint SDE problem in the Itô sense satisfies the differential equation:

$$\mathrm{d}\boldsymbol{a}_{\psi}(t) = -\left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\psi(t)}\right)\left(K_{\psi(t)} - 2C^{\mathrm{IS}}_{\psi(t)}\right)\mathrm{d}t - \left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\psi(t)}\right) M_{\psi(t)} \mathrm{d}W(t), \tag{18}$$

with the initial condition:

$$\boldsymbol{a}_{\psi}(t_N) = \nabla_{\psi(t_N)} \mathfrak{L}(\{|\psi(t_i)\rangle\}), \tag{19}$$

where the standard conversion factor $C^{\mathrm{IS}}_{\psi(t)}$ in equation (18) accounts for the required transformation from the Itô to the Stratonovich sense, and vice versa [54]. The gradients of the loss function $\boldsymbol{a}_{\boldsymbol{\theta}}(t_0) = \nabla_{\boldsymbol{\theta}} \mathfrak{L}$ with respect to $\boldsymbol{\theta}$ are then determined by the integration of

$$\mathrm{d}\boldsymbol{a}_{\boldsymbol{\theta}}(t) = -\left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\boldsymbol{\theta}}\right)\left(K_{\psi(t)} - 2C^{\mathrm{IS}}_{\psi(t)}\right)\mathrm{d}t - \left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\boldsymbol{\theta}}\right) M_{\psi(t)} \mathrm{d}W(t), \tag{20}$$

with the initial condition $\boldsymbol{a}_{\boldsymbol{\theta}}(t_N) = \mathbf{0}_{\mathrm{Dim}[\theta]}$. This continuous adjoint sensitivity method will be used in section 3.2. Further details regarding the adjoint method and our Julia implementation [55, 56] within the SciML ecosystem [11, 16, 40] are discussed in appendix C.2.

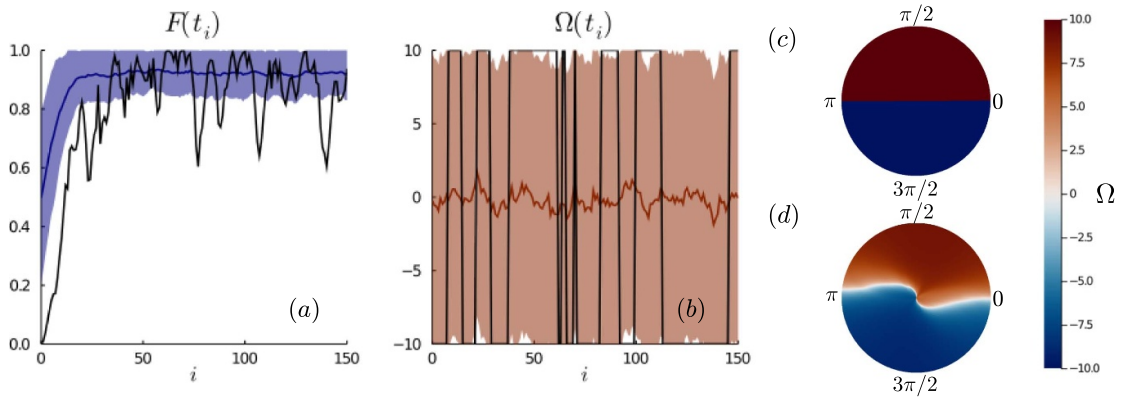

Figure 3. Preparation of the state $|e\rangle$ for the continuously monitored qubit described by the SDE (8). (a) Mean fidelity (blue solid line) and corresponding standard deviation (shaded) for the hand-crafted strategy summarized in algorithm 3 as a function of the time steps i at which the checkpoints are stored. (b) Corresponding drive amplitudes. The black lines in (a), (b) visualize the time evolution of the initial state $|g\rangle$. Stereographic projection, equation (21), of the applied drive Ω with respect to states $|\psi\rangle$ of the southern hemisphere of the Bloch sphere for (c) the hand-crafted strategy and (d) the optimized NN described in section 3.2. Note that panel (d) is the stereographic projection of the Bloch sphere of the inset of figure 4(a).

Algorithm 3: Hand-crafted controller

Input: $\psi(t)$
Result: $\Omega(t)$
if $\langle\sigma_y\rangle_{\psi(t)} > 0$ **then**
  $\Omega(t) \leftarrow \Omega_{\max}$
else
  $\Omega(t) \leftarrow -\Omega_{\max}$
end

3. SDE control based on full knowledge of the state

We first investigate the scenario in which the controller maps the quantum state $|\psi(t)\rangle$ to a new control parameter, $\mathcal{C} : |\psi(t)\rangle \mapsto \Omega(t) \in [-\Omega_{\max}, \Omega_{\max}]$. From a learning perspective, this is a major simplification, because the NN does not need to learn how to filter the homodyne current $J(t)$ to determine the state $|\psi(t)\rangle$. From the practical perspective of controller design, this approach assumes that there is already a filter module, which allows one to predict the state of the system from the measurement record and past control actions. In section 2.1 and appendix A.4, we discuss the implementation of such a filter in our case of a qubit subjected to homodyne detection. Note that if the initial state of the qubit is unknown, only a mixed state ρ_t can be obtained if the detection record is too short. Nevertheless, it is always possible to obtain a pure state estimate $\rho_t \mapsto |\psi(t)\rangle$ and recover our scenario, e.g., by projecting ρ_t onto the surface of the Bloch sphere. Alternatively, one can straightforwardly generalize our approach to the case of a stochastic quantum master equation describing the evolution of the system's density matrix in the presence of homodyne detection, see appendix A.5.

For all numerical experiments, we fix the parameters of the physical model at $\Delta = 20\kappa$ and $\Omega_{\max} = 10\kappa$. In appendix D, we discuss the variation of these parameters and their impact on the fidelity obtained.

3.1. Hand-crafted strategy

The control operator σ_x in equation (1) induces a rotation about the x-axis. Therefore, a simple but very intuitive strategy to move the state upward at each time step is to compute the expectation value $\langle\sigma_y\rangle_{\psi(t)}$ and to choose the direction of rotation depending on its sign. Specifically, if $\langle\sigma_y\rangle > 0$ (or $\langle\sigma_y\rangle < 0$), the controller should rotate (counter-) clockwise about the x-axis, as summarized in algorithm 3.

Figures 3(a) and (b) show the fidelity and the associated drive Ω for this control scheme. Although conceptually straightforward, this hand-crafted control function is very efficient. The mean fidelity over the whole control interval is $F_{\text{hc}} = 0.90 \pm 0.13$. We visualize the control strategy in figure 3(c), where we show the applied drive Ω as a function of the state on the Bloch sphere. The Bloch sphere [with spherical coordinates ϑ, ϕ, see also equation (11)] is mapped onto the tangential plane at $z = 0$, described by polar coordinates (R, Φ), using a stereographic projection:

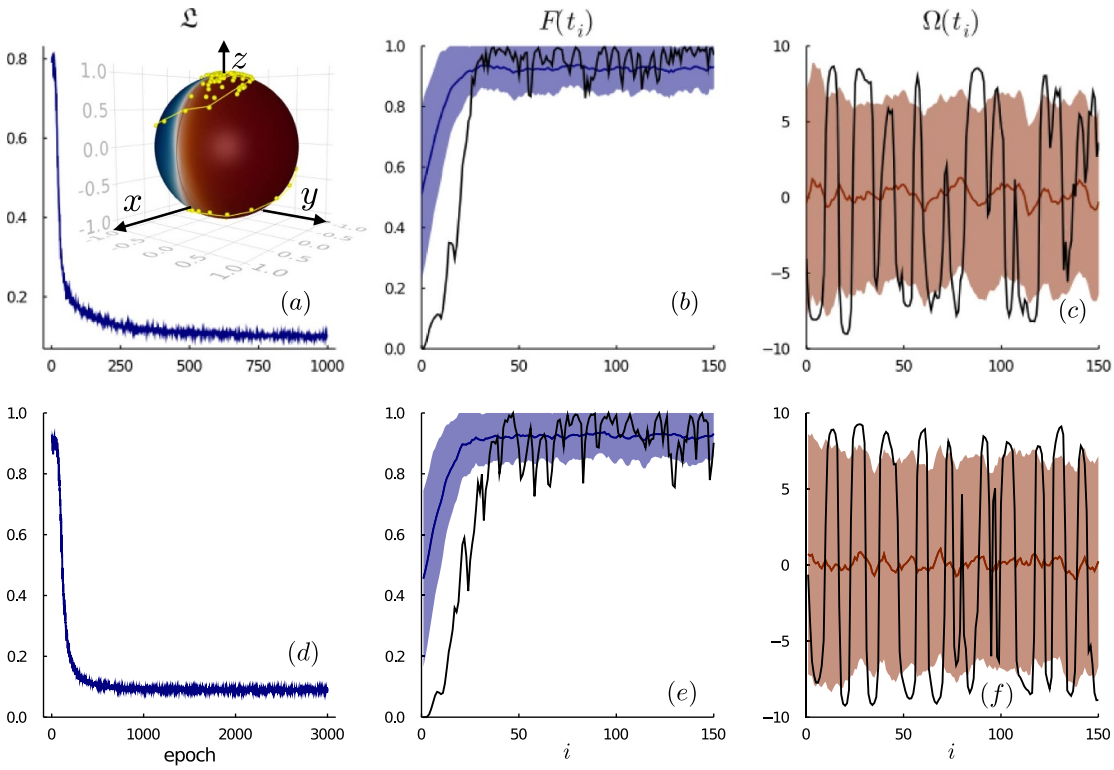

Figure 4. Preparation of the state $|e\rangle$ for the continuously monitored qubit described by the SDE (8), based on (a)–(c) continuously updated control parameters and (d)–(f) piecewise-constant control parameters. We compute the gradients of the loss function with respect to the parameters of the NN using the continuous adjoint sensitivity method in (a)–(c) and the nested discrete AD method in (d)–(f), see section 2.4. (a), (d) Smooth evolution of the loss as a function of the number of training epochs. (b)–(f) Performance of the trained NN when it is applied to a set of 256 randomly sampled initial states on the Bloch sphere. The solid blue [red] line in (b), (e) [(c), (f)] shows the mean fidelity $F(t_i)$ [control drive $\Omega(t_i)$] as a function of the time steps t_i at which the checkpoints are stored. The shaded regions represent the corresponding standard deviation which is large in panels (c) and (f) because the controller chooses a different sequence of controls for each quantum state. The black lines in (b)–(f) visualize the results for an initial state $|g\rangle$. Inset in panel (a): visualization of the control parameters Ω (in color code) as a function of the current state $|\psi(t)\rangle$ on the Bloch sphere. The color palette indicates rotation clockwise (blue) or anti-clockwise (red) about the x-axis. The yellow points indicate the trajectory starting at the initial state $|g\rangle$; the first 50 points are connected by a line. The hyperparameters are listed in table B1.

$$(R, \Phi) = \left(\cot\frac{\vartheta}{2}, \phi\right). \tag{21}$$

Evidently, the stereographic projection maps the south pole to $R = 0$ and the north pole to $R = \infty$. Throughout this paper, we truncate the value of R to an interval $[0, 1]$, so we plot the applied drive values for the states of the southern hemisphere, see figures 3(c) and (d).

3.2. Continuously updated control drive

We now apply a NN as the controller. We use a controller which changes $\Omega(t)$ at every time step of the solver based on the current state $|\psi(t)\rangle$. This implements a high-frequency feedback loop but renders discrete AD methods very inefficient, since all parameters of the NN contribute to the feedback loop. Consequently, we use the continuous adjoint sensitivity method described in section 2.4 to calculate the gradients of the loss function.

Figure 4(a) shows the smooth evolution of the loss function throughout the training of the (fully connected) NN, which converges to a configuration that can quickly reach fidelities with a mean value of about 0.9 and a modest standard deviation, see figure 4(b). The mean fidelity over the whole control interval is $F_c = 0.90 \pm 0.13$. Figure 4(c) illustrates the drive Ω applied during this time evolution. The inset of figure 4(a) visualizes the control strategy on the Bloch sphere. The same is depicted in figure 3(d) using stereographic projection, see equation (21). The controlled evolution of the initial state $|\psi(t_0)\rangle = |g\rangle$, corresponding to the black lines in figures 4(b) and (c), is marked by yellow points. These points show how the controller first transfers the state from the south pole to the north pole region and then stabilizes it in the vicinity of the target state $|e\rangle$.

Algorithm 4: Homodyne current

Input: $\psi(t_0)$, t_0, $\Omega(t_0)=0$
Result: $\mathfrak{L}$
for $i=0:N-1$ **do**
  compute and store checkpoints
  $\Omega(t_{i+1}) \leftarrow \mathrm{NN}_{\boldsymbol{\theta}}(\{\boldsymbol{J}_\tau(t), \boldsymbol{\Omega}_m(t)\})$
  $\psi(t_{i+1}) \leftarrow \mathrm{solve}(\psi(t_i), \Omega(t_{i+1}))$ for SDE (8) in $[t_i, t_{i+1}]$
end
$\mathfrak{L} \leftarrow \mathrm{loss}(\{\psi(t_i), \Omega(t_i)\})$

3.3. Piecewise-constant control drive

We now reduce the control frequency and assume that the controller changes the action $\Omega(t)$ only every N_{sub} time steps. This scenario is crucial in many physical situations, where the control loop is not fast enough to follow high-frequency changes in the physical system. In practice, this means that we evolve the state for N_{sub} substeps between checkpoints $\{|\psi(t_i)\rangle, \Omega(t_i)\}$ using a fixed value of $\Omega(t_i)$. The resulting piecewise-constant control scheme allows us to use the discrete forward-mode adjoint sensitivity method through the SDE solver combined with an outer reverse-mode AD for the rest of the control loop, as described in section 2.4. Although we restrict the rate at which $\Omega(t)$ can change, we find that the NN converges to a similar learning behavior with large fidelities $F(t)$ similar to those of section 3.2, see figure 4(e). The mean fidelity over the whole control interval is $F_{\mathrm{pw}} = 0.89 \pm 0.10$.

3.4. Comparison of the control strategies

Based on the results from a set of 256 trajectories, all three control strategies perform nearly equally well in terms of their average fidelities $F \approx 0.9$. The piecewise-constant controller slightly outperforms the other two approaches by having the smallest relative dispersion of the mean fidelity F_{pw}, which we attribute to the larger number of training epochs and the larger NN, see table B1.

The hand-crafted strategy achieves a large average fidelity but generates sudden jumps in the drive $\Omega(t)$, as shown in figure 3(b). Such a drive is hardly feasible from an experimental perspective. We find that NNs with moderate depths provide a smooth mapping between the input states and the drive, see figures 3(c) vs. 3(d), while maintaining high fidelity in the control interval. The signals generated by these protocols are experimentally more accessible, see figures 4(d) and (f). When necessary, specific terms can be added to $\mathfrak{L}$ in equation (12) to strengthen various requirements on the controller's performance (e.g., the smoothness of the drive or bounds on the power input) as discussed in section 2.2. This is not the case for hand-crafted strategies, where an efficient implementation of these constraints might be impossible. Furthermore, in some cases, e.g., the mountain car problem, it is not straightforward to come up with any hand-crafted strategy to start with. In contrast, the RL and ∂P approach is easily adjustable to different physical systems.

There are two principal reasons why the fidelity F only reaches 0.9 in our setup. First, F is averaged over time and the controller needs some time to align the qubit to a target state (starting from an arbitrary state). Second, the controlled qubit rotates about an axis in the x–z-plane whose direction depends on the ratio Δ/Ω. Hence, even in the case $\Omega \gg \Delta$ when the qubit rotates solely about the x-axis, the drive can only bring the qubit all the way up to the north pole of the Bloch sphere if the current state lies on a great circle perpendicular to this axis. Therefore, the ratio Δ/κ and the maximum value of Ω set a limit on how close the qubit can come to the target state on average. In appendix D, we study scenarios with lower κ and observe that the average fidelity can increase and approach unity. We can thus conclude that $F \approx 0.9$ is not a limit of our design but rather originates from the restricted capability of the control operations considered here.

4. SDE control based on homodyne current

In this section, we will construct a controller which directly obtains the (noisy) measurement record of the homodyne current and determines the optimal control field $\Omega(t_i)$ at each time interval $[t_i, t_{i+1}]$, see algorithm 4. We consider a controller formed by a slightly augmented NN architecture with fully connected layers (see figure B1). The acquisition of input data for the NN consists of the following steps:

(a) The controller generates a piecewise-constant drive $\Omega(t_i)$ between two checkpoints at times t_i and t_{i+1}, as described in section 3.3. In the time window $[t_i, t_{i+1}]$, we integrate the SDE (8) using N_{sub} substeps of length δt, as outlined in figure 5(a). We label these substeps with k. The input that the NN uses to predict the next $\Omega(t_{i+1})$ is the vector $\boldsymbol{J}_\tau(t_i) = [\delta J_{i1}, \ldots, \delta J_{iN_{\mathrm{sub}}}]^T$, where $\tau = N_{\mathrm{sub}}\delta t$ is the length of the time interval over which we gather the data. Experimentally, δt can be interpreted as the detection time

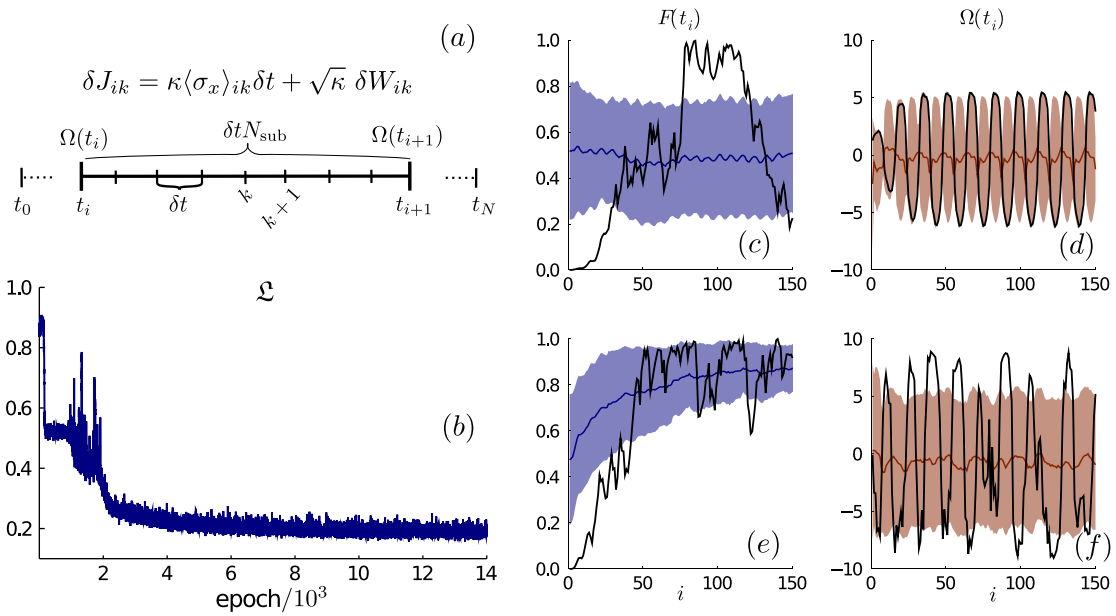

Figure 5. Preparation of the state $|e\rangle$ using the piecewise-constant control scheme with the discrete AD method to compute the gradients $\nabla_{\boldsymbol{\theta}}\mathfrak{L}$ (see section 2.4), where only the homodyne current $J(t)$ is used as an input to the controller. (a) An outline of the acquisition process of the input data for the NN controller, see section 4. (b) Evolution of the loss function $\mathfrak{L}$ as a function of the training epochs. (c), (d) Performance of the NN after 400 training epochs [in the first plateau of $\mathfrak{L}$ in panel (b)] for a set 256 trajectories: (c) mean fidelity (blue solid lines) and corresponding standard deviation (shaded area), (d) Corresponding drive amplitudes. (e), (f) The same quantities as in panels (c) and (d), but for the fully trained NN, i.e., after 14 000 training epochs. Black lines in panels (c)–(f) show the results for an initial state $|g\rangle$. The hyperparameters are listed in table B1 in appendix B.

window of the photodetectors. According to equations (4) and (5), the homodyne measurement at the kth substep is

$$\delta J_{ik} = \kappa\langle\sigma_x\rangle_{ik}\delta t + \sqrt{\kappa}\,\delta W_{ik}. \tag{22}$$

The first term corresponds to the quadrature signal $\langle\sigma_x\rangle_{ik}$ measured over the detection window δt, which we assume to be approximately constant on timescales of the order of δt. The second term, $\delta W_{ik} \equiv W_{i(k+1)} - W_{ik}$, is the Wiener increment during the kth substep. Note that the quantity δJ_{ik} is dimensionless and solely represents the number of detected photons, as discussed in appendix A.

(b) In addition, we provide the NN with information about the m last control parameters $\boldsymbol{\Omega}_m(t_i) = [\Omega(t_{i-1}), \ldots, \Omega(t_{i-m})]^T$. This equips the NN with a 'memory' of its actions, such that it can take into account how the sequence of preceding control drive amplitudes affected the performance. We empirically choose m such that the length of the input vector $\boldsymbol{\Omega}_m(t_i)$ corresponds to one tenth of the length of $\boldsymbol{J}_\tau(t_i)$.

Given these inputs, the function of the controller is to provide an optimal mapping $\mathcal{C} : \{\boldsymbol{J}_\tau(t), \boldsymbol{\Omega}_m(t)\} \mapsto \Omega(t) \in [-\Omega_{\max}, \Omega_{\max}]$. Figure 5(b) shows the evolution of the loss function $\mathfrak{L}$ during the learning phase as a function of the training epochs. Note that there are two distinct plateaus. First, after a couple of hundred epochs, the NN develops a general strategy consisting of the application of a periodic drive to prevent the qubit from decaying to the ground state. Figures 5(c) and (d) show examples of the performance of the NN at epoch 400 to illustrate this phase of the training. This strategy is state-independent, yet it manages to maintain the mean fidelity of the simulated trajectories at around 0.5 over the whole control interval. Around epoch 2500, after some transition phase where $\mathfrak{L}$ oscillates substantially, the NN starts to provide state-sensitive control fields. The final performance of the controller in figures 5(e) and (f) reaches the mean fidelity $F_J = 0.79 \pm 0.17$ over the whole control interval. During the last 50 time steps in the control interval, which we specifically focus on during training using an adjusted loss function term of the type shown in (13) for these steps, the average fidelity is $F_J^{50} = 0.86 \pm 0.12$.

At this stage, we infer that the NN learns how to extract the signal from noisy data before it develops a similar learning strategy to that seen in section 3. In contrast to the filter mentioned in section 2.1 and appendix A.4, this universal approach based on ∂P will also work if some parameters of the model are *a priori* unknown. For example, if the detuning Δ were unknown, one could train the controller on an ensemble of M randomly chosen parameters $\{\Delta_k\}_{k=1}^M$, such that it learned how to deal with the general

situation of arbitrary detuning. In this case, a straightforward filtering of the signals to obtain the state is infeasible, as this would require solving the filter for all possible values of the model's parameters; see equation (A.33). Other options for signal filtering would be (recursive) backward filtering methods [57, 58] or recurrent NNs, because their structure allows them to capture temporal correlations in the data [39]. Note that such filter methods are compatible with the two-step control approach described in section 3.

5. Discussion

In this work, we proposed a framework based on ∂P to automatically design feedback control protocols for stochastic quantum dynamics. As a test bed, we used a qubit subjected to homodyne detection, whose dynamics was given by a stochastic Schrödinger equation. Note, however, that our method can straightforwardly be applied to different physical systems and that it can be generalized to the case of stochastic quantum master equations.

In section 3, we demonstrated that a controller using an NN can be trained to prepare and stabilize a target state starting from an arbitrary initial state of the system when the NN obtains full knowledge of the instantaneous state at any time. The method generates a smooth drive $\Omega(t)$ while maintaining high fidelity with the target state over the whole control interval. Additional constraints on the performance of the controller can be implemented by adding further terms to the loss function. This makes the ∂P approach more versatile than tailoring control functions manually, which requires a unique approach for every new system and can even be infeasible for large quantum systems.

The key feature of our ∂P framework is the inclusion of an explicit model of the system's dynamics in the computation of the gradients of the loss function with respect to the parameters of the NN, i.e., the controller. Specifically, in section 3.2, we employed the recently developed continuous adjoint sensitivity method for gradient estimation through an SDE solver, which is memory efficient and, thus, allows us to study a high-frequency controller. One of the authors implemented these new continuous adjoint sensitivity methods in the DiffEqSensitivity.jl package within the open-source SciML ecosystem [55].

In section 4, we showed that the feedback control scheme can be based on the direct provision of a record of the homodyne current measurements to the NN without the need to filter the information using the actual state beforehand. Therefore, the NN must first learn how to filter the input data (with a poor signal-to-noise ratio) before it can predict the optimal state-dependent values of the control drive. Ultimately, the trained NN was able to reach fidelities above 85% in a target time interval for random initial states.

In future studies, the optimization of the loss function based on stochastic trajectories using adjoint sensitivity methods could be compared to alternative approaches. First, the solution to the stochastic optimal control problem in the specific case of Markovian feedback (as in section 3) is a Hamilton–Jacobi–Bellman equation [11, 54]. The solution of this partial differential equation, with the same dimensions as the original SDE, may directly give the optimal drive [59]. However, solving this partial differential equation with a mesh-based technique is computationally demanding, and mesh-free methods, e.g., those based on NNs, also require a (potentially costly) training procedure [60]. Second, the expected values of the loss functions could be optimized by leveraging the Koopman expectation for direct computation of the expected values from stochastic and uncertain models [61]. Additionally, one could approach this control problem by using an SDE moment expansion to generate ordinary differential equations for the moments and apply a closure relationship [62]. Additional research is required to ascertain the efficiency of these approaches in comparison to our method.

The results reported in this paper imply that ∂P is a powerful tool for the automated design of quantum control protocols. Further experimental needs, e.g., finite time lag between the measurement and the applied drive, finite-temperature effects, or imperfect homodyne detection, can be straightforwardly incorporated into this method. Thus, our work introduces a new perspective on how prior physical knowledge can be encoded into machine learning tools to construct a universal control framework. Besides the control application demonstrated here, the ∂P paradigm can also be adopted to solve other inverse problems, such as estimating model parameters from experimental data [63–65]. An interesting perspective for future work is to extend our framework to control-assisted quantum sensing and metrology.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/frankschae/Control-of-Stochastic-Quantum-Dynamics-with-Differentiable-Programming [66].

Acknowledgments

We would like to thank Niels Lörch, Eliska Greplova, Moritz Schauer, and Chris Rackauckas for helpful discussions. We acknowledge financial support from the Swiss National Science Foundation (SNSF) and the NCCR Quantum Science and Technology. Parts of the computations were performed at the sciCORE (scicore.unibas.ch) scientific computing core facility at the University of Basel.

Appendix A. Continuous homodyne detection

A.1. Quantum trajectories: monitoring the spontaneous emission of a qubit

Consider a two-level atom interacting with a free photon. In the case of discrete photon modes, the interaction is given by the usual Hamiltonian $\bar{H}_{\rm int} = {\rm i}g(\hat{\sigma}_+\hat{a} - \hat{\sigma}_-\hat{a}^\dagger)$, where $\hat{a}^\dagger$ and $\hat{a}$ are the bosonic creation and annihilation operators, respectively, fulfilling the commutation relation $[\hat{a}, \hat{a}^\dagger] = 1$, and where g is a coupling constant with the energy dimension. In contrast to the notation used in the main text, in this appendix, we will mark operators by a hat to distinguish them from scalars. This Hamiltonian results from the dipole interaction written in the form $\bar{H}_{\rm int} \propto (\hat{a} + \hat{a}^\dagger)\hat{\sigma}_y$ and application of the rotating wave approximation. However, when addressing a decay to the continuum, the photons are represented by quantum field operators with the commutation relation $[\hat{a}_t, \hat{a}_s^\dagger] = \delta(t-s)$, which thus have units of the square root of energy (or, equivalently, the inverse square root of time). The interaction Hamiltonian takes the form

$$\hat{H}_{\rm int} = {\rm i}\sqrt{\kappa}(\hat{\sigma}_+\hat{a}_t - \hat{\sigma}_-\hat{a}_t^\dagger), \tag{A.1}$$

where κ is the decay rate, again, with a energy dimension such as g.

The field propagator in the interaction picture can be formally written as $\hat{U}_\tau = \mathcal{T}\left[{\rm e}^{-{\rm i}\int_t^{t+\tau}\hat{H}_{\rm int}(s){\rm d}s}\right]$, where $\mathcal{T}$ is the time-ordering operator. Expanding $\hat{U}_{\delta t}$ for short times δt, we find

$$\begin{aligned}\hat{U}_{\delta t} =& 1 + \sqrt{\kappa}\int_t^{t+\delta t}(\hat{\sigma}_+\hat{a}_s - \hat{\sigma}_-\hat{a}_s^\dagger){\rm d}s \\ &- \frac{\kappa}{2}\int_t^{t+\delta t}(\hat{\sigma}_-\hat{\sigma}_+\hat{a}_s^\dagger\hat{a}_{s'} + \hat{\sigma}_+\hat{\sigma}_-\hat{a}_s\hat{a}_{s'}^\dagger){\rm d}s{\rm d}s'.\end{aligned} \tag{A.2}$$

Here, the second-order terms provide an important $O(\delta t)$ contribution. Higher-order terms only contribute according to $O(\delta t^{3/2})$ and can be safely neglected, and we also assume that δt is short on the timescale of the internal qubit dynamics. Assuming that the free field is originally in the vacuum state $|0\rangle$, we can write

$$\hat{U}_{\delta t}|0\rangle = \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e|\right)|0\rangle + \sqrt{\kappa\,\delta t}\hat{\sigma}_-|1\rangle_t, \text{where } |1\rangle_t = \frac{1}{\sqrt{\delta t}}\int_t^{t+\delta t}\hat{a}_s^\dagger{\rm d}s|0\rangle \tag{A.3}$$

is a properly normalized state of the field, and $|e\rangle$ denotes the excited qubit state.

The simplest way to monitor the qubit state $|\psi\rangle$ is then to continuously measure the intensity of the outgoing field. This defines a Poissonian process, as the probability of detecting a photon in a time interval δt reads

$$\begin{aligned}p_1 &= {\rm tr}_{\rm qubit}\left\{\langle 1|\hat{U}_{\delta t}|0\rangle\langle 0|\otimes|\psi\rangle\langle\psi|\hat{U}_{\delta t}^\dagger|1\rangle\right\} \\ &= \kappa\delta t\,{\rm tr}\{\hat{\sigma}_-|\psi\rangle\langle\psi|\hat{\sigma}_+\} = \kappa\delta t|\langle e|\psi\rangle|^2,\end{aligned} \tag{A.4}$$

and the probability of no detection is $p_0 = 1 - p_1$. Thus, we obtain the standard unraveling of the spontaneous decay process:

$$\begin{aligned}&\text{one photon emitted :} && \sqrt{\kappa\delta t}\hat{\sigma}_-|\psi\rangle, \\ &\text{no photons emitted :} && \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e|\right)|\psi\rangle.\end{aligned} \tag{A.5}$$

A.2. Weak homodyning

An alternative to photon-counting detection is to mix the outgoing light with coherent light from a local oscillator laser using a beamsplitter, and to measure the intensities at both output ports of the beamsplitter. To start with, we first consider the situation where the intensity of the local oscillator field is comparable to that of the emitted light. It is given by the expansion of a coherent state using only vacuum and single-photon states

$$\left|\sqrt{\delta t}\beta\right\rangle = \left(1 - \frac{|\beta|^2}{2}\delta t\right)|0\rangle + \beta\sqrt{\delta t}\,|1\rangle\,, \tag{A.6}$$

where the single-photon state of the $\hat{b}$-mode is defined identically to that of mode $\hat{a}$. Mixing the two states using a 50:50 beamsplitter gives

$$\begin{aligned}\hat{U}_{\mathrm{BS}}\hat{U}_{\delta t}|0\rangle\left|\sqrt{\delta t}\beta\right\rangle = &\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| - \frac{|\beta|^2\delta t}{2}\right)|0,0\rangle \\ &+ \sqrt{\frac{\delta t}{2}}\left(\beta + \sqrt{\kappa}\hat{\sigma}_-\right)|1,0\rangle \\ &+ \sqrt{\frac{\delta t}{2}}\left(\beta - \sqrt{\kappa}\hat{\sigma}_-\right)|0,1\rangle\,.\end{aligned} \tag{A.7}$$

Introducing the photocurrent variable $q = n - m$, which measures the intensity difference of the two outputs, we see that there are three possibilities, $q = -1, 0, 1$. The respective probabilities of the three outcomes are given by the norms of the three different terms in equation (A.7),

$$\begin{aligned}p_{q=\pm1} &= \frac{\delta t}{2}\langle\beta^2 \pm \beta\sqrt{\kappa}\hat{\sigma}_x + \kappa|e\rangle\langle e|\rangle_\psi\,, \\ p_{q=0} &= 1 - \delta t\beta^2 - \delta t\kappa\langle|e\rangle\langle e|\rangle_\psi\,.\end{aligned} \tag{A.8}$$

The three possible states of the system after the measurement are proportional to

$$\begin{aligned}q = \pm1: &\quad \left(\beta \pm \sqrt{\kappa}\hat{\sigma}_-\right)|\psi\rangle\,, \\ q = 0: &\quad \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| - \frac{|\beta|^2\delta t}{2}\right)|\psi\rangle\,.\end{aligned} \tag{A.9}$$

By setting $\beta = 0$ (i.e., no mixing with the local oscillator field), we recover the previous case where the qubit simply has a chance to decay:

$$q = \pm1: \qquad \sqrt{\kappa}\hat{\sigma}_-|\psi\rangle\,, \tag{A.10}$$

and $p_{q=1} + p_{q=-1} = p_1$. The only difference is that the emitted photon is randomly split between the two detectors, with no information about the state of the qubit contained in the sign of q.

A.3. Strong homodyning

Finally, we consider the situation treated in the main text. The standard homodyne measurement requires mixing the signal with a strong coherent local oscillator field $|\beta| \gg 1$:

$$|\beta\rangle = \sum_{n=0}^{\infty} c_n(\beta)\,|n\rangle\,,\ c_n(\beta) = \mathrm{e}^{-\beta^2/2}\frac{\beta^n}{\sqrt{n!}}. \tag{A.11}$$

Recall that, here, the Fock states $|n\rangle = \frac{\hat{b}^{\dagger n}}{\sqrt{n!}}|0\rangle$ are defined using the creation operator $\hat{b}^\dagger = \frac{1}{\sqrt{\delta t}}\int_t^{t+\delta t}\hat{b}_s^\dagger\,\mathrm{d}s$, which describes the field impinging on the lower port of the beamsplitter during the interval δt [see figure 1(a)]. For modes $\hat{a}$ and $\hat{b}$ to match, the local oscillator laser β has to be in resonance with the drive laser Ω because, in the lab frame, the field a_t picks up the phase of $\hat{\sigma}_-$ rotating at the frequency of the drive laser.

When the modes match, the 50:50 beamsplitter transformation takes the form

$$\hat{U}_{\mathrm{BS}}\begin{pmatrix}\hat{a}^\dagger \\ \hat{b}^\dagger\end{pmatrix} = \begin{pmatrix}\frac{\hat{a}^\dagger - \hat{b}^\dagger}{\sqrt{2}} \\ \frac{\hat{a}^\dagger + \hat{b}^\dagger}{\sqrt{2}}\end{pmatrix}. \tag{A.12}$$

In particular, this implies $\hat{U}_{\mathrm{BS}}|0,\beta\rangle = \left|\frac{\beta}{\sqrt{2}}, \frac{\beta}{\sqrt{2}}\right\rangle$, i.e., a coherent state $|\beta\rangle$ of mode b is split into two coherent states at half the intensity when mixed with the vacuum state $|0\rangle$ of mode a by a 50:50 beamsplitter. Using the

last two expressions, we can write down the overall state of the qubit, the spontaneously emitted radiation, and the homodyne laser field after the beamsplitter. To do so, let us compute

$$\begin{aligned}\tilde{\mathbb{U}}_{\delta t} &= (\hat{\mathbb{1}}_{\text{qubit}} \otimes \hat{U}_{\text{BS}})(\hat{U}_{\delta t} \otimes \hat{\mathbb{1}}_{\text{laser}})\,|0\rangle\,|\beta\rangle \\ &= (\hat{U}_{\text{BS}} \otimes \hat{\mathbb{1}}_{\text{qubit}})\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \sqrt{\kappa\delta t}\hat{a}^\dagger\hat{\sigma}_-\right)|0,\beta\rangle \\ &= \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \sqrt{\frac{\kappa\delta t}{2}}(\hat{a}^\dagger - \hat{b}^\dagger)\hat{\sigma}_-\right)\left|\frac{\beta}{\sqrt{2}}, \frac{\beta}{\sqrt{2}}\right\rangle, \end{aligned} \tag{A.13}$$

which is the operator mapping the state of the qubit at time t to the state of the qubit and the two detected modes at time $t+\delta t$. We can further simplify this expression by noting that

$$\begin{aligned}\hat{a}^\dagger\,|\beta\rangle &= \mathrm{e}^{-\beta^2/2}\sum_{n=0}\frac{\beta^n}{\sqrt{n!}}\hat{a}^\dagger\,|n\rangle = \mathrm{e}^{-\beta^2/2}\sum_{n=0}\frac{\beta^n}{\sqrt{n!}}\sqrt{n+1}\,|n+1\rangle \\ &= \mathrm{e}^{-\beta^2/2}\sum_{n=0}\frac{\beta^{n+1}}{\sqrt{(n+1)!}}\frac{n+1}{\beta} = \mathrm{e}^{-\beta^2/2}\sum_{n=0}\frac{\beta^n}{\sqrt{n!}}\frac{n}{\beta}\,|n\rangle \\ &= \frac{\hat{n}}{\beta}\,|\beta\rangle\,, \end{aligned} \tag{A.14}$$

where $\hat{n} = \hat{a}^\dagger\hat{a}$ is the photon number operator. Labeling the photon number operator for the $\hat{b}$ mode using $\hat{m} = \hat{b}^\dagger\hat{b}$, we can then rewrite equation (A.13) in a very intuitive form:

$$\tilde{\mathbb{U}}_{\delta t} = \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \sqrt{\kappa\delta t}\,\frac{\hat{n}-\hat{m}}{\beta}\hat{\sigma}_-\right)\left|\frac{\beta}{\sqrt{2}}, \frac{\beta}{\sqrt{2}}\right\rangle. \tag{A.15}$$

Again, $\tilde{\mathbb{U}}_{\delta t}\,|\psi_t\rangle$ directly gives us the state of the qubit and the detected modes, while

$$\langle n,m|\,\tilde{\mathbb{U}}_{\delta t}\,|\psi_t\rangle = \sqrt{p_{n,m}}\,|\psi_{t+\delta t|n,m}\rangle \tag{A.16}$$

gives the conditional state of the qubit together with the probability of detecting the n and m photons respectively. In the measurement process, the superpositions of states with different photon numbers collapse such that the state after the post-measurement is a classical statistical mixture of the possible outcomes:

$$\sum_{n,m=0}^{\infty} p_{n,m}|\psi_{t+\delta t|n,m}\rangle\langle\psi_{t+\delta t|n,m}| \otimes |n,m\rangle\langle n,m|. \tag{A.17}$$

One easily sees from equation (A.15) that $|\psi_{t+\delta t|n,m}\rangle = |\psi_{t+\delta t|n-m}\rangle$, i.e., only the difference $(n-m)$ between the photon counts reveals information about the state of the qubit, while the sum $(n+m)$ only describes the shot noise of the local oscillator. It is therefore sufficient to keep the difference $q = n-m$ and discard the sum, which defines the state:

$$\sum_{q=-\infty}^{\infty} p_q|\psi_{t+\delta t|q}\rangle\langle\psi_{t+\delta t|q}| \otimes |q\rangle\langle q|, \qquad \text{where} \qquad p_q = \sum_{n=\max(0,-q)}^{\infty} p_{n,n+q}. \tag{A.18}$$

In a slight abuse of notation, we can formally introduce the joint quantum state of the qubit and the different in the counts, q, as $\hat{\mathbb{U}}_{\delta t}\,|\psi_t\rangle$ with

$$\hat{\mathbb{U}}_{\delta t} = \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \sqrt{\kappa\delta t}\,\frac{\hat{q}}{\beta}\hat{\sigma}_-\right)\left|\tilde{\Phi}_\beta\right\rangle \quad \text{where} \qquad \left|\tilde{\Phi}_\beta\right\rangle = \sum_{q=-\infty}^{\infty}\sqrt{\mu_j(\beta)}\,|q\rangle\,, \tag{A.19}$$

$\hat{q}\,|q\rangle = q\,|q\rangle$, which gives rise to the same post-measurement state. Here,

$$\mu_q(\beta) = \sum_{n=\max(0,-q)}^{\infty} c_n^2(\beta/\sqrt{2})c_{n+q}^2(\beta/\sqrt{2}) = \mathrm{e}^{-\beta^2}I_q(\beta^2), \tag{A.20}$$

where $I_n(z)$ is the modified Bessel function of the first kind.

Next, we make use of the fact that the distribution of q on the right-hand side of equation (A.20) is closely approximated by the normal distribution $\mathcal{N}(0,\beta^2)$ in the limit $\beta \gg 1$. Therefore, we can replace the state $\left|\tilde{\Phi}_\beta\right\rangle$ of an integer-valued q in equation (A.19) with

$$|\Phi_\beta\rangle = \int_{-\infty}^{\infty} \left[\frac{1}{\sqrt{2\pi}\beta}\exp\left(-\frac{q^2}{2\beta^2}\right)\right]^{1/2} |q\rangle \,\mathrm{d}q, \tag{A.21}$$

of a continuously valued and normally distributed q, and we also introduce a continuously valued operator $\hat{q}$. In the regime of interest $\beta \gg 1$, the actual value of β does not play a role. To get rid of it, recall that $\beta^2 = I\delta t$ is the intensity of the local oscillator laser in the time window δt, so it is more convenient to work with the laser power I, which is independent of the choice of the time window δt. Then, we can rescale the photon count difference to

$$j = \sqrt{\frac{\kappa}{I}}q, \tag{A.22}$$

to eliminate the laser intensity. Equation (A.19) now reads

$$\mathbb{U}_{\delta t} = \left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \hat{j}\hat{\sigma}_-\right)|\Phi\rangle\,, \tag{A.23}$$

where the initial state $|\Phi\rangle$ of the rescaled j can be obtained from equation (A.21) and satisfies

$$P_0(j) = |\langle j|\Phi\rangle|^2 = \frac{1}{\sqrt{2\pi\kappa\delta t}}\exp\left(-\frac{j^2}{2\kappa\delta t}\right). \tag{A.24}$$

This also allows us to define the homodyne current of the main text, $J = j/\delta t$, as the photon count difference per time. At this point, note that equation (A.23) already implies equation (3). Let us now show how the measured value of j is distributed. The marginal distribution of j after the measurement reads

$$\begin{aligned} P(j) &= \left|\left|\langle j|\,\hat{\mathbb{U}}_{\delta t}\,|\psi_t\rangle\right|\right|^2 = \left|\left|\langle j|\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + \hat{j}\hat{\sigma}_-\right)|\Phi\rangle\,|\psi_t\rangle\right|\right|^2 \\ &= P_0(j)\left|\left|\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + j\hat{\sigma}_-\right)|\psi_t\rangle\right|\right|^2 \\ &= P_0(j)\,\langle\psi_t|\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + j\hat{\sigma}_+\right)\left(1 - \frac{\kappa\delta t}{2}|e\rangle\langle e| + j\hat{\sigma}_-\right)|\psi_t\rangle \\ &= P_0(j)\left(1 + (j^2 - \kappa\delta t)\langle\hat{\sigma}_+\hat{\sigma}_-\rangle_\psi + j\langle\hat{\sigma}_x\rangle_\psi\right). \end{aligned} \tag{A.25}$$

From $P(j)$, one easily deduces the expected value and the variance of j:

$$\mathbb{E}(j) = \kappa\delta t\langle\hat{\sigma}_x\rangle_\psi \qquad \mathrm{Var}(j) = \kappa\delta t. \tag{A.26}$$

Clearly, it has the form of a Wiener process with drift:

$$j = \kappa\langle\hat{\sigma}_x\rangle_\psi\delta t + \sqrt{\kappa}\,\delta W, \tag{A.27}$$

where δW is the increment of a Wiener process for an interval δt (a normally distributed random variable with a zero mean and a variance δt). The last step is to include the Hamiltonian of the qubit $\hat{H} = \frac{1}{2}\left(\Delta\hat{\sigma}_z + \Omega(t)\hat{\sigma}_x\right)$ and set $\mathrm{d}t = \delta t$ such that $I^{-1} \ll \mathrm{d}t \ll \kappa^{-1}, \Delta^{-1}, \Omega^{-1}$ in order to get

$$\begin{aligned} \hat{\mathbb{U}}_{\mathrm{d}t} &= \left(1 - \left(\mathrm{i}\Delta\hat{\sigma}_z + \mathrm{i}\Omega(t)\hat{\sigma}_x + \kappa\,\hat{\sigma}_+\hat{\sigma}_-\right)\frac{\mathrm{d}t}{2} + \hat{j}\hat{\sigma}_-\right)|\Phi\rangle\,, \\ &\text{with} \qquad |\Phi\rangle = \int_{-\infty}^{\infty}\sqrt{P_0(j)}\,|j\rangle\,\mathrm{d}j. \end{aligned} \tag{A.28}$$

A.4. Formal solution of the stochastic dynamics as a filter

The expression for the infinitesimal time evolution $\hat{\mathbb{U}}_{\mathrm{d}t}$ we just derived can be composed for all time intervals to define the evolution over a long period $[0, t]$:

$$\hat{\mathbf{U}}_t = \hat{\mathbb{U}}_{\mathrm{d}t}(t - \mathrm{d}t) \ldots \hat{\mathbb{U}}_{\mathrm{d}t}(\mathrm{d}t)\hat{\mathbb{U}}_{\mathrm{d}t}(0), \tag{A.29}$$

where each time step introduces a new quantum system for the photon detection difference measured during the corresponding infinitesimal interval. The joint state of the qubit and all values of the observed homodyne current for $s \in [0, t]$ at the final time t reads

$$\hat{\mathbf{U}}_t |\psi_0\rangle = \mathcal{T}\left[\exp\left(\int_0^t \left(-\mathrm{i}\frac{\Delta}{2}\hat{\sigma}_z - \mathrm{i}\frac{\Omega(s)}{2}\hat{\sigma}_x - \frac{\kappa}{2}\hat{\sigma}_+\hat{\sigma}_- + J(s)\hat{\sigma}_-\right)\mathrm{d}s\right)\right] \bigotimes_s |\Phi_s\rangle\,|\psi_0\rangle\,. \tag{A.30}$$

Hence, for any fixed values of the measured current $J(s) = j_s/\mathrm{d}s$, we can find the (unnormalized) state of the qubit that is conditioned on these outcomes:

$$\begin{aligned} &\bigotimes_s \langle \mathrm{j}_s|\,\hat{\mathbf{U}}_t\,|\psi_0\rangle = c_{\mathbf{J}}\hat{D}_t\,|\psi_0\rangle\,, \quad \text{where} \\ &\hat{D}_t = \mathcal{T}\left[\exp\left(\int_0^t \left(-\mathrm{i}\frac{\Delta}{2}\hat{\sigma}_z - \mathrm{i}\frac{\Omega(s)}{2}\hat{\sigma}_x - \frac{\kappa}{2}\hat{\sigma}_+\hat{\sigma}_- + J(s)\hat{\sigma}_-\right)\mathrm{d}s\right)\right] \end{aligned} \tag{A.31}$$

and $c_{\mathbf{J}} = \prod_s \langle j_s|\Phi_s\rangle$ is a scalar that is independent of the input state $|\psi_0\rangle$. Thus, the state of the qubit at time t conditioned on the homodyne detection record reads

$$|\psi_t\rangle \propto \hat{D}_t\,|\psi_0\rangle\,, \tag{A.32}$$

and for mixed initial states, $\hat{\rho}_t \propto \hat{D}_t\hat{\rho}_0\hat{D}_t^\dagger$. The expression of the operator:

$$\hat{D}_t = \mathcal{T}\left[\exp\left(\frac{1}{2}\int_0^t \begin{pmatrix} -\mathrm{i}\Delta & -\mathrm{i}\Omega(s) + 2J(s) \\ -\mathrm{i}\Omega(s) & \mathrm{i}\Delta - \kappa \end{pmatrix}\mathrm{d}s\right)\right] \tag{A.33}$$

can be thought of as a filter relating the record of the drive fields $\boldsymbol{\Omega}_t$ and the values of the measured homodyne current $\boldsymbol{J}_t$ to the map between the states of the qubit at times 0 and t, represented here by a two-by-two complex matrix. In a real experiment, $\hat{D}_t$ can be computed by, e.g., discretizing the integral.

Notably, even if the initial state of the qubit is unknown, e.g., $\hat{\rho}_0 = \frac{1}{2}\hat{\mathbb{1}}$, after a certain characteristic time the state:

$$\hat{\rho}_t = \frac{\hat{D}_t\hat{\rho}_0\hat{D}_t^\dagger}{\mathrm{tr}\left[\hat{D}_t\hat{\rho}_0\hat{D}_t^\dagger\right]} \tag{A.34}$$

becomes pure. This is because the stochastic dynamics essentially decouples the state of the system at time t from its state in the remote past. On the other hand, the measured homodyne current $\boldsymbol{J}_t$ reveals information about the unknown initial state and we have just shown how to filter this information, as

$$\frac{P(\boldsymbol{J}_t|_{|\psi_0\rangle})}{P(\boldsymbol{J}_t|_{|\phi_0\rangle})} = \frac{\|\hat{D}_t\,|\psi_0\rangle\,\|^2}{\|\hat{D}_t\,|\phi_0\rangle\,\|^2}, \tag{A.35}$$

for any two initial states $|\psi_0\rangle$ and $|\phi_0\rangle$.

A.5. Derivation of the norm-preserving stochastic Schrödinger equation

We start with the stochastic Schrödinger equation (3) which does not preserve the norm of the state $|\psi(t)\rangle$. Using

$$\mathrm{d}\tilde{\rho} = |\mathrm{d}\tilde{\psi}\rangle\langle\tilde{\psi}| + |\tilde{\psi}\rangle\langle\mathrm{d}\tilde{\psi}| + |\mathrm{d}\tilde{\psi}\rangle\langle\mathrm{d}\tilde{\psi}|, \tag{A.36}$$

we can derive the corresponding stochastic quantum master equation:

$$\mathrm{d}\tilde{\rho} = -\mathrm{i}[\hat{H}, \tilde{\rho}]\mathrm{d}t + \kappa\mathcal{D}[\hat{\sigma}_-]\tilde{\rho}\mathrm{d}t + \sqrt{\kappa}J(t)\left(\hat{\sigma}_-\tilde{\rho} + \tilde{\rho}\hat{\sigma}_-^\dagger\right)\mathrm{d}t, \tag{A.37}$$

which does not preserve the norm of the density matrix. Note that the last term in equation (A.36) contains a contribution of the order $\mathrm{d}t$, since $\mathrm{d}W^2 = \mathrm{d}t$. The last term in equation (A.37) shows that the state $\tilde{\rho}$ depends

Table B1. Hyperparameters employed to train the neural networks in the case of continuously updated control parameters (section 3.2), piecewise-constant control parameters with knowledge of the state $|\psi\rangle$ (section 3.3), piecewise-constant control parameters with knowledge of the homodyne current (section 4), and the closed-system dynamics described by the Schrödinger equation (appendix C.1). For all simulations, the number of checkpoints is $N = 150$. The value of N_{sub} gives the number of solver substeps between checkpoints. The coefficient c_{F50} refers to the modification of the loss function (13) which is limited to the last 50 steps of the control interval. The specifications of the solvers are provided in the SciML documentation [40]. 'LLS n', 'LLA n' and 'LLC n' denote the nth linear-layer of the state-aware, action-aware, and combination-aware networks, respectively. The number of input and output channels of the layers are specified in brackets.

	$\|\psi(t)\rangle$ (SDE)		$J(t)$ (SDE)	$\|\psi(t)\rangle$ (ODE)
	Section 3.2	Section 3.3	Section 4	Appendix C.1
Solver				
Scheme	EulerHeun	RKMil	RKMil	Tsit5
N_{sub}	200	20	80	
dt	$10^{-4}\kappa^{-1}$	$10^{-3}\kappa^{-1}$	$2.5\times10^{-4}\kappa^{-1}$	adaptive
Loss function				
c_F	1	0.8	1.2	1
c_{F50}	0	$1.8N/50$	$0.8N/50$	0
c_Ω	0	10^{-3}	10^{-3}	0
Adam optimizer				
learning rate	0.0015	0.0001	0.0001	0.0015
batchsize b	64	64	64	256
epochs	1000	3000	14 000	400
NN				
LLS 1	(4, 256)	(4, 256)	(N_{sub}, 256)	(4, 256)
LLS 2	(256, 64)	(256, 128)	(256, 256)	(256, 64)
LLS 3	(64, 1)	(128,64)	(256, 128)	(64, 1)
LLS 4		(64, 1)		
LLA 1			(N_{sub}/10, 128)	
LLA 2			(128, 128)	
LLC 1			(256, 64)	
LLC 2			(64, 32)	
LLC 3			(32, 1)	

on the homodyne signal $J(t)$, but it does not preserve the norm of $\tilde{\rho}$ during the time evolution. This can be compensated for by adding a correction term:

$$-\sqrt{\kappa}\,(J(t)\mathrm{d}t - \mathrm{d}W)\left(\hat{\sigma}_-\tilde{\rho} + \tilde{\rho}\hat{\sigma}_-^\dagger\right) - \sqrt{\kappa}\langle\hat{\sigma}_- + \hat{\sigma}_+\rangle\tilde{\rho}\mathrm{d}W, \tag{A.38}$$

which cancels the last term in equation (A.37) without introducing any new norm-nonconserving terms. On the level of the stochastic Schrödinger equation, this term can be generated by adding a contribution,

$$\left[-\frac{\kappa}{2}\langle\hat{\sigma}_- + \hat{\sigma}_+\rangle\hat{\sigma}_-\mathrm{d}t - \frac{\kappa}{8}\langle\hat{\sigma}_- + \hat{\sigma}_+\rangle^2\mathrm{d}t - \frac{\sqrt{\kappa}}{2}\langle\hat{\sigma}_- + \hat{\sigma}_+\rangle\mathrm{d}W\right]|\psi\rangle\,, \tag{A.39}$$

which gives rise to equation (8) of the main text.

Appendix B. Neural network architectures and hyperparameters

The hyperparameters used in the control tasks are summarized in table B1. The architecture of the NN used in section 4 is shown in figure B1. In all NNs, we use ReLUs as activation functions for the hidden layers and the softsign activation function for the last layer. Modifications of this simple architecture, e.g., through the application of recurrent neural networks to capture temporal correlations in the homodyne current for the SDE control in section 4, might be essential for the control of complex many-body quantum systems.

In this work, we used NNs as universal function approximators. While this approach is very general, other controller choices, e.g., Chebychev polynomials or Fourier basis expansions, could boost the performance for low-dimensional inputs, as in the case of a qubit, because they can be optimized to the problem at hand. The use of sparse identification for the dynamic system method [67–69] or symbolic regression tools [70] could further allow one to replace the trained NNs by a symbolic description based on a pre-defined library of operators.

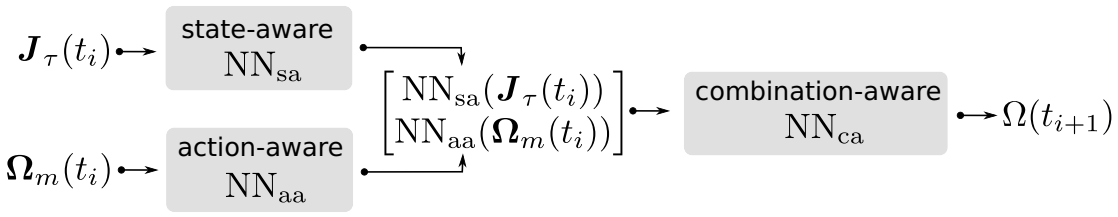

Figure B1. Scheme of the NN used in section 4, which consists of three segments of fully connected networks. The first one, the state-aware $\mathrm{NN}_{\mathrm{sa}}$, obtains the vector $\boldsymbol{J}_\tau(t_i)$ which bears information about the state of the controlled system in the form of its quadrature $\langle\sigma_x\rangle$ over the time interval $[t_i-\tau, t_i]$. Similarly, the action-aware $\mathrm{NN}_{\mathrm{aa}}$ takes as an input a vector $\boldsymbol{\Omega}_m(t_i)$ of the last m actions taken prior to the time t_i. The outputs of $\mathrm{NN}_{\mathrm{sa}}$ and $\mathrm{NN}_{\mathrm{aa}}$ are concatenated into a vector which enters the last segment, which we refer to as the combination-aware $\mathrm{NN}_{\mathrm{ca}}$. The final output is the next drive value $\Omega(t_{i+1})$ to be applied. The parameters of the network can be found in table B1.

Appendix C. Continuous adjoint sensitivity method for ODEs and SDEs

In this appendix, we discuss the continuous adjoint sensitivity method for ordinary differential equations (ODEs) and its generalization to the SDEs used in the main text. In appendix C.1, the continuous adjoint sensitivity method for ODEs is used to compare the stochastic control scenario discussed in section 3.2 with the associated unitary control scenario in the case of a closed system. In appendix C.2, we provide an intuitive understanding of the stochastic adjoint process and discuss technical details regarding the continuous adjoint sensitivity method for SDEs and its implementation [55]. In all implementations, we use an isomorphism to map the complex amplitudes of the quantum state to real numbers, as required by the AD backend [71].

C.1. Quantum control of a qubit in a closed system

In this section, we aim to control the closed-system dynamics of a qubit given by the Schrödinger equation:

$$\mathrm{d}|\psi(t)\rangle = -\mathrm{i}H^{\mathrm{CS}}|\psi(t)\rangle\mathrm{d}t =: K^{\mathrm{CS}}_{\psi(t)}\mathrm{d}t, \tag{C.1}$$

with the Hamiltonian:

$$H^{\mathrm{CS}} = \frac{\Delta}{2}\sigma_z + \frac{\Omega(t)}{2}\sigma_x, \tag{C.2}$$

where Δ is the qubit transition frequency [10]. As in section 3.2, we choose an NN with parameters $\boldsymbol{\theta}$ as the controller ansatz and we allow the NN to change the control drive $\Omega(t)$ at every time step, based on the state $|\psi(t)\rangle$. The initial states are uniformly distributed on the Bloch sphere, see equation (11). We compute the forward pass, equation (C.1), with the adaptive Tsitouras 5/4 embedded Runge–Kutta (Tsit5) scheme as implemented in the DifferentialEquations.jl package [40]. Given this solution, we use the loss function (12) with weights $c_F = 1, c_\Omega = 0$. As discussed in section 2.3, it depends on the checkpoints at times $\{t_i\}_{i=1}^N$, $\mathfrak{L} = \mathfrak{L}(\{|\psi(t_i)\rangle\})$.

As discussed in section 2.4, the continuous adjoint sensitivity method circumvents the memory issues of discrete reverse-mode AD and scales better with the number of parameters than forward-mode AD. To derive this adjoint method, one first adds a zero to the loss function, equation (12), and rewrites it as a time integral:

$$I(\boldsymbol{\theta}) = \int_{t_0}^{t_N}\left[\frac{1}{N}\sum_{i=0}^{N}\left(1-|\langle\psi(t_i)|\psi_{\mathrm{tar}}\rangle|^2\right)\delta(t-t_i) - \boldsymbol{\lambda}^\dagger(t)\left(|\dot{\psi}\rangle - K^{\mathrm{CS}}_{\psi(t)}(\boldsymbol{\theta})\right)\right]\mathrm{d}t, \tag{C.3}$$

where we inserted $\mathfrak{L}_F$, as defined in equation (13), and introduced the Lagrange multiplier $\boldsymbol{\lambda}$, such that $I(\boldsymbol{\theta}) = \mathfrak{L}(\boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} I(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}\mathfrak{L}(\boldsymbol{\theta})$. After an integration by parts and re-arrangement of terms for $\nabla_{\boldsymbol{\theta}} I(\boldsymbol{\theta})$, one finds that computing the gradients $\nabla_{\boldsymbol{\theta}}\mathfrak{L}$ requires an evaluation of the time evolution of the quantity $\boldsymbol{a}_\psi(t) = \boldsymbol{\lambda}^\dagger(t)$. This leads to the gradients $\nabla_{\psi(t)}\mathfrak{L}$ of the loss function with respect to the state $|\psi(t)\rangle$ for all times t and is called the adjoint process:

$$\boldsymbol{a}_\psi(t) = \nabla_{\psi(t)}\mathfrak{L}. \tag{C.4}$$

The associated adjoint ODE problem satisfies the differential equation [11, 51, 72–74]:

$$\mathrm{d}\boldsymbol{a}_\psi(t) = -\left(\boldsymbol{a}^\dagger_\psi(t)\cdot\nabla_{\psi(t)}\right)K^{\mathrm{CS}}_{\psi(t)}\mathrm{d}t,$$

$$\boldsymbol{a}_\psi(t_0) = \boldsymbol{a}_\psi(t_N) + \int_{t_N}^{t_0} \left[\frac{d\boldsymbol{a}_\psi(t)}{dt} - \sum_{i \neq N} \delta(t - t_i) \nabla_{\psi(t_i)} \mathfrak{L} \right] \mathrm{d}t, \tag{C.5}$$

with the initial condition:

$$\boldsymbol{a}_\psi(t_N) = \nabla_{\psi(t_N)} \mathfrak{L}(\{|\psi(t_i)\rangle\}). \tag{C.6}$$

This adjoint ODE is defined backwards in time from t_N to t_0. To compute the vector-Jacobian products in equation (C.5), one needs to know the value of the state $|\psi(t)\rangle$ along its entire trajectory, which was computed in the forward pass. Thus, we must store these states or recompute them by solving the ODE backwards in time starting from the final value $|\psi(t_N)\rangle$,

$$|\psi(t)\rangle = |\psi(t_N)\rangle + \int_{t_N}^{t} K^{\mathrm{CS}}_{\psi(t')} \mathrm{d}t', \tag{C.7}$$

together with the adjoint process. This does not introduce a significant memory overhead. Computing the gradients $\nabla_{\boldsymbol{\theta}} \mathfrak{L}$ requires yet another integral, which depends on the original and the adjoint process, $|\psi(t)\rangle$ and $\boldsymbol{a}_\psi(t)$, respectively. With the initial condition:

$$\boldsymbol{a}_{\boldsymbol{\theta}}(t_N) = \mathbf{0}_{\mathrm{Dim}[\theta]}, \tag{C.8}$$

the gradients $\nabla_{\boldsymbol{\theta}} \mathfrak{L} = \boldsymbol{a}_{\boldsymbol{\theta}}(t_0)$ are determined by

$$\begin{aligned} \mathrm{d}\boldsymbol{a}_{\boldsymbol{\theta}}(t) &= -\left(\boldsymbol{a}^\dagger_\psi(t) \cdot \nabla_{\boldsymbol{\theta}} \right) K^{\mathrm{CS}}_{\psi(t)} \mathrm{d}t, \\ \boldsymbol{a}_{\boldsymbol{\theta}}(t_0) &= \boldsymbol{a}_{\boldsymbol{\theta}}(t_N) + \int_{t_N}^{t_0} \frac{d\boldsymbol{a}_{\boldsymbol{\theta}}(t)}{dt} \mathrm{d}t. \end{aligned} \tag{C.9}$$

Therefore, the gradients of the loss function $\nabla_{\boldsymbol{\theta}} \mathfrak{L}$ with respect to the neural network parameters $\boldsymbol{\theta}$ can be obtained by a single (adjoint) ODE with an augmented state given by $|\psi(t)\rangle$, $\boldsymbol{a}_\psi(t)$, and $\boldsymbol{a}_{\boldsymbol{\theta}}(t)$.

Because of the reversion of the ODE, equation (C.7) may be numerically unstable. Therefore, we use an interpolating adjoint algorithm to increase stability [11]. When solving the augmented state backwards in time, we recompute the forward pass sequentially for all time intervals $[t_i, t_{i+1}]$ between two checkpoints. Then, fourth-order interpolations of the recomputed forward pass are used to compute the vector-Jacobian products for the reverse pass.

The results of the closed-system control task are shown in figure C1. We observe a very fast and smooth learning process with our physics-informed RL framework. The target state $|e\rangle$ is an eigenstate of the Hamiltonian (C.2) for $\Omega = 0$, therefore, the control drive Ω is switched off once the target state $|e\rangle$ is reached. Figures C1(d) and (e) show that the control strategy, illustrated on a Bloch sphere and using a stereographic projection, resembles the stochastic case discussed in the main text.

C.2. Technical details of the continuous adjoint sensitivity method for SDEs

To generalize the continuous adjoint sensitivity method for ODEs to Itô SDEs, one first needs to determine how the sample path of an SDE can be reversed, i.e., how one can reconstruct the forward pass of the state $|\psi(t)\rangle$ from time t_0 to t_N by a reversed time evolution from t_N to t_0 launched at $|\psi(t_N)\rangle$. The reversion of an SDE:

$$\mathrm{d}|\psi\rangle = \tilde{K}_{\psi(t)} \mathrm{d}t + M_{\psi(t)} \circ \mathrm{d}W(t), \tag{C.10}$$

defined in the Stratonovich sense, as indicated by the $\circ$ is given by [52, 53]

$$|\psi(t)\rangle = |\psi(t_N)\rangle + \int_{t_N}^{t} \tilde{K}_{\psi(t')} \mathrm{d}t' + \int_{t_N}^{t} M_{\psi(t')} \circ \mathrm{d}W(t'), \tag{C.11}$$

with noise values $W(t)$ identical to those sampled in the forward pass. This closely resembles the reversion in the case of an ODE shown in equation (C.7). To restore the noise, we use a dense noise grid of the noise values used in the forward pass. The memory overhead caused by using a noise grid could be traded against speed by using a virtual Brownian tree [52] or a Brownian interval [53], which would enable the reconstruction of $W(t)$, and only require the storage of very little information, such as the seed of the pseudo-random number generator used. Despite the allocation of the noise values, the continuous stochastic adjoint sensitivity method is still much more memory efficient than discrete reverse-mode AD backpropagation through the solver operations.

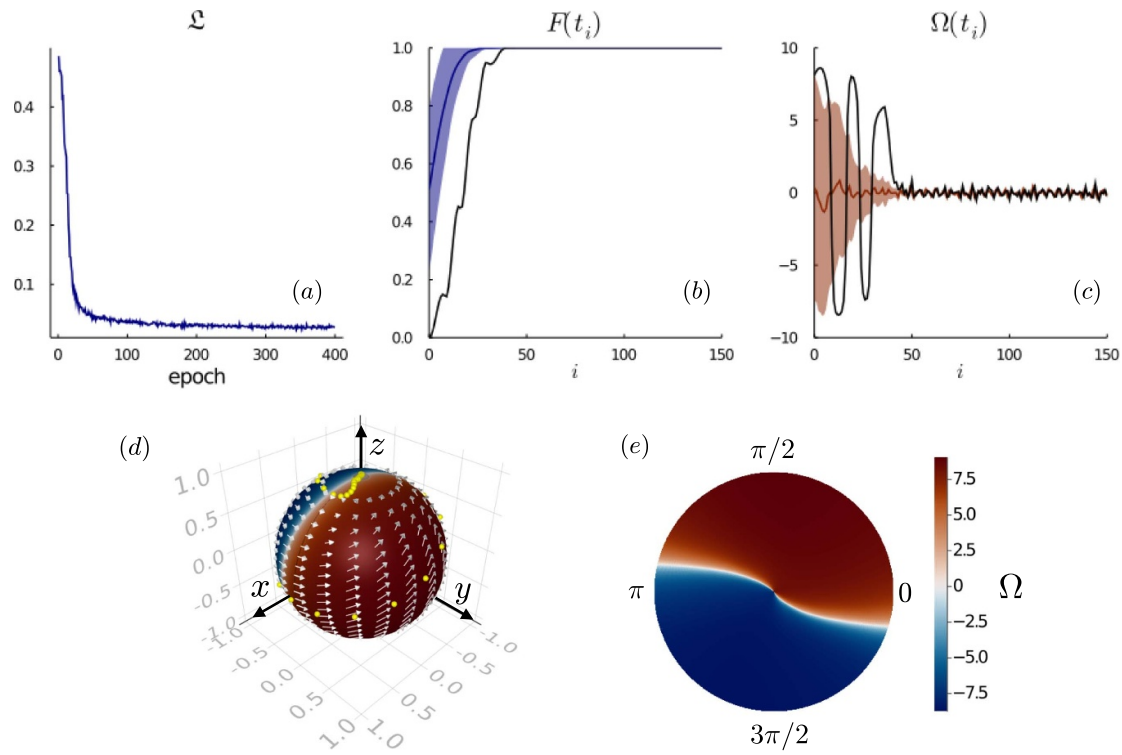

Figure C1. Preparation of the state $|e\rangle$ in a closed system described by the Hamiltonian, equation (C.2), based on continuously updated control parameters and the continuous (interpolating) adjoint sensitivity method. (a) Loss function, equation (12), as a function of training epochs. Means (blue/red) and standard deviations (shaded) for (b) fidelity and (c) control parameters, respectively, as a function of the steps i at which checkpoints are stored. The black line in (c), (d) as well as the yellow dots in (d) highlight the special case of the initial state $|g\rangle$. (d) Bloch sphere with color coding representing Ω and a vector field showing the infinitesimal evolution of the state $|\psi\rangle$ at that position. (e) Stereographic projection of the southern hemisphere of the Bloch sphere according to equation (21). The hyperparameters used can be found in table B1 in appendix B.

From the reverse Stratonovich SDE, equation (C.11), we can straightforwardly obtain the reverse Itô SDE:

$$|\psi(t)\rangle = |\psi(t_N)\rangle + \int_{t_N}^{t} \left(K_{\psi(t')} - 2C^{\mathrm{IS}}_{\psi(t)}\right) \mathrm{d}t' + \int_{t_N}^{t} M_{\psi(t')} \mathrm{d}W(t'), \tag{C.12}$$

of the monitored qubit, equation (8), where the standard conversion rule:

$$C^{\mathrm{IS}}_{\psi(t)} = \frac{1}{2} \left(M_{\psi(t)} \cdot \nabla_{\psi(t)}\right) M_{\psi(t)}, \tag{C.13}$$

in equation (C.12) accounts for the required transformation from Itô to the Stratonovich sense and vice versa [54].

Analogously to the ODE case, taking the scalar loss function $\mathfrak{L}$ [equation (12)], the adjoint process:

$$\boldsymbol{a}_{\psi}(t) = \nabla_{\psi(t)} \mathfrak{L}(\{|\psi(t_i)\rangle\}), \tag{C.14}$$

to compute the gradients of $\mathfrak{L}$ with respect to the state $|\psi(t)\rangle$ is now a strong solution [54] of the adjoint Itô SDE:

$$\mathrm{d}\boldsymbol{a}_{\psi}(t) = -\left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\psi(t)}\right) \left(K_{\psi(t)} - 2C^{\mathrm{IS}}_{\psi(t)}\right) \mathrm{d}t - \left(\boldsymbol{a}^{\dagger}_{\psi}(t) \cdot \nabla_{\psi(t)}\right) M_{\psi(t)} \mathrm{d}W(t), \tag{C.15}$$

with the initial condition:

$$\boldsymbol{a}_{\psi}(t_N) = \nabla_{\psi(t_N)} \mathfrak{L}(\{|\psi(t_i)\rangle\}). \tag{C.16}$$

Again, the value of the state $|\psi(t)\rangle$ of the forward pass is seen to be embedded in the vector-Jacobian products within the backward pass and, therefore, knowledge of the state $|\psi(t)\rangle$ along its trajectory is necessary. Using equation (C.12), we can recompute the state $|\psi(t)\rangle$ without having to store the full trajectory of the forward pass.

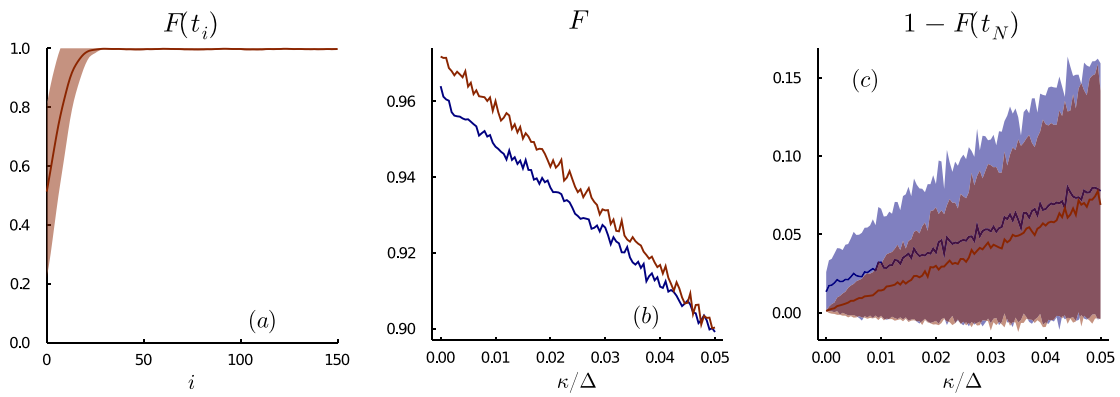

Figure D1. Comparison of the preparation of the state $|e\rangle$ for the continuously monitored qubit described by the SDE (8), based on continuously updated control parameters for different values of κ/Δ. (a) The solid red line shows the mean fidelity $F(t_i)$ as a function of the time steps t_i at which checkpoints are stored in the case of $\kappa/\Delta = 0.001$ and the neural network trained using the continuous adjoint sensitivity method with $\kappa/\Delta = 0.05$ of section 3.2 (red). The shaded region represents the corresponding standard deviation. (b) Average fidelity over all time steps i as a function of κ/Δ for the neural network from (a) (red) and the hand-crafted control strategy of section 3.1 (blue). (c) Average final-state infidelity (solid lines) and corresponding standard deviation (shaded regions) as a function of κ/Δ for the neural network from (a) (red) and the hand-crafted control strategy of section 3.1 (blue). The mean values and standard deviations are computed for a set of 512 randomly sampled initial states on the Bloch sphere. The hyperparameters are listed in table B1.

This SDE can be augmented by the additional quantity $\boldsymbol{a}_{\boldsymbol{\theta}}(t)$ to compute the gradients $\boldsymbol{a}_{\boldsymbol{\theta}}(t_0) = \nabla_{\boldsymbol{\theta}}\mathfrak{L}$. It is the solution of the SDE:

$$\mathrm{d}\boldsymbol{a}_{\boldsymbol{\theta}}(t) = -\left(\boldsymbol{a}_{\psi}^{\dagger}(t)\cdot\nabla_{\boldsymbol{\theta}}\right)\left(K_{\psi(t)} - 2C^{\mathrm{IS}}_{\psi(t)}\right)\mathrm{d}t - \left(\boldsymbol{a}_{\psi}^{\dagger}(t)\cdot\nabla_{\boldsymbol{\theta}}\right)M_{\psi(t)}\mathrm{d}W(t), \tag{C.17}$$

with the initial condition:

$$\boldsymbol{a}_{\boldsymbol{\theta}}(t_N) = \mathbf{0}_{\mathrm{Dim}[\theta]}. \tag{C.18}$$

As a consequence of the scalar noise character of the forward SDE [equation (8)], the adjoint SDE with an augmented state according to equations (C.12), (C.15), and (C.17) also has scalar noise. Similarly to the ODE setting, solving SDEs backwards is not guaranteed to be stable. Thus, to improve the stability, we modify this approach by resetting the reverse integration using the checkpoints $\{|\psi(t_i)\rangle\}$.

Appendix D. The effect of the decay rate κ on the fidelity F

In the main text, we reported mean fidelities of about $F \approx 0.9$ averaged over the whole control interval, including the very low fidelities due to the transient initial dynamics. This value of the fidelity is determined by the physical parameters of the quantum system (Δ, κ), as well as by the experimental limitations of the control scheme (e.g., Ω_{max} and the feedback control frequency). We note that these parameters are setup-specific and do not represent a hard fidelity limit for our proposed control scheme. In figure D1, we demonstrate that the mean fidelity over the full time interval as well as the final state infidelity can easily be improved by decreasing the decay rate κ with respect to the detuning Δ. In the limit $\kappa/\Delta \to 0$, one recovers a fidelity close to unity comparable to the closed-system case (see appendix C.1). We further see that the difference between the trained NN and the hand-crafted strategy becomes more pronounced as κ is decreased.

ORCID iDs

Frank Schäfer https://orcid.org/0000-0003-2684-4984
Pavel Sekatski https://orcid.org/0000-0001-8455-020X
Martin Koppenhöfer https://orcid.org/0000-0003-0162-3261
Michal Kloc https://orcid.org/0000-0002-4575-7723

References

[1] D'Alessandro D 2008 *Introduction to Quantum Control and Dynamics* (London: Chapman and Hall)
[2] Wiseman H M and Milburn G J 2009 *Quantum Meas. Control* (Cambridge: Cambridge University Press)

[3] Glaser S J et al 2015 Training Schrödinger's cat: quantum optimal control Eur. Phys. J. D 69 1–24
[4] Zhang J, xi Liu Y, Wu R-B, Jacobs K and Nori F 2017 Quantum feedback: theory, experiments and applications Phys. Rep. 679 1–60
[5] Sutton R S and Barto A G 2018 Reinforcement Learning: An Introduction 2nd edn (Cambridge, MA: MIT Press)
[6] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D and Wierstra D 2016 Continuous control with deep reinforcement learning Int. Conf. on Learning Representations (Poster)
[7] Niu M Y, Boixo S, Smelyanskiy V N and Neven H 2019 Universal quantum control through deep reinforcement learning npj Quantum Inf. 5 33
[8] Bukov M 2018 Reinforcement learning for autonomous preparation of Floquet-engineered states: inverting the quantum Kapitza oscillator Phys. Rev. B 98 224305
[9] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control Phys. Rev. X 8 031086
[10] Schäfer F, Kloc M, Bruder C and Lörch N 2020 A differentiable programming method for quantum control Machine Learn.: Sci. Technol. 1 035009
[11] Rackauckas C, Ma Y, Martensen J, Warner C, Zubov K, Supekar R, Skinner D and Ramadhan A 2020 Universal differential equations for scientific machine learning (arXiv:2001.04385)
[12] Leung N, Abdelhafez M, Koch J and Schuster D 2017 Speedup for quantum optimal control from automatic differentiation based on graphics processing units Phys. Rev. A 95 042318
[13] Abdelhafez M, Schuster D I and Koch J 2019 Gradient-based optimal control of open quantum systems using quantum trajectories and automatic differentiation Phys. Rev. A 99 052327
[14] Rackauckas C, Edelman A, Fischer K, Innes M, Saba E, Shah V B and Tebbutt W 2020 Generalized physics-informed learning through language-wide differentiable programming AAAI Symp.: MLPS
[15] Liao H-J, Liu J-G, Wang L and Xiang T 2019 Differentiable programming tensor networks Phys. Rev. X 9 031041
[16] Rackauckas C, Innes M, Ma Y, Bettencourt J, White L and Dixit V 2019 Diffeqflux.jl—a Julia library for neural differential equations (arXiv:1902.02376)
[17] Wu R-B, Ding H, Dong D and Wang X 2019 Learning robust and high-precision quantum controls Phys. Rev. A 99 042327
[18] Coopmans L, Luo D, Kells G, Clark B K and Carrasquilla J 2020 Protocol discovery for the quantum control of Majoranas by differential programming and natural evolution strategies (arXiv:2008.09128)
[19] Breuer H-P and Petruccione F 2002 The Theory of Open Quantum Systems (Oxford: Oxford University Press)
[20] Briant T, Cohadon P, Pinard M and Heidmann A 2003 Optical phase-space reconstruction of mirror motion at the attometer level Eur. Phys. J. D 22 131–40
[21] Iwasawa K, Makino K, Yonezawa H, Tsang M, Davidovic A, Huntington E and Furusawa A 2013 Quantum-limited mirror-motion estimation Phys. Rev. Lett. 111 163602
[22] Wieczorek W, Hofer S G, Hoelscher-Obermaier J, Riedinger R, Hammerer K and Aspelmeyer M 2015 Optimal state estimation for cavity optomechanical systems Phys. Rev. Lett. 114 223601
[23] Wiseman H M and Milburn G J 1993 Quantum theory of optical feedback via homodyne detection Phys. Rev. Lett. 70 548–51
[24] Mancini S, Vitali D and Tombesi P 1998 Optomechanical cooling of a macroscopic oscillator by homodyne feedback Phys. Rev. Lett. 80 688–91
[25] Hofmann H F, Mahler G and Hess O 1998 Quantum control of atomic systems by homodyne detection and feedback Phys. Rev. A 57 4877–88
[26] Doherty A C and Jacobs K 1999 Feedback control of quantum systems using continuous state estimation Phys. Rev. A 60 2700–11
[27] Wilson D J, Sudhir V, Piro N, Schilling R, Ghadimi A and Kippenberg T J 2015 Measurement-based control of a mechanical oscillator at its thermal decoherence rate Nature 524 325
[28] Nha H and Carmichael H J 2004 Entanglement within the quantum trajectory description of open quantum systems Phys. Rev. Lett. 93 120408
[29] Viviescas C, Guevara I, Carvalho A R R, Busse M and Buchleitner A 2010 Entanglement dynamics in open two-qubit systems via diffusive quantum trajectories Phys. Rev. Lett. 105 210502
[30] Koppenhöfer M, Bruder C and Lörch N 2018 Unraveling nonclassicality in the optomechanical instability Phys. Rev. A 97 063812
[31] Koppenhöfer M, Bruder C and Lörch N 2020 Heralded dissipative preparation of nonclassical states in a Kerr oscillator Phys. Rev. Res. 2 013071
[32] Bose S, Knight P L, Plenio M B and Vedral V 1999 Proposal for teleportation of an atomic state via cavity decay Phys. Rev. Lett. 83 5158–61
[33] Greplova E, Mølmer K and Andersen C K 2016 Quantum teleportation with continuous measurements Phys. Rev. A 94 042334
[34] Ficheux Q, Jezouin S, Leghtas Z and Huard B 2018 Dynamics of a qubit while simultaneously monitoring its relaxation and dephasing Nat. Commun. 9 1926
[35] Vijay R, Macklin C, Slichter D, Weber S, Murch K, Naik R, Korotkov A N and Siddiqi I 2012 Stabilizing Rabi oscillations in a superconducting qubit using quantum feedback Nature 490 77–80
[36] Armen M A, Au J K, Stockton J K, Doherty A C and Mabuchi H 2002 Adaptive homodyne measurement of optical phase Phys. Rev. Lett. 89 133602
[37] Naghiloo M, Foroozani N, Tan D, Jadbabaie A and Murch K 2016 Mapping quantum state dynamics in spontaneous emission Nat. Commun. 7 1–7
[38] Bouten L, Van Handel R and James M R 2007 An introduction to quantum filtering SIAM J. Control Optim. 46 2199–241
[39] Flurin E, Martin L S, Hacohen-Gourgy S and Siddiqi I 2020 Using a recurrent neural network to reconstruct quantum dynamics of a superconducting qubit from physical observations Phys. Rev. X 10 011006
[40] Rackauckas C and Nie Q 2017 Differentialequations.jl—a performant and feature-rich ecosystem for solving differential equations in Julia J. Open Res. Softw. 5 15
[41] Rackauckas C and Nie Q 2017 Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory Discrete Contin. Dyn. Syst. B 22 2731
[42] Rackauckas C and Nie Q 2020 Stability-optimized high order methods and stiffness detection for pathwise stiff stochastic differential equations 2020 IEEE High Performance Extreme Conf. (HPEC) (Piscataway, NJ: IEEE) pp 1–8
[43] Caneva T, Calarco T and Montangero S 2011 Chopped random-basis quantum optimization Phys. Rev. A 84 022326
[44] Glynn P W 1990 Likelihood ratio gradient estimation for stochastic systems Commun. ACM 33 75–84
[45] Yang J and Kushner H J 1991 A Monte Carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems SIAM J. Control Optim. 29 1216–49

[46] Kleijnen J P and Rubinstein R Y 1996 Optimization and sensitivity analysis of computer simulation models by the score function method *Eur. J. Oper. Res.* **88** 413–27
[47] Williams R J 1992 Simple statistical gradient-following algorithms for connectionist reinforcement learning *Mach. Learn.* **8** 229–56
[48] Baydin A G, Pearlmutter B A, Radul A A and Siskind J M 2017 Automatic differentiation in machine learning: a survey *J. Machine Learn. Res.* **18** 5595–637
[49] Wengert R E 1964 A simple automatic derivative evaluation program *Commun. ACM* **7** 463–4
[50] Rackauckas C, Ma Y, Dixit V, Guo X, Innes M, Revels J, Nyberg J and Ivaturi V 2018 A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions (arXiv:1812.01892)
[51] Pontryagin L S 2018 *Mathematical Theory of Optimal Processes* (Oxford: Routledge)
[52] Li X, Wong T-K L, Chen R T and Duvenaud D 2020 Scalable gradients for stochastic differential equations (arXiv:2001.01328)
[53] Kidger P, Foster J, Li X, Oberhauser H and Lyons T 2021 Neural SDEs made easy: SDEs are infinite-dimensional GANs *Int. Conf. on Learning Representations* (submitted)
[54] Kloeden P E and Platen E 2013 *Numerical Solution of Stochastic Differential Equations* (Berlin: Springer Science & Business Media)
[55] Schäfer F 2020 High weak order solvers and adjoint sensitivity analysis for stochastic differential equations (available at: https://summerofcode.withgoogle.com/archive/2020/projects/5076877036748800/)(Accessed 15 February 2021)
[56] Bezanson J, Karpinski S, Shah V B and Edelman A 2012 Julia: a fast dynamic language for technical computing (arXiv:1209.5145)
[57] van der Meulen F and Schauer M 2017 Continuous-discrete smoothing of diffusions (arXiv:1712.03807)
[58] van der Meulen F and Schauer M 2020 Automatic backward filtering forward guiding for Markov processes and graphical models (arXiv:2010.03509)
[59] Gough J, Belavkin V P and Smolyanov O G 2005 Hamilton–Jacobi–Bellman equations for quantum optimal feedback control *J. Opt. B: Quantum Semiclass. Opt.* **7** S237–44
[60] Sirignano J and Spiliopoulos K 2018 DGM: a deep learning algorithm for solving partial differential equations *J. Comput. Phys.* **375** 1339–64
[61] Gerlach A R, Leonard A, Rogers J and Rackauckas C 2020 The Koopman expectation: an operator theoretic method for efficient analysis and optimization of uncertain hybrid dynamical systems (arXiv:2008.08737)
[62] Lamperski A, Ghusinga K R and Singh A 2018 Analysis and control of stochastic systems using semidefinite programming over moments *IEEE Trans. Autom. Control* **64** 1726–31
[63] Greplova E, Andersen C K and Mølmer K 2017 Quantum parameter estimation with a neural network (arXiv:1711.05238)
[64] Valenti A, van Nieuwenburg E, Huber S and Greplova E 2019 Hamiltonian learning for quantum error correction *Phys. Rev. Res.* **1** 033092
[65] Krastanov S, Zhou S, Flammia S T and Jiang L 2019 Stochastic estimation of dynamical variables *Quantum Sci. Technol.* **4** 035003
[66] Schäfer F, Sekatski P, Koppenhöfer M, Bruder C and Kloc M 2021 Control of stochastic quantum dynamics with differentiable programming (available at: https://github.com/frankschae/Control-of-Stochastic-Quantum-Dynamics-with-Differentiable-Programming) (Accessed 15 February 2021)
[67] Brunton S L, Proctor J L and Kutz J N 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems *Proc. Natl Acad. Sci.* **113** 3932–7
[68] Boninsegna L, Nüske F and Clementi C 2018 Sparse learning of stochastic dynamical equations *J. Chem. Phys.* **148** 241723
[69] Kaiser E, Kutz J N and Brunton S L 2018 Sparse identification of nonlinear dynamics for model predictive control in the low-data limit *Proc. R. Soc.* A **474** 20180335
[70] Cranmer M, Sanchez-Gonzalez A, Battaglia P, Xu R, Cranmer K, Spergel D and Ho S 2020 Discovering symbolic models from deep learning with inductive biases *NeurIPS 2020*
[71] Innes M, Edelman A, Fischer K, Rackauckus C, Saba E, Shah V B and Tebbutt W 2019 Zygote: a differentiable programming system to bridge machine learning and scientific computing (arXiv:1907.07587)
[72] Chen R T, Rubanova Y, Bettencourt J and Duvenaud D K 2018 Neural ordinary differential equations *Advances in Neural Information Processing Systems* (Red Hook, NY: Curran Associates, Inc.) pp 6571–83 (https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf)
[73] Johnson S G 2007 Notes on adjoint methods for 18.336
[74] Jia J and Benson A R 2019 Neural jump stochastic differential equations *Advances in Neural Information Processing Systems* (Red Hook, NY: Curran Associates, Inc.) pp 9847–58 (https://proceedings.neurips.cc/paper/2019/file/59b1deff341edb0b76ace57820cef237-Paper.pdf)